



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

MÁSTER EN INGENIERÍA EN TECNOLOGÍAS DE
TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

**DIGITALIZACIÓN DE MÁQUINAS
MEDIANTE OPC UA PARA SER
INTEGRABLES EN SISTEMAS BAJO
FILOSOFÍA INDUSTRIA 4.0**

Autor: María Angustias Díaz Romero

Director: José Antonio Rodríguez Mondéjar

Madrid

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
“DIGITALIZACIÓN DE MÁQUINAS MEDIANTE OPC UA PARA SER
INTEGRABLES EN SISTEMAS BAJO FILOSOFÍA INDUSTRIA 4.0” en la ETS de
Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2022/2023 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.

Fdo.: María Angustias Díaz Romero

Fecha: 31/08/2023

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: José Antonio Rodríguez Mondéjar

Fecha://



COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

MÁSTER EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

DIGITALIZACIÓN DE MÁQUINAS MEDIANTE OPC UA PARA SER INTEGRABLES EN SISTEMAS BAJO FILOSOFÍA INDUSTRIA 4.0

Autor: María Angustias Díaz Romero

Director: José Antonio Rodríguez Mondéjar

Madrid

DIGITALIZACIÓN DE MÁQUINAS MEDIANTE OPC UA PARA SER INTEGRABLES EN SISTEMAS BAJO FILOSOFÍA INDUSTRIA 4.0

Autor: Díaz Romero María Angustias

Director: Rodríguez Mondéjar José Antonio

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

El objetivo en un principio trataba del desarrollo de un dispositivo con protocolo OPC UA sobre una plataforma Raspberry o una plataforma Siemens IoT2050. Sin embargo, se ha optado por utilizar un ordenador personal. Este desarrollo se realizará mediante las herramientas proporcionadas por la OPC Foundation. El propósito de este dispositivo es transformar una máquina con una automatización clásica en una máquina apta para integrarse en la industria 4.0.

Palabras clave: Digitalización de máquinas, Industria 4.0, Arquitectura Unificada OPC (OPC UA), Automatización, Node-red, PLC, TCP/IP.

1. Introducción

La digitalización de máquinas mediante OPC UA es un paso importante para lograr la integración de las máquinas en sistemas bajo filosofía Industria 4.0. OPC UA es un protocolo de comunicación industrial que permite la integración de diferentes sistemas de automatización y control, y que se está utilizando cada vez más en la industria para lograr la interconexión de los sistemas de automatización.

2. Definición del proyecto

En este presente proyecto se desarrolla una aplicación que permite convertir una máquina con automatización clásica en una máquina apta para ser integrada en sistemas bajo el paradigma en la Industria 4.0. Para ello la aplicación dota a la máquina de la comunicación con otros sistemas utilizando el protocolo **OPC UA**. El único requisito que debe cumplir la máquina original es disponer de algún protocolo de comunicación con otros sistemas. Las capacidades a través de OPC UA están limitadas por las características del protocolo original. Como caso de uso, se utiliza un ejemplo de máquina con un **controlador** (PLC) de la marca **Siemens** cuyo sistema de automatización ha sido desarrollado bajo la herramienta **TIA Portal**.

El desafío se trata por tanto en conseguir integrar la tecnología OPC UA en un sistema que previamente no cuenta con esta capacidad de comunicación. En este sentido,

4. Resultados

Se ha conseguido el desarrollo de una aplicación, la cual ha sido sometida a pruebas para un caso ejemplo específico, donde se empleó el protocolo de comunicación original de la máquina, basado en el método PUT/GET. Además, se diseñó y construyó la función FP utilizando Node-RED, proporcionando una implementación concreta y funcional para el caso en cuestión.

En este contexto, se ha abordado el desarrollo de Función de servidor OPC UA (FO) y la función de conexión (FC) de manera genérica, estos componentes fueron configurados en función a las necesidades del caso ejemplo. Para llevar a cabo las pruebas, se ha hecho uso del entorno de simulación PLCSIM Advanced que permitió emular el comportamiento del controlador de la planta. También se ha hecho uso de TIA Portal, para poder simular el entorno de la planta, empleando forzado de variables para recrear condiciones específicas y observar el comportamiento de la aplicación en situaciones determinadas.

Mostramos la comunicación entre el “UA Sample Server” y el “UA Expert”, para ilustrar cómo se establece una conexión entre un servidor OPC UA de Visual Studio y un cliente OPC UA de UA Expert, siendo esto relevante para mostrar la funcionalidad de comunicación A través de este enlace de comunicación, el cliente OPC UA, puede acceder y obtener datos del servidor.

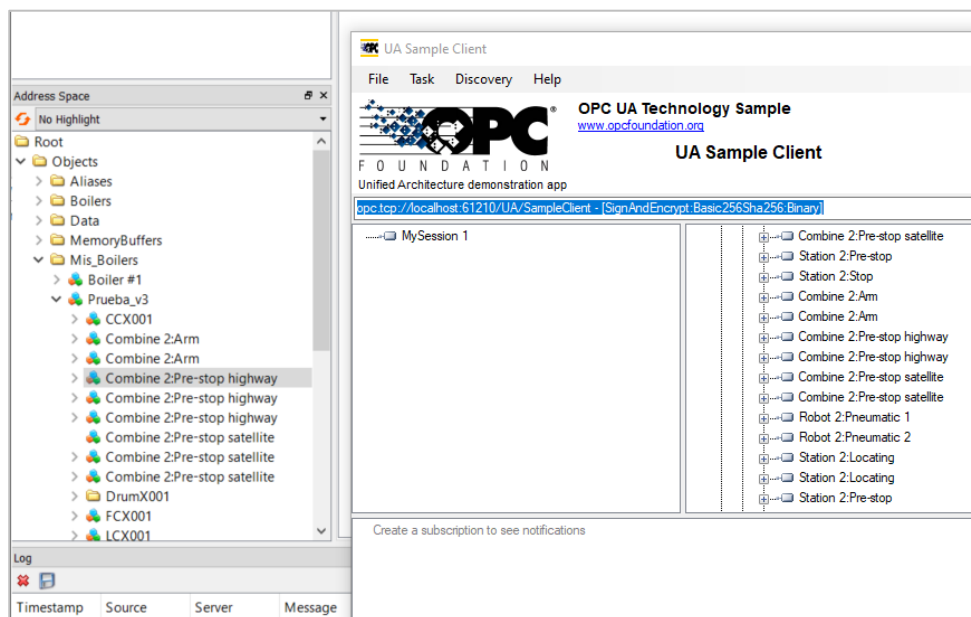


Ilustración B: Conexión entre servidor OPC UA de Visual Studio (UA Sample Client) y el cliente OPC UA (UA Expert)

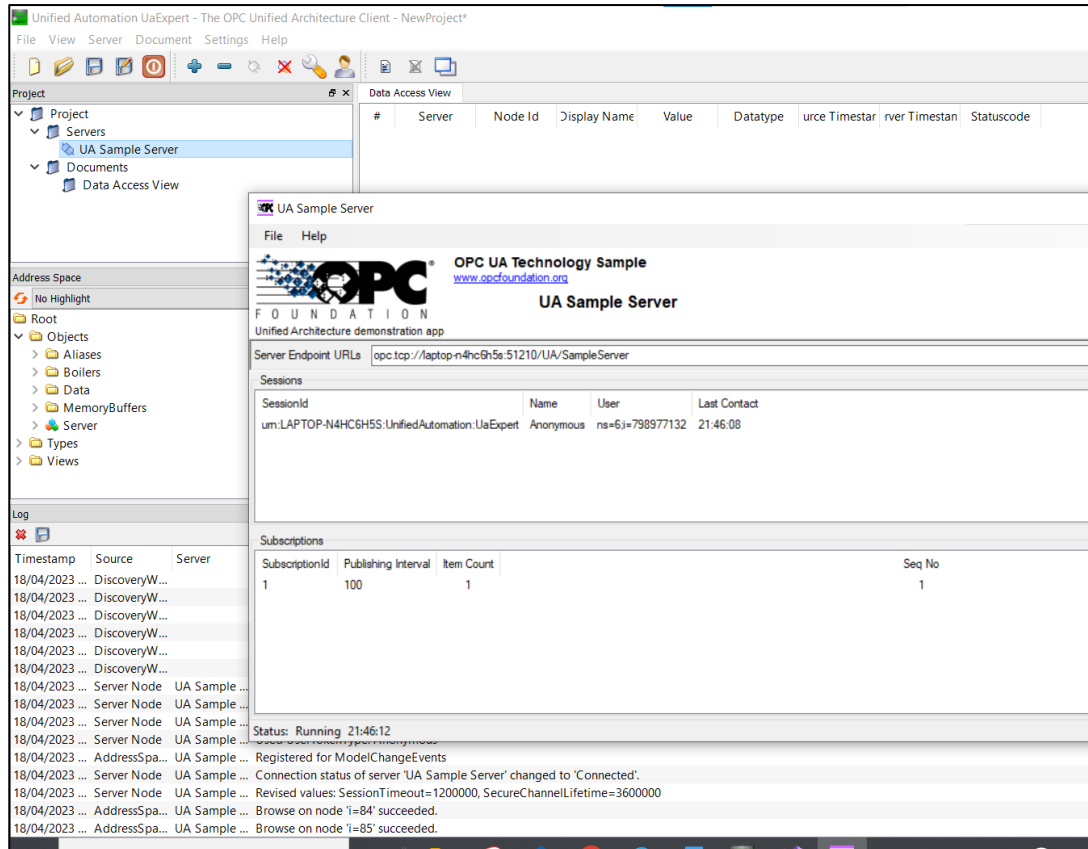


Ilustración C: Conexión entre servidor OPC UA de Visual Studio (UA Sample Server) y el cliente OPC UA (UA Expert)

De manera que se ha conseguido conectar con FO mediante un cliente OPC UA (UA Expert) pudiendo enviar y recibir información del controlador original. Se ha implementado la tecnología OPC UA en un entorno industrial no previamente compatible. A través de la creación de un servidor OPC UA desarrollado en Visual Studio y su conexión con una máquina específica, se logró establecer una comunicación bidireccional para la lectura y escritura de datos en tiempo real utilizando node-RED.

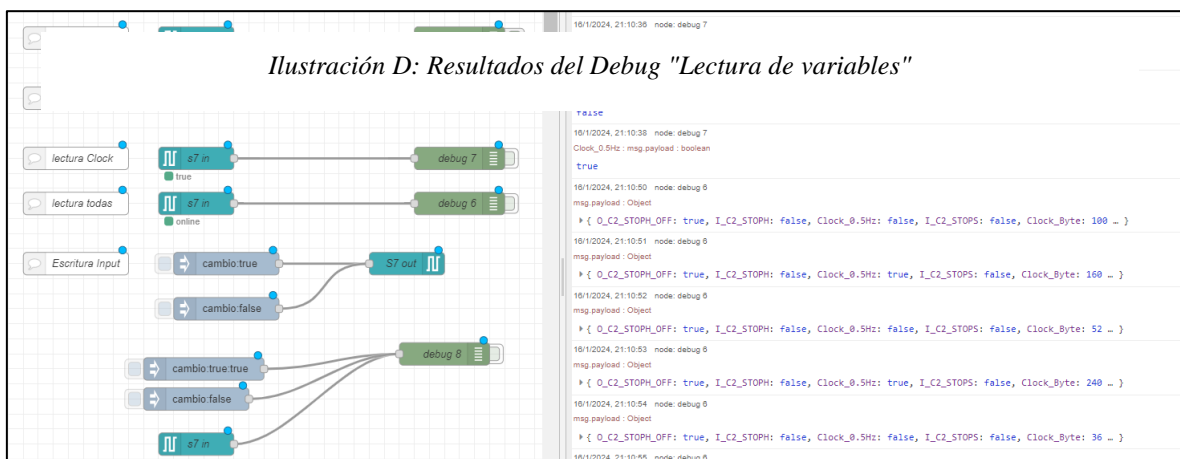


Ilustración D: Resultados del Debug "Lectura de variables"

De esta manera po

demos observar los resultados en tiempo real en el “Debug” de la herramienta deNode-RED para asegurarnos que todo está operando como se espera.

5. Conclusiones

La implementación exitosa de OPC UA en un sistema que previamente no contaba con esta capacidad de comunicación demuestra la flexibilidad y adaptabilidad de esta tecnología, además abre la puerta a futuras aplicaciones y expansiones en la digitalización de sistemas industriales. La habilidad para incorporar OPC UA en sistemas no diseñados originalmente para ello es un testimonio del potencial transformador de esta tecnología en el panorama de la Industria 4.0, al permitir la conectividad y la obtención de datos en tiempo real de manera eficaz y segura.

En resumen, la implementación exitosa de OPC UA en un entorno no preexistente demuestra la capacidad de esta tecnología para adaptarse y mejorar sistemas tradicionales, contribuyendo a la transformación de la industria hacia la era de la automatización avanzada y la conectividad total.

DIGITIZATION OF MACHINES THROUGH OPC UA FOR INTEGRATION INTO INDUSTRY 4.0 PHILOSOPHY SYSTEMS.

Autor: Díaz Romero María Angustias

Director: Rodríguez Mondéjar José Antonio

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

ABSTRACT

The goal is to develop a device with the OPC UA protocol on a Raspberry platform or a Siemens IoT2050 platform (instead, a personal computer has been chosen for use), using tools from the OPC Foundation. This device will enable the transformation of a machine with classical automation into a machine capable of being integrated into Industry 4.0."

Key words: Machine Digitization, Industry 4.0, Unified OPC Architecture (OPC UA), Automation, Node-red, PLC, TCP/IP.

1. Introduction

The digitization of machines through OPC UA is a significant step toward achieving the integration of machines into systems under the Industry 4.0 philosophy. OPC UA is an industrial communication protocol that enables the integration of different automation and control systems, and it is increasingly being used in the industry to achieve the interconnection of automation systems.

2. Project Definition

In this present project, an application is developed that enables the transformation of a machine with classical automation into a machine capable of being integrated into systems under the Industry 4.0 paradigm. To achieve this, the application provides the machine with the capability to communicate with other systems using the **OPC UA** protocol. The only requirement that the original machine must fulfill is to have some form of communication protocol with other systems. The capabilities through OPC UA are limited by the characteristics of the original protocol. As a use case, an example machine with a Siemens brand controller (**PLC**) is employed, whose automation system has been developed using the **TIA Portal** tool.

The challenge lies in successfully integrating OPC UA technology into a system that previously lacked this communication capability. In this regard, this project serves as a guide for others to understand how to carry out the integration of OPC UA in similar situations.

3. Model Description

In order for the application to be adaptable to any type of machine that meets certain communication requirements, it has been developed following the following functional architecture. It consists of three components, FP, FO, and FC, which refers to the connection between them.

It is important to note that the configurations of FO and FC are reserved for the final implementation phase, as their nature is closely tied to the specific context of each application. On the other hand, the programming of FP should be tailored to the original communication protocol of the machine in question, in case it differs from the example use case that has been used in the project.

The functional architecture will be represented by the Plant **Interaction Function (FP)**, representing the interaction with the actual physical controller, and the **OPC UA Server Function (FO)**, representing the implemented OPC UA server in Visual Studio. The connection between FP and FO is the communication link between the application and the actual physical controller, with this connection function between the plant and the OPC UA server (FC) functioning as an intermediary system between FP and FO.

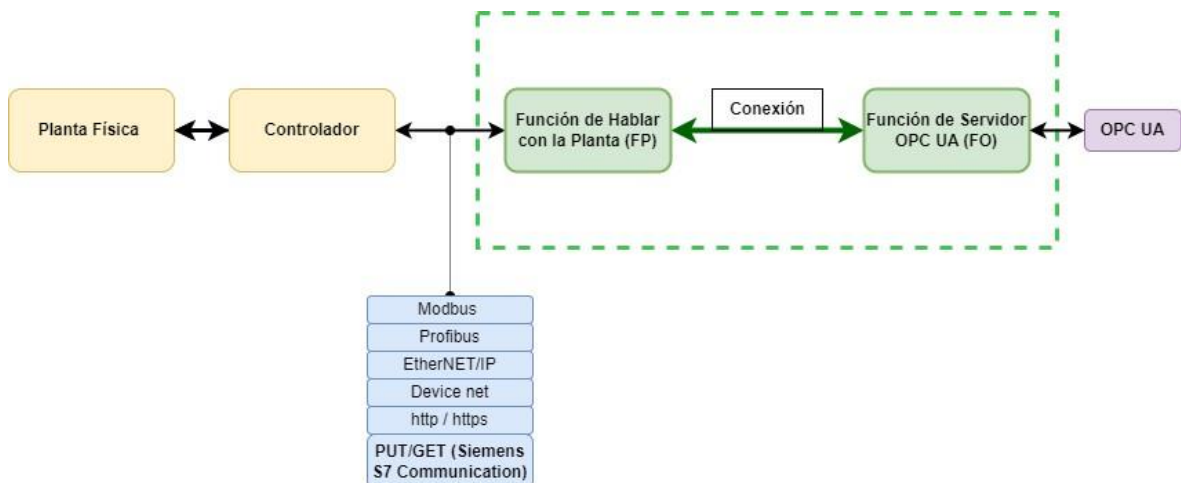


Illustration A: Functional Architecture of the Project

4. Results

The development of an application has been achieved, which has undergone testing for a specific example case, where the machine's original communication protocol, based on the PUT/GET method, was used. Additionally, the FP function was designed and constructed using Node-RED, providing a specific and functional implementation for the particular case.

In this context, the development of the OPC UA Server Function (FO) and the Connection Function (FC) was approached in a generic manner; these components were configured according to the needs of the example case. To conduct the tests, the PLCSIM Advanced simulation environment was used, which allowed for emulating the behavior of the plant's controller. Additionally, TIA Portal was employed to simulate the plant environment, using variable forcing to recreate specific conditions and observe the application's behavior in particular situations.

We illustrate the communication between the "UA Sample Server" and the "UA Expert" to demonstrate how a connection is established between a Visual Studio OPC UA server and a UA Expert OPC UA client. This is relevant to showcase the communication functionality. Through this communication link, the OPC UA client can access and retrieve data from the server.

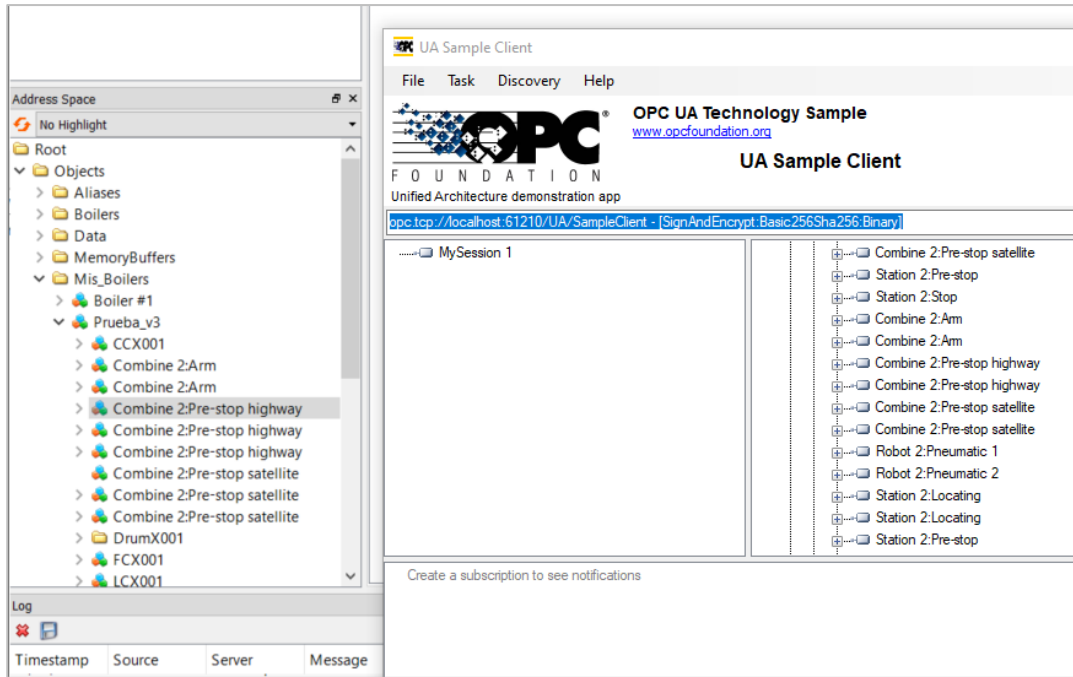


Illustration B: Connection between Visual Studio OPC UA Server (UA Sample Client) and OPC UA Client (UA Expert)

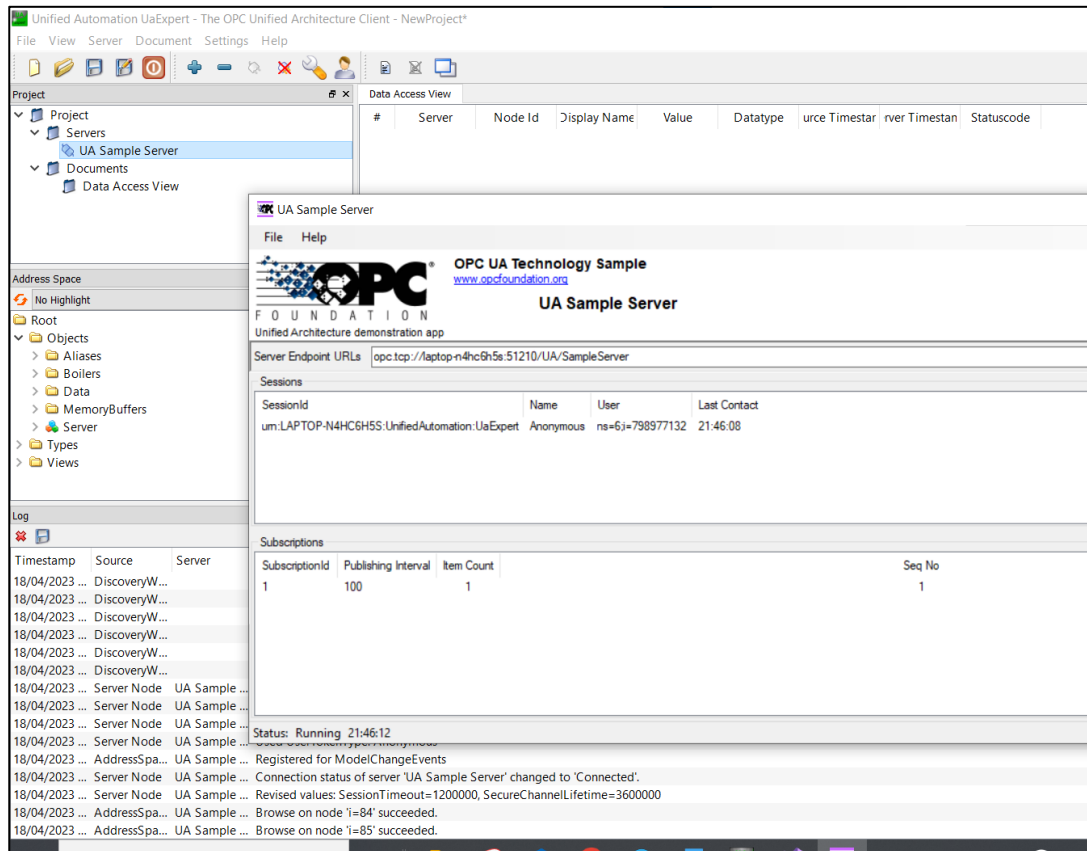


Illustration C: Connection between Visual Studio OPC UA Server (UA Sample Client) and OPC UA Client (UA Expert)

So, we have managed to connect to FO through an OPC UA client (UA Expert), enabling the sending and receiving of information from the original controller. OPC UA technology has been implemented in a previously incompatible industrial environment. Through the creation of an OPC UA server developed in Visual Studio and its connection to a specific machine, bidirectional communication for real-time data reading and writing was established using Node-RED.

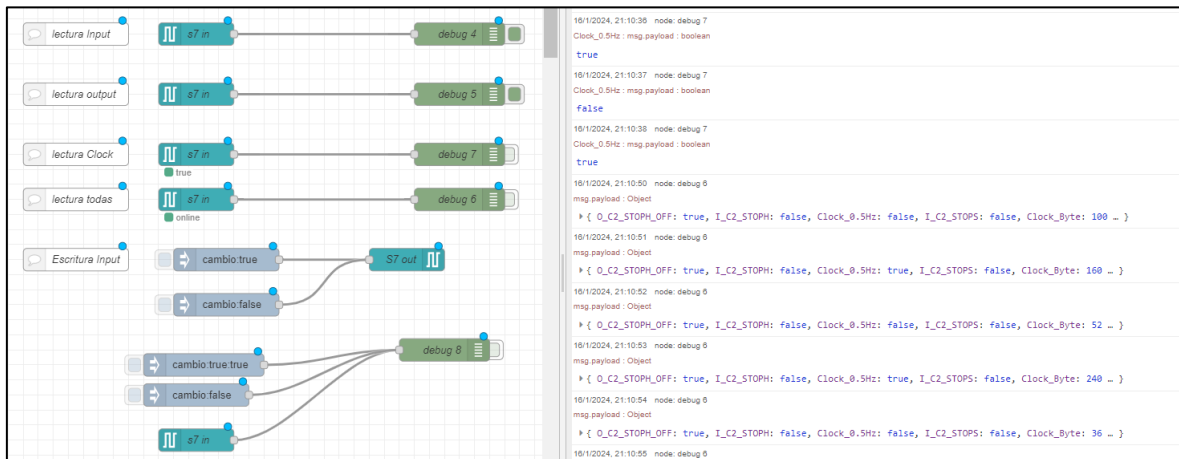


Illustration D: Debug Results - 'Reading of different variables''

In this way, we can observe real-time results in the "Debug" of the Node-RED tool to ensure that everything is operating as expected.

5. Conclusions

The successful implementation of OPC UA in a system that previously lacked this communication capability demonstrates the flexibility and adaptability of this technology. Furthermore, it paves the way for future applications and expansions in the digitization of industrial systems. The ability to incorporate OPC UA into systems not originally designed for it is a testament to the transformative potential of this technology in the Industry 4.0 landscape, enabling effective and secure connectivity and real-time data acquisition.

In summary, the successful implementation of OPC UA in a non-preexisting environment showcases the capacity of this technology to adapt and enhance traditional systems, contributing to the industry's transformation toward the era of advanced automation and complete connectivity.

Índice de la memoria

Capítulo 1. Introducción	7
1.1 Motivación del proyecto.....	7
1.2 Objetivos	8
1.3 Estructura del trabajo.....	10
1.3.1 Introducción	10
1.3.2 Estado de la técnica.....	10
1.3.3 descripción de las tecnologías.....	10
1.3.4 Desarrollo del proyecto.....	10
1.3.5 Gestión del proyecto.....	11
1.3.6 Conclusiones.....	11
1.3.7 Bibliografía.....	11
1.3.8 Anexos	11
Capítulo 2. filosofía de la industria 4.0.....	12
2.1 La industria 4.0.....	12
2.1.1 La importancia de la industria 4.0.....	13
2.2 Ejemplos de industria 4.0	14
Capítulo 3. Digitalización de máquinas mediante OPC-UA.....	18
3.1 Arquitectura Unificada opc (opc ua)	18
3.1.1 Diferencias entre OPC UA y OPC Convencional	21
3.1.2 Características y Beneficios de opc ua para la industria.....	24
3.2 OPC foundation.....	25
3.3 Desafíos y consideraciones de la digitalización mediante opc ua en la industria 4.0	27
Capítulo 4. entorno de aplicación	29
4.1 Introducción de la descripción de las tecnologías	29
4.2 Entorno de Visual Studio	29
4.2.1 Qué es .net.....	30
4.3 Librería de opc foundation	31
4.3.1 Repositorio opc ua .net estándar de github.....	31
4.4 Herramienta de Siemens: Tia Portal.....	33

4.4.1 PLC Virtual	34
4.5 Node-Red	35
4.6 Otras herramientas y recursos	36
4.6.1 Webinars de OPC UA.....	37
4.6.2 UA Expert.....	37
4.6.3 Gestión de certificados: Open SSL.....	38
Capítulo 5. Desarrollo de la aplicación	40
5.1 Arquitectura Funcional.....	40
5.1.1 Función de hablar con la planta (FP).....	42
5.1.2 Función de servidor OPC UA (FO)	43
5.1.3 Conexión entre función de planta (FP) y función del servidor OPC UA (FO)- FC.....	43
5.2 Esquema de comunicación OPC UA.....	44
5.3 Puesta en marcha y Validación de Bibliotecas OPC UA con UA Expert	46
5.3.1 UA Sample Client	46
5.3.2 UA Sample Server.....	50
5.4 Implementación de FO - Visual Studio.....	51
5.4.1 Configuración de señales (entradas y salidas) a utilizar vía OPC UA.....	52
5.5 Desarrollo de la Función de Hablar con la Planta (FP)	56
5.5.1 Configuración y desarrollo con Node RED: “node-red-contrib-s7”	56
5.5.2 Configuración adicional (S7-1200/1500).....	58
5.5.3 Lectura de datos – Comunicación S7.....	59
5.6 Caso de uso.....	65
5.6.1 Configuración comunicación PLC- Node-RED	66
5.6.2 Definición de variables remotas.....	67
5.6.3 Configuración del servidor OPC UA Visual.....	69
5.6.4 Configuración de Node-RED como cliente OPC UA.....	70
5.6.5 Procesamiento de datos en Node-RED	71
5.6.6 Verificación y comprobación del conjunto.....	71
5.6.7 Observaciones adicionales.....	73
Capítulo 6. Gestión del proyecto	74
6.1 Metodología.....	74
6.2 Gestión del proyecto y planificación.....	76

6.3	Costos del Proyecto	77
6.3.1	<i>Costos de Software</i>	77
6.3.2	<i>Costos de Hardware</i>	77
Capítulo 7. Conclusiones		78
7.1	Conclusiones Generales	78
7.2	Análisis de los objetivos.....	80
7.3	Trabajo Futuro	81
7.4	Contribución a los Objetivos del Desarrollo Sostenible.....	82
Capítulo 8. Bibliografía		84
ANEXO I: instalación plataforma .net		87
ANEXO II: Instalación de Node-Red		90
ANEXO III: Configuración de certificados con OpenSSL		92

Índice de figuras

Ilustración 1: Pilares básicos Industria 4.0 [3]	14
Ilustración 2. Mecanismo cliente-servidor [7].....	19
Ilustración 3. Modelo de publicación-suscripción [7]	19
Ilustración 4 Conectividad de OPC UA [8].....	20
Ilustración 5. Comparativa entre OPC y OPC UA [11]	23
Ilustración 6. Proyectos de “UA Sample Applications”- Visual Studio.....	32
Ilustración 7. Entorno de TIA Portal- práctica Grafcet	34
Ilustración 8. S7 PLCSIM Advanced V4	35
Ilustración 9. Webinars de OPC UA [18].....	37
Ilustración 10. Entorno de UA Expert	38
Ilustración 11. Versión OpenSSL 3.1.0.....	39
Ilustración 12. Arquitectura Funcional	41
Ilustración 13. Esquema de comunicación OPC-UA [22].....	45
Ilustración 14. OPC UA Sample Client.....	47
Ilustración 15. Añadir servidor en UA Expert.....	48
Ilustración 16. Address Space (UA Expert)	49
Ilustración 17. Conexión entre cliente-servidor	49
Ilustración 18. OPC UA Sample Server	50
Ilustración 19. Conexión establecida entre cliente-servidor (UA Epxert).....	51
Ilustración 20. Ejecución OPC.UA Compiler.exe.....	53
Ilustración 21. Código en Visual Studio: Señales Minifábrica- Robot 2 (BoilerDesign.xml)	54
Ilustración 22: Señales Inputs y Outputs Minifábrica - Robot 2	55
Ilustración 23: Señales entrada Minifábrica - Robot 2 (Address Space).....	55
Ilustración 24. Nodos de la librería "Node-red-contrib-s7"	57
Ilustración 25. Configuración "node-red-contrib-S7" (1).....	58
Ilustración 26. Configuración "node-red-contrib-S7" (2).....	59
Ilustración 27. Esquema flujos Node-RED: lectura de variable "Clock_1Hz"	60

Ilustración 28. Configuración nodo "s7 in"	60
Ilustración 29. Configuración endpoint	61
Ilustración 30. Dirección IP proyecto GrafcetUA	62
Ilustración 31. Rack y Slot proyecto GrafcetUA.....	62
Ilustración 32. Declarar variables node-RED.....	63
Ilustración 33. Conversión de direcciones de la librería "node-red-contrib-s7"	64
Ilustración 34. Configuración nodo "Debug".....	64
Ilustración 35: Esquema de comunicación	65
Ilustración 36: paleta node-red-contrib-s7.....	66
Ilustración 37: Configuración endpoint node-red (I).....	66
Ilustración 38: Configuración PLCSIM Advanced V4.0	67
Ilustración 39: Visualizador variables TIA Portal	68
Ilustración 40: Direccionamiento de variables s7-comm	68
Ilustración 41: Configuración endpoint node-RED (II)	69
Ilustración 42: Configuración del Address Space	69
Ilustración 43: Dirección – OPC UA.....	70
Ilustración 44: Dirección Node-RED	71
Ilustración 45: Flujo de Node-RED.....	71
Ilustración 46: Comprobación de resultados	72
Ilustración 47: Visualización variables (TIA Portal).....	72
Ilustración 48: Verificación de comunicación.....	73
Ilustración 49. Archivos de programa para la instalación de .NET.....	87
Ilustración 50 .Salida tras ejecución del comando "dotnet"	88
Ilustración 51: Administrados paquetes Nuget.....	89
Ilustración 52 Terminal para abrir Node-red	90
Ilustración 53. Información tras ejecución de comando de Node-RED	90
Ilustración 54. Interfaz gráfica de Node-RED.....	91
Ilustración 55. ejecutable- OpenSSL v3.1.0	92
Ilustración 56. Editor variables del entorno - OpenSSL.....	93
Ilustración 57. Versión instalada de OpenSSL.....	93

Ilustración 58. Información adicional para crear CSR	94
Ilustración 59. Archivos y certificados generados.....	96
Ilustración 60. Administración de certificados UA Expert	96

Índice de tablas

Tabla 1: Diferencias entre OPC Convencional y OPC UA [12]	23
Tabla 2. Tabla informativa de parámetros utilizados	54
Tabla 3.Función nodos - librería "Node-red-contrib-s7"	57
Tabla 4. Variable Clock_1Hz	64
Tabla 5. Diagrama de Gantt.....	76
Tabla 6. Análisis de Objetivos Generales.....	81

Capítulo 1. INTRODUCCIÓN

En este primer capítulo se realizará una breve introducción sobre el proyecto, se describirá por tanto la motivación de este, abarcando los objetivos que se pretenden alcanzar y la estructura del proyecto que se va a seguir.

1.1 MOTIVACIÓN DEL PROYECTO

Nos encontramos en un entorno de constante evolución exponencial, ante una revolución industrial caracterizada por la aparición de nuevas tecnologías que van a crear un profundo impacto y cambio en la sociedad incluso en el modelo económico. El acceso a Internet y a dispositivos tecnológicos por parte de los usuarios, el uso de la Inteligencia Artificial, el aprendizaje de las máquinas y más, están derivando en una **revolución industrial** que realizará grandes e importantes cambios y mejoras en la sociedad mediante el uso de las máquinas.

Esto nos sitúa en los inicios de la llamada **Cuarta Revolución Industrial**, conocida también como **Industria 4.0**. Este concepto va a desarrollarse y a acompañarnos a lo largo del presente proyecto. La Industria 4.0 combina avanzadas técnicas de producción y tecnologías inteligentes que se integrarán en las diferentes organizaciones, empresas y en la vida de todos nosotros

La cuarta revolución industrial está teniendo un fuerte impacto en la sociedad, y está transformando la industria, es por ello por lo que exige un nuevo tipo de profesional capacitado en las tecnologías apropiadas de producción. Cada vez más se trata con la **digitalización de las máquinas**. Diferentes áreas serán las afectadas en esta nueva etapa de Industrialización: Servicios y modelos de negocio, Seguridad y productividad continua, Seguridad TI, Seguridad de maquinaria, Ciclos de vida de productos, Cadena de valor de la industria, Habilidades de los trabajadores, Ámbito socioeconómico... Podríamos seguir mencionando muchas áreas más, de esta manera vemos, que esta nueva etapa no afecta

solamente a la industria de la maquinaria. La Industria 4.0 implica por lo tanto la **digitalización** de las cadenas de valor, haciendo uso de diversas tecnologías de procesamiento de datos, software inteligente y sensores entre otros. Generando así una cadena de valor que afectará desde proveedores hasta clientes. Podemos decir que, el objetivo que tiene la Industria 4.0 es el de tratar de **digitalizar el mundo de la fabricación**, suponiendo que habrá que integrar distintas técnicas en las actividades de producción y logística, así como en el uso de Internet en los procesos industriales.

Podemos por lo tanto tener claro que la Industria 4.0 ha llegado para quedarse, y la gran importancia de la **digitalización de la maquinaria**. Para poder comenzar en esta nueva revolución industrial se necesitarán redes seguras, convergentes y de extremo a extremo para cumplir con los requisitos que se exigen hoy en día. Se necesitan redes flexibles y escalables, dinámicas y capaces de adaptar su uso a miles de usuarios que las utilizará. Para todo lo indicado y más, el uso del componente **OPC UA** es la solución que se utilizará.

En este presente proyecto se tratará de desarrollar un dispositivo utilizando el protocolo **OPC UA** y haciendo uso de una plataforma **Siemens (TIA Portal)**, permitiendo así convertir una máquina con una automatización clásica en una máquina que pueda ser totalmente integrable en esta nueva cuarta revolución industrial o **Industria 4.0**.

1.2 OBJETIVOS

El principal objetivo del proyecto trata del aprendizaje y uso del **protocolo OPC UA** haciendo uso de las herramientas y el código que ofrece la **OPC Foundation**, que permitan por tanto convertir una máquina con una automatización clásica, en una máquina integrable bajo filosofía OPC UA.

Durante el desarrollo de este proyecto, el desafío más destacado se trata de integrar la tecnología de **OPC UA** en un sistema que previamente no cuenta con esta capacidad de comunicación. A medida que se vaya avanzando, se irán explorando posibilidades para conseguir transformar una maquinaria tradicional en un componente integral de la industria

4.0 implementando el protocolo que permita la conexión necesaria de manera segura y eficiente.

A continuación, se encuentran los objetivos particulares que se pretenden alcanzar en el proyecto:

1. Estudio de la importancia del protocolo OPC UA en la digitalización ya sea Industria 4.0 o Industrial 5.0. Explorando las diferencias de la automatización clásica y la Industria 4.0, estudiando de esta manera las necesidades de integrar esta nueva Industria en nuestro entorno tanto de fabricación como en la vida cotidiana.
2. Implementar el protocolo OPC UA a partir de lo que ofrece la OPC Foundation. Se aprenderá a utilizar esta herramienta, haciendo uso de las herramientas y librerías que nos ofrece OPC Foundation, para lograr la digitalización de máquinas y sistemas industriales, permitiendo una integración y comunicación fluida en un entorno de industria 4.0.
3. Aplicación hacia un caso práctico ejemplo, procesando y analizando datos obtenidos de la aplicación.
4. Establecer una comunicación eficiente y confiable entre los diversos dispositivos y sistemas, utilizando el protocolo OPC UA como herramienta clave para lograr la integración y la interoperabilidad. Para ilustrar la aplicación del protocolo, se empleará la plataforma de Siemens (TIA Portal) como caso ejemplo, pero la metodología y los conceptos utilizados son aplicables de manera similar a otras herramientas, como *Codesys*, en caso de ser necesario.
5. Crear una guía para convertir una máquina con automatización clásica en una máquina apta para ser integrada en la Industria 4.0, o bien a una máquina con salida OPC UA.

1.3 ESTRUCTURA DEL TRABAJO

A continuación, se hará una breve descripción de los diferentes apartados que componen el trabajo.

1.3.1 INTRODUCCIÓN

En esta sección se encuentra una breve descripción de la motivación, y las razones que llevan a desarrollar el presente proyecto, así como los objetivos a cumplir en el mismo.

1.3.2 ESTADO DE LA TÉCNICA

En este segundo apartado se encuentra toda la información previa necesaria para el desarrollo del proyecto. Se introducen conceptos importantes sobre la Industria 4.0. Se analizan diferentes ejemplos de la Industria 4.0 y propuestas de aplicación integrables en sistemas bajo esta filosofía. Se realizará además una explicación sobre la digitalización de máquinas mediante OPC UA, tratando de especificar en qué consiste y como funciona este protocolo de comunicación, exponiendo asimismo las diferencias entre OPC UA y OPC Convencional.

1.3.3 DESCRIPCIÓN DE LAS TECNOLOGÍAS

Este apartado se dedicará a describir las diferentes herramientas de las que se ha hecho uso como por ejemplo el entorno de *Visual Studio* o la herramienta de *UAGateway Expert* además de los protocolos de comunicación utilizadas y las librerías como la *OPC Foundation* para poder facilitar la comprensión del desarrollo del proyecto.

1.3.4 DESARROLLO DEL PROYECTO

Esta sección describe el funcionamiento del sistema que se realizará. Incluirá un caso práctico en el que se demostrará la conexión del sistema utilizando las herramientas proporcionadas por la OPC Foundation. A través de la implementación y utilizando el protocolo OPC UA, se logrará desarrollar e interconectar de manera eficiente dicho sistema. Encontraremos por tanto la descripción general del sistema, los requisitos y casos del uso de

este. Se analizan también los requerimientos funcionales y no funcionales. Se presenta además el diseño tanto Software como Hardware del sistema. Por último, se desarrollan pruebas realizadas que comprueban el correcto funcionamiento del sistema.

1.3.5 GESTIÓN DEL PROYECTO

En este apartado se describe la metodología utilizada para la realización del proyecto. Se añade la gestión que se ha llevado a cabo, incluyendo una lista de tareas y un diagrama visual (*Diagrama de Gantt*). Por último, encontraremos la estimación de costes que ha tenido el desarrollo del sistema.

1.3.6 CONCLUSIONES

En este apartado se hace una breve reflexión sobre el proyecto en general, los objetivos, el desarrollo y el aprendizaje adquirido durante la realización. También se añade un apartado de ‘Trabajo Futuro’ donde se describe la evolución que puede llegar a tener el sistema y como se podría desarrollar en el futuro.

1.3.7 BIBLIOGRAFÍA

Por último, se incluyen todas las fuentes de las que se ha hecho uso en el proyecto para la obtención de información.

1.3.8 ANEXOS

Se incluye un apartado de anexos donde encontramos todas las especificaciones de las diferentes herramientas y tecnologías empleadas a lo largo del proyecto.

Capítulo 2. FILOSOFÍA DE LA INDUSTRIA 4.0

En este presente capítulo iniciaremos destacando la importancia de la cuarta revolución industrial, también llamada **Industria 4.0**. Se destaca la importancia de la Industria 4.0 en la transformación de la industria y cómo se relaciona con el presente proyecto. Haciendo referencia a cómo la digitalización de las máquinas y la integración de los sistemas son elementos clave en la implementación de la Industria 4.0.

2.1 LA INDUSTRIA 4.0

La **Industria 4.0** considerada como la **cuarta revolución Industrial**, consigue integrar la digitalización e internet con los procesos industriales integrando diversas tecnologías que han permitido dar el salto tecnológico en la gestión y optimización de procesos industriales.

La Industria 4.0 está marcada por la aparición de nuevas tecnologías, como son la Inteligencia Artificial, la robótica o el IoT (*Internet of Things*) entre otros. Es así como el mundo industrial se enfrenta diariamente a nuevos retos que cambian cada vez más. La digitalización y la automatización ofrecen soluciones efectivas para abordar estos desafíos de manera más eficiente, ya que permiten comprender y aprovechar la gran cantidad de datos generados, así como dominar las técnicas de análisis necesarias para ello. La nueva revolución de la Industria 4.0 da como resultado una solución, combinando el mundo real con el digital, utilizando a la vez diferentes recursos y conectándolos entre ellos de una manera eficiente y sostenible. En este contexto, la digitalización de las máquinas y la integración de sistemas se vuelven fundamentales. La digitalización implica convertir los datos físicos en información digital y utilizarla para optimizar los procesos, mejorar la calidad y tomar decisiones informadas. Por otro lado, la integración de sistemas permite la comunicación fluida y la interoperabilidad entre diferentes componentes y sistemas en el entorno industrial. [1]

En nuestro proyecto, nos enfocamos en utilizar el protocolo **OPC UA** como herramienta clave para convertir máquinas con automatización clásica en máquinas totalmente integrables en la **Industria 4.0**. El controlador (**PLC**) de la marca *Siemens* formará parte del caso ejemplo utilizado para ilustrar esta implementación. Aprovechamos los beneficios de OPC UA, como la seguridad, la interoperabilidad y la conectividad multiplataforma, para establecer una comunicación eficiente y confiable entre los diversos dispositivos y sistemas involucrados.

A lo largo de este capítulo, exploraremos los conceptos clave de la Industria 4.0, sus beneficios, desafíos y consideraciones, así como casos de éxito que demuestran su aplicación en diferentes sectores industriales. Con esto, estableceremos una base sólida para comprender cómo nuestro proyecto se alinea con los principios y objetivos de la Industria 4.0, y cómo utilizamos OPC UA y la plataforma Siemens para impulsar la digitalización y la integración en sistemas.

2.1.1 LA IMPORTANCIA DE LA INDUSTRIA 4.0

Es importante entender el potencial que tiene esta cuarta revolución industrial, pues no solo afectará a los procesos de fabricación, si no que, afectará a todas las industrias y sectores e incluso a la sociedad. La industria 4.0 es capaz de mejorar las operaciones de negocio, causando un crecimiento de ingresos, transformando productos, mejorando la cadena de suministro e incluso las expectativas de los clientes. De esta forma, afectará a la manera en la que los clientes interactúan con las empresas, por lo que probablemente afecte y cambie la manera en la que hacemos las cosas.

El uso de diversas tecnologías relacionadas con la Industria 4.0 nos conducirá a cambios en la cadena de suministro, pues el uso de sensores, dispositivos, protocolos y diferentes entornos, provocarán mejoras en los productos, desde la creación de prototipos y pruebas hasta la incorporación de conectividad en diversos sistemas.

Cabe destacar que el impacto que puede tener la industria 4.0 se pueden notar en diversos niveles como en diferentes ecosistemas, organizaciones e individuos. [2]

2.2 EJEMPLOS DE INDUSTRIA 4.0

En este apartado demostraremos algunas de las aplicaciones y pilares que demuestran la evolución de la Industria 4.0 en diferentes sectores industriales. Para ello, comenzamos observando la siguiente imagen.



Ilustración 1: Pilares básicos Industria 4.0 [3]

En la *Ilustración 1* se muestran los **pilares básicos** de la industria 4.0. En ella figuran: robots, simulación, sistema integración, IoT (*Internet of Things*), ciberseguridad, cloud computing, realidad aumentada y Big Data. Siendo la **conectividad** entre ellos el pilar fundamental, pues para poder implementar soluciones IoT y Cloud, las fábricas deberán contar con un sistema óptimo de conectividad. Describiremos a continuación algunos de los pilares básicos que componen la industria 4.0.

- **Conectividad- Internet de las cosas:**

El IoT (*Internet of Things*) determina el primer pilar de la industria 4.0. La cantidad de dispositivos IoT conectados a internet crece exponencialmente cada vez más, pues en la actualidad prácticamente cualquier objeto cotidiano puede convertirse en un objeto del IoT. El Internet de las Cosas puede verse como la interacción y comunicación llevada a cabo entre los dispositivos que generan e intercambian datos con las cosas u

objetos del mundo real. Permiten un procesamiento de datos en tiempo real, siendo esto ventajoso a la hora de obtener mediciones de cualquier tipo, para poder analizar la información de manera más eficiente. Los sensores inteligentes, la tecnología de control y la capacidad de interactuar en tiempo real, permite una mayor eficiencia entre los distintos procesos.

Existen muchos ejemplos que ilustran la integración de IoT con la Industria 4.0, pues se puede exponer cualquier proceso de fabricación que contenga sensores conectados a internet para recopilar datos en tiempo real en una fábrica. Un ejemplo podría ser una fábrica de automóviles, pues, en la línea de producción se utilizan una gran variedad de sensores que recopilan datos como la temperatura, la presión, la vibración y otros parámetros relevantes. Estos sensores están conectados a través de la red de Internet de las Cosas (IoT) y envían los datos en tiempo real a una plataforma centralizada, la cual se encargará de analizar los parámetros que se quieran, consiguiendo de esta manera monitorear el rendimiento, predecir fallos y optimizar la eficiencia de los procesos de producción, facilitando así incluso la toma de decisiones y pudiendo mejorar la calidad de dicho proceso de fabricación. [4][4]

- **Capacidad informática- Big data y Cloud Computing:**

Hablaremos de estas dos tecnologías en conjunto, big data y cloud computing, ya que ambas trabajan en conjunto para permitir el procesamiento y el análisis de grandes cantidades de datos generados por dispositivos conectados y sistemas en el entorno industrial. En este escenario, la capacidad informática es fundamental, la conectividad en tiempo real de los datos ya que va a permitir una administración mucho más eficiente y efectiva, además de una mejor toma de decisiones, más rápida y basada en los datos recopilados. La integración de estas tecnologías en la industria 4.0 nos va a ofrecer escalabilidad, pues la capacidad de almacenamiento y procesamiento en la nube va a permitir manejar los grandes volúmenes de datos sin restricciones de recursos locales. [5]

- **Gestión inteligente- aprendizaje automático (ML)**

Los algoritmos de inteligencia artificial y aprendizaje automático (ML) son el tercer pilar de la industria 4.0. Implican técnicas y algoritmos matemáticos para resolver problemas complejos. La Industria 4.0 supone una revolución ante este escenario, sobre todo porque va a afectar al modelo productivo generando grandes beneficios, algunos tales como: el ML permite la optimización de procesos al analizar grandes volúmenes de datos para identificar patrones y tendencias, lo que lleva a una mayor eficiencia y reducción de costos. Además, el ML facilita el mantenimiento predictivo al analizar datos de sensores para detectar posibles fallos, lo que permite programar tareas de mantenimiento de manera proactiva, reduciendo el tiempo de inactividad y maximizando la vida útil de los equipos. También mejora el control de calidad al detectar anomalías en tiempo real, optimiza la cadena de suministro mediante el análisis de datos de la demanda y los precios, y personaliza la experiencia del cliente al utilizar datos para ofrecer productos y servicios adaptados a las necesidades individuales. [6]

- **Ciberseguridad**

Por último, también está presente la ciberseguridad en la industria 4.0, pues con la digitalización y la interconexión de sistemas en la industria 4.0, aumenta considerablemente la superficie de exposición a riesgos y ataques cibernéticos. Pues adoptar tecnologías como el *Internet de las Cosas*, la computación en la nube y la conectividad de red provoca que haya más oportunidades de ciberataques. Más aun en la Industria 4.0, donde las máquinas, los sistemas y los dispositivos están interconectados y se comunican entre sí, es en este escenario donde la ciberseguridad juega un papel fundamental para proteger los activos digitales, los datos confidenciales y los sistemas críticos. Una brecha de seguridad en un entorno industrial puede tener consecuencias graves, como el robo de información sensible, el sabotaje de procesos de producción o la interrupción de operaciones. La ciberseguridad en la Industria 4.0 implica, por lo tanto, la implementación de medidas de seguridad proactivas para prevenir y mitigar los riesgos cibernéticos. Por

ello en este presente proyecto, se tendrá en cuenta las distintas configuraciones de seguridad para utilizar el protocolo OPC UA en un entorno industrial de una manera segura, protegiendo la integridad, confidencialidad y autenticidad de los datos.

Capítulo 3. DIGITALIZACIÓN DE MÁQUINAS

MEDIANTE OPC-UA

3.1 ARQUITECTURA UNIFICADA OPC (OPC UA)

En la industria 4.0 cada vez hay más sistemas y protocolos de comunicación entre máquinas, conexiones en remoto e incluso computación en la nube. Esta tendencia se debe a un crecimiento exponencial en la automatización industrial como ya se ha visto en anteriores secciones.

Cuando hablamos de OPC UA se trata de la **Arquitectura Unificada OPC** (*Open Platform Communications Unified Architecture, OPC UA*), pues es la evolución de la tecnología OPC Clásica. Es un protocolo de comunicación independiente del proveedor para aplicaciones de automatización industrial, multiplataforma, abierta, orientada a servicios, segura y con diversos modelos de información disponibles dependiendo de la aplicación. Podemos por tanto definir la **Arquitectura Unificada de Comunicaciones de Plataforma Abierta (OPC UA)** como un protocolo de comunicación entre máquinas que permite intercambiar información entre máquinas de manera sencilla. Esta tecnología es mantenida por la **OPC Fundación**, de la que hablaremos con más detalle en secciones posteriores (Parte I3.2 OPC foundation)

OPC UA se ha basado en el principio de **cliente-servidor**, donde un cliente solicita información y recibe una respuesta de un servidor, permitiendo una comunicación continua y sin interrupción entre sensores y actuadores. Podemos entender mejor este mecanismo en la siguiente imagen. *Ilustración 2.*

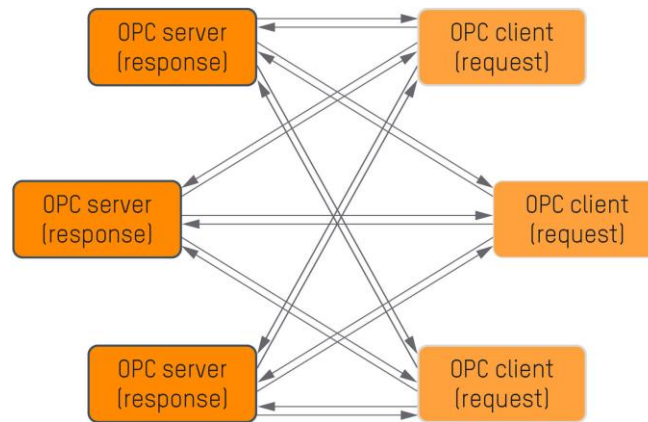


Ilustración 2. Mecanismo cliente-servidor [7]

El problema viene cuando la red incluye muchos nodos, entonces este sistema de cliente-servidor resulta limitado. Por esta razón se decidió añadir una arquitectura de **publicación-suscripción (Pub-Sub)** a la solución de comunicación de OPC UA. De esta manera, el modelo de publicación-suscripción permite una comunicación “*de uno a muchos*” y de “*muchos a muchos*”. Un servidor envía sus datos a la red (publicador) y todos los clientes pueden recibirlos (suscriptor), de esta manera, el servidor puede publicar datos periódicamente o cuando ocurre un evento, y los clientes pueden suscribirse para recibir esos datos de manera automática sin necesidad de realizar solicitudes explícitas. [7]

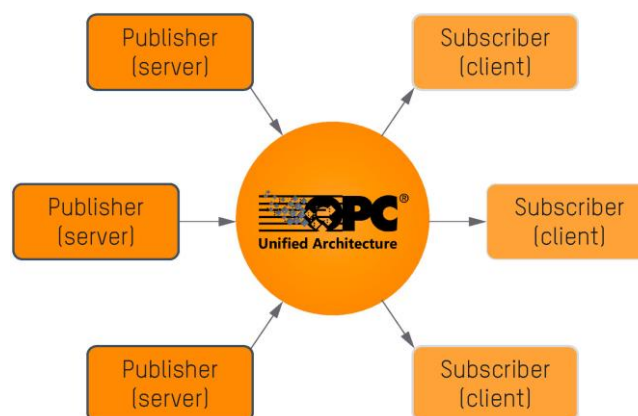


Ilustración 3. Modelo de publicación-suscripción [7]

El protocolo es además **independiente** a la plataforma resultando esto ser una ventaja competente, además dispone de mecanismos de seguridad integrados. La flexibilidad e

independencia de OPC UA hacen que este protocolo de comunicación sea ideal para la implementación en la **Industria 4.0**.

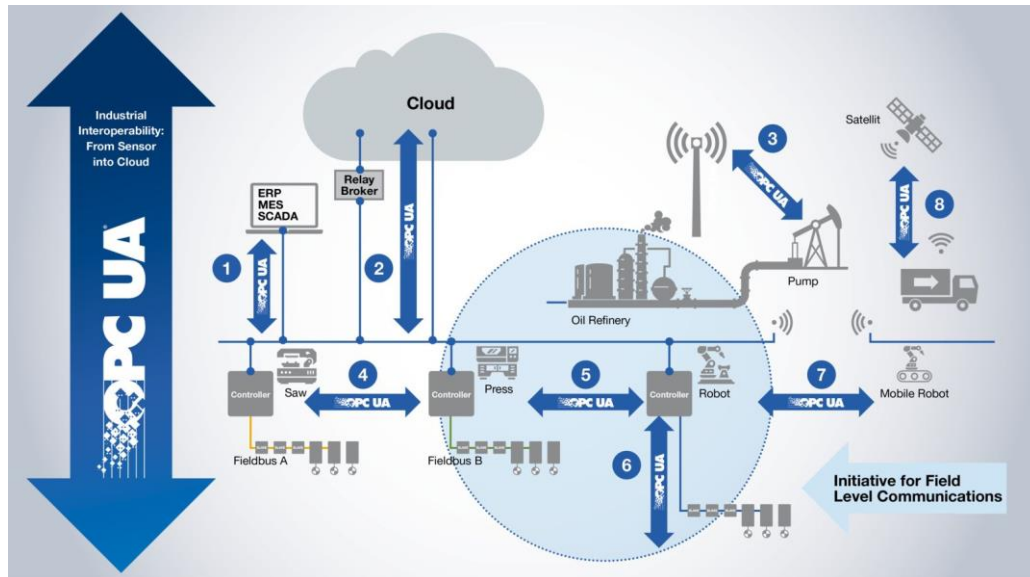


Ilustración 4 Conectividad de OPC UA [8]

OPC UA va a solucionar sobre todo los problemas basados entre la IP y la planta de producción. Pues solventará problemas como, por ejemplo, los interfaces, las puertas de enlace y la consiguiente pérdida de información:

- **Interfaces:** proporcionando un modelo de objetos y una arquitectura estándar, lo que facilita la integración de los sistemas. Elimina la necesidad de interfaces personalizadas y facilita así la interoperabilidad entre equipos y proveedores.
- **Puertas de enlace:** En algunos casos, es necesario utilizar puertas de enlace (*gateways*) para traducir los protocolos de comunicación entre dispositivos que pueden no ser compatibles. OPC UA evita la necesidad de puertas de enlace adicionales, ya que proporciona un protocolo estandarizado.
- **Pérdida de información:** OPC UA establece conexiones directas y seguras entre cliente y servidor, lo que minimiza la pérdida de información durante la transferencia de datos.

OPC UA transfiere todos los datos del proceso de producción mediante un único protocolo. Además, permite diferentes conexiones: dentro de una misma máquina, entre máquinas o entre una máquina y una base de datos en la nube. OPC UA elimina por tanto la necesidad de utilizar los tradicionales sistemas de bus de campo a nivel de fábrica. [9]

OPC UA proporciona por tanto transportes abiertos, mejor seguridad y un modelo de información completo, aporta un mecanismo muy flexible y adaptable para mover datos entre sistemas de tipo empresarial y tipos de controles, dispositivos de monitoreo y sensores que interactúan con datos en el mundo real.

3.1.1 DIFERENCIAS ENTRE OPC UA Y OPC CONVENCIONAL

En la década de los 2000, los servidores **OPC** se extendieron por todas partes, y se implementaron en la industria de la automatización con mucho éxito. Sin embargo, OPC presentaba ciertos problemas relacionados con la seguridad, la dependencia de las plataformas de Microsoft, la ineficiencia y los costosos métodos para transferir datos, así como problemas de mantenimiento. Por esta razón, surgió posteriormente **OPC UA**, que abordó y solucionó gran parte de los problemas mencionados anteriormente. Esta nueva generación de la tecnología de OPC proporciona mayor seguridad a la hora de transferir información entre servidores y clientes, un modelo de datos más completo, un mecanismo más flexible y mayor adaptación entre otros beneficios. [10]

En esta sección se va a realizar un análisis y comparación entre el OPC Convencional y el OPC UA actual para de esta manera poder entender el porqué de esta evolución del protocolo, en cuanto a necesidades y características. Para ello se irán mencionando a continuación características en las que se especificarán las diferencias entre ambos protocolos.

- **Sistemas Operativos**

OPC Clásico está basado en tecnología Windows, ya que utiliza COM/DCOM (*Distribution COM*) para mover datos entre aplicaciones, siendo esta una tecnología

más antigua que Microsoft, sin embargo, OPC UA, se puede implementar sobre cualquier sistema operativo ya sea Android, Linux o Windows.

- **Tecnología embebida**

En OPC UA, los servidores pueden residir dentro de los mismos PLC (Autómata Programable), formando parte del hardware. En OPC Clásico, los servidores son drivers, los cuales deben instalarse sobre una máquina que se comunique directamente sobre las máquinas.

- **Seguridad**

Cada vez somos más conscientes de la importancia de proteger los datos. OPC UA permite la encriptación de las comunicaciones mediante certificados de seguridad que los interlocutores deben conocer antes de empezar a comunicarse, siendo esto más seguro.

- **Comportamiento de FireWall**

El OPC convencional necesitaba tener muchos puertos abiertos ya que necesitaba muchas configuraciones especiales, sin embargo, OPC UA necesita solo un puerto para la comunicación entre aplicaciones solventando de esta manera los problemas de DCOM de FireWall.

- **Modelo de información**

El OPC clásico carece de la capacidad de representar adecuadamente los tipos de datos, información y relaciones entre elementos de datos y sistemas que son esenciales en el mundo conectado en el que hoy vivimos. Además, tiene el problema de que OPC clásico utiliza diferentes tecnologías para cada tipo de datos: OPC DA para el tiempo real, OPC HDA para historización de datos y OPC A&E para alarmas y eventos. El inconveniente es que estas tecnologías no pueden comunicarse entre sí. OPC UA resuelve este problema, ya que consigue combinar todas ellas en una sola tecnología. [12]

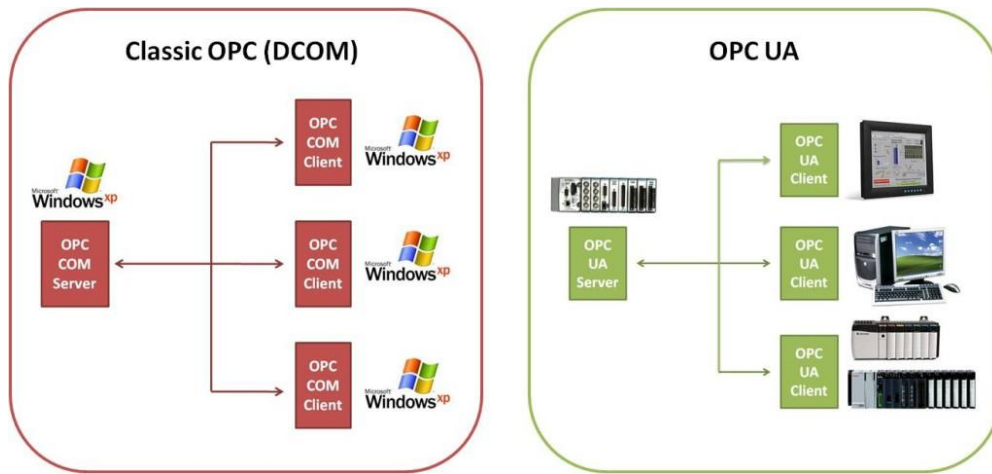


Ilustración 5. Comparativa entre OPC y OPC UA [11]

En la siguiente tabla se recogen las principales diferencias entre OPC UA y OPC clásico.

	<i>OPC UA</i>	<i>OPC Clásico</i>
Sistema Operativo	Cualquier sistema operativo	Windows (COM/DCOM)
Tecnología embebida	Puede residir dentro del PLC y formando parte del software de este	Debe instalarse sobre una máquina
Modelo de Información	Combina todas las opciones en una sola tecnología	Diferentes tecnologías para cada tipo de fuente de datos. No se pueden comunicar entre sí
Firewall	Sencillo de configurar	Complejo de configurar
Seguridad	Encriptación de las comunicaciones	Comunicaciones más vulnerables y menos seguras

Tabla 1: Diferencias entre OPC Convencional y OPC UA [12]

En resumen, mientras que el protocolo **OPC convencional** podía provocar fallos de vulnerabilidad, la nueva versión **OPC UA**, con la arquitectura unificada resuelve este conflicto además de aportar una mayor compatibilidad. Con el sistema OPC Convencional, únicamente nos limitábamos a enviar datos entre sistemas SCADA y sensores, con OPC UA podemos enviar datos a cualquier tipo de sistema (bases de datos, hojas Excel, aplicaciones móviles, etc.) de una manera más sencilla y segura.[13]

3.1.2 CARACTERÍSTICAS Y BENEFICIOS DE OPC UA PARA LA INDUSTRIA

Al haber analizado ya las diferencias entre el OPC Convencional y la nueva versión OPC UA, podemos definir de manera más clara y concisa los beneficios que trae consigo el protocolo OPC UA para la industria. OPC UA ha sido una mejora significativa respecto al antiguo estándar OPC, ha sido diseñado para mejorar y optimizar la comunicación entre diversos sistemas activos presentes en una planta industrial.

A diferencia del OPC convencional que se enfoca principalmente en la comunicación máquina a máquina (M2M), OPC UA va más allá y ofrece la posibilidad de procesar datos en el contexto del Internet de las cosas (**IoT**), es decir, OPC UA no solo facilita la transferencia de datos entre máquinas, sino que también brinda la capacidad de realizar análisis, almacenamiento y procesamiento de datos para aplicaciones IoT.

Además, el protocolo de comunicación OPC UA va a permitir a empresas pequeñas, las cuales tienen un menor **presupuesto**, que puedan incorporar sus soluciones, permitiendo un cambio hacia la industria 4.0 con costes de integración menores, pues tiene un estándar abierto y gratuito, independencia de plataforma y facilidad de integración. Esto hace que sea una opción atractiva y accesible para empresas con presupuestos más limitados que deseen adoptar tecnologías avanzadas. [13]

OPC UA utiliza además plataformas **escalables**, múltiples modelos de seguridad, diferentes capas de transporte y un modelo de información que permite controlar e interactuar libremente con diversas aplicaciones de servidor a alto nivel. De esta manera OPC UA puede comunicar cualquier tipo de datos, sin importar su complejidad ni la cantidad

de información, pudiendo ser ésta muy elevada, como por ejemplo una planta completa de fábrica. [10]

Son muchas las características y beneficios que esta tecnología nos ofrece. A continuación, se numeran alguna de las características más relevantes:

- **Escalabilidad:** OPC UA es escalable e independiente. Admite diversos servidores y sensores.
- **Flexibilidad:** admite un espacio de direcciones flexible. Se organiza en torno al concepto de un objeto.
- **Codificación y transporte común:** utiliza estándares garantizando así la conectividad.
- **Seguridad:** Implementa encriptación de datos además de autenticación entre cliente y servidor.
- **Capacidad:** alta capacidad a través de internet.
- **Servicios:** contiene un conjunto robusto de servicios para eventos, alarmas, lectura, escritura y más.
- **Interoperabilidad:** OPC UA tiene interoperabilidad certificada.
- **Modelo de información:** sofisticado, compatible y sencillo de entender y utilizar.

3.2 OPC FOUNDATION

La **OPC Foundation** es una organización dedicada a la promoción y el desarrollo de estándares para la comunicación y la interoperabilidad de sistemas en la industria y automatización. Se dedica por tanto al desarrollo y mantenimiento del estándar **OPC** (*Open Platform Communications*) y promover su adopción en la industria. Estos estándares incluyen **OPC UA (Unified Architecture)**, que como ya hemos visto anteriormente, es una especificación independiente para el intercambio seguro y confiable de información entre diferentes sistemas. [14]

En el presente proyecto, se va a hacer por tanto uso de las tecnologías y especificaciones desarrolladas por la **OPC Foundation**, particularmente del estándar **OPC UA** (*Open Platform Communications Unified Architecture*). El objetivo de utilizar OPC UA en el presente proyecto, es lograr la digitalización de máquinas y sistemas industriales, permitiendo una integración y comunicación fluida en un entorno de industria 4.0. Mediante el uso de las herramientas que nos ofrece la **OPC Foundation** se conseguirá implementar una solución que facilite la recopilación de datos de las máquinas, el intercambio de información en tiempo real y la monitorización y control remoto que ayudará a una toma de decisiones basada en datos en mi entorno industrial.

La OPC Foundation proporciona varias librerías para diferentes lenguajes de programación y para distintas plataformas. Podemos encontrar por lo tanto numerosas librerías con las que podremos trabajar. No obstante, cada librería tiene sus propias características y requisitos de instalación, por lo que es importante elegir la librería adecuada para el proyecto. En este proyecto se va a trabajar con la librería **OPC UA .NET Standard Library**, la cual nos proporcionará una implementación de OPC UA para la plataforma .NET. En secciones posteriores se detallarán las especificaciones y funcionalidades de esta biblioteca, así como los pasos necesarios para su configuración y utilización eficiente. (Parte I4.3: Librería de opc foundation)

En resumen, la OPC Foundation y su estándar OPC UA desempeñan un papel fundamental en mi proyecto al proporcionar las herramientas y especificaciones necesarias para la digitalización de máquinas y la integración de sistemas bajo la filosofía de la Industria 4.0. Se utilizará OPC UA para establecer una comunicación segura y eficiente entre los diferentes componentes del sistema, permitiendo la optimización de procesos, el mantenimiento predictivo y la toma de decisiones basada en datos en mi entorno industrial.

3.3 DESAFÍOS Y CONSIDERACIONES DE LA DIGITALIZACIÓN MEDIANTE OPC UA EN LA INDUSTRIA 4.0

Cabe destacar la importancia de los diversos desafíos y consideraciones que surgen al implementar la digitalización en máquinas utilizando el protocolo OPC UA en el marco de la industria 4.0. Por ello es importante antes de comenzar con la descripción de tecnologías y desarrollo de la aplicación, hacer un breve estudio y tener en cuenta algunos aspectos clave que pueden obstaculizarnos. Se numeran y explican con brevedad a continuación:

- **Interoperabilidad y compatibilidad**

Es importante investigar previamente la interoperabilidad entre los dispositivos, sistemas y proveedores que utilizan OPC UA en un entorno de Industria 4.0. Será necesario asegurarse de que las máquinas y los sistemas puedan comunicarse de manera fluida mediante OPC UA, teniendo en cuenta los protocolos y estándares existentes utilizados en el entorno. Esto nos facilitará la integración y comunicación fluida entre los distintos componentes del sistema.

- **Seguridad y ciberseguridad**

Se repite este apartado sobre ciberseguridad, pues resulta necesario implementar las correctas medidas de seguridad al utilizar OPC UA. Sobre todo, si se va a dar uso en un entorno que trabaja con abundantes datos que pueden ser confidenciales. El empleo de autenticación robusta, cifrados de datos o uso de claves y certificados van a ser necesarios para un correcto uso de los datos.

- **Configuración y gestión de la infraestructura de comunicación OPC UA**

Configurar y administrar la infraestructura de comunicación OPC UA puede presentar desafíos en términos de la configuración correcta de servidores y clientes OPC UA, así como la gestión de certificados y claves de seguridad. Será necesario comprender en detalle los requisitos de configuración y administración para garantizar una comunicación confiable y segura entre las máquinas y los sistemas.

- **Integración de datos y análisis**

Aunque la gestión de datos y el análisis de estos no sean una responsabilidad directa del proyecto, se han de tener en cuenta, pues es importante considerar y saber el tipo de datos con los que se va a trabajar para una correcta configuración de la lectura y modificación de los datos. Tener conocimiento de los datos con los que se trabaja ayudará además a mejorar los procesos y eficiencia en general.

- **Capacitación y adopción de los flujos de trabajo**

Una vez que se implementa la solución de OPC UA en la digitalización de las máquinas, es muy importante asegurarse de que los miembros del equipo comprendan los conceptos básicos de OPC UA para saber utilizar y aprovechar la funcionalidad. Es necesario que se familiaricen con los nuevos flujos de trabajo y tengan las habilidades necesarias para mantener la infraestructura de comunicación basada en OPC UA.

Esto son solo algunas consideraciones a tener en cuenta, no obstante, a lo largo del proyecto se irán teniendo más desafíos y diferentes obstáculos que se irán nombrando y explicando en las diferentes situaciones y entornos. Tener en cuenta estos desafíos y consideraciones ayudaran a conseguir una implementación exitosa en la digitalización de máquinas mediante OPC UA.

Capítulo 4. ENTORNO DE APLICACIÓN

4.1 INTRODUCCIÓN DE LA DESCRIPCIÓN DE LAS TECNOLOGÍAS

En el contexto de digitalización de máquinas resulta fundamental comprender y conjuntar el sistema completo. La convergencia de tecnologías, protocolos y herramientas en este entorno puede resultar compleja, pero también ofrece oportunidades para mejorar la eficiencia y la interoperabilidad.

Comprender y conjuntar todo el sistema puede ser un desafío, ya que implica la integración de tecnologías específicas, como la **librería .NET de OPC Foundation**, el entorno de desarrollo **Visual Studio**, el repositorio **OPC UA .NET Standard** de **GitHub** y herramientas adicionales como **UA Expert** y **Node-RED**. Cada uno de estos componentes juega un papel importante en la comunicación y la integración de las máquinas en el contexto de la Industria 4.0.

Por lo tanto, en este capítulo se proporcionará una descripción de las tecnologías, protocolos, herramientas y librerías que se van a tratar a lo largo del proyecto para poder facilitar la lectura, comprensión y el trabajo realizado. Se describirán cada una de las tecnologías creando finalmente un entorno de aplicación conectado en conjunto. Además, se abordarán los desafíos y consideraciones específicos relacionados con cada una de estas herramientas, proporcionando algunas recomendaciones y mejoras prácticas.

4.2 ENTORNO DE VISUAL STUDIO

Visual Studio es un entorno de desarrollo integrado (**IDE**), que se utiliza para desarrollar cualquier tipo de aplicación o aprender un lenguaje de programación. Además de la edición de código, el IDE de Visual Studio contiene diseñadores gráficos, compiladores, diversas herramientas para la ejecución de código, control de código fuente, extensiones y muchas más características.

Visual Studio proporciona una plataforma robusta y familiar para trabajar con el código fuente y las herramientas de desarrollo relacionadas con **OPC UA**. La versión de Visual Studio que se va a utilizar en este presente proyecto será la 17.4.3. [15]

4.2.1 QUÉ ES .NET

.NET es una plataforma creada por **Microsoft** que sirve para poder desarrollar código y crear diferentes tipos de aplicaciones. Es de **código abierto**, siendo esta una de sus grandes ventajas. Con **.NET** se puede usar varios lenguajes, editores y bibliotecas para crear aplicaciones web, móviles, de escritorio, juegos, IoT, etc...

En el presente proyecto es necesaria la instalación y configuración de esta librería ya que haremos uso del repositorio **OPC UA .NET Standard de Github**, siendo esta una implementación basada en la plataforma **.NET** siguiendo el estándar OPC UA definido por la OPC Foundation.

.NET ofrece características avanzadas del lenguaje, como genéricos, junto con amplias bibliotecas de clases y compatibilidad con varios lenguajes para desarrolladores, en el caso del presente proyecto se hace uso de **C#**. Además, se puede usar para cualquier tipo de aplicación por lo que nos resulta muy útil poder utilizar esta plataforma que será compatible con las librerías y repositorios de OPC UA. [16]

La *librería .NET* y la *librería OPC UA .NET Standard* están estrechamente relacionadas porque la implementación de OPC UA en la plataforma **.NET** se basa en la infraestructura y las características proporcionadas por la librería **.NET**. La librería OPC UA **.NET Standard** utiliza las capacidades de la plataforma **.NET** para ofrecer una implementación de OPC UA compatible y eficiente. Será por lo tanto de vital importancia familiarizarnos con esta librería y el entorno de desarrollo para poder realizar la implementación necesaria, para ello se han seguido una serie de seminarios y de videotutoriales que podemos encontrar en el siguiente enlace: (<https://learn.microsoft.com/es-es/training/>). Además, en el **Anexo I** se especificará los pasos a seguir para la instalación de la librería **.NET** [ANEXO I: instalación plataforma .net

4.3 LIBRERÍA DE OPC FOUNDATION

La librería de **OPC Foundation** es un conjunto de herramientas, bibliotecas y estándares desarrollados y mantenidos por la OPC Foundation. Estas librerías proporcionan un conjunto de funcionalidades y capacidades para implementar una comunicación basada en OPC UA.

4.3.1 REPOSITORIO OPC UA .NET ESTÁNDAR DE GITHUB

El repositorio que utilizaremos en el proyecto será “**OPC UA .NET Standard máster**” de **GitHub**, disponible en la siguiente dirección: <https://github.com/OPCFoundation/UA-.NETStandard>. Se trata de un repositorio de código abierto que contiene la implementación de la especificación OPC UA en .NET Standard, desarrollado y mantenido por la **OPC Foundation**. Proporciona las bibliotecas, clases y funcionalidades necesarias para desarrollar aplicaciones compatibles con OPC UA en entornos basados en .NET.

Este repositorio está organizado en varios proyectos de **Visual Studio** (14 proyectos en total) que se encargan de diferentes aspectos de la implementación de OPC UA, contiene diferentes herramientas y ejemplos de uso como el protocolo de comunicación, la serialización de datos, la seguridad, entre otros.

Para este proyecto se ha hecho uso del ejemplo “*UA Sample Applications*” que se encuentra en el repositorio de muestras: <https://github.com/OPCFoundation/UA-.NETStandard-Samples>. Este repositorio se enfoca específicamente en ejemplos y muestras de código relacionados con la implementación de OPC UA en entornos basados en .NET utilizando la librería OPC UA .NET Standard. La solución “UA Sample Applications.sln” incluye varios proyectos, algunos de los cuales se mencionan a continuación con una breve descripción.

- **OPC UA Sample Client:** Un proyecto que muestra cómo desarrollar una aplicación cliente que se conecta a un servidor OPC UA y lee o escribe datos en el servidor.

- **OPC UA Sample Server:** Un proyecto que muestra cómo desarrollar un servidor OPC UA que expone datos y funcionalidades a los clientes que se conectan a él.
- **OPC UA Sample Controls:** Un proyecto que muestra cómo utilizar los controles y componentes proporcionados por la librería OPC .NET Standard para facilitar el desarrollo de aplicaciones OPC UA.
- **OPC UA Sample Discovery Server:** Un proyecto que muestra cómo implementar un servidor de descubrimiento OPC UA que permite a los clientes descubrir los servidores OPC UA disponibles en la red.

Cada uno de los proyectos tienen su propio código fuente, archivos de configuración y otros recursos necesarios para su ejecución.

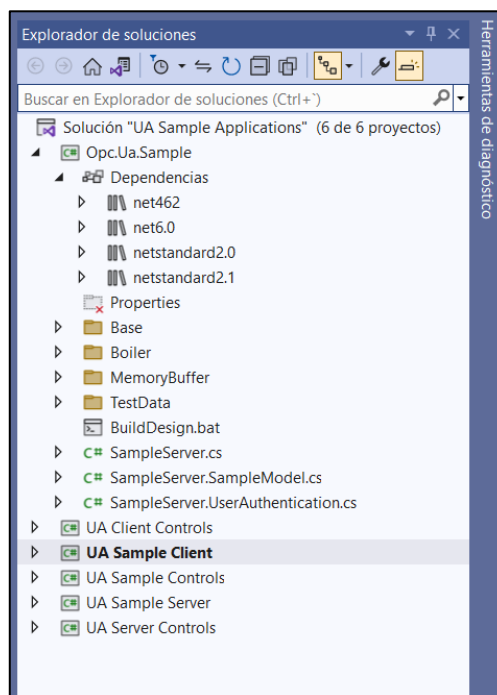


Ilustración 6. Proyectos de “UA Sample Applications”- Visual Studio

La elección de este ejemplo particular del repositorio “UA Sample Applications”, será para tener como referencia para la implementación de comunicación de **cliente-servidor** de OPC UA.

En resumen, el **repositorio OPC UA .NET Standard** de **GitHub** es una implementación de la especificación OPC UA en .NET Standard, que permite desarrollar **clientes y servidores OPC UA** en la **plataforma .NET**. El repositorio contiene varios proyectos que son necesarios para una implementación completa de la especificación OPC UA.

4.4 HERRAMIENTA DE SIEMENS: TIA PORTAL

TIA Portal (*Totally Integrated Automation Portal*) es una herramienta de ingeniería integral desarrollada por **Siemens**. Proporciona un entorno de desarrollo unificado para la programación, configuración y supervisión de los sistemas de automatización de Siemens, incluyendo PLCs (Controladores Lógicos Programables), HMI (Interfaces Hombre-Máquina), variadores de velocidad y otros dispositivos. [17]

Es importante destacar que TIA Portal se utiliza en el caso ejemplo, pero en otros contextos se emplearían plataformas de otros fabricantes de controladores como, por ejemplo: *ABB, Schneider, Rockwell* etc. TIA Portal no es una herramienta gratuita, se comercializa como un producto con licencia. Por ello se usará una versión de prueba gratuita llamada "TIA portal Trial" pero hay que tener en cuenta que tiene una duración limitada. También se podrá usar el software adquirido en los ordenadores de la Universidad. Se explorará el desarrollo y la configuración de PLCs en TIA Portal a través de ejemplos de prácticas como "*Práctica Grafset y acceso con OPC UA*" y "*Robot_Planta*", que utilizan el lenguaje de GRAFCET (GRAPH en argot de Siemens) para aplicar los conceptos de TIA Portal en la automatización y control.

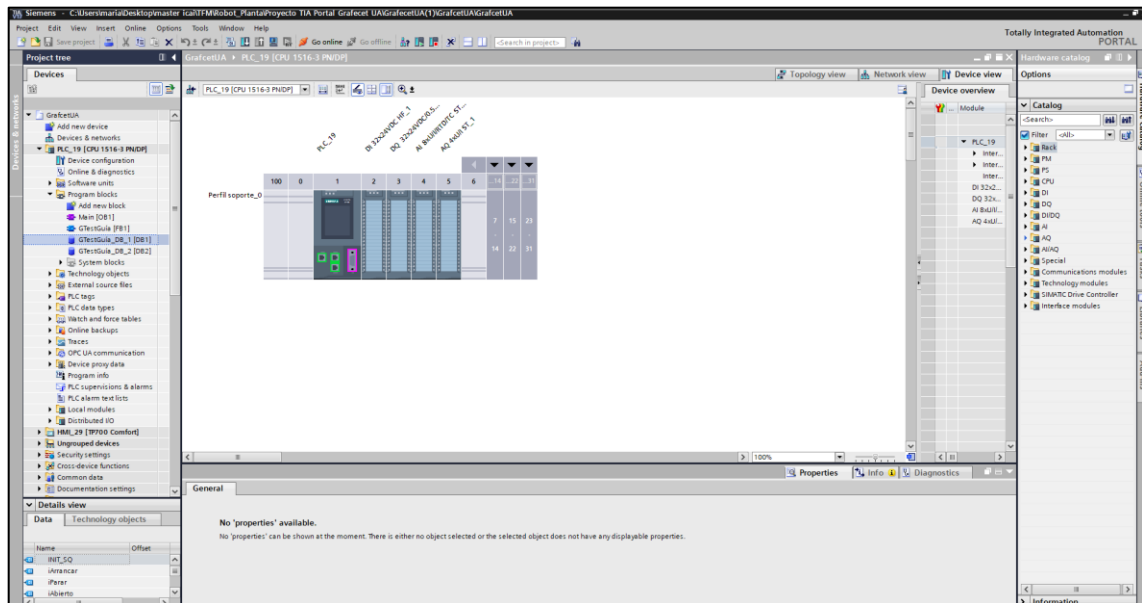


Ilustración 7. Entorno de TIA Portal- práctica Grafset

4.4.1 PLC VIRTUAL

En el proyecto usaremos además un PLC virtual “S7 PLCSIM Advanced V4”. Se trata de una herramienta que permite crear PLC virtuales, es decir, consigue simular el PLC físico en un entorno virtual, sin necesidad de tener acceso a un PLC físico o estar físicamente en el entorno de la máquina. Este PLC virtual es una solución proporcionada por Siemens que permitirá emular un PLC (Controlador lógico programable) proporcionando características similares a un PLC físico, como la capacidad de ejecutar ciclos de programa, interactuar con dispositivos de entrada y salida virtuales, y simular el comportamiento de una máquina o proceso.

Es importante destacar que el uso de este PLC virtual está relacionado con el caso ejemplo del proyecto. En otros contextos o con otros fabricantes, se utilizarían diferentes simuladores que emulan PLCs. Por ejemplo, fabricantes como *ABB*, *Schneider* o *Rockwell* podrían ofrecer simuladores específicos para sus sistemas de control, permitiendo realizar pruebas y desarrollo en ambientes virtuales antes de la implementación en sistemas reales.

El empleo de un PLC físico sería esencial en un entorno de producción real, pero el uso del PLC virtual ha sido particularmente beneficioso para la ejecución del proyecto a distancia, eliminando la necesidad de acceso físico a un PLC real o a la máquina en sí.

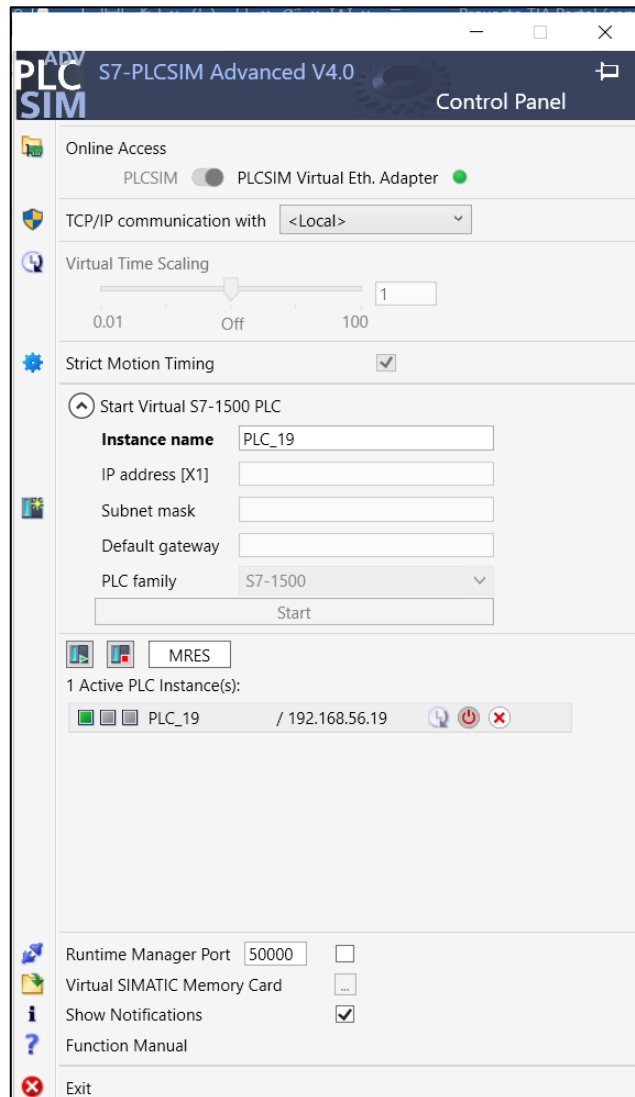


Ilustración 8. S7 PLCSIM Advanced V4

4.5 ***NODE-RED***

Node-Red es una herramienta de programación visual basada en flujos que permite conectar y automatizar dispositivos, servicios y sistemas de manera sencilla y rápida pues

proporciona una interfaz gráfica intuitiva en la que puedes arrastrar y soltar nodos para crear flujos de trabajo, lo que lo convierte en una opción popular para la integración de sistemas y la creación de aplicaciones IoT.

Node-red ofrece una interfaz gráfica visual y una amplia gama de nodos y librerías que permiten trabajar en un entorno basado en nodos. Se usará esta herramienta para poder leer los valores de las variables que se necesiten a través del proyecto de TIA Portal. Se ha de instalar node-red correctamente, en el **Anexo II** se encuentran los pasos que se han seguido para la instalación de la herramienta [ANEXO II: Instalación de Node-Red] para a continuación configurar los nodos necesarios y poder conectarlos al proyecto de TIA Portal. Se deberán definir los flujos y la lógica específica en Node-red para lograr la interacción deseada con las variables y generar las respuestas adecuadas.

Al describir la implementación de Node-RED en el contexto de nuestro caso ejemplo, es crucial destacar que estamos trabajando específicamente con el protocolo propietario **PUT/GET** de Siemens. Esta elección se alinea con el entorno del caso ejemplo que estamos siguiendo, donde la comunicación se establece de acuerdo con el protocolo original de la máquina. Por tanto, el uso de Node-RED se adapta a estas circunstancias particulares.

Este enfoque subraya cómo estamos integrando el protocolo PUT/GET de Siemens en nuestra implementación a través de Node-RED. Aunque Node-RED es una herramienta versátil y admite una variedad de protocolos de comunicación, estamos aplicando este conjunto de nodos para abordar el caso específico que hemos elegido. Esta decisión es esencial para comprender cómo estamos logrando la comunicación entre el sistema y el controlador.

4.6 OTRAS HERRAMIENTAS Y RECURSOS

En el desarrollo de este proyecto, además de las herramientas y tecnologías fundamentales que se han debatido a lo largo de este capítulo, también se ha dedicado tiempo

a explorar y utilizar otros recursos esenciales que han permitido mejorar la comprensión y orientación sobre OPC UA y su aplicación en el contexto de la Industria 4.0.

4.6.1 WEBINARS DE OPC UA

Los **webinars de OPC UA** son eventos en línea organizados por expertos y organizaciones que se centran en brindar información, compartir conocimientos y proporcionar orientación sobre temas relacionados con OPC UA. Podemos encontrar estos recursos en la propia página oficial de *Unified Automation* (<https://www.unified-automation.com/services/training/ua-webinar.html>).

Estos webinars han sido una oportunidad para profundizar en temas específicos relacionados con OPC UA, conocer las últimas tendencias y mejores prácticas, y aprender de expertos en el campo. Aunque no han sido el foco central del proyecto, han contribuido significativamente a nuestra formación y comprensión general del protocolo.

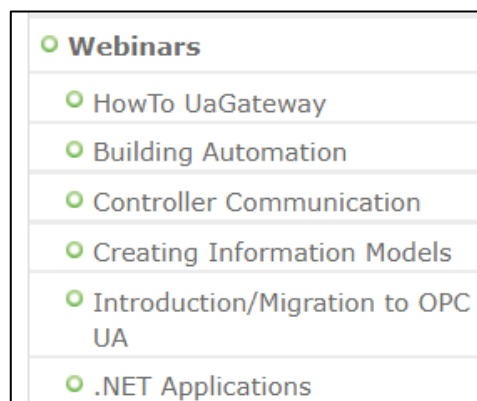


Ilustración 9. Webinars de OPC UA [18]

4.6.2 UA EXPERT

UA Expert es una herramienta de software desarrollada por *Unified Automation*, una empresa especializada en el desarrollo de soluciones y tecnologías relacionadas con la automatización industrial y la comunicación basada en OPC UA. Esta herramienta se utiliza para diagnosticar y visualizar sistemas que implementan el estándar de OPC UA. Con ella se pueden explorar, acceder a datos, monitorear y realizar pruebas en sistemas de OPC UA,

de manera que a lo largo del proyecto se ha podido verificar por medio de esta herramienta que el servidor y/o cliente responden correctamente cuando se envían solicitudes de escritura y lectura. [19]

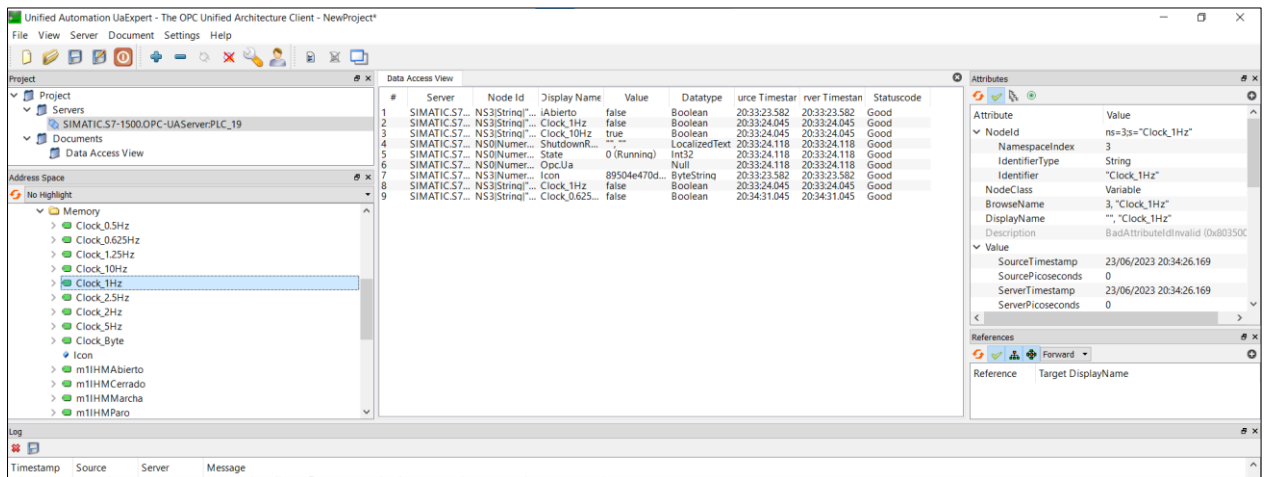


Ilustración 10. Entorno de UA Expert

UA Expert ha sido por tanto un recurso esencial para mejorar la comprensión del funcionamiento del sistema, y sobre todo para realizar pruebas conforme se iba avanzando en el proyecto.

4.6.3 GESTIÓN DE CERTIFICADOS: OPEN SSL

Es necesario utilizar y configurar un **certificado** en UA Expert para garantizar la seguridad de la comunicación entre los diferentes nodos de OPC UA. Sin la autenticación adecuada, cualquier persona podría hacerse pasar por el servidor y comprometer la integridad y privacidad de los datos. Por lo que la configuración de un certificado es fundamental para garantizar la seguridad de la comunicación. Para manejar los certificados y la encriptación en UA Expert, se tiene que usar una herramienta que genere el certificado, en este proyecto se ha usado **OpenSSL**.

OpenSSL es una herramienta de código abierto que contiene funciones útiles de criptografía para aplicaciones y sistemas operativos, con ella generaremos los certificados necesarios. Luego se configura el certificado generado en el servidor OPC UA, en este caso,

UA Expert. Se ha de configurar la encriptación, para habilitar mensajes utilizando TLS. La versión que se ha usado en este proyecto es “*OpenSSL 3.1.0*”.

```
C:\Users\maria>openssl version  
OpenSSL 3.1.0 14 Mar 2023 (Library: OpenSSL 3.1.0 14 Mar 2023)
```

Ilustración 11. Versión OpenSSL 3.1.0

En el **ANEXO III**: se encuentran los pasos que se han seguido para una correcta descarga de OpenSSL, instalación y configuración de los certificados.

Capítulo 5. DESARROLLO DE LA APLICACIÓN

En este capítulo se aborda el desarrollo de la aplicación para la digitalización de máquinas utilizando OPC UA. Se analizarán, por tanto, las diferentes funciones que realizará la aplicación y los distintos componentes y configuraciones necesarias para lograr una integración exitosa. Diferenciaremos por tanto entre la arquitectura funcional del sistema, es decir, su estructura y funcionamiento con diferentes roles, y la configuración necesaria para introducir las distintas señales de entrada y salida.

5.1 ARQUITECTURA FUNCIONAL

Entender la arquitectura funcional del sistema es clave para comprender cómo se estructura la aplicación, cómo funciona y cómo deben de interactuar los distintos componentes con sus respectivas funcionalidades para que funcione todo de manera correcta.

La arquitectura funcional está formada por tres componentes, **FP**, **FO** y **FC**. Es esencial distinguir los distintos roles y responsabilidades que ocupan, pues tendremos dos roles diferenciados, el del diseñador, encargado de configurar FO y FC, y el rol de desarrollador encargado de desarrollar la función de “hablar” con la planta (FP), es decir, la comunicación con la planta basada en la interfaz definida para FC. En el proyecto se pretende diseñar y desarrollar la aplicación final que permitirá al integrador añadir a una planta automatizada un servidor OPC UA, es decir, se desarrollan todas las partes que no van a cambiar de una planta a otra y se darán las pautas y metodología para las partes que sí cambian.

La arquitectura funcional se va a representar mediante la **Función de Hablar con la planta (FP)** representando la interacción con el controlador físico real y la **Función de Servidor OPC UA (FO)** representando el servidor OPC UA implementado en Visual

Studio. La conexión entre FP y FO es el enlace de comunicación entre la aplicación y el controlador físico real.

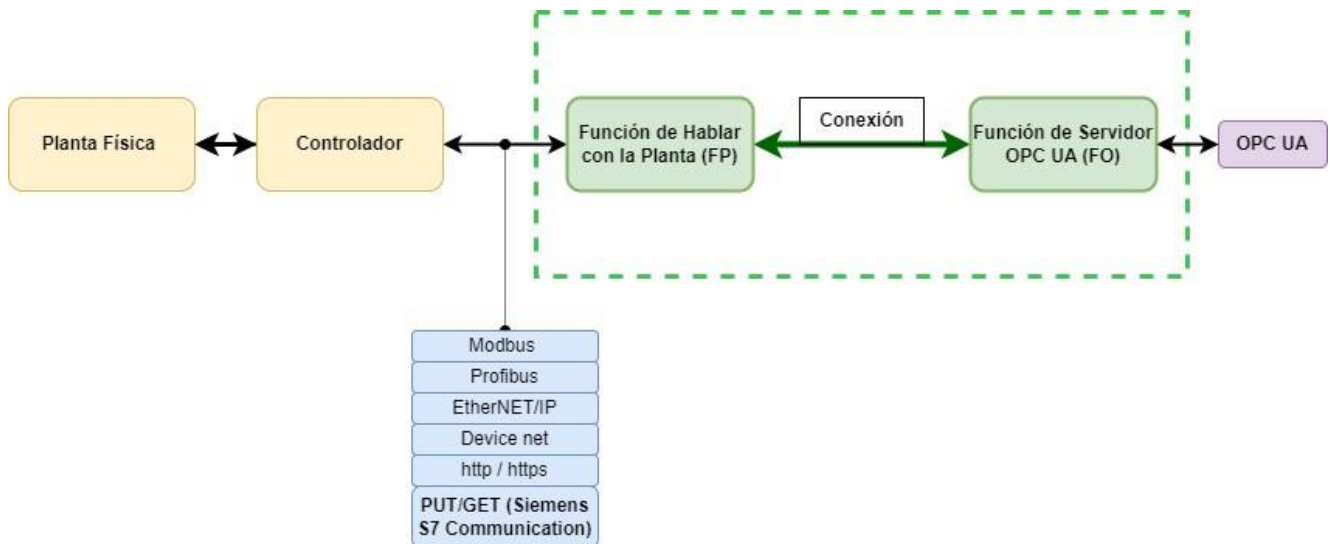


Ilustración 12. Arquitectura Funcional

El anterior esquema [Ilustración 12] muestra la comunicación de izquierda a derecha desde la máquina y el controlador que no hablan con OPC UA hasta la función FP, la conexión FP con FO y la función FO.

- **Planta física:** Representa la maquinaria o planta física con su respectivo controlador.
- **Controlador:** Este controlador no tiene capacidades nativas de comunicación con OPC UA por lo que utiliza su propio protocolo de comunicación. Estos protocolos propios del controlador hacen referencia a protocolos de comunicación específicos como por ejemplo pueden ser Modbus, Profibus, EtherNET/IP, DeviceNet, http, o PUT/GET. En particular la comunicación del ejemplo se basa en el protocolo propietario PUT/GET de Siemens S7, siendo este un protocolo que utiliza específicamente Siemens para permitir la interacción con sus controladores y transferencia de datos.
- **Función “FP”:** Representa la función de hablar con la planta, actúa como cliente OPC UA y se comunica con el controlador físico mediante su propio protocolo.

- **Función “FO”:** Representa la función de servidor OPC UA implementado en Visual Studio utilizando la librería *OPC UA .NET Standard*.

Este diagrama de flujo muestra además la **conexión** entre FP y FO. Estos tres últimos elementos son los que se desarrollan en el presente proyecto (señalados mediante un recuadro con líneas discontinuas de la *Ilustración 12*). Además, notar que las flechas del flujo han de ser **bidireccionales**, lo que significa que la comunicación puede ocurrir en ambas direcciones, permitiendo tanto la escritura como la lectura de datos entre máquina, controlador, y funciones.

5.1.1 FUNCIÓN DE HABLAR CON LA PLANTA (FP)

Llamaremos **FP** a la función de comunicación con la planta. Esta función se refiere a la capa o módulo encargado de interactuar directamente con la máquina física (PLC) o planta. El objetivo principal será recopilar datos de los sensores y actuadores de la máquina, así como enviar comandos y recibir respuestas de esta.

Distinguimos las responsabilidades de cada rol. El diseñador será responsable por tanto de la comprender los requisitos específicos de la planta y definir la interfaz de comunicación requerida, además deberá proporcionar una especificación clara y detallada de los datos de entrada y salida que se van a manejar en FP. Por el contrario, nuestro rol, se ocupará de implementar la Función de planta (FP) basándose en la interfaz definida por el diseñador. Debiendo utilizar la especificación proporcionada y desarrollar la metodología necesaria y el código para establecer la comunicación bidireccional con FO.

Esta función será controlada por el controlador, en nuestro caso, el PLC. FP se comunica con el controlador mediante el cliente, el cual establece comunicación con el servidor OPC UA implementado en Visual Studio. Esto permite acceder a los datos y controlar la planta o máquina de manera efectiva. En este caso en particular, para establecer la comunicación con el PLC, se hará uso del software de programación *TIA Portal*, no obstante, en otros casos podrían utilizarse otras plataformas de otros fabricantes como se ha comentado anteriormente (ABB; Schneider, Rockwell...).

5.1.2 FUNCIÓN DE SERVIDOR OPC UA (FO)

Llamaremos **FO** a la función del servidor OPC UA. Esta función se refiere al servidor OPC UA que actúa como intermediario entre la planta y los sistemas externos. El objetivo principal será por tanto proporcionar una interfaz estándar y segura para la comunicación entre la planta y otros dispositivos o sistemas.

En este caso, el diseñador será el responsable de adquirir y configurar el servidor OPC UA adecuado para la planta. Esto incluye, la instalación y configuración del servidor con la estructura de la planta. Por otro lado, nuestro rol será el de implementar la herramienta necesaria para configurar FO según la planta, es decir, utilizar el servidor OPC UA configurado por el diseñador y conseguir establecer la comunicación bidireccional entre FO y FP asegurándome que los datos se intercambien correctamente.

La función FO representa el Servidor OPC UA implementado en Visual Studio con el uso de la librería *OPC UA .NET Standard*. Este servidor proporciona los datos y servicios a los clientes que se conectan a él. El servidor en este caso es el responsable de gestionar el *Address Space*, y la configuración de nodos correspondientes a las señales.

5.1.3 CONEXIÓN ENTRE FUNCIÓN DE PLANTA (FP) Y FUNCIÓN DEL SERVIDOR OPC UA (FO)- FC

A esta función de conexión entre planta y servidor OPC UA lo denominaremos **FC**, este funcionará como un sistema intermedio ya que su función, como se indica se refiere a la conexión o interfaz entre la función de hablar con la planta (FP) y la función del servidor OPC UA (FO). El objetivo principal será por tanto establecer la comunicación bidireccional entre la planta (FP) y el servidor OPC UA (FO), siendo esta comunicación esencial para establecer la transferencia de datos. Esta conexión implica la configuración de los puntos de acceso y nodos del *Address Space* del servidor OPC UA para reflejar la estructura y datos de la planta.

En este caso, el rol del diseñador se encargará de adquirir y configurar FC, esto incluirá por tanto establecer reglas de mapeo de datos, asegurar la sincronización adecuada

y gestionar cualquier transformación de datos. Nuestro rol como desarrolladores será, implementar la conexión de FP y FO.

La conexión entre el PLCSIM y **FP** en el caso ejemplo simula la comunicación con la planta digitalizada.

Se establece una conexión entre **Node-RED** y el Servidor OPC UA de Visual Studio permitiendo así acceder a los datos proporcionados por el servidor y visualizar la información de la planta o máquina en tiempo real. Node-RED se utiliza para configurar y desarrollar los nodos y flujos que permiten la conexión y el intercambio de información con el controlador, utilizando el protocolo propietario PUT/GET de Siemens. Esta conexión se realiza a través de la implementación de un flujo en Node-RED que interactúa con el servidor. Posteriormente se indica cómo se crea este flujo de nodos “s7in” de la librería de “node-red-contrib-s7” para realizar la lectura de variables.

5.2 ESQUEMA DE COMUNICACIÓN OPC UA

OPC UA es el estándar que hoy día la industria elige para la comunicación entre la máquina y los sistemas de gestión. Pues como se ha ido comentando, es un protocolo diseñado para atraer datos del mundo industrial al mundo IT. Los datos pueden pasar por firewall o conectarse con un PLC directamente facilitando además el envío de datos entre cliente y servidor. [20]

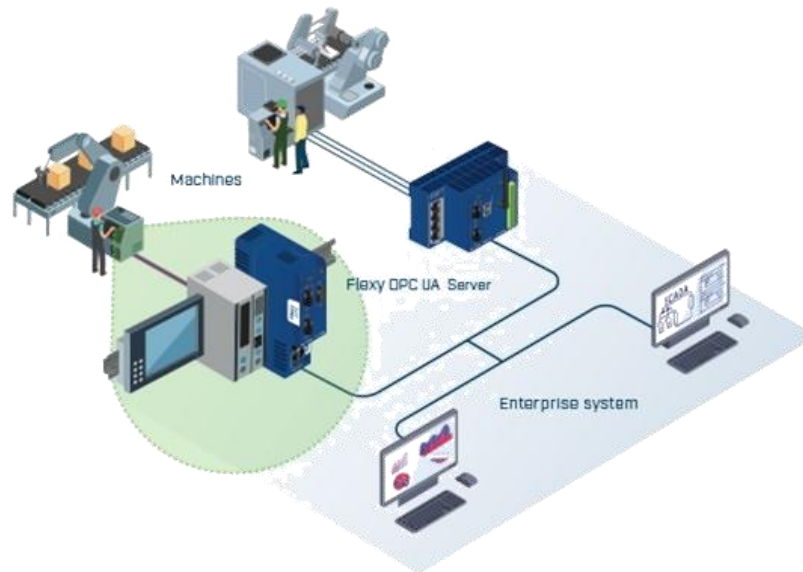


Ilustración 13. Esquema de comunicación OPC-UA [22]

En la *Ilustración 13* podemos ver un esquema genérico del funcionamiento de OPC UA, de manera que las máquinas, se encuentran equipadas con un controlador, nosotros utilizaremos en este caso un PLC, el simulador **S7 PLCSIM**. El PLC funcionará como servidor y será el que se encargue de acceder a los datos y funcionalidades del servidor mediante el software de simulación **TIA Portal**. El **servidor** OPC UA actúa como intermediario entre las máquinas y el sistema empresarial. Recopila y gestiona los datos que recoge el PLC y los presenta en una estructura unificada, el *Address Space*. También puede proporcionar información a las diferentes máquinas y recibir comandos de ellas a través de OPC UA.

El **cliente** es el que realizará las solicitudes al servidor para leer datos de las variables, escribir en ellas, acceder a objetos, suscribirse a eventos y alarmes, entre otras funcionalidades. En este caso, elegimos **Node-RED** para actuar como cliente, ya que nos permite conectar diferentes fuentes de datos y servicios, incluidos los servidores OPC UA, para crear flujos de trabajo automatizados.

Por último, acorde con la Ilustración 13 el **sistema empresarial** es el que se va a comunicar con el servidor OPC UA para obtener información en tiempo real y datos de producción de las máquinas.

5.3 PUESTA EN MARCHA Y VALIDACIÓN DE BIBLIOTECAS OPC UA CON UA EXPERT

Para poner en marcha y validar las bibliotecas OPC UA, comenzaremos implementando una comunicación cliente-servidor utilizando **UA Expert**. Esta comunicación cliente-servidor es un concepto esencial en la implementación de sistemas basados en OPC UA, ya que es el modelo fundamental de interacción entre dispositivos y sistemas en esta arquitectura. Para el diseño del cliente y servidor del sistema se hará uso del repositorio “OPC UA .NET Standard Sample master” de GitHub, más específicamente de la solución “**UA Sample Applications.sln**” explicada en el *apartado 4.3.1* Incluye varios proyectos, a continuación, ponemos en marcha los proyectos “**UA Sample Client**” y “**UA Sample Server**”

5.3.1 UA SAMPLE CLIENT

Para el diseño del cliente del proyecto usando UA Expert se ha hecho uso del ejemplo “*UA Sample Client*” del repositorio “UA .NET Standard Sample”. Para poder acceder a los datos de la planta o máquina que se quiere digitalizar se han de seguir unos pasos que se comentan a continuación:

1. **Configuración del cliente OPC UA:** Estableciendo la dirección IP y el puerto del servidor OPC UA al que se conectará el cliente
2. **Conexión al servidor OPC UA:** Iniciar la conexión con el servidor utilizando las configuraciones adecuadas.
3. **Explorar y acceder a los nodos:** esto permitirá conocer la estructura y jerarquía de la planta.

Al ejecutar “UA Sample Client” desde Visual Studio se abrirán dos ventanas emergentes, una ventana de “UA Expert” y otra del cliente de OPC Foundation.

“UA Expert” es la herramienta que permite a los clientes OPC UA descubrir los servidores OPC UA disponibles en la red, mostrando los detalles de conexión con el servidor OPC UA.

Cuando el ‘**OPC UA Samples Client**’ se ejecuta por primera vez, intentará descubrir los servidores OPC UA disponibles en la red mediante el uso de *Discovery Server*. Si encuentra algún servidor, se mostrará en dicha ventana y se podrá seleccionar para conectarse. Podremos por tanto interactuar con el servidor a través de la aplicación cliente. La ventana de aplicación cliente, lo que muestra es la estructura del servidor, de esta manera se permitirá al usuario explorar los nodos, visualizar los valores de dichos nodos e incluso realizar operaciones como escribir nuevos valores de los datos, realizar cambios etc.

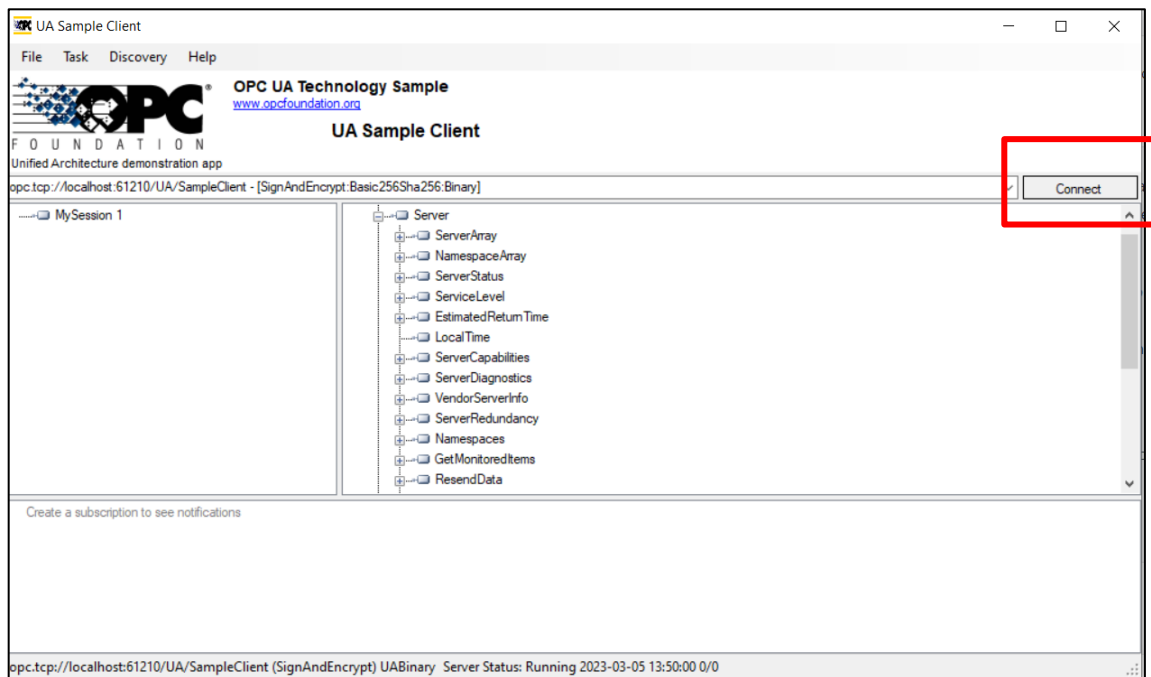


Ilustración 14. OPC UA Sample Client

En la anterior imagen, *Ilustración 14*, vemos la ventana del cliente de OPC Foundation, donde los nodos se refieren a los objetos o variables que están disponibles en el

servidor OPC UA al que se ha conectado el cliente. Los nodos se organizan en una jerarquía de nodos, similar a un árbol, en la que cada nodo puede tener nodos secundarios.

En general, los nodos se dividen en dos categorías: los nodos de objetos y los nodos de variables. Donde los **nodos de objetos** representan objetos en el servidor, como dispositivos físicos o áreas funcionales, éstos también pueden contener más nodos secundarios. Por otra parte, los **nodos de variables** representan valores que pueden ser medidos o calculados y que contienen la información actual de la variable, como su valor, tipo de dato, rango y unidad de medida.

Además, en la *Ilustración 14*, al darle a “Connect” podremos conectar con UA Expert configurando el puerto, que se indica en la ventana (opc.tcp://localhost:61210).

Por otra parte, conectamos con “UA Expert” mediante “Discovery Server” para poder establecer la comunicación y poder así leer los valores de las distintas variables.

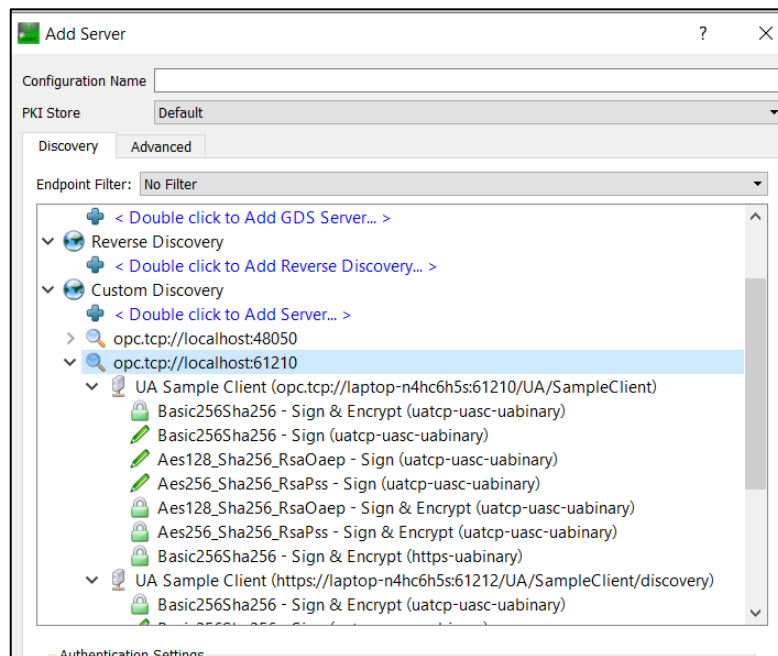


Ilustración 15. Añadir servidor en UA Expert

Una vez que se establece la conexión podremos navegar en el árbol de la derecha de UA Expert, el *Address Space*, siendo este un espacio donde aparecen los nodos y objetos

disponibles en un servidor OPC UA. Es una estructura que muestra cómo los datos y funcionalidades están organizados y disponibles para ser accedidos por los clientes que se conectan al servidor OPC UA.

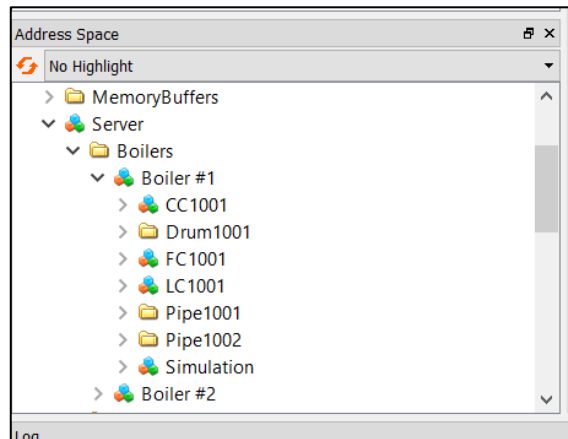


Ilustración 16. Address Space (UA Expert)

Podemos comprobar la correcta conexión ya que en ambas ventanas coinciden los nodos y objetos de manera que podremos acceder a los valores de las variables e interactuar con ellas de una manera visual.

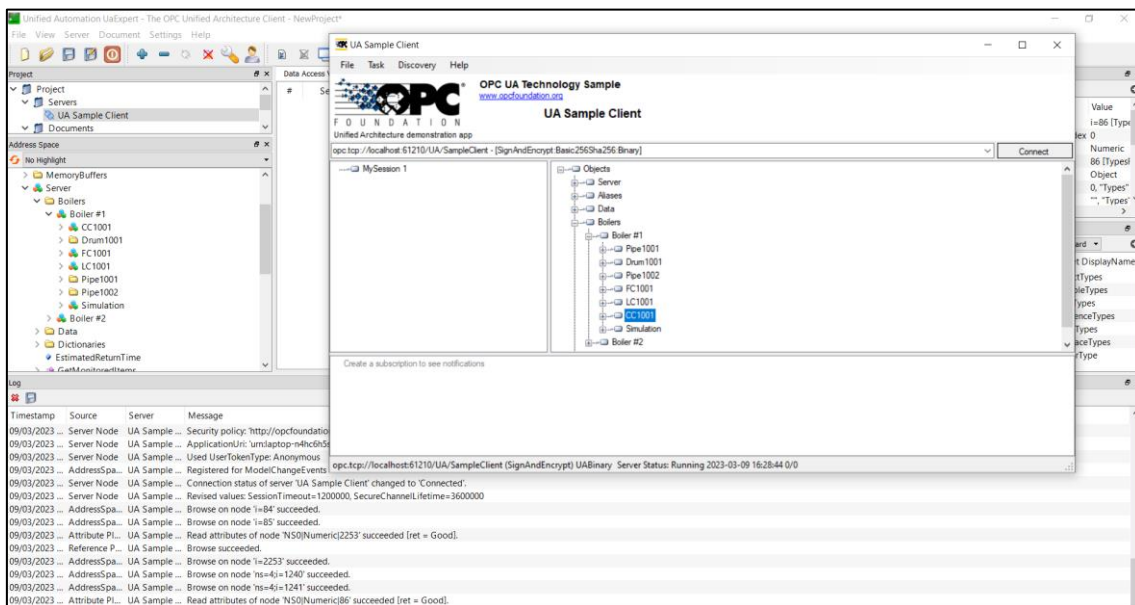


Ilustración 17. Conexión entre cliente-servidor

5.3.2 UA SAMPLE SERVER

El diseño del servidor para establecer una comunicación mediante **UA Expert** es similar. Se hace uso del ejemplo “*UA Sample Server*” del repositorio “UA .NET Standard Sample” para poder acceder a los datos de la planta o máquina que se quiere utilizar. Para una correcta configuración del servidor, se siguen los siguientes pasos:

1. Seleccionar la plataforma de desarrollo adecuada, en este caso se elige **Visual Studio**.
2. Elegir una librería OPC UA, pues existen muchas como se ha visto. La **librería OPC UA .NET Standard** es la que usaremos.
3. Configurar el **Address Space** adecuadamente, con una estructura y jerarquía de datos.
4. Definir las variables y objetos que el servidor proporcionará a los clientes OPC UA.
5. Implementar el servidor.

Se hará uso por tanto del proyecto muestra “UA Sample Server” desde Visual Studio. Tras su ejecución se abrirá la ventana emergente de la siguiente imagen.

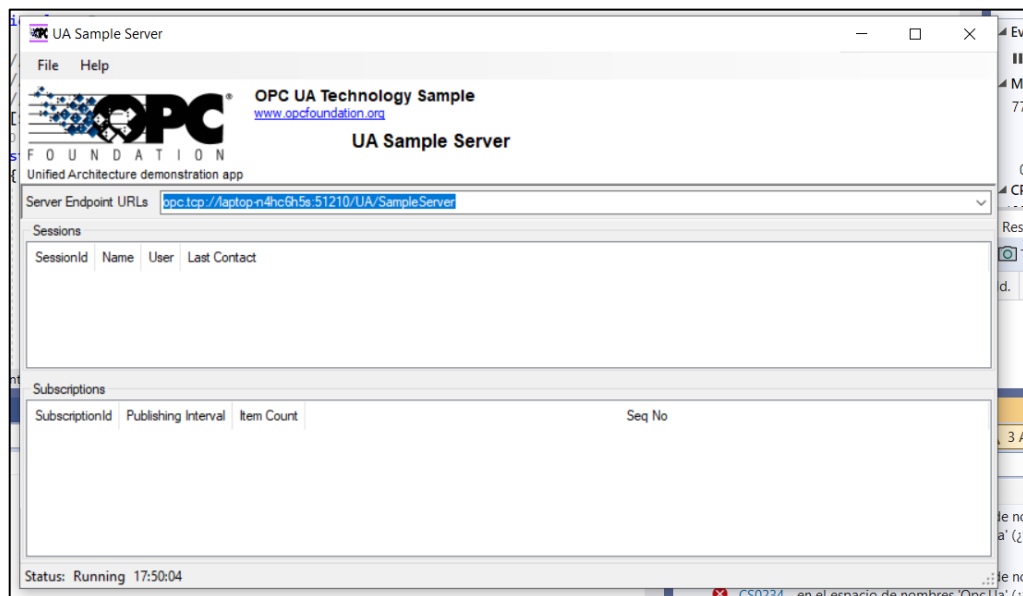


Ilustración 18. OPC UA Sample Server

De manera que habrá que hacer una configuración similar a la que se ha realizado con “UA Sample Client”, añadiendo el servidor en “UA Expert” y configurando el puerto que se indica establecemos la conexión.

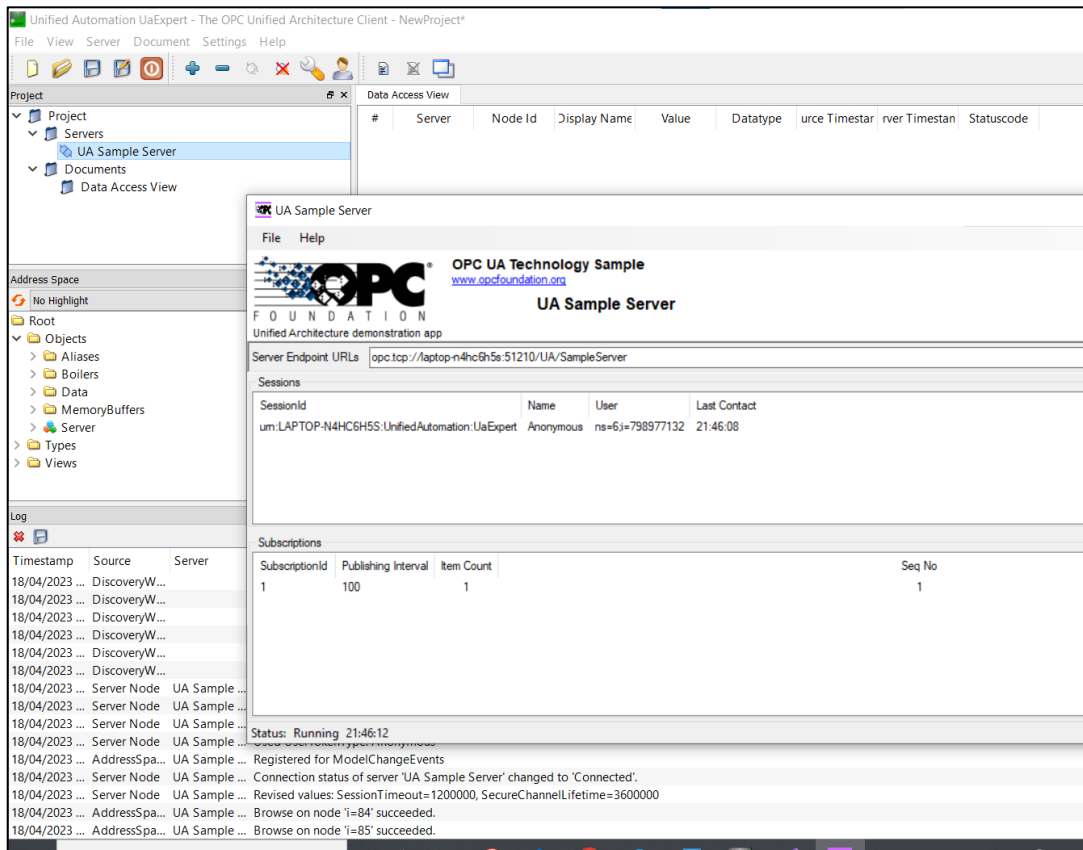


Ilustración 19. Conexión establecida entre cliente-servidor (UA Expert)

5.4 IMPLEMENTACIÓN DE FO - VISUAL STUDIO

Después de comprobar la comunicación y asegurar el correcto funcionamiento utilizando la herramienta de UA Expert, se decide desarrollar el servidor OPC UA utilizando **Visual Studio** con la librería “**OPC UA .NET Standard**”. De esta manera se tiene un control más completo de la implementación del servidor y se aprovecharán las funcionalidades avanzadas que ofrece dicha librería ya que se trabaja directamente con el código desarrollado en Visual Studio.

5.4.1 CONFIGURACIÓN DE SEÑALES (ENTRADAS Y SALIDAS) A UTILIZAR VÍA OPC UA

Primero que todo, para poder definir señales de entrada y salida en un servidor de OPC UA, es necesario modificar el *Address Space* del servidor. El *Address Space* es una representación lógica de los sistemas en el servidor OPC UA y se organiza en objetos y nodos. Para añadir entradas y salidas al servidor OPC UA habrá que **modificar** el código del servidor para agregar los nodos correspondientes al *Address Space*.

5.4.1.1 Modificar el *Address Space*

Para poder modificar el *Address Space* nos encontramos con diversos **problemas** de implementación. Pues el proyecto del que se está haciendo uso “UA Sample Applications.sln” tiene asociados 6 proyectos en su interior, muchos archivos y por tanto muchas dependencias de diferentes variables, objetos y nodos entre cada proyecto. Al cambiar cualquier valor de una variable, o intentar introducir un nuevo objeto en el código, es necesario cambiarlo en todos los archivos por igual, siendo esto un proceso muy tedioso y casi imposible de realizar sin tener errores de compilación.

La solución ante este problema se ha encontrado gracias a una herramienta proporcionada por la OPC Foundation en su propio repositorio de GitHub “UA – **Model Compiler**”.

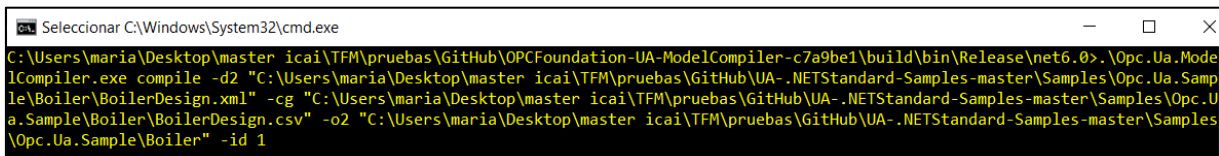
5.4.1.1.1 UA – **Model Compiler**

Un componente crítico en nuestro proyecto es el **UA- Model Compiler**, esta herramienta desempeña un papel fundamental al permitirnos llevar a cabo las modificaciones necesarias en los modelos de información pues va a ser la solución para poder realizar los cambios pertinentes que queremos realizar. Esta es una herramienta que se utiliza para compilar modelos de información en el formato OPC UA en lenguajes de programación específicos, de manera que nos va a permitir adaptar y personalizar el modelo de información que usamos en la implementación del proyecto.

En nuestro caso ejemplo, el modelo de información de OPC UA define una estructura determinada, haciendo uso de atributos, objetos y datos que se pueden intercambiar a través de comunicación por OPC UA. Estos modelos se definen utilizando el lenguaje de descripción de información propio de OPC UA llamado “*OPC UA Information Modeling Language, UANodeSet*” en un **archivo XML**.

Por lo tanto, pasamos a instalar el compilador “**UA- Model Compiler**” a través del GitHub oficial de la fundación OPC UA (<https://github.com/OPCFoundation/UA-ModelCompiler/tree/master>). El “**UA-ModelCompiler**” toma el archivo XML antes mencionado que describe el modelo de información y lo compila en clases y bibliotecas de código fuente en diferentes lenguajes de programación, en este caso, nos sirve con el lenguaje que estamos utilizando C#. Este compilador nos facilita por tanto la generación de código para la implementación de la aplicación cliente-servidor OPC UA que trabaja con objetos y datos definidos en el modelo de información. Para llevar a cabo esta compilación se siguen los siguientes pasos descritos:

1. Descargar el repositorio directamente desde GitHub en el entorno de desarrollo local en el que estamos trabajando.
2. Abrir el proyecto denominado “*Model Compiler Solutions.sln*” y lo compilamos. Este proceso solo se tendrá que realizar una única vez.
3. Tras completar la compilación sin errores, obtenemos el archivo binario que necesitamos, el cual estará disponible en la carpeta de salida **bin/Release**, desde donde abriremos una terminal de comandos y ejecutamos el archivo “**OPC.UA.Model Compiler.exe**” haciendo uso de la siguiente línea de comandos que se muestra en la *Ilustración 20*.



```
Seleccionar C:\Windows\System32\cmd.exe
C:\Users\maria\Desktop\master icai\TFM\pruebas\GitHub\OPCFoundation-UA-ModelCompiler-c7a9be1\build\bin\Release\net6.0>. \OpC.Ua.ModelCompiler.exe compile -d2 "C:\Users\maria\Desktop\master icai\TFM\pruebas\GitHub\UA-.NETStandard-Samples-master\Samples\OpC.Ua.Sample\Boiler\BoilerDesign.xml" -cg "C:\Users\maria\Desktop\master icai\TFM\pruebas\GitHub\UA-.NETStandard-Samples-master\Samples\OpC.Ua.Sample\Boiler\BoilerDesign.csv" -o2 "C:\Users\maria\Desktop\master icai\TFM\pruebas\GitHub\UA-.NETStandard-Samples-master\Samples\OpC.Ua.Sample\Boiler" -id 1
```

Ilustración 20. Ejecución OPC.UA Compiler.exe

Para entender qué se ha realizado en la anterior línea de comandos se adjunta la siguiente tabla informativa de cada parámetro utilizado [Tabla 2].

-d2	Indica la ruta del archivo “ModelDesign” o “NodeSet2” que contiene el modelo de información de la UA. EL primer archivo especificado es el modelo a generar. El resto son modelos incluidos. La ruta del archivo puede ser seguida del prefijo del espacio de nombre y un nombre corto para el modelo. Las comas separan cada campo
-cg	Crea el archivo identificador si no existe
-o2	El directorio de salida para los archivos generados
-id 1	El primer identificador que se utilizará para asignar los nuevos ids a los nodos

Tabla 2. Tabla informativa de parámetros utilizados

- Tras utilizar este comando con los parámetros indicados en la Tabla 2, el ejecutable toma un archivo OPC UA “*ModelDesign*” y genera un *NodeSet* y código para la pila **.NETStandard**.
- Ya se pueden realizar los cambios pertinentes en el proyecto de “*UA Sample Applications.sln*”, como crear nuevos nodos. Añadimos las señales de entrada y salida que queremos en el *BoilerDesign.xml*

Mostraremos como se realiza un ejemplo de cambio. Se muestra cómo se han añadido las distintas señales del robot2 de la minifábrica en el *BoilerDesign.xml*.

```
<opc:Object SymbolicName="Minifabrica_robot2" TypeDefinition="ua:FolderType" SupportsEvents="true">
  <opc:Children>
    <opc:Object SymbolicName="Inputs" TypeDefinition="BoilerType">
      <opc:BrowseName>Inputs</opc:BrowseName>
      <opc:Children>
        <opc:Object SymbolicName="I_C2_STOPH">
          <opc:DisplayName>I_C2_STOPH</opc:DisplayName>
        </opc:Object>
        <opc:Object SymbolicName="I_C2_STOPS">
          <opc:DisplayName>I_C2_STOPS</opc:DisplayName>
        </opc:Object>
        <opc:Object SymbolicName="I_S2_PSTOP">
          <opc:DisplayName>I_S2_PSTOP</opc:DisplayName>
        </opc:Object>
        <opc:Object SymbolicName="I_S2_STOP">
          <opc:DisplayName>I_S2_STOP</opc:DisplayName>
        </opc:Object>
      </opc:Children>
    </opc:Object>
  </opc:Children>
</opc:Object>
```

Ilustración 21. Código en Visual Studio: Señales Minifábrica- Robot 2 (*BoilerDesign.xml*)

Podemos comprobar que se ha añadido el nodo que hemos cambiado en el **Address Space** [¡Error! No se encuentra el origen de la referencia.].

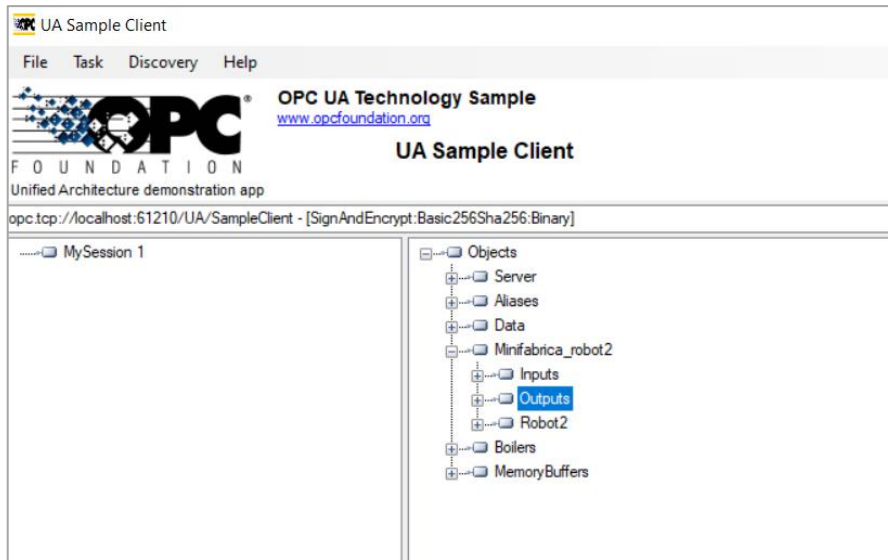


Ilustración 22: Señales Inputs y Outputs Minifábrica - Robot 2

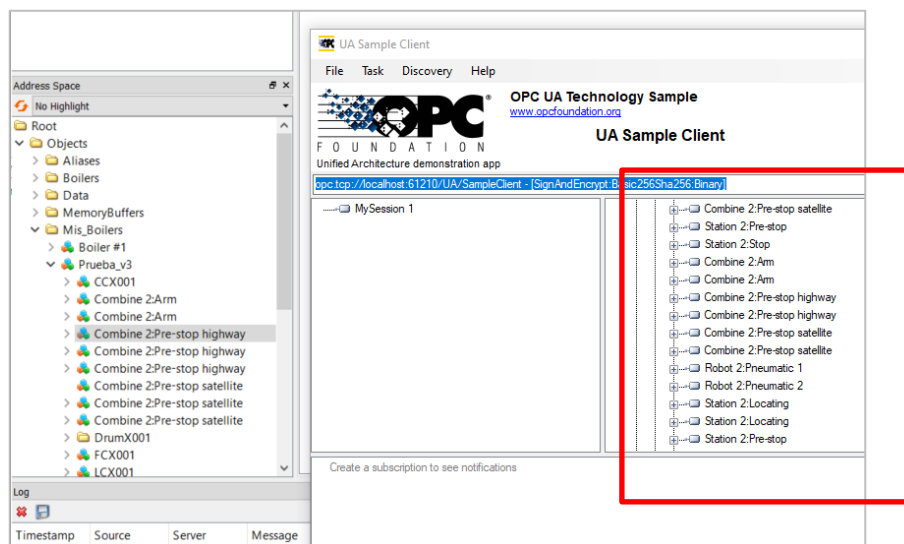


Ilustración 23: Señales entrada Minifábrica - Robot 2 (Address Space)

De esta manera comprobamos que se ha conseguido modificar el *Address Space*, pues el árbol de nodos se ha modificado correctamente.

5.5 DESARROLLO DE LA FUNCIÓN DE HABLAR CON LA PLANTA (FP)

En el contexto del desarrollo de la “Función de Hablar con la Planta” (FP) para un PLC de Siemens que permite acceso externo vía HTTP, se ha utilizado Node-RED como herramienta cliente para interactuar con el controlador. En el **ANEXO II: Instalación de Node-Red** se explican los pasos a seguir para una correcta instalación de la herramienta.

Se aborda la configuración y desarrollo de nodos y flujos con Node-RED que posibilita establecer la conexión entre el controlador y realizar tanto lectura como escritura de los datos de manera eficiente.

5.5.1 CONFIGURACIÓN Y DESARROLLO CON NODE RED: “NODE-RED-CONTRIB-S7”

Se va a hacer uso de **Node-RED** como cliente para interactuar con la planta o máquina que tiene el controlador. Una vez que podemos acceder a la interfaz gráfica de Node-RED se pasa a la configuración específica de nodos y flujos para poder establecer la conexión entre el controlador y poder realizar la lectura y escritura de datos.

Para la configuración se hará uso de la librería “**node-red-contrib-s7**” que permite la comunicación con controladores Siemens S7 a través del protocolo de comunicación S7, basándose en la comunicación **Ethernet**.

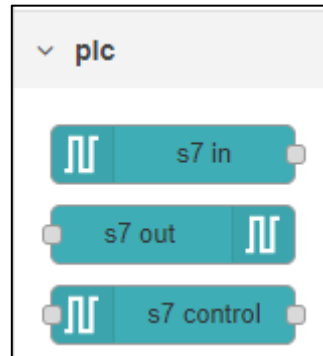


Ilustración 24. Nodos de la librería "Node-red-contrib-s7"

Podemos ver en la *Ilustración 24* los nodos que se utilizan en la librería de “*node-red-contrib-s7*”. A continuación, se expone una tabla con una breve explicación de cada nodo.

<i>Nodo</i>	<i>Función</i>
S7 in	Este nodo se utiliza para leer (nodo de lectura) datos del PLC. Permite establecer una conexión con el controlador y configurar la dirección de las variables que desees leer.
S7 out	Este nodo se utiliza para escribir (nodo de escritura) datos en el PLC. También establece una conexión con el controlador y te permite enviar información para modificar las variables en el PLC.
S7 control	Este nodo es responsable de la gestión de la conexión con el PLC. Permite establecer los parámetros de comunicación, como la dirección IP del controlador y el tiempo de espera.

Tabla 3. Función nodos - librería "Node-red-contrib-s7"

Las especificaciones de dicha librería se pueden encontrar en: <https://flows.nodered.org/node/node-red-contrib-s7>.

Algunas características y ventajas del uso de la librería “*node-red-contrib-s7*” en el presente proyecto:

- Compatible con controladores Siemens S7, permitiendo la comunicación utilizando el control virtual S7 PLCSIM Advanced que se usa en el proyecto
- Es fácil de usar, se pueden crear flujos agregando nodos específicos para poder comunicarnos con el controlador.
- La librería soporta tanto lectura como escritura de datos desde y hacia el controlador. Esto es esencial para el proyecto, ya que se necesita una comunicación bidireccional.
- Node-red actúa como cliente en la comunicación cliente-servidor con OPC UA, gracias a node-red-contrib-s7 se podrán obtener datos y enviarlos al servidor implementado en Visual Studio.
- La librería ofrece flexibilidad para adaptar y personalizar el flujo de trabajo de acuerdo con las necesidades específicas de comunicación.

5.5.2 CONFIGURACIÓN ADICIONAL (S7-1200/1500)

Siguiendo las especificaciones de la librería *node-red-contrib-s7* son necesarios algunos pasos de configuración adicionales en el PLC, ya que los PLC nuevos ofrecen una versión “extendida” del protocolo S7, pero nosotros solo tenemos la versión básica del mismo.

- Hace falta deshabilitar el "*acceso a bloque optimizado*" para las bases de datos a las que queremos acceder.

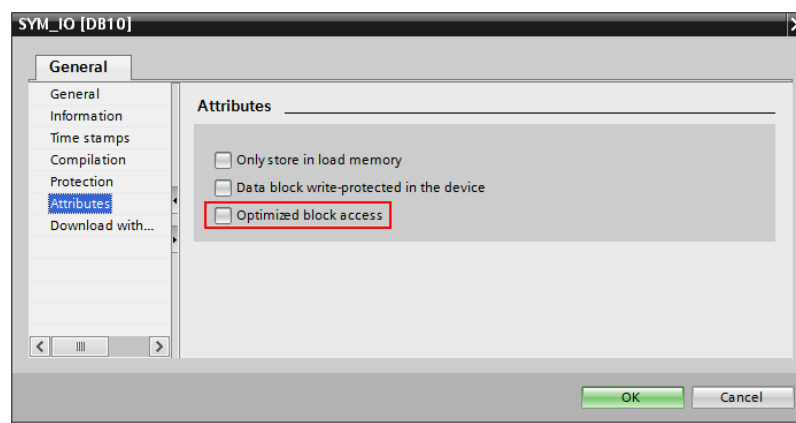


Ilustración 25. Configuración "node-red-contrib-S7" (1)

- En la sección "*Protección*" de las Propiedades de la CPU, hay que activar la casilla de verificación "*Permitir acceso con PUT/GET*"

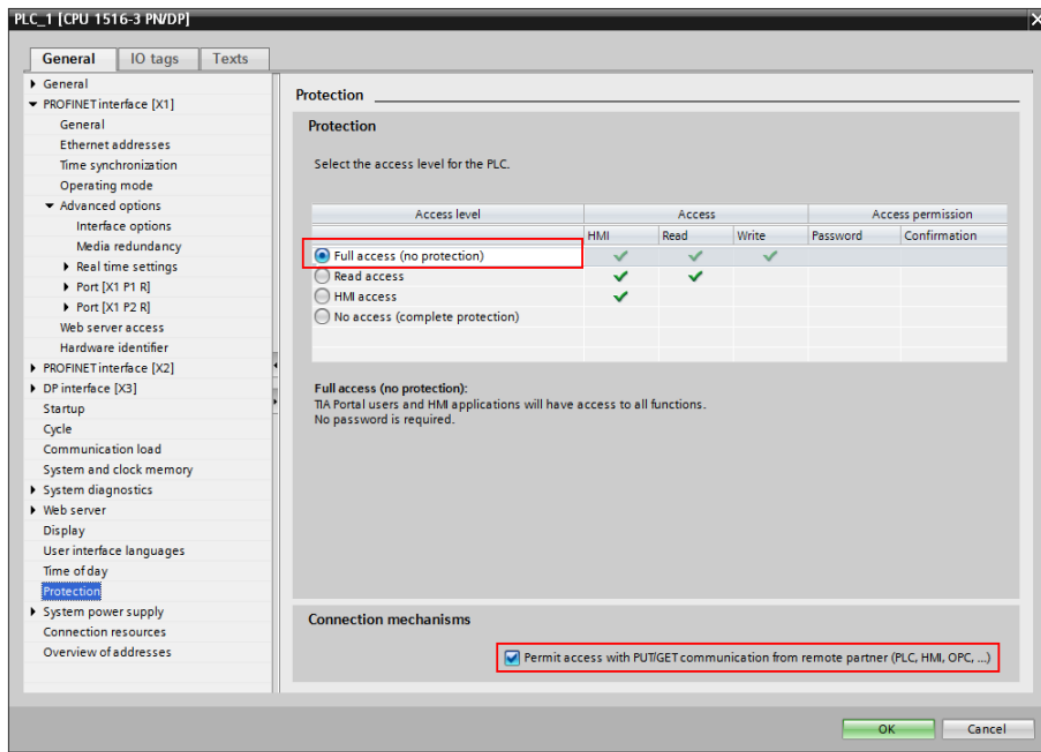


Ilustración 26. Configuración "node-red-contrib-S7" (2)

5.5.3 LECTURA DE DATOS – COMUNICACIÓN S7

En este apartado se explica el uso de la librería “*node-red-contrib-s7*” en Node-RED, para la lectura de datos del proyecto “*Grafcet*”. Esta integración de comunicación S7 -IIoT (*Industrial Internet of Things*) nos permitirá obtener información en tiempo real desde el controlador en el entorno de desarrollo. Veremos cómo configurar el nodo “S7 endpoint” para establecer la comunicación con el PLC virtual, así como el nodo “Read” para leer los datos de una variable específica.

Esta tabla contiene todas las variables y datos relevantes para el funcionamiento del controlador, y nos permitirá la lectura de datos. El esquema que se crea en Node-RED de ejemplo es para la lectura de la variable “*Clock_1Hz*” de la tabla de “*PLC Tags*”. Se hace uso del nodo “*s7 in*” quedando el siguiente esquema a modo ejemplo de lectura de variable.

Un flujo muy básico para poder leer una variable “*Clock*” en Node-RED puede ser el siguiente.

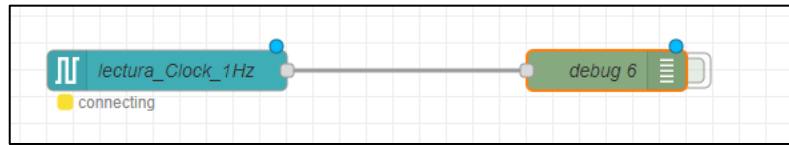


Ilustración 27. Esquema flujos Node-RED: lectura de variable "Clock_1Hz"

Para crear el esquema de flujos en Node-RED, simplemente se tiene que arrastrar el nodo desde la paleta de nodos al flujo.

Lo primero será crear un nuevo *endpoint* y configurarlo. En Node-RED, el *endpoint* se refiere a una URL única que identifica un recurso específico. Es la dirección a la que se pueden enviar peticiones HTTP para interactuar con los flujos. Cada nodo o conjunto de nodos se configura con un *endpoint* específico.

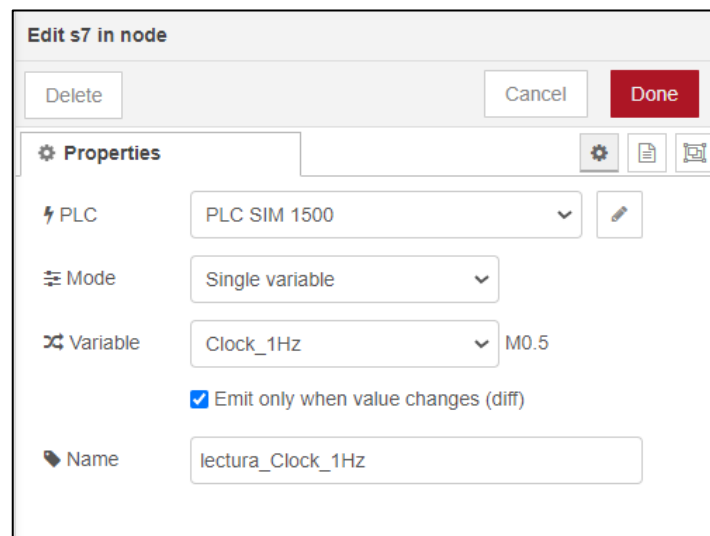


Ilustración 28. Configuración nodo "s7 in"

En la Ilustración 28 se indica el nombre que queremos ponerle al endpoint, seleccionamos si se quieren leer una o varias variables a la vez. También podemos seleccionar la opción "Emit only when value changes (diff)" que quiere decir que solo se va a realizar la lectura de la variable únicamente cuando el valor de dicha variable cambie.

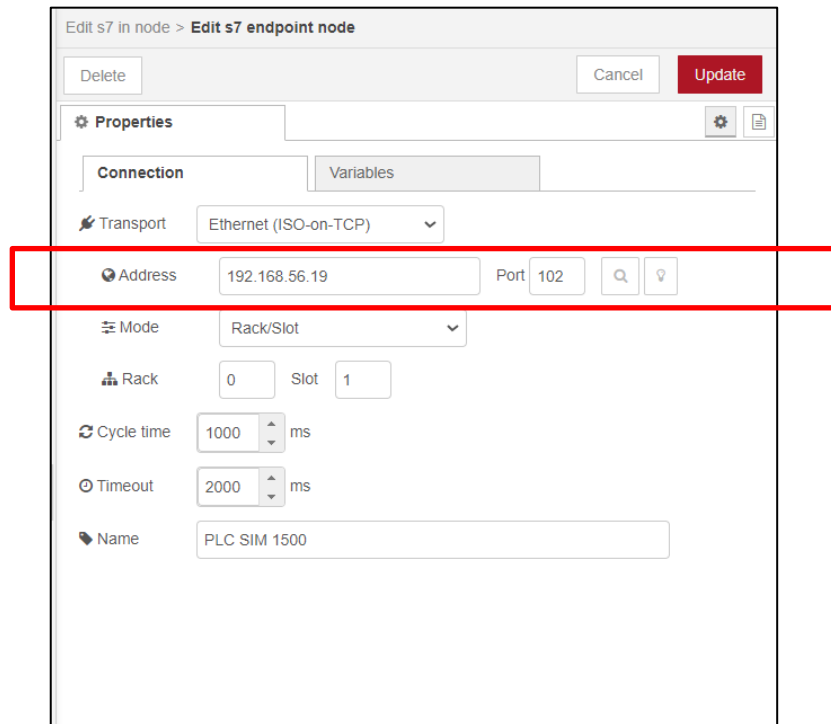


Ilustración 29. Configuración endpoint

La comunicación se realiza mediante **Ethernet**, y la **dirección IP** a la que tenemos que conectar la encontraremos en el proyecto de TIA Portal [Ilustración 30]. El puerto 102 se deja por defecto.

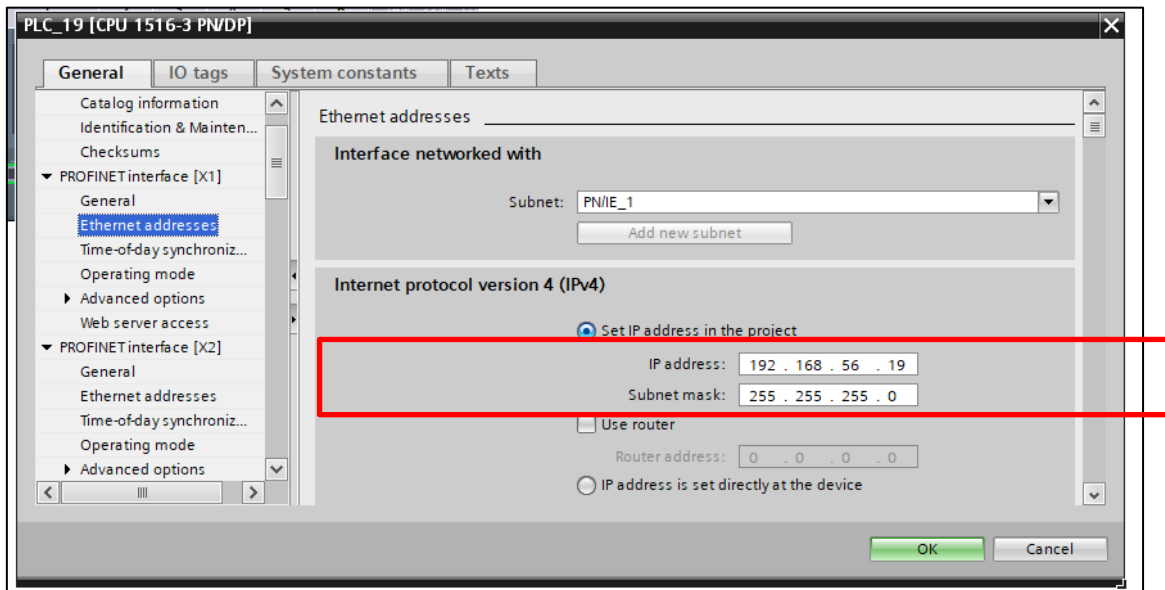


Ilustración 30. Dirección IP proyecto GrafcetUA

Para poder determinar el **rack** y **slot** tenemos que mirar en el proyecto de TIA Portal de Grafcet. Podemos ver en la Ilustración 31 que el “slot” será el denominado “Perfil soporte_0”, y el “rack” el 1.

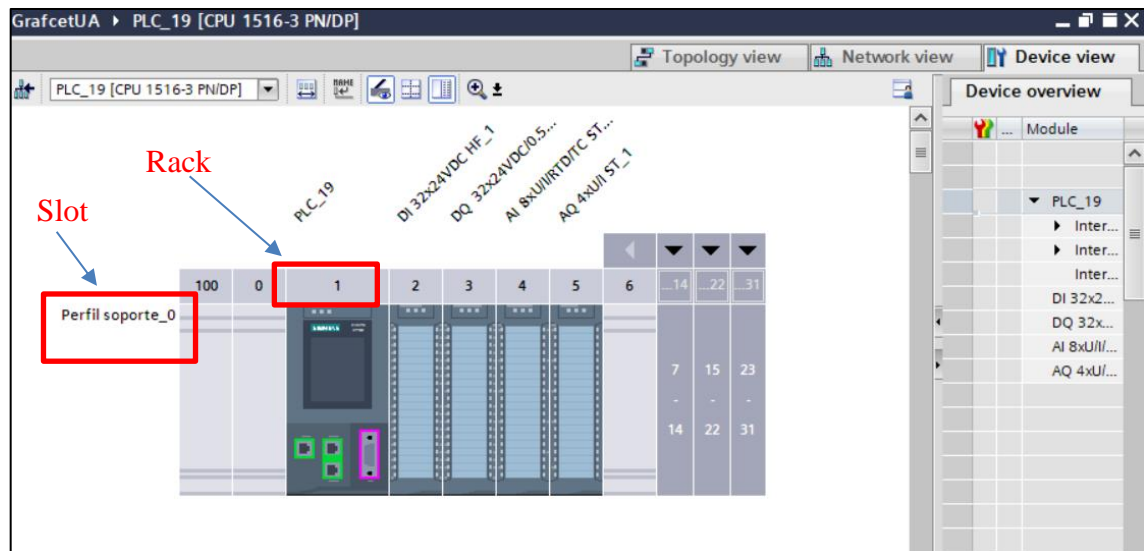


Ilustración 31. Rack y Slot proyecto GrafcetUA

Siguiendo con la configuración del endpoint de la Ilustración 29, encontramos el tiempo de ciclo, “*cycle time*”, este se refiere al intervalo de tiempo en el que Node-RED va

a enviar una solicitud de lectura o escritura, es decir, cada cuanto tiempo va a realizar la lectura el nodo “Clock_1Hz” sobre el PLC, si lo configuramos en 2000 ms (2 segundos), se enviará una solicitud de lectura cada segundo. Definir la frecuencia con la que node-RED interactúa con el dispositivo es importante ya que un ciclo de tiempo más corto puede proporcionar una actualización más rápida de los datos, pero también puede generar una mayor carga de comunicación en la red, por otro lado, un ciclo de tiempo más largo puede reducir la carga de comunicación, pero la actualización de datos será menos frecuente.

Por otro lado, tenemos el tiempo de espera “*Timeout*”, siendo el tiempo máximo en el que Node-RED espera para recibir respuesta del dispositivo, lo configuramos a un tiempo un poco mayor para dar margen, timeout=3000ms.

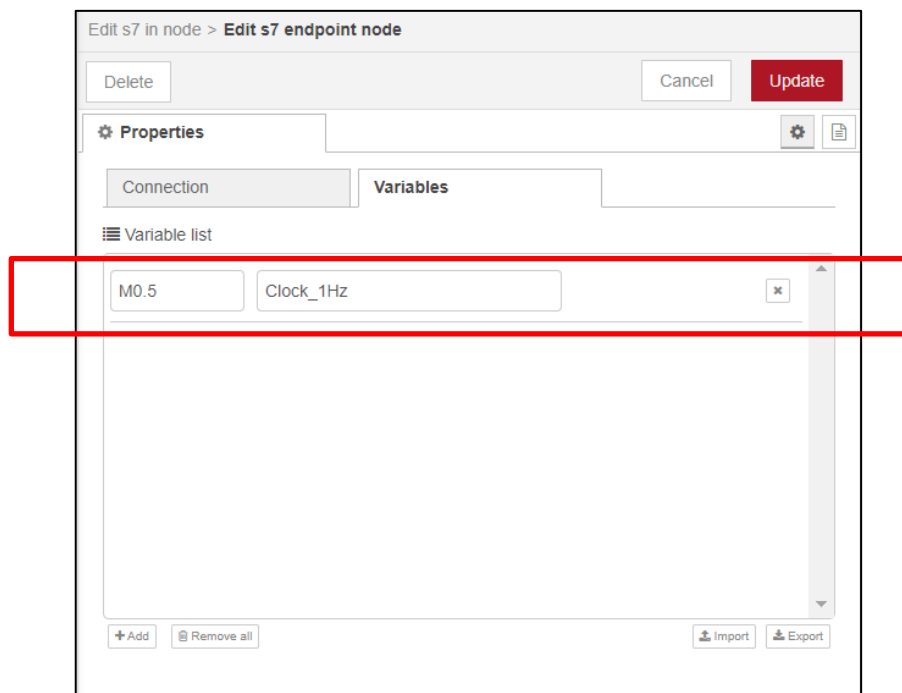


Ilustración 32. Declarar variables node-RED

En la Ilustración 32 se declaran las variables que queremos leer, en este caso vamos a leer la variable de “**Clock_1Hz**”.

<i>Variable</i>	<i>Tipo dato</i>	<i>Dirección</i>
Clock_1Hz	Bool	%M0.5

Tabla 4. Variable Clock_1Hz

Atendiendo a la librería de “node-red-contrib-s7” y dependiendo del tipo de dato, la sintaxis de la variable de la que queremos hacer la lectura puede variar (Ilustración 33). En este caso, el tipo de dato al ser booleano la dirección de conversión no cambia.

I1.0 or E1.0	I1.0 or E1.0	Boolean	Bit 0 of byte 1 of input area
Q2.1 or A2.1	Q2.1 or A2.1	Boolean	Bit 1 of byte 2 of output area
M3.2	M3.2	Boolean	Bit 2 of byte 3 of memory area

Ilustración 33. Conversión de direcciones de la librería “node-red-contrib-s7”

Finalmente se añade un nodo de “Debug” al esquema de node-RED para poder ver el valor en pantalla.

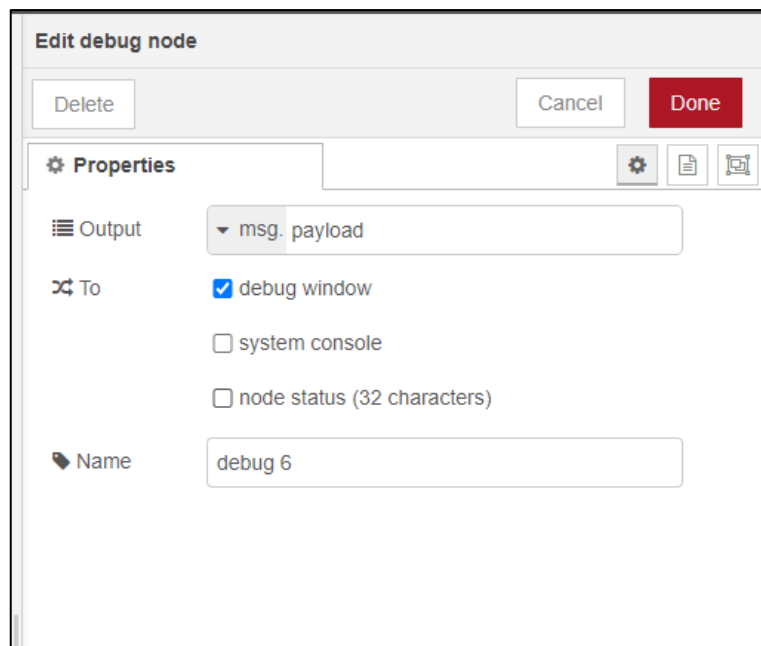


Ilustración 34. Configuración nodo “Debug”

5.6 CASO DE USO

El propósito principal de este caso de uso es presentar un ejemplo genérico de los pasos esenciales para habilitar un servidor **OPC UA** en una máquina controlada por un **PLC**. Se ilustra el caso utilizando la *Minifábrica ICAI* como referencia de la máquina y un *PLC Siemens S7-1500* para su control. Se considera que la única capacidad de comunicación del PLC con otros sistemas es a través del **protocolo S7COMM**, propio de Siemens. Para simplificar el ejemplo, se emplea la herramienta **PLCSIM Advanced** para simular el PLC, permitiendo su implementación en un entorno virtual, sin requerir hardware adicional en un PC.

A modo resumen y para entender el proceso, se va a seguir el siguiente esquema de comunicación.

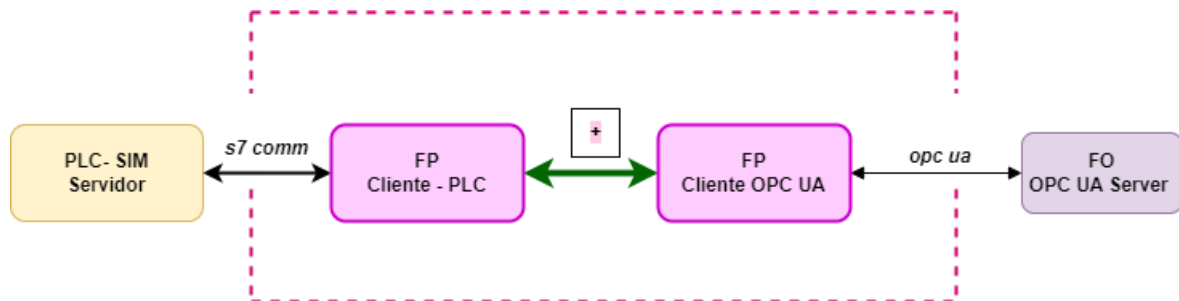


Ilustración 35: Esquema de comunicación

El flujo inicia desde el PLC SIM (actuando como servidor OPC UA). Este es el servidor que expone las variables y datos que deben estar disponibles para su lectura o escritura por otros sistemas. Siguiendo el esquema, encontramos una comunicación *s7 comm* con dos bloques de cliente FP, estos corresponden a los nodos en Node-RED configurados para conectarse con el servidor OPC UA. Duplicar el cliente en Node-RED nos permitirá una comunicación bidireccional entre el servidor OPC UA y Node-RED.

Se desglosará paso a paso como configurar el flujo de Node-RED para lograr una lectura y escritura de datos desde el servidor OPC UA para comprobar y demostrar una correcta comunicación. Los pasos a seguir son los siguientes.

5.6.1 CONFIGURACIÓN COMUNICACIÓN PLC- NODE-RED

Primero se debe establecer la conexión entre **Node-RED** y el PLC *Siemens* utilizando el **protocolo S7** para la comunicación.

- En Node-RED, se utilizará el **protocolo S7**, con sus nodos correspondientes para establecer la conexión entre Node-RED y el PLC Siemens.

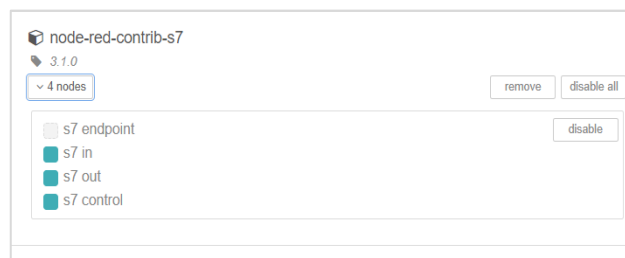


Ilustración 36: paleta node-red-contrib-s7

- Es necesario tener la configuración correcta de la **dirección IP (Address)**, el número de **rack** y **slot** del PLC en Node-RED como se ha explicado en secciones anteriores

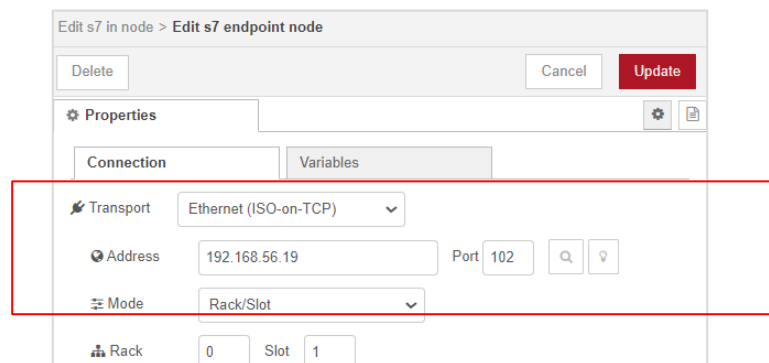


Ilustración 37: Configuración endpoint node-red (I)

- Configurar correctamente el PLC Virtual (**PLCSIM Advanced V4.0**) el cual no soporta OPC UA.

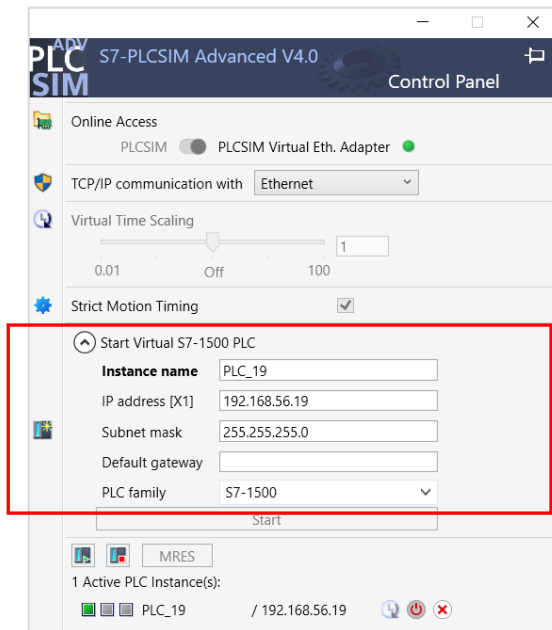


Ilustración 38: Configuración PLCSIM Advanced V4.0

Su configuración está explicada en detalle en secciones anteriores de la memoria.

5.6.2 DEFINICIÓN DE VARIABLES REMOTAS

Para ello **identificamos** primero las variables a las que queremos acceder a través de OPC UA.

- Analizar y seleccionar las variables específicas que se desean leer desde el servidor OPC UA.
- Asegurarse de que las variables están disponibles en el PLC, esto se puede verificar abriendo el proyecto en el visualizador TIA Portal.

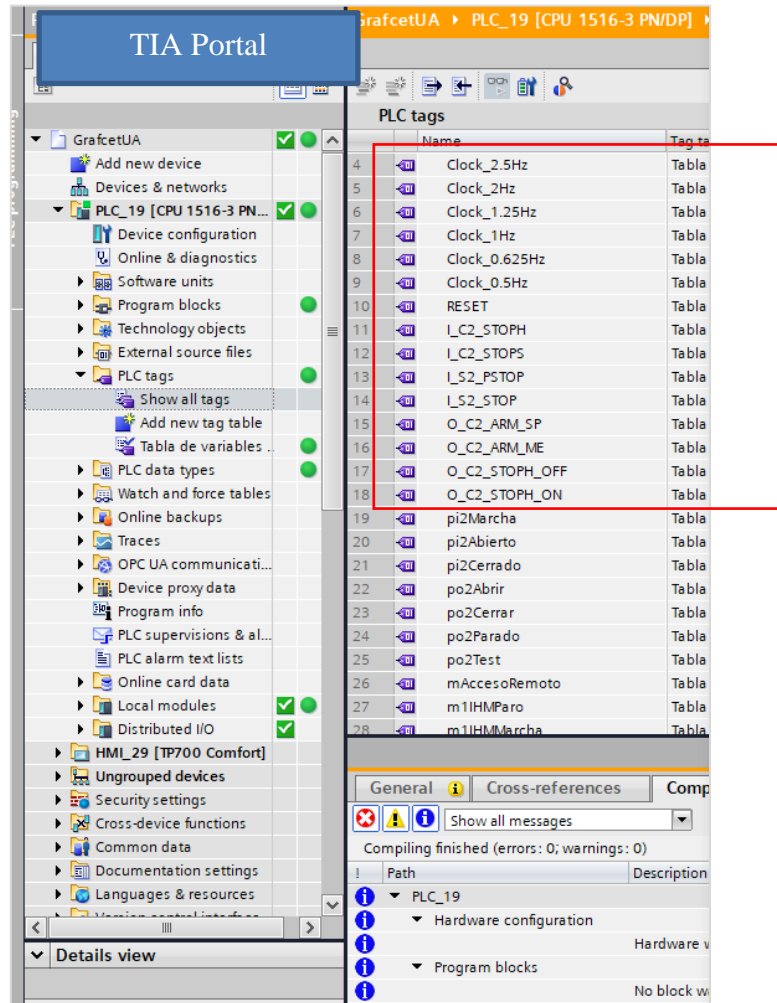


Ilustración 39: Visualizador variables TIA Portal

- Definir las variables a las que queremos acceder en Node-RED mediante identificadores únicos (NodeID). Teniendo en cuenta además el **direccionamiento de variables** del protocolo de comunicación s7.

Address	Step7 equivalent	JS Data type	Description
DB5.X0.1	DB5.DBX0.1	Boolean	Bit 1 of byte 0 of DB 5
DB23.B1 or DB23.BYTE1	DB23.DBB1	Number	Byte 1 (0-255) of DB 23
DB100.C2 or DB100.CHAR2	DB100.DBB2	String	Byte 2 of DB 100 as a Char
DB42.I3 or DB42.INT3	DB42.DBW3	Number	Signed 16-bit number at byte 3 of

Ilustración 40: Direccionamiento de variables s7-comm

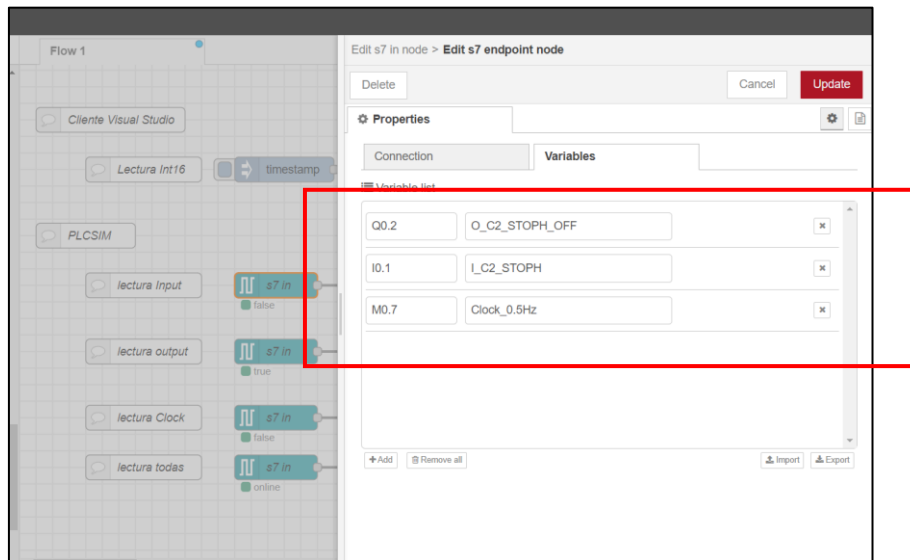


Ilustración 41: Configuración endpoint node-RED (II)

5.6.3 CONFIGURACIÓN DEL SERVIDOR OPC UA VISUAL

Para ello se han de definir las variables seleccionadas en el servidor OPC UA.

- Configuración del Servidor **OPC UA en Visual Studio** utilizando las variables previamente identificadas. (El proceso de cómo añadir estas variables están explicadas en el apartado de la memoria: 5.4.1)

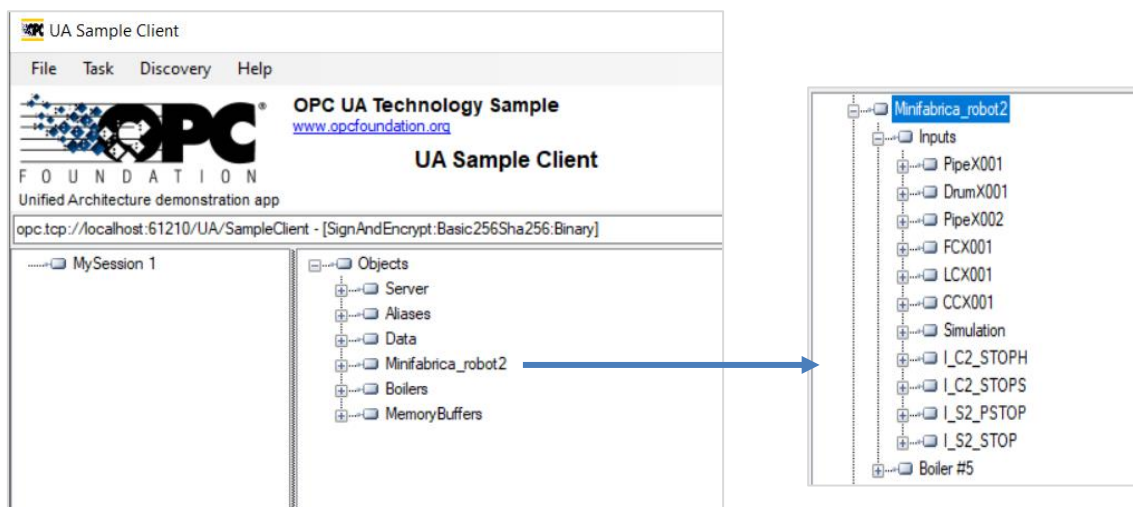


Ilustración 42: Configuración del Address Space

- Asignar las mismas direcciones, tipos de dato y nombre que en el PLC.

5.6.4 CONFIGURACIÓN DE NODE-RED COMO CLIENTE OPC UA

Configurar Node-RED para que actúe como cliente de OPC UA.

- Establecer una conexión segura y confiable entre Node-RED y el Servidor OPC UA Visual para actuar de pasarela entre las variables que se leen/escriben en Node-RED (desde el PLC) y lo que debe aparecer en el servidor OPC Visual.
- Ingresar la dirección del servidor, credenciales de acceso, identificar y determinar el *endpoint* y detalles de seguridad.

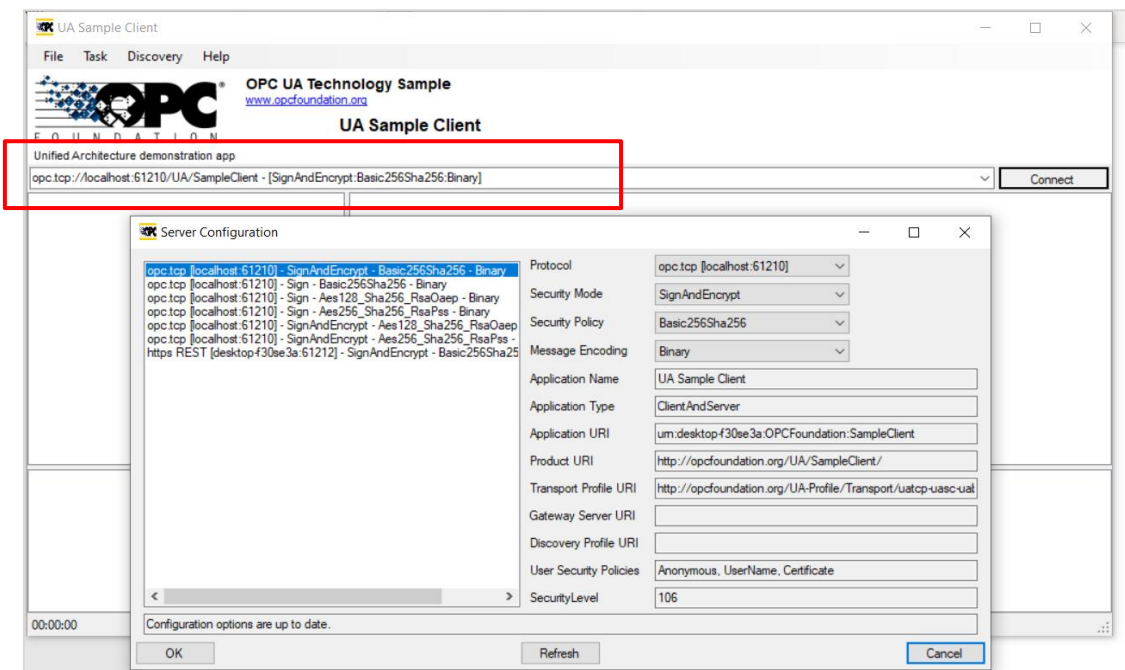


Ilustración 43: Dirección – OPC UA

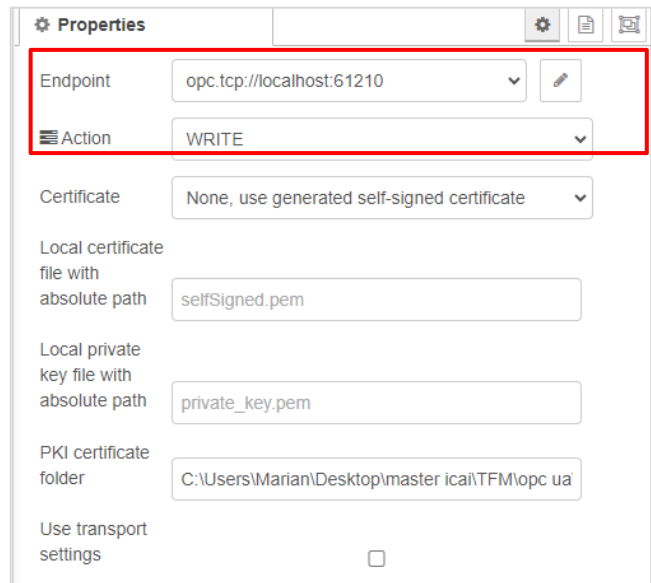


Ilustración 44: Dirección Node-RED

5.6.5 PROCESAMIENTO DE DATOS EN NODE-RED

- Utilizando los nodos de función, switch, o filtros según sea necesario para procesar los datos recibidos.

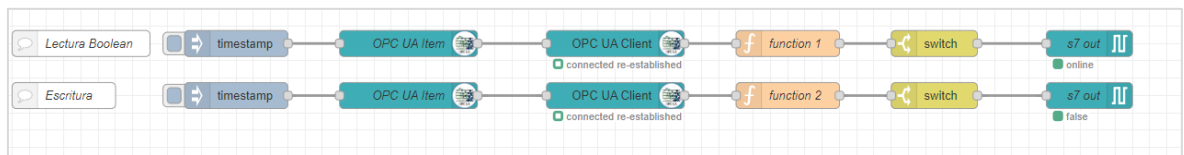


Ilustración 45: Flujo de Node-RED

- Se pueden aplicar diferentes filtros para poder adaptar los datos a las necesidades específicas de cada proyecto.

5.6.6 VERIFICACIÓN Y COMPROBACIÓN DEL CONJUNTO

Comprobar que el conjunto funciona correctamente realizando pruebas para que se confirme que la comunicación entre Node-RED, el PLC y el servidor OPC UA Visual es exitosa. Tenemos diferentes formas de visualizar los datos.

- Utilizando nodos Dashboard para mostrar los datos leídos en paneles gráficos. Poder así configurar los paneles para visualizar valores o indicadores.
- Se pueden visualizar los datos de una manera más directa mediante la propia ventana de “Debug” de node-RED.

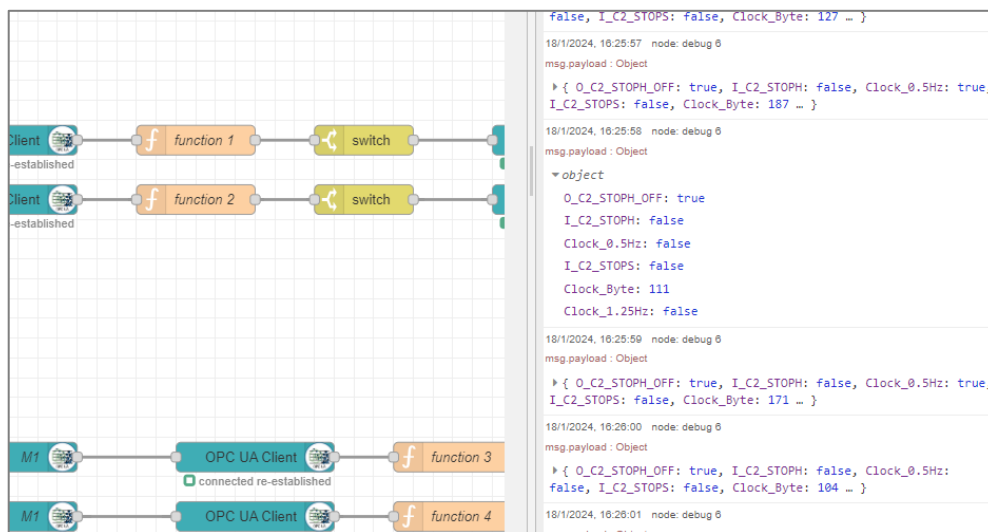


Ilustración 46: Comprobación de resultados

- Se pueden visualizar las variables mediante **TIA Portal**.

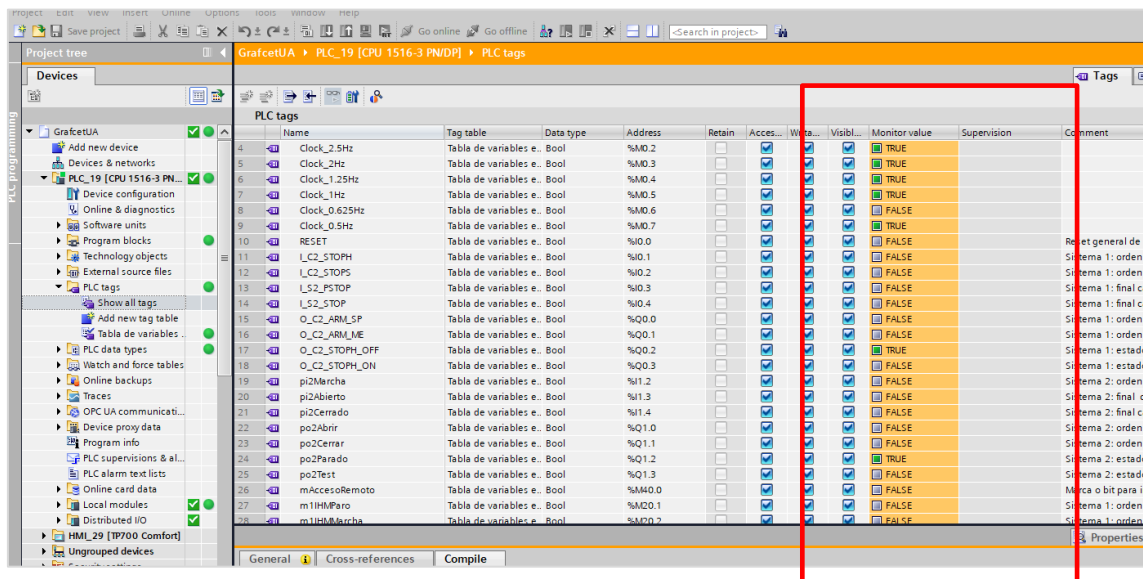


Ilustración 47: Visualización variables (TIA Portal)

5.6.7 OBSERVACIONES ADICIONALES

- **Verificación de comunicación:** Para verificar la comunicación con Node-RED se han de realizar pruebas de conectividad, pudiendo usar nodos de diagnóstico en Node-RED. También se puede comprobar por medio de la herramienta de **UA Expert**. Para ello se configura indicando el **endpoint**, pero **solo** nos será útil para **comprobar** que la comunicación principal a través de Node-RED funciona.

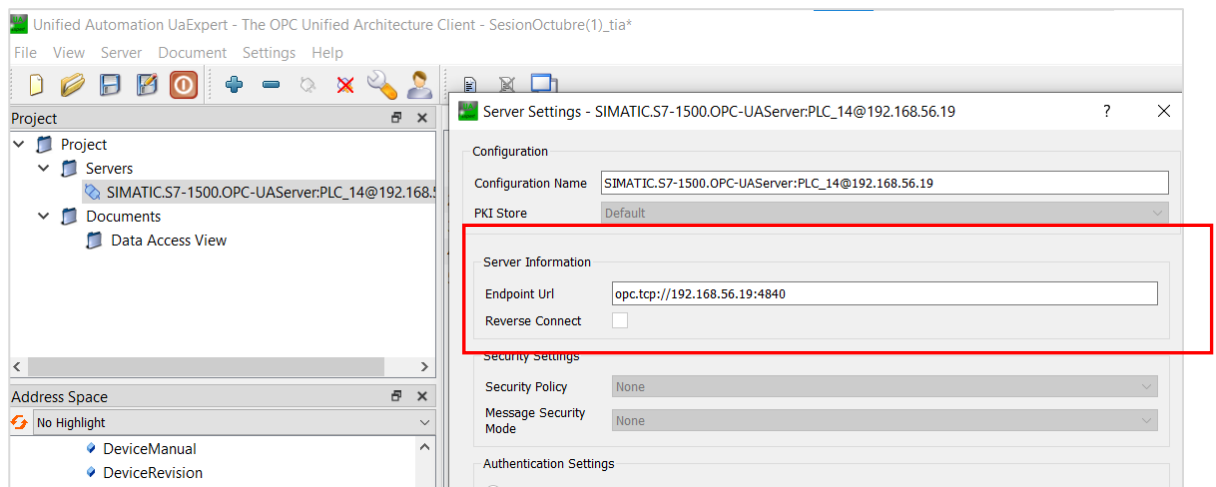


Ilustración 48: Verificación de comunicación

- **Ambiente de Trabajo:** Asegurarse de trabajar con un entorno que emule un PLC Siemens para este caso de uso, como el PLC virtual generado a través de PLCSIM Advanced.
- **Uso del PLC Siemens:** Aclarar que el PLC Siemens no funcionará como servidor OPC UA. Se empleará solo para probar la comunicación a través de Node-RED.

Capítulo 6. GESTIÓN DEL PROYECTO

6.1 METODOLOGÍA

Para este proyecto se ha seguido una **metodología en cascada “Waterfall”**, de manera que el desarrollo del proyecto a seguido una secuencia lineal de etapas, donde cada etapa se completa antes de pasar a la siguiente. Cada etapa del proceso representa un conjunto de actividades específicas, desde la definición de requisitos hasta la implementación y despliegue del producto final. En el contexto del presente proyecto, sobre digitalización de máquinas mediante OPC UA para ser integrables en sistemas bajo filosofía Industria 4.0, la metodología en cascada nos ha proporcionado un arco organizado para abordar cada etapa del desarrollo de manera efectiva asegurando que cada fase se va completando antes de avanzar a la siguiente. Las etapas que se han seguido en el proyecto siguiendo la metodología en cascada son las siguientes:

1. **Definición de objetivos:** se llevó a cabo una investigación sobre los requisitos y objetivos del proyecto. Recogidos en el “Anexo B” documento que se debió de entregar tiempo atrás. Se estableció una comunicación con los interesados y usuarios finales para comprender la necesidad de implantar la solución propuesta, incluyendo la integración de máquinas en el entorno Industria 4.0 mediante OPC UA.
2. **Aprendizaje de las herramientas:** esta etapa es la que más tiempo se ha dedicado, ya que se necesitó mucho tiempo para la formación y adquisición de conocimientos para poder familiarizarse con las diferentes herramientas y tecnologías utilizadas en el proyecto.
3. **Diseño del sistema:** Con los requisitos definidos y teniendo un conocimiento amplio sobre el funcionamiento de las herramientas, se procedió a diseñar la arquitectura del sistema. Se elaboraron diferentes esquemas para entender la comunicación entre las máquinas y el servidor OPC UA. Se planificó y se trató de entender cómo se deben de conectar los diferentes componentes y los datos que se compartirían entre ellos.

4. **Implementación:** Una vez implementado el diseño, se inició la fase de implementación. Para ello se utilizó Visual Studio y la librería .NET Standard de OPC Foundation para desarrollar el servidor OPC UA. Se configuró el PLC virtual (PLCSIM Advanced) utilizando TIA Portal para que actuara como el controlador de la máquina. También se instaló y configuró Node-RED para que actuara como el cliente encargado de comunicarse con el servidor OPC UA.
5. **Pruebas y validación:** Por último, se procedió al despliegue del sistema haciendo uso del ejemplo de “GrafcetUA” donde se ejecutó y se realizó la comprobación de comunicación entre los distintos componentes y sistemas de manera que la comunicación funcionó exitosamente.

Al seguir la metodología en cascada, se logró un desarrollo organizado, secuencial y estructurado del proyecto, lo que ha facilitado cumplir con los requisitos establecidos y asegurar una integración exitosa de las máquinas en el entorno Industria 4.0 mediante OPC UA.

6.2 GESTIÓN DEL PROYECTO Y PLANIFICACIÓN

Para realizar una organización y seguimiento de actividades del proyecto se ha utilizado una herramienta denominada “**Diagrama de Gantt**”, de manera que se ha representado gráficamente las distintas tareas del proyecto. En el eje horizontal se encuentran las tareas y en el eje vertical el tiempo dedicado en cada una de ellas. El proyecto comenzó a realizarse a partir de diciembre hasta julio, comprendiendo una media de 212 días en total.

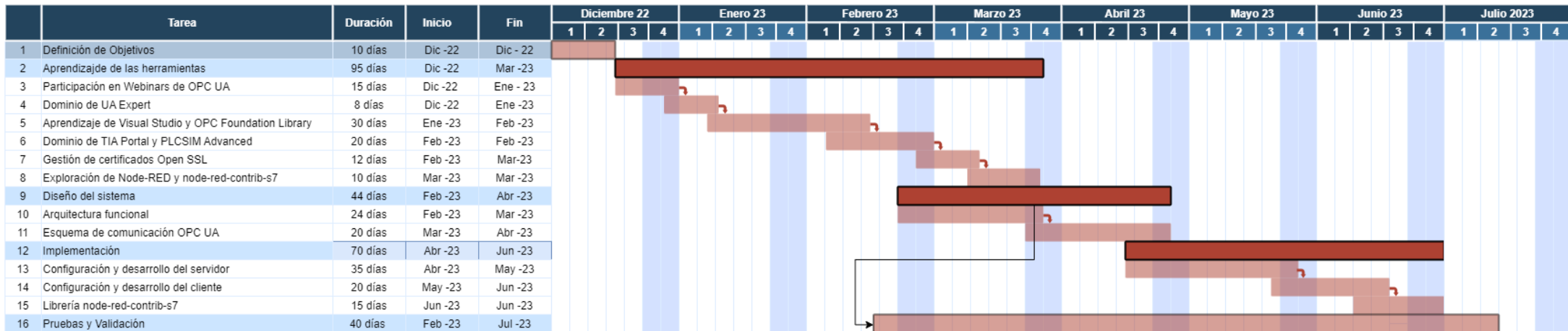


Tabla 5. Diagrama de Gantt

6.3 COSTOS DEL PROYECTO

En esta sección se detallan los costos asociados con el desarrollo e implementación del proyecto. Se incluyen tanto los costos de software como de hardware utilizados.

6.3.1 COSTOS DE SOFTWARE

1. **Visual Studio:** La licencia de Visual Studio gratuita ofrece todas las funcionalidades y herramientas necesarias.
2. **Tia Portal:** El costo de la licencia de TIA Portal depende de la versión y opciones seleccionadas. Puede oscilar entre 1.000€ a 5.000€, según el paquete adquirido.
3. **Node-RED:** Es una plataforma de código abierto y gratuita. No requiere ningún costo asociado para su uso básico.
4. **UA Expert:** UA Expert de Unified Automation generalmente ofrece una versión de prueba gratuita, pero para obtener la licencia completa, puede costar alrededor de 500€ a 1.000€ dependiendo de las opciones seleccionadas.
5. **Licencias y Derechos de Uso:** La librería OPC UA.NET Standard de la OPC Foundation es de código abierto y se distribuye bajo la licencia MIT, por lo que no conlleva gastos ni costos de licencias.
6. **Webinars de OPC UA:** los webinars suelen ofrecerse de forma gratuita por diversas organizaciones, como la OPC Foundation, lo que no implica costos directos.

6.3.2 COSTOS DE HARDWARE

1. **PLC Virtual (PLCSIM Advanced):** Esta es una herramienta de simulación proporcionada por Siemens, y su costo está incluido en la licencia de TIA Portal mencionada anteriormente.
2. **Servidor y Equipamiento de Red:** el costo de los servidores y equipamiento de red puede variar según las especificaciones y requisitos específicos de cada proyecto.

Capítulo 7. CONCLUSIONES

7.1 CONCLUSIONES GENERALES

El proyecto de “Digitalización de Máquinas mediante OPC UA para ser Integrables en Sistemas bajo Filosofía Industria 4.0” ha sido un proceso enriquecedor que ha permitido profundizar en el mundo de la Industria 4.0 y las tecnologías emergentes de comunicación industrial. A lo largo del desarrollo, se han logrado alcanzar los objetivos planteados y se han obtenido conclusiones acerca del impacto de la implementación de OPC UA en la digitalización de máquinas industriales.

Durante el desarrollo de este proyecto, uno de los desafíos más destacados fue la integración de la tecnología **OPC UA** en un sistema que previamente no contaba con esta capacidad de comunicación. A medida que explorábamos las posibilidades de transformar una maquinaria tradicional en un componente integral de la Industria 4.0, surgió la necesidad de implementar un protocolo de comunicación que permitiera una interconexión segura y eficiente.

Para lograr esto, se llevó a cabo una investigación y análisis de las soluciones disponibles en el ámbito de la automatización industrial y la digitalización. A pesar de que el sistema no tenía una estructura originalmente compatible con OPC UA, se tomó la decisión de utilizar Visual Studio para desarrollar un servidor OPC UA y establecer una comunicación bidireccional con una máquina específica. Esta elección permitió superar las limitaciones iniciales y brindar la capacidad de leer y escribir datos en tiempo real en una máquina previamente no conectada.

La implementación exitosa de OPC UA en este contexto no preexistente no solo demuestra la **flexibilidad** y **adaptabilidad** de esta tecnología, sino que también abre la puerta a futuras aplicaciones y expansiones en la digitalización de sistemas industriales. La habilidad para incorporar OPC UA en sistemas no diseñados originalmente para ello es una

clara oportunidad para integrar esta tecnología en el mundo de la Industria 4.0, al permitir la conectividad y la obtención de datos en tiempo real de manera eficaz y segura.

Se ha comprobado que OPC UA es un protocolo de comunicación altamente efectivo y versátil que facilita la integración y comunicación eficiente entre diferentes sistemas industriales. Gracias a su estructura escalable y su seguridad robusta, OPC UA se ha posicionado como una herramienta fundamental en la convergencia de la planta de producción y los sistemas de la Industria 4.0.

De la misma manera, la utilización de las diversas herramientas (Visual Studio, TIA Portal, Node-RED y UA Expert) que han permitido desarrollar y configurar el sistema cliente-servidor que se adapte a las necesidades específicas. La combinación de estas herramientas ha demostrado ser una solución efectiva para la supervisión de una planta industrial, así como para el procesamiento de datos y comunicación entre máquinas.

En cuanto al uso de librerías específicas, como la librería OPC UA .NET Standard de la OPC Foundation y la librería de “node-red-contrib-s7”, se ha comprobado que estas proporcionan una base confiable para la implementación que han permitido agilizar y optimizar el desarrollo de la solución. Además, el acceso a los webinars y recursos proporcionados por la OPC Foundation y Unified Automation han resultado ser de gran ayuda para comprender y dominar las características más avanzadas de OPC UA y maximizar así su potencial.

A lo largo del desarrollo del proyecto también se ha tenido en cuenta la seguridad. De manera que se ha incorporado una gestión de certificados necesaria para garantizar una comunicación segura.

Finalmente, el proyecto ha demostrado la viabilidad y beneficios de la digitalización de maquinarias mediante OPC UA. La combinación de herramientas, tecnologías y protocolos utilizados ha permitido conseguir una mayor eficiencia, interoperabilidad y adaptabilidad en la industria.

7.2 ANÁLISIS DE LOS OBJETIVOS

A continuación, se inserta una tabla analizando los objetivos establecidos en el Capítulo 1 del proyecto, añadiendo además algunos objetivos más específicos que se han ido desarrollando a lo largo del proyecto.

<i>Objetivo</i>	<i>Nivel de logro</i>	<i>Observaciones</i>
Estudio del protocolo OPC UA en la digitalización.	100%	Se ha logrado explorar las diferencias de la automatización clásica y la Industria 4.0, estudiando las necesidades de integrar una nueva Industria en nuestro entorno de fabricación.
Lograr la digitalización de máquinas industriales	100%	Se ha conseguido encontrar una solución de comunicación para un control de la planta de producción a través del uso del protocolo OPC UA y otras herramientas avanzadas
Implementación de OPC UA	100%	Se ha logrado implementar el protocolo a partir de lo que ofrece de la OPC Foundation.
Desarrollo del Cliente-Servidor	90%	Aunque se ha conseguido desarrollar el cliente y servidor, habría sido preferible que se hubiera programado un servidor desde cero. En lugar, se ha hecho uso del servidor que proporciona el repositorio opc ua .net standard de github, haciendo modificaciones.
Integración de tecnologías	100%	Se ha evaluado y combinado diversas herramientas como Visual Studio, TIA Portal y Node-RED, para conseguir una integración y un correcto funcionamiento del sistema.
Seguridad y Ciberseguridad. Comunicación eficiente y confiable.	100%	Se ha considerado y aplicado certificados mediante OpenSSL

Aplicación hacia un caso práctico	90%	Se ha hecho uso de la práctica “Grafcet”, en su lugar, habría sido mejor usar un ejemplo más complejo, o un caso práctico nuevo.
Uso de la plataforma de Siemens (TIA Portal)	100%	Se ha logrado manejar el uso de la plataforma.

Tabla 6. Análisis de Objetivos Generales

Mediante el análisis de los objetivos planteados, se concluye que se han alcanzado en su mayoría de manera satisfactoria. La implementación de OPC UA, junto con las tecnologías y herramientas seleccionadas, han permitido una digitalización efectiva.

7.3 TRABAJO FUTURO

Tras haber realizado un análisis de los objetivos del proyecto, a pesar de que se hayan alcanzado de manera satisfactoria, siempre existen oportunidades para mejorar y expandir el alcance de la implementación, presentamos algunas áreas de trabajo futuro que podrían ser consideradas para una evolución del sistema.

En cuanto a **escalabilidad**, si bien el proyecto se ha enfocado a una planta de producción específica, se puede explorar la posibilidad de escalar la solución a un entorno más amplio, es decir, poder adaptar esta solución a cualquier arquitectura de planta. Además, se pueden realizar muchas mejoras de optimización y rendimiento, mejorando por ejemplo los tiempos de respuesta y la eficiencia de la comunicación, para lograr más agilidad y sincronización entre las máquinas y el sistema.

Se podría además tratar de integrar más **dispositivos**, de manera que se pudiera usar una variedad más amplia de dispositivos, sensores y actuadores presentes en la planta de producción, e integrarlos todos en el sistema.

En cuanto al **análisis de datos**, se podrían incorporar técnicas de análisis de datos avanzados para obtener análisis más eficientes de los datos generados por las máquinas, facilitando así la toma de decisiones basados en datos, incluso poder prevenir posibles fallas.

Por último, se ha pensado en poder integrar con **Sistemas de Gestión Empresarial (ERP)** existentes, permitiendo una mayor coordinación y sincronización entre operaciones de producción y operaciones comerciales.

Podemos concluir afirmando que el sistema y la solución propuesta tiene una gran evolución futura, ya que se pueden añadir diversas funciones adicionales y mejoras futuras.

7.4 CONTRIBUCIÓN A LOS OBJETIVOS DEL DESARROLLO SOSTENIBLE

Este proyecto de implementación en un entorno de automatización industrial se alinea de manera significativa con varios Objetivos del Desarrollo Sostenible (ODS) establecidos por las Naciones Unidas. A continuación, se van a destacar los vínculos específicos entre el proyecto y ciertos ODS:

1. **ODS 9: Industria, Innovación e Infraestructura:** Este proyecto fomenta la innovación al incorporar tecnología OPC UA en un entorno no originalmente diseñado para ello, demostrando flexibilidad y adaptabilidad de esta tecnología.
2. **ODS 11: Ciudades y Comunidades Sostenibles:** Con un mayor desarrollo y evolución puede llegar a contribuir a la creación de comunidades más sostenibles al mejorar la conectividad y la eficiencia en la gestión de sistemas industriales
3. **ODS 12: Producción y Consumo Responsable:** La implementación de OPC UA permite una gestión más eficiente de los recursos, contribuyendo a prácticas de producción y consumo más responsables.
4. **ODS 13: Acción por el Clima:** Al facilitar la comunicación eficiente entre sistemas, el proyecto ayuda a optimizar procesos, reduciendo potencialmente el consumo de energía y la huella de carbono asociada.
5. **ODS 17: Alianzas para lograr los Objetivos:** Al utilizar estándares abiertos y trabajar con tecnologías ampliamente aceptadas, el proyecto promueve la colaboración y la creación de alianzas para impulsar el desarrollo sostenible.

En conjunto, esta iniciativa no solo aborda desafíos técnicos, sino que también busca contribuir a la construcción de un futuro más sostenible, aprovechando la tecnología para mejorar la eficiencia y reducir el impacto ambiental y social de los procesos industriales.

Capítulo 8. BIBLIOGRAFÍA

- [1] Siemens (Ed.). (n.d.). Digital enterprise. Siemens España. https://www.siemens.com/es/es/empresa/temas-clave/fabrica-digital.html?gclid=Cj0KCQjwnrmlBhDHARIsADJ5b_IEpQVRDTgrH0z-
- [2] *¿Qué es la industria 4.0?*: Deloitte España. Deloitte Spain. (2018, January 25). <https://www2.deloitte.com/es/es/pages/manufacturing/articles/que-es-la-industria-4.0.html>
- [3] *Industria 4.0*. Elion. (2022, December 16). <https://www.elion.es/tecnologias/industry40/>
- [4] *¿Qué es el internet de las cosas (IOT)? ¿Qué es el Internet de las cosas (IoT)?* | Oracle España. (n.d.). <https://www.oracle.com/es/internet-of-things/what-is-iot/#:~:text=La%20Internet%20de%20las%20cosas,sistemas%20a%20trav%C3%A9s%20de%20Internet>
- [5] Quiñonero, J. A. M. (n.d.). *Prototipo para mejorar la automatización de procesos en las infraestructuras en el sector del agua*. <https://openaccess.uoc.edu/bitstream/10609/137869/1/jaminarroTFM0122memoria.pdf>
- [6] Festo. (n.d.). *Software para la producción industrial*. Software para la producción industrial | Festo ES. https://www.festo.com/es/es/e/soluciones/digitalizacion/software-id_253294/?fwacid=166c25c1fce3e9bf&fwkeyword=industria+4.0&gad=1&gclid=Cj0KCQjwnrmlBhDHARIsADJ5b_15edVeH5YDN28b8jqoXcPasnkKW3gE1c0z08tDTUuyx4hu6jKCxj4aAiZkEALw_wcB
- [7] B&R Industrial Automation GmbH. B&R: *Perfection in Automation*. (n.d.). <https://www.br-automation.com/es-es/tecnologias/opc-ua/>
- [8] *FAQs acerca de OPC UA sobre TSN*. B&R: Perfection in Automation. (n.d.-b). <https://www.br-automation.com/es-es/tecnologias/opc-ua/faqs-acerca-de-opc-ua-sobre-tsn/>
- [9] B&R Industrial Automation GmbH. (n.d.-a). *Qué es opc ua*. B&R: Perfection in Automation. <https://www.br-automation.com/es-es/tecnologias/opc-ua/>

- [10] Flores, D., & Zuniga, J. (2022, September 27). *OPC Ua Transferir Información Entre servidores y clientes*. Blog Logicbus. <https://www.logicbus.com.mx/blog/servidor-opc-ua-integrado/>
- [11] Protocolo OPC UA. Características y Aplicaciones en scada. Canal Gestión Integrada. (2016, February 22). <https://www.inesem.es/revistadigital/gestion-integrada/protocolo-opc-ua-caracteristicas-y-aplicaciones/>
- [12] Ruiz, D. (n.d.). *Diferencias OPC Clasico Ua*. D.ruiz. Scribd. <https://es.scribd.com/document/542230262/Diferencias-OPC-Clasico-UA-druiz>
- [13] Atvise. (2022, March 20). *¿Qué es opc ua? beneficios de la arquitectura unificada*: Atvise®. atvise® - Sistema SCADA Web. <https://atvise.vesterbusiness.com/news/que-es-opc-ua-arquitectura-unificada/>
- [14] Home Page - OPC Foundation. *OPC Foundation - The Industrial Interoperability Standard™*. (2023, April 15). <https://opcfoundation.org/>
- [15] *Descargar Visual studio*. Visual Studio. (2023, July 17). <https://visualstudio.microsoft.com/es/thank-you-downloading-visual-studio/?sku=Community&channel=Release&version=VS2022&source=VSLandingPage&cid=2030&passive=false>
- [16] *¿Por qué elegir la plataforma para desarrolladores de .net?*. Microsoft. (n.d.). <https://dotnet.microsoft.com/es-es/platform/why-choose-dotnet>
- [17] Gútiez, I., & Iñigo Gútiez: *Enseño a programar PLC de Siemens a través de mis cursos*. (2023, April 21). *TIA portal: ¿Qué es?* Programación Siemens. https://programacionsiemens.com/que-es-tia-portal/#Y_lo_Simatic_S7-200
- [18] *Ua Webinar – unified automation*. unified. (n.d.). <https://www.unified-automation.com/services/training/ua-webinar.html>
- [19] *Services - unified automation*. unified. (n.d.-a). <https://www.unified-automation.com/services.html>
- [20] Bada, C. R. (2022, March 3). *OPC-UA & Mqtt: Dos protocolos y un destino*. Automática e Instrumentación - La revista de la Industria 4.0. <https://www.automaticaeinstrumentacion.com/texto-diario/mostrar/3474519/opc-ua-mqtt-protocolos-destino>

- [21] MatrikonOPC. (n.d.). *Que Es un servidor opc?*. OPC News.
<https://www.matrikonopc.es/opc-servidor/index.aspx>
- [22] Bada, Cesar Rincón. (2022, March 3). OPC-UA & Mqtt: *Dos protocolos y un destino*. Automática e Instrumentación - La revista de la Industria 4.0.
<https://www.automaticaeinstrumentacion.com/texto-diario/mostrar/3474519/opc-ua-mqtt-protocolos-destino>

ANEXO I: INSTALACIÓN PLATAFORMA .NET

En este anexo se explica los pasos que se han seguido para instalar la plataforma .NET para poder hacer uso de ella y poder así utilizar la librería de *OPC UA .NET Standard*.

Primero se ha tenido que descargar e instalar .NET, junto a un SDK de .NET (Kit de desarrollo de software), para así poder empezar a compilar aplicaciones .NET ya que este último ejecutable nos permitirá utilizar el comando “dotnet” (<https://dotnet.microsoft.com/es-es/download>).

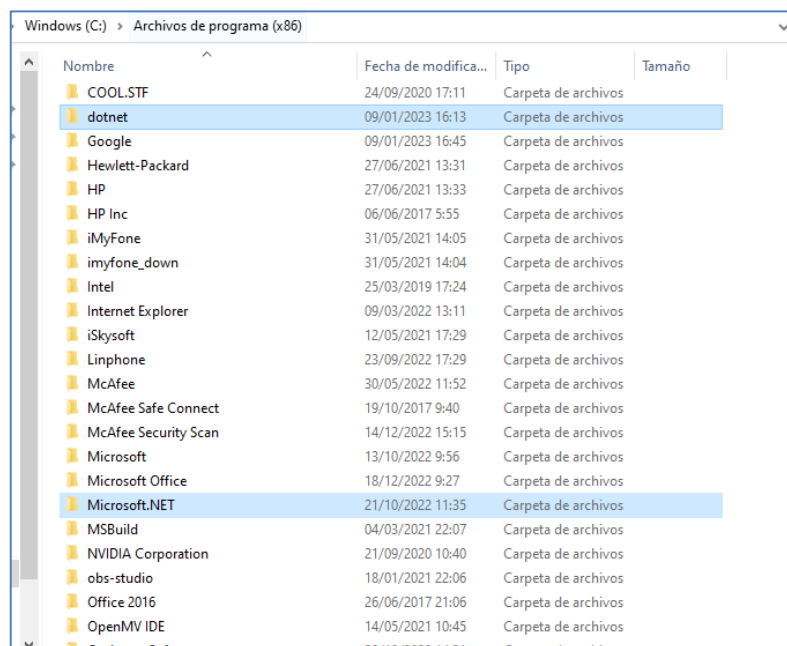


Ilustración 49. Archivos de programa para la instalación de .NET

Tras la descarga y siguiendo los pasos que se indican en la instalación, se han de crear dos carpetas en “Archivos de programa” una dedica a “dotnet” y la otra a la librería “Microsoft.NET”. Para comprobar la correcta instalación se ha de abrir un símbolo de sistema y ejecutar el comando “dotnet”.

```
C:\WINDOWS\system32>dotnet

Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
  -h|--help           Display help.
  --info              Display .NET information.
  --list-sdks         Display the installed SDKs.
  --list-runtimes    Display the installed runtimes.

path-to-application:
  The path to an application .dll file to execute.
```

Ilustración 50 .Salida tras ejecución del comando “dotnet”

Si obtenemos la salida que vemos en la anterior imagen (Ilustración 50 .Salida tras ejecución del comando “dotnet”), comprobamos el correcto funcionamiento e instalación de la librería .NET.

Tras la correcta instalación de .NET será necesaria la gestión de los paquetes **Nuget**. Siendo estos unos paquetes de código abierto para el framework de desarrollo de Microsoft .NET, que permiten la descarga de las diferentes bibliotecas y componentes necesarios para el proyecto. Los paquetes Nuget se alojan en un repositorio en línea y se pueden descargar directamente desde la propia aplicación de Visual Studio.

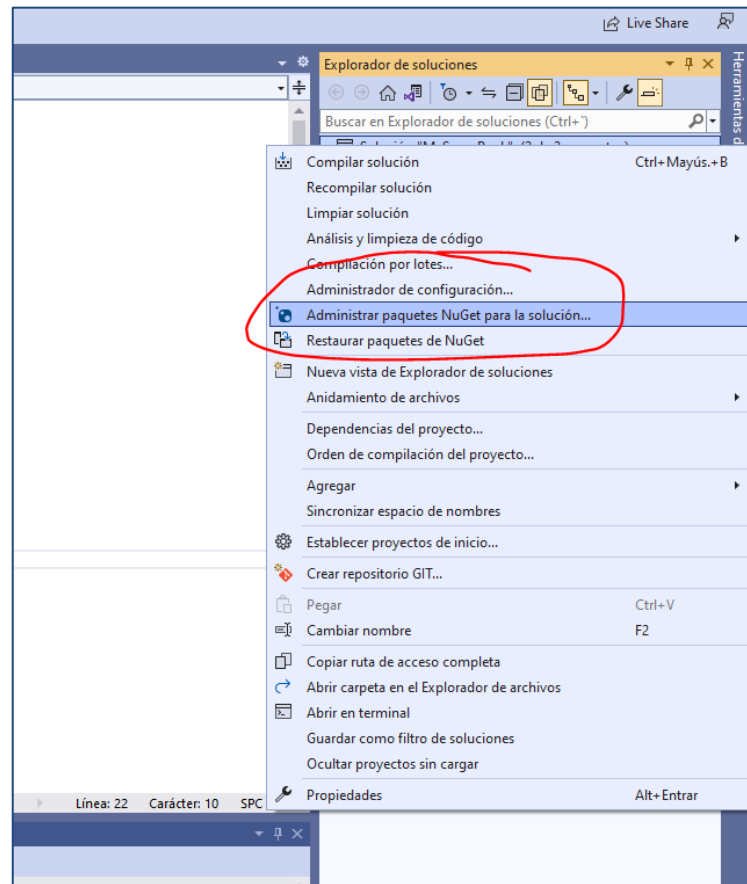


Ilustración 51: Administrados paquetes Nuget

ANEXO II: INSTALACIÓN DE NODE-RED

En este anexo se explica los pasos que se han seguido para instalar la herramienta de programación Node-RED.

1. Primero se ha descargado la instalación desde el siguiente enlace: <https://nodejs.org/en>, y se han seguido las instrucciones de instalación.
2. Una vez que se ha instalado correctamente, se ha de abrir una terminal o línea de comandos desde Node.js

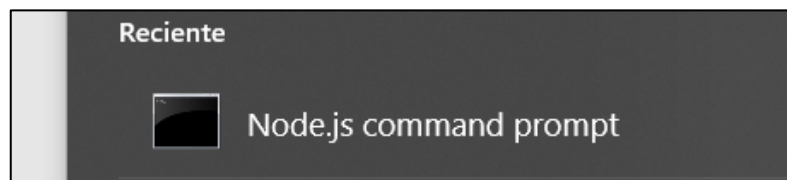


Ilustración 52 Terminal para abrir Node-red

3. Se ha de ejecutar el comando “node-red”.
4. Node-red se ejecutará y mostrará información sobre la versión y la URL de acceso en la terminal.

```
You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
23 Jun 19:10:46 - [warn] Encrypted credentials not found
23 Jun 19:10:46 - [info] Server now running at http://127.0.0.1:1880/
23 Jun 19:10:46 - [info] Starting flows
23 Jun 19:10:46 - [info] Started flows
23 Jun 19:35:41 - [info] Installing module: node-red-contrib-opcua, version: 0.2.307
23 Jun 19:37:05 - [info] Installed module: node-red-contrib-opcua
23 Jun 19:37:08 - [info] Added node types:
23 Jun 19:37:08 - [info] - node-red-contrib-opcua:OpcUa-Item
23 Jun 19:37:08 - [info] - node-red-contrib-opcua:OpcUa-Client
23 Jun 19:37:08 - [info] - node-red-contrib-opcua:OpcUa-Browser
23 Jun 19:37:08 - [info] - node-red-contrib-opcua:OpcUa-Server
23 Jun 19:37:08 - [info] - node-red-contrib-opcua:OpcUa-Endpoint
23 Jun 19:37:08 - [info] - node-red-contrib-opcua:OpcUa-Event
```

Ilustración 53. Información tras ejecución de comando de Node-RED

5. Abrir el navegador web y acceder a la dirección indicada <http://127.0.0.1:1880> para acceder a la interfaz gráfica de Node-RED.

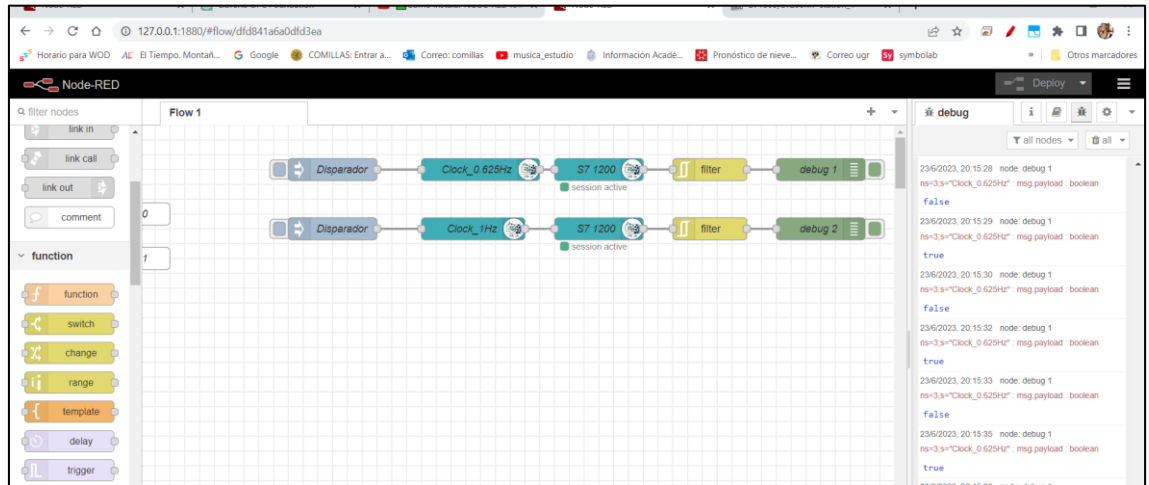


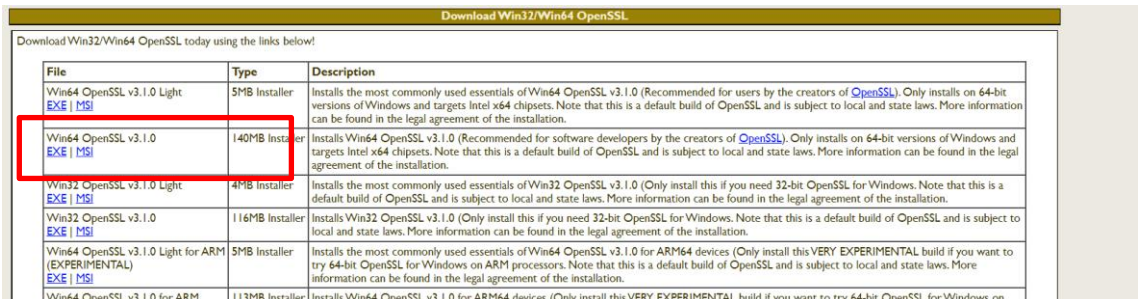
Ilustración 54. Interfaz gráfica de Node-RED

ANEXO III: CONFIGURACIÓN DE CERTIFICADOS CON OPENSSL

En este anexo se explica los pasos que se han seguido para instalar la herramienta de **OpenSSL** y con ella la configuración de **certificados** necesarios.

Paso 1: Descarga e instalación de OpenSSL

1. Primero se ha descargado el ejecutable de OpenSSL (versión v3.1.0) en el sitio web oficial: <https://slproweb.com/products/Win32OpenSSL.html>.



File	Type	Description
Win64 OpenSSL v3.1.0 Light EXE MSI	5MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.1.0 (Recommended for users by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.1.0 EXE MSI	140MB Installer	Installs Win64 OpenSSL v3.1.0 (Recommended for software developers by the creators of OpenSSL). Only installs on 64-bit versions of Windows and targets Intel x64 chipsets. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.1.0 Light EXE MSI	4MB Installer	Installs the most commonly used essentials of Win32 OpenSSL v3.1.0 (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win32 OpenSSL v3.1.0 EXE MSI	116MB Installer	Installs Win32 OpenSSL v3.1.0 (Only install this if you need 32-bit OpenSSL for Windows. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.1.0 Light for ARM EXE MSI	5MB Installer	Installs the most commonly used essentials of Win64 OpenSSL v3.1.0 for ARM64 devices (Only install this VERY EXPERIMENTAL build if you want to try 64-bit OpenSSL for Windows on ARM processors. Note that this is a default build of OpenSSL and is subject to local and state laws. More information can be found in the legal agreement of the installation.
Win64 OpenSSL v3.1.0 for ARM	113MB Installer	Installs Win64 OpenSSL v3.1.0 for ARM64 devices (Only install this VERY EXPERIMENTAL build if you want to try 64-bit OpenSSL for Windows on

Ilustración 55. ejecutable- OpenSSL v3.1.0

Tras la descarga se han de seguir las instrucciones específicas del instalador según el sistema operativo.

2. Para que OpenSSL funcione correctamente y sea accesible desde cualquier ubicación en la línea de comandos, se agrega la ruta del directorio de instalación como una variable de entorno en las “Propiedades del sistema”.

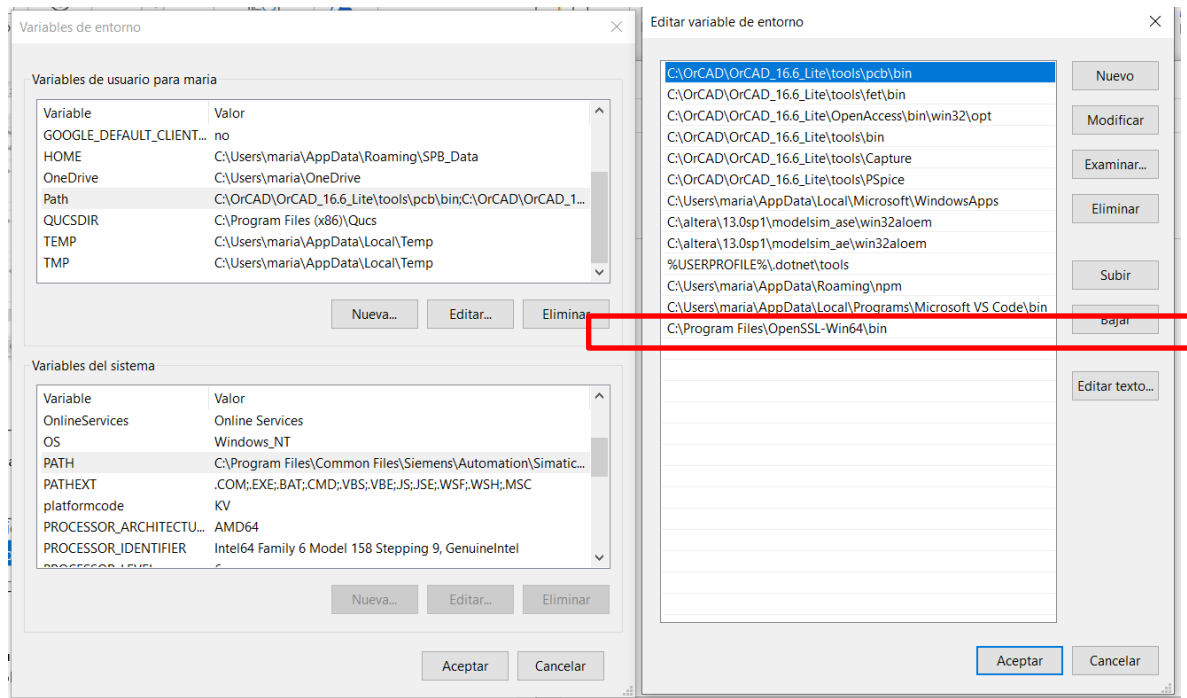


Ilustración 56. Editor variables del entorno - OpenSSL

3. Para comprobar que hemos instalado correctamente OpenSSL, basta con abrir una terminal de comandos y verificar la última versión instalada.

```
C:\Users\maria>openssl version
OpenSSL 3.1.0 14 Mar 2023 (Library: OpenSSL 3.1.0 14 Mar 2023)
```

Ilustración 57. Versión instalada de OpenSSL

Paso 2: Generar certificado

1. Para ello primero será necesario que se genere una clave privada. Se abre una terminal y se ejecuta el siguiente comando

```
openssl genrsa -des3 -out private_key.key 2048
```

Esto creará una clave privada con formato PKCS#1, donde el comando "des3" habilita la contraseña para la clave privada, "private_key.key" es el nombre que se

le da al archivo de la clave privada nueva y “2048” indica el tamaño de la clave en bits.

1. Creo una solicitud de firma de **certificado CSR**. Esta solicitud está asociada a la clave privada y posteriormente se transforma en un certificado. Usando el siguiente comando:

```
openssl req -new -key private_key.key -out CSR.csr
```

Tras ejecutar este comando se nos solicitará una información adicional, que ayudará a los usuarios a identificar el certificado y garantizar su fiabilidad.

```
-----  
Country Name (2 letter code) [AU]:ES  
State or Province Name (full name) [Some-State]:Mad  
Locality Name (eg, city) []:Madrid  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ICAI_TFM  
Organizational Unit Name (eg, section) []:ICAI_TFM  
Common Name (e.g. server FQDN or YOUR name) []:madiaz44  
Email Address []:  
  
Please enter the following 'extra' attributes  
to be sent with your certificate request  
A challenge password []:9746  
An optional company name []:Comillas
```

Ilustración 58. Información adicional para crear CSR

Como podemos ver en la anterior imagen, la información que se pide es por ejemplo el nombre de dominio, el país, estado, etc. Obtenemos ahora 2 archivos, uno que contendrá la clave privada (**.key**) y la solicitud de la firma de certificado (**.csr**). Ahora solo queda firmar la solicitud para poder transformarla en un certificado.

Paso 3: Firma del certificado

1. Tras crear la solicitud de **firma de certificado (CSR)**, debemos firmarla por la entidad emisora de certificados (CA). Creamos entonces una CA con el siguiente comando:

```
openssl req -new -newkey rsa:2048 -nodes -out CA_CSR.csr -keyout  
CA_private_key.key -sha256
```

2. Creamos un certificado para la CA privada. Se creará un certificado (.arm) que será útil para poder firmar CSR.

```
openssl x509 -signkey CA_private_key.key -days 360 -req -in CA_CSR.csr -out  
CA_certificate.arm -sha256
```

3. Utilizamos ahora el certificado CA para firmar la solicitud de firma de certificado que hemos creado antes.

```
openssl x509 -req -days 360 -in CSR.csr -CA CA_certificate.arm -CAkey  
CA_private_key.key -out certificate.arm -set_serial 01 -sha256
```

De esta manera se ha firmado la solicitud de firma de certificado y se ha obtenido un certificado nuevo.

4. Una vez obtenido el certificado, lo guardamos en un archivo con extensión “.pem”, al que llamaremos “certificateCopia.pem”.
5. Combinamos ahora la clave privada y el certificado en un solo archivo con el siguiente comando.

```
openssl pkcs12 -export -in certificateCopia.pem -inkey private_key.key -out  
mycert.pfx
```

Esto va a crear un archivo “mycert.pfx” que contendrá tanto la clave privada como el certificado. Los archivos que nos quedan de certificados y claves finalmente son los siguientes:

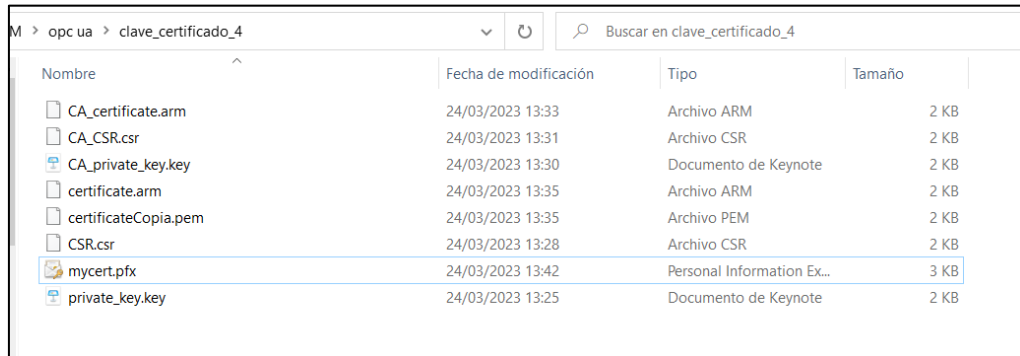


Ilustración 59. Archivos y certificados generados

Paso 4: Configuración del certificado en el servidor

Una vez obtenido el certificado firmado, se deberá añadir en la aplicación del servidor. Los pasos para añadir los certificados varían según la aplicación. En el caso de UA Expert se ha de importar para poder usarlo y conectar con servidores de OPC UA de forma segura.

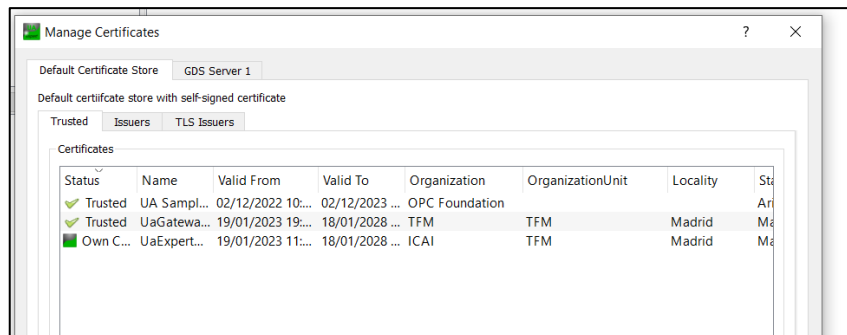


Ilustración 60. Administración de certificados UA Expert