

*GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y
AUTOMÁTICA*

Escuela superior de ingeniería y tecnología

**Simulación de SCADA de diversas funciones de control
y automatización de un buque para su empleo en las
prácticas de la asignatura Fundamentos y Equipos
Automáticos del Buque**

Trabajo de Fin de Grado



Autor:

Carlos Pérez Hernández

Tutor:

Ricardo Mesa Cruz

Cotutor:

Pedro Toledo Delgado

Septiembre 2020

INDICE

1.	Resumen	3
1.	Summary	4
2.	Introducción.	5
3.	Objetivos	8
4.	Tecnologías	9
4.1	Comunicación.	10
4.1.1	OPC-UA	11
4.2	Tecnologías de simulación y control de procesos	15
4.2.1	Selección de software de control y simulación de procesos.	16
4.3	Sistema Scada.	22
5.	Descripción del entorno de Twincat 3.	24
5.1	Twincat 3 eXtended Automation Engineering (XAE)	24
5.2	TE-2000 TC3 HMI	26
5.3	TF-6100	28
5.3.1	Twincat Connectivity Project	29
5.3.2	UAGateway Administration	30
5.3.3	Sample client	31
6.	Implementación de un prototipo.	32
6.1	Sistema de control de Nivel.	33
6.2	configuración previa de Twincat 3	33
6.3	Simulación de la planta	34
6.4	PLC simulado	35
6.5	Sistema SCADA+HMI	37
6.6	Servidor OPC-UA	39
6.6.1	Configuración del servidor OPC-UA	43
6.6.1.1	Conexión de la planta simulada con el servidor OPC-UA	44
7.	Conclusiones	49
8.	Coclusions	50
8.	Líneas futuras	51
	Bibliografía	52
	Fuente de las figuras	53
	Anexos	55

1. Resumen

En este proyecto se presenta una solución a las necesidades crecientes a nivel formativo en materia de automatización de procesos. Se trata de un complemento a las herramientas de aprendizaje por parte de alumnos como de profesionales en el sector para que dispongan de los recursos necesarios para llevar a cabo su formación en procesos industriales automáticos.

La importancia de este proyecto se fundamenta principalmente en el elevado coste que supone adquirir un prototipo de una planta física con todos sus componentes, además de los gastos de mantenimiento. Por tanto, nace la necesidad de usar alternativas más económicas y sencillas de llevar a la práctica en el campo de la docencia de la automatización industrial. Por otro lado, el surgimiento de la industria 4.0 (internet de las cosas, sistemas interconectados, big data entre otros) hacen que las necesidades aumentan y sea muy difícil darle respuesta con los modelos actuales. Poniendo de manifiesto la necesidad de buscar un modelo formativo que sea expandible y actualizable para dar una respuesta rápida a estos cambios que se producen en el sector.

El proyecto da una solución a este problema mediante una simulación de una determinada planta y su control mediante un PLC simulado. Además, se permite la visualización de los distintos parámetros que genere el control de la planta mediante un sistema SCADA. Asimismo, se logra una comunicación entre los distintos elementos que componen la solución, de tal forma que permite la sustitución de algunos de los componentes elegidos por otros sin que esto afecte al resto de los elementos. Igualmente, pudiéndose añadir componentes extra sin que esto suponga un problema a la hora de establecer la comunicación. De esta forma se logra un proyecto ampliable que se adapte a las necesidades formativas que se presenten, añadiendo los elementos que se requieran al sistema que se plantea.

1. Summary

The most significant conclusion that has been reached in this project is that the multiple tools available to us thanks to the advancement of industrial technologies allow us to complement the training of students in a relevant way. As we have seen, these technologies can be adapted to the needs of students and allow the creation of complex industrial systems without the need for a high economic cost.

In this project, it has been explained how to use the OPC-UA communication standard to connect plant simulation platforms with an industrial automation programming platform and a SCADA system. With this solution the aim is to achieve control and monitoring of an industrial plant, which allows students to become familiar with an industrial environment. It should be noted that the interesting thing about this system is that it has been made following an open communication standard, so any device or application adapted to work with this technology can be connected to this server, thus allowing the project presented here to be expanded. or adapt it to the needs that arise. Likewise, the characteristics of the adoption by the industries of standardized communication protocols have been presented, and what it is understood that their future will be.

The project presented satisfies the objectives established in section 3 of this document, since it has revealed a new training model in matters related to industrial automation. More specifically, a complement to learning is presented that allows greater versatility when carrying out practices of these subjects. In contrast, this project represents a beginning in what is considered to be the final model that reaches the students. As explained in section 8 of this document, this project can be expanded and further adapted to the needs of the students. Therefore, it is a precedent that serves to show the benefits of using these technologies and how important it can become if you expand on this idea even more.

2. Introducción.

El sector de la producción se basa fundamentalmente en la política de reducir costes y aumentar la producción. Por este motivo surge la necesidad de mejorar cada vez más los elementos que componen el sistema productivo, apareciendo nuevas tecnologías que de una forma u otra cambian el modelo productivo y lo hacen más eficiente. En este ámbito de mejora continua de los sistemas de producción surge la industria 4.0.

El nacimiento de la industria 4.0 supone la aparición de nuevos modelos de control industrial, cuyo principal propósito es digitalizar la industria y cualquier ámbito relacionado con ella. De esta forma se mejoran la eficiencia de las plantas industriales mediante el control total de todos los medios que la componen. Esto se explica debido al aumento en la interconectividad de todos los departamentos de las empresas que permiten que la información recogida en cada uno de los elementos que conforman los sistemas de producción (instrumentación, PLC, sistemas de análisis de datos, etc.), mantengan una comunicación directa. Esta mejora en la conectividad genera un gran volumen de datos (big data) que, junto con las herramientas analíticas modernas, basadas en inteligencias artificiales, permiten tomar las decisiones correctas para mejorar la eficiencia de los procesos de producción, disminuyendo los costes y aumentando la calidad del producto final. . Es lo que se conoce como el internet de las cosas industrial o IIoT industrial por sus siglas en inglés. A esto se une el avance en la equipos de instrumentación y control que componen la planta)

En la figura 2.1 que se presenta a continuación, se establece la pirámide de automatización en la que se recogen los distintos niveles tecnológicos que componen a un sistema industrial moderno (industria 4.0). Esta pirámide se caracteriza porque a medida que se avanza en ella aumenta el nivel de decisión que se posee, abarcando desde los operarios de la planta industrial hasta el nivel ejecutivo de la empresa.

- Nivel de campo: Engloba todos los sensores y actuadores que posee la planta industrial.
- Nivel de control: Engloban los PLC (computadores lógico-programables) que realizarán el control sobre la información recibida de los sensores y actuadores

GIEIA- trabajo de fin de grado

- Nivel de supervisión: engloban el SCADA y el HMI. El primero permite la supervisión, control y adquisición de datos provenientes de los niveles de control y de campo. El segundo actúa como interfaz entre Hombre-máquina para visualizar lo ejecutado por SCADA.
- Nivel de operación: engloban los sistemas MES que monitorizan y registran todos los procesos de producción de la planta dirigidos a aumentar la eficiencia y disminuir los costes.
- Nivel de información: Engloban los sistemas ERP y a diferencia del anterior nivel monitorizan y registran toda la información de la distribución y recursos humanos. Normalmente el sistema MES suelen estar integrado dentro del sistema ERP.
- Comunicación: Es uno de los pilares básicos de los sistemas productivos. Mantener una rápida comunicación vertical resulta fundamental si se quiere mejorar la toma de decisiones.

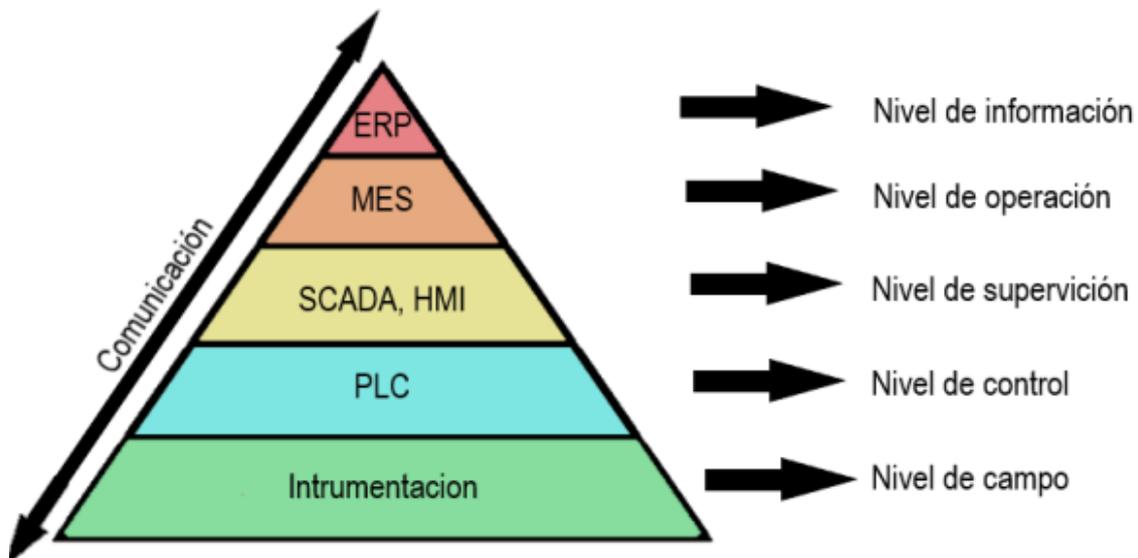


Figura 2.1. Pirámide de la automatización.

Desde el punto de vista docente se puede concluir que la industria 4.0 pone de manifiesto la necesidad de crear herramientas formativas que permitan a los alumnos universitarios familiarizarse con estas tecnologías (Sistemas MES, Scada, ERP, estándares de comunicación...). Además, debido a la rapidez con la que se producen avances tecnológicos en este sector, es importante buscar un modelo formativo que sea expandible y actualizable para dar una respuesta rápida a estos cambios que se producen en el sector.

GIEIA- trabajo de fin de grado

Este proyecto se desarrolla con el fin de ser un complemento a las herramientas de aprendizaje por parte del alumnado con el fin de que dispongan de los recursos necesarios para llevar a cabo su formación en procesos industriales automáticos, sin necesidad de utilizar una planta física. Para ello, se hará uso de un simulador de planta y un simulador de autómeta industrial. Los alumnos tendrán la tarea de programar el controlador y comprobar cómo se comporta el sistema. Además, se presenta un sistema SCADA que permite supervisar los procesos.

La comunicación entre planta, autómeta y SCADA se realizará mediante un estándar de comunicación abierto y multiplataforma. De esta forma se consigue que el proyecto no esté ligado a una única plataforma de simulación de planta o controlador en específico, pudiendo utilizar tecnologías de distintos creadores o intercambiar tanto la planta como el PLC simulado por elementos físicos independientes del fabricante sin que esto suponga ningún tipo de problema en el resto de componentes del proyecto.

3. Objetivos

El objetivo principal de este proyecto es ofrecer una herramienta de simulación de un entorno industrial.

Para conseguir este objetivo general, se plantean los siguientes subobjetivos:

1. Disponer de una herramienta de simulación de un PLC.
2. Disponer de una herramienta de simulación de una planta.
3. Disponer de un sistema SCADA+HMI.
4. Lograr un sistema de comunicación entre los componentes.

Con estos objetivos se espera conseguir que el proyecto pueda:

1. Servir de herramienta educativa para los estudiantes en materias relacionadas con la automatización industrial.
2. Ser un complemento al aprendizaje, motivando a los alumnos a realizar sus propias simulaciones y control de sistemas automáticos.
3. Suponer un ahorro económico, en un futuro, para la formación en el campo de la automatización industrial, puesto que ya no será imprescindible contar con plantas físicas para la realización de estas prácticas.
4. Servir de precedente en futuros proyectos que requieran de esta tecnología, motivando a ampliar el proyecto que se presenta.

Para cumplir con los objetivos, se plantean además los siguientes requisitos no funcionales:

- Las distintas tecnologías que se presenten deben tener una licencia gratuita. Para así facilitar la accesibilidad a los alumnos o su implementación en el centro docente.

GIEIA- trabajo de fin de grado

- Crear un proyecto ampliable, adaptándose a las necesidades docentes que se presenten.

4. Tecnologías

Este proyecto se centrará en simular los 3 primeros niveles de la pirámide de automatización: el nivel de campo, de control y de supervisión. La arquitectura de la solución es la que se muestra en la Figura 4.1.

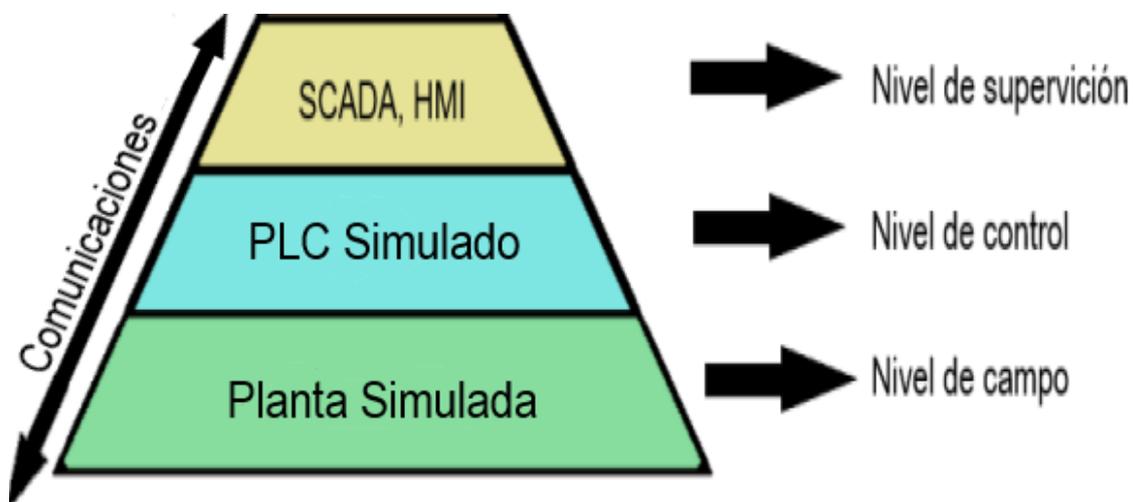


Figura 4.1. Partes que componen la solución.

Para realizar este proyecto se hará uso de un simulador de plantas industriales, que mantendrá una comunicación con el controlador y con un sistema SCADA.

En cuanto al software requerido, deberán incluir una plataforma de control y simulación de plantas, una plataforma HMI+SCADA y un sistema de comunicación entre los distintos elementos.

Por tanto, este apartado se centra en escoger una plataforma de simulación de plantas, de simulación de PLC, un sistema Scada y un medio de comunicación entre los distintos elementos. Para ello se realiza un estudio de las distintas plataformas existentes seleccionando la que mejor se ajuste a este proyecto.

4.1 Comunicación.

La comunicación es de gran importancia en este proyecto, debido a que tiene el papel de realizar el intercambio de información entre una plataforma y otra. Por tanto, es uno de los pilares básicos en el que se asienta el desarrollo del proyecto.

En vista de los objetivos marcados en el apartado 2 y teniendo en cuenta la arquitectura de la solución, la comunicación debe satisfacer las siguientes características:

1. Mantener una rápida comunicación entre los distintos elementos que componen la solución. Esto es debido a la necesidad de una respuesta rápida sobre la planta por parte del controlador.
2. Permitir la conexión en una misma red de distintas plataformas con independencia del desarrollador. Logrando de esta forma elegir la mejor opción respecto a los demás elementos que componen la arquitectura, sin necesidad de adaptar las comunicaciones entre sí. Por tanto, este protocolo de comunicación debe estar extendido, estandarizado y ser de actualidad.
3. Gratuito o en su defecto suponer el menor coste económico.

Para satisfacer la característica 2 se debe emplear un estándar de comunicación industrial. Un estándar de comunicación permite entre otras cosas, una comunicación total entre plataformas o equipos de distintos desarrolladores eliminando la necesidad de adaptar la información en cada uno de los receptores.

En vista de las características descritas, la mejor alternativa a la hora de realizar este proyecto es la implementación de un estándar de comunicación OPC-UA, que es el estándar de referencia industrial. En el siguiente apartado se pone en relevancia la importancia de la aparición de este estándar de comunicación, así como las características que posee y por qué es la mejor alternativa para el desarrollo de este proyecto.

4.1.1 OPC-UA

Con el propósito de comprender lo que supuso la aparición del estándar de comunicación OPC-UA en el ámbito de las comunicaciones industriales se ha realizado la Figura 4.2.

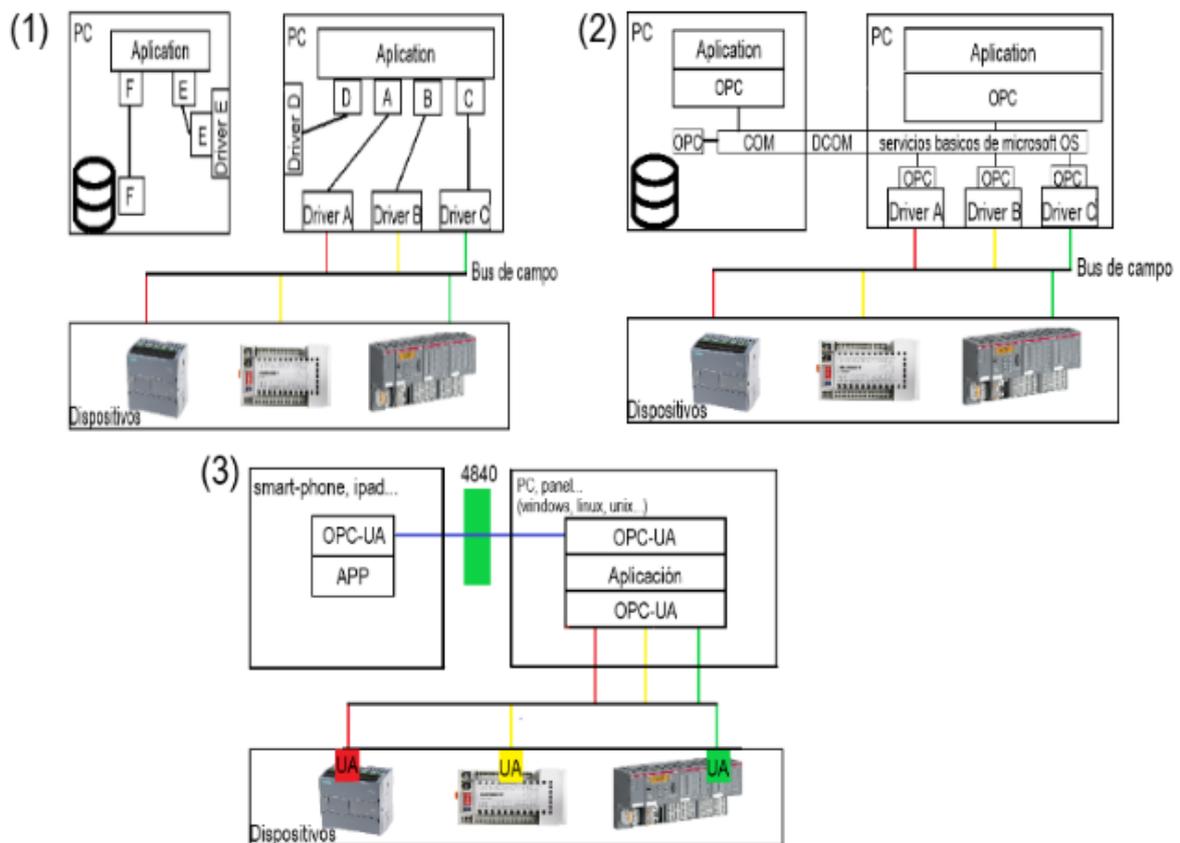


Figura 4.2. Tipos de redes de comunicación industrial. (1) Sin estandarización. (2) Estableciendo un estándar de comunicación OPC. (3) Estableciendo un estándar de comunicación OPC-UA.

En la imagen superior izquierda (1) de la Figura 4.2 se muestra cómo se realizaría la conexión en red sin existir una estandarización de protocolos. En este caso, cada uno de los equipos y aplicaciones que componen la red utilizan sus propios protocolos de comunicación (los llamados protocolos propietarios). De esta manera tanto los PLCs de distintos fabricantes, como las aplicaciones

(procesamiento de datos, sistemas Scada, bases de datos, etc.) necesitan de costosos drivers propietarios específicos para adaptar los datos a las demás aplicaciones o equipos que componen la red. Además, estos drivers no son reutilizables y si se intercambiase alguno de los elementos que componen la red por nuevos equipos o aplicaciones habría que adquirir nuevos drivers para establecer la comunicación. Esto lleva consigo un elevado coste económico y una baja eficiencia de la red ya que cada uno de los autómatas debe establecer comunicación por separado con las distintas aplicaciones, y estas aplicaciones al conectarse con otros equipos deben establecer drivers de comunicación adicionales tal y como se observa gráficamente en la figura 4.2. Por lo que traería consigo que dos o más aplicaciones pueden estar solicitando los mismos datos al PLC simultáneamente y cada una de estas aplicaciones esté adaptando la información por separado a través de los drivers. Este problema hace que aumente considerablemente la carga de trabajo en el dispositivo llegando a colapsar si el número de peticiones es superior al número de conexiones simultáneas que soporta. Por lo tanto, aparece la necesidad de establecer una capa intermedia entre los distintos elementos que adapte la información, eliminando la necesidad adquirir drivers específicos y estandarizando las comunicaciones. [1], [2]

OPC clásico, cuyo funcionamiento se muestra gráficamente en la imagen superior derecha (2) de la figura 4.2, resuelve estos problemas introduciendo una capa intermedia, el servidor OPC. El servidor OPC permite la encapsulación de la información, accediendo el cliente OPC a ésta a través de una interfaz. De esta forma se estandarizan los protocolos de comunicación, permitiendo a las tecnologías de distintos desarrolladores compartir información y eliminar los costosos drivers propietarios antes mencionados. Además, se reduce considerablemente la carga de trabajo de los dispositivos, ya que un conector OPC requiere una única conexión con la fuente de datos, pudiendo compartir esta información con múltiples aplicaciones simultáneamente. Otro problema que se soluciona es que aumenta la vida útil de las aplicaciones y equipos que componen la red, debido a que al incluir la tecnología OPC cualquier elemento antiguo que componga la red puede comunicarse con softwares nuevos. Por otro lado, este estándar de comunicación presenta la desventaja de que utiliza la tecnología COM/DCOM de Windows (tecnología que permite desarrollar componentes de software distribuidos sobre varias computadoras) para el flujo de datos. COM/DCOM presenta algunos problemas y actualmente se encuentra en desuso. Algunos de estos problemas son su difícil configuración, sus largos tiempos de espera y su problema con el Firewall, se requerían muchos puertos abiertos para realizar la conexión. Esto se une a que DCOM, al ser una tecnología propia de Windows, el servidor OPC necesitaba de este sistema operativo para funcionar. Por tanto, no resulta eficiente implementarlo si lo que se quiere es comunicar con servidores o clientes remotos o que utilicen sistemas operativos independientes de Windows. Todos estos problemas se solucionaron con el desarrollo de OPC-UA.[1], [2]

El funcionamiento de OPC-UA se muestra gráficamente en la imagen inferior (3) de la figura 4.2. OPC-UA es la evolución de los servidores OPC clásicos y nace de la necesidad de mover datos entre múltiples dispositivos (páginas web, dispositivos móviles...). Esta tecnología deja de utilizar la tecnología DCOM de Windows y por lo tanto permite la comunicación con cualquier dispositivo con independencia del sistema operativo incluso si este no existiera, sirviendo de esta forma para todos los elementos de un sistema, desde un sensor hasta una aplicación en la nube. De esta característica aparece una nueva posibilidad y es la de poder realizar sistemas embebidos, ya que no es necesario disponer de un ordenador intermedio con un sistema operativo específico para el intercambio de datos, sino que se posibilita que esta función la realice un procesador. Otra característica es que este estándar de comunicación utiliza un protocolo TCP/IP binario pensado para datos que requieren un procesamiento rápido sobre todo a nivel de control y un protocolo HTTP más XML para mover los datos de una forma más lenta, pero en grandes paquetes, pensado para aplicaciones de análisis de datos (MES, bases de datos...). Asimismo, los servidores OPC-UA, a diferencia de OPC, requieren de un único puerto para pasar los datos, lo que simplifica enormemente la conexión desde otros equipos a este servidor frente al OPC clásico. [1],[2],[3]

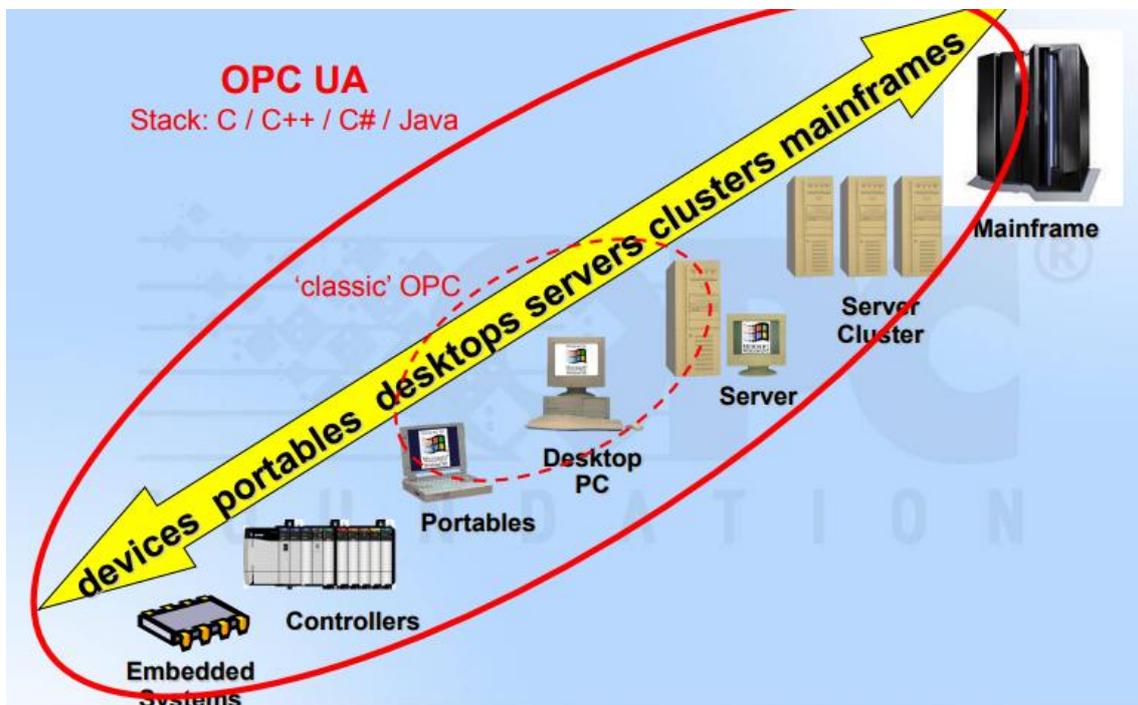


Figura 4.2. Comparación gráfica entre los distintos niveles de OPC clásico frente a la tecnología OPC-UA

Las ventajas de utilizar este sistema en cuanto a la realización de este proyecto se resumen en el siguiente listado:

- Es un protocolo de comunicación estandarizado. No está sujeto a un sistema operativo ni hardware específico, desde sensores hasta servidores basados en la nube.
- La mayoría de software de control y simulación de plantas modernas incluyen esta tecnología.
- Permite la conexión entre redes e incluso a través de internet
- Posee retro compatibilidad con sistemas OPC antiguos
- Posee una fácil configuración y mantenimiento.
- Presenta la posibilidad de conectar distintos PLC, programas de simulación o sistemas SCADA en una única red con independencia del fabricante o desarrollador. De esta manera se permite la comunicación con otras aplicaciones de recogida y procesamiento de datos tales como sistemas MES, bases de datos etc. Quitando la necesidad de la instalación de drivers que realicen este proceso, actuando como una capa intermedia entre los distintos elementos que conforman la comunicación. Este servidor es el que se encarga de recibir los datos y distribuirlos a los clientes.
- Es gratuito.
- Se trata de un protocolo de comunicación en constante crecimiento y uso. También presenta continuas actualizaciones que solucionan errores e introducen mejoras.
- Permite una integración total de todos los sistemas productivos.

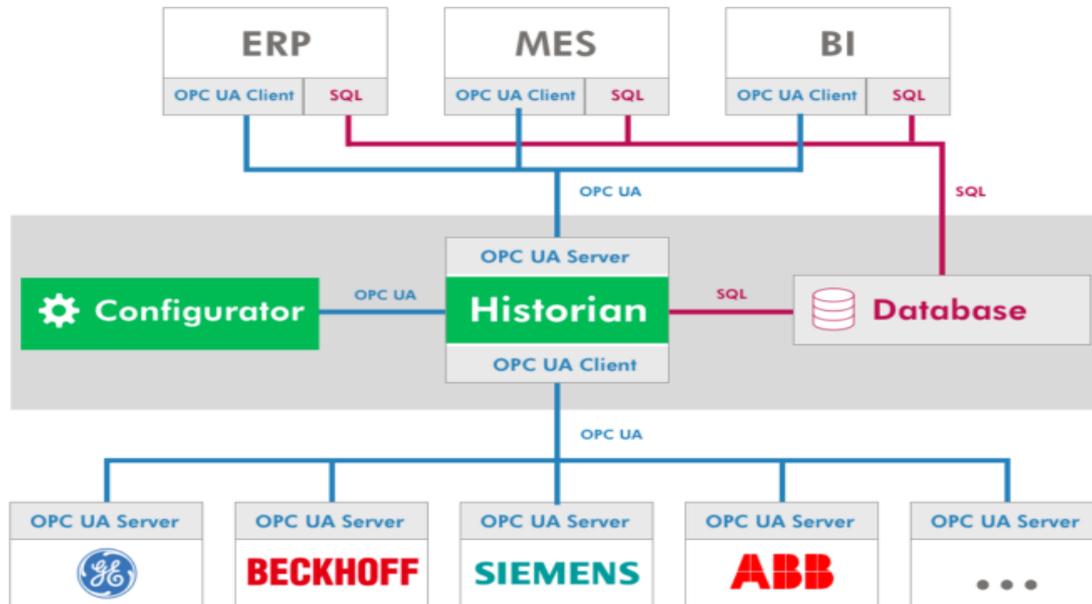


Figura 4.3. Diagrama de conexiones de un servidor OPC-UA.

4.2 Tecnologías de simulación y control de procesos

En cuanto a las tecnologías de control y simulación de plantas, se fijan los siguientes requisitos mínimos que se tendrán en cuenta a la hora de realizar este estudio:

- Disponer de las herramientas necesarias para realizar las labores de control y simulación de plantas en la misma plataforma, o en su defecto que permita la comunicación con un tercer programa que realice la labor de simulación de la planta.
- El requerimiento en cuanto a software debe ser bajo.
- El poder implementar una comunicación OPC-UA.
- Se encuentre disponible en la Universidad de la Laguna, para aquellos alumnos que no dispongan de ordenador propio puedan acceder al programa.
- Desarrollado por una compañía sólida que garantice una buena gestión de los fallos que se puedan presentar en el software.

GIEIA- trabajo de fin de grado

- El nivel de integración en Windows 10. Se valorará que no se tenga que hacer uso de máquinas virtuales para acceder al entorno.
- Posea su propio software HMI, facilitando la integración con el PLC.
- Presente múltiples lenguajes de programación, preferiblemente el estándar IEC-61131-3 (Lenguaje de diagrama de contactos, lenguaje de lista de instrucciones, graficet y texto estructurado).
- Ser gratuito.
- Software reciente y sujeto a soporte técnico.

4.2.1 Selección de software de control y simulación de procesos.

En este apartado se llevará a cabo una comparativa entre los distintos softwares de control y simulación de plantas. Se presentan en este estudio las ventajas y desventajas de estos programas para realizar este proyecto, en función de los requisitos mínimos fijados en el apartado anterior.

El ISPSOft de delta posee las siguientes características:

- Permite el control y la simulación de la planta en la misma plataforma.
- Posee múltiples herramientas para controlar o simular sistemas automáticos ya que incorpora el lenguaje de programación estándar internacional IEC 61131.
- Es multitarea y permite la ejecución de proyectos paralelamente.
- No requiere de un hardware sofisticado, ni tampoco su HMI.
- Posee programación cíclica basada en tareas que resulta bastante útil a la hora de simular plantas.
- Posee su propio software HMI (dopsoft)

GIEIA- trabajo de fin de grado

- Es gratuito, por lo que permite a los alumnos utilizar el software sin adquirir el uso de licencia. Además, se podrá instalar la plataforma en los ordenadores del centro para el alumnado que requiera su uso.
- Permite el control y la simulación de la planta en la misma plataforma y no requiere de software adicional para simular el PLC.
- Es un software actual que se encuentra sometido a soporte técnico.
- El nivel de requerimientos en cuanto al software es bajo.

Posee la limitación de que sólo pueden implementarse servidores OPC, por tanto, el proyecto estará limitado por el uso de esta tecnología.

CoDeSys V3 presenta las siguientes ventajas:

- Posee su propio software HMI.
- Permite crear servidores OPC-UA.
- Es multitarea y permite la ejecución de proyectos paralelamente.
- Posee una programación cíclica basada en tareas lo que incrementa las posibilidades a la hora de simular.
- Es gratuito.
- El nivel de requerimiento en cuanto a software es bajo.
- Posee además múltiples herramientas para simular o programar PLC ya que incorpora el lenguaje de programación estándar internacional IEC 61131.
- Es posible conectar con Matlab y Simulink, lo que aumenta considerablemente las posibilidades a la hora de simular.
- La CoDeSys Automation Alliance (CAA) nace con el objetivo final de ofrecer un conjunto de dispositivos para diferentes tareas industriales con un único instrumento de programación. Esta alianza cuenta con cerca de 100 miembros. Por tanto, si los alumnos aprenden el entorno de CoDeSys, tendrán las facultades necesarias para programar múltiples PLC. Programas como Twincat 3 que se plantean en esta comparativa están basados principalmente en esta compañía.

GIEIA- trabajo de fin de grado

- Permite el control y la simulación de la planta en la misma plataforma y no requiere de software adicional para simular el PLC ni drivers específicos.

Presenta el inconveniente de que su software SCADA+HMI no permite establecer estándar de comunicación OPC-UA. Sin embargo, su software de programación de PLC si, por lo tanto, se tendría que hacer uso de programas SCADA de terceros para resolver esta limitación.

El Somachine Basic de la compañía Schneider presenta las siguientes ventajas:

- Posee lenguaje de programación lenguaje de diagrama de contactos, lenguaje de lista de instrucciones y grafcet.
- Posee su propio simulador de PLC integrado.
- Sus requerimientos en cuanto a hardware son bajos.
- Es gratuito.

Posee múltiples desventajas:

- No dispone de HMI ni SCADA gratuito, por lo tanto, habría que optar por otro desarrollador.
- No es posible establecer una conexión OPC-UA.
- No es multitarea, y solo permite crear un programa principal.
- No dispone de texto estructurado para la programación de PLC, lo que le resta funcionalidad.
- Carece de la posibilidad de instalar complementos y escasos bloques funcionales por lo que las posibilidades de simulación son muy limitadas.
- Entorno establecido solo para programar un modelo de PLC.

El STEP7 de Siemens tiene como ventajas para la realización de este proyecto:

- Es el sistema de programación de PLC más usado en el mundo, por lo que goza de bastante contenido en la red.
- Es un sistema multitarea por lo que permite lanzar más dos programas a la vez.
- Sus requerimientos en cuanto a hardware son muy bajos.
- Posee múltiples herramientas para simular o programar PLC ya que incorpora el lenguaje de programación estándar internacional IEC 61131.
- Posee multitud de bloques funcionales que ayudan no solo a la hora de programar PLC, si no a la hora de simular plantas.
- Permite crear servidores OPC-UA.
- Posee un sistema SCADA propio, Wincc.
- Es posible conectar con Matlab y Simulink, lo que aumenta considerablemente las posibilidades a la hora de simular.
- Se encuentra disponible en los equipos del centro donde se presenta el trabajo.

Sin embargo, la forma en la que está pensado el software no es la idónea para realizar este proyecto ya que por su estructura está más enfocado a la simulación o programación de PLC que a la simulación de procesos, para este caso habría que hacer uso de un tercer programa que se encargue de realizar la simulación de la planta. Aunque se encuentre disponible en el centro, la licencia no es gratuita, requisito fundamental que se marcó en los objetivos de este proyecto.

En cuanto a Twincat 3 de Beckhoff las ventajas que presenta son:

- Utiliza el entorno de programación de Microsoft Visual Studio por lo que la integración en Windows es prácticamente total.
- Permite el control y la simulación de la planta en la misma plataforma (Microsoft Visual Studio) y no requiere de software adicional para simular el PLC.

GIEIA- trabajo de fin de grado

- Su Software HMI se encuentra también integrado en Microsoft Visual Studio, por lo que resulta más cómodo para la programación.
- Es totalmente gratuito, por lo que favorece que el alumnado pueda hacer uso de esta plataforma desde sus ordenadores personales.
- El nivel de requerimientos de hardware es bajo.
- Incorpora un lenguaje de programación basado en CoDeSys, que utiliza las normas de programación estándar internacional IEC 61131-3. Además de estos lenguajes incluye gráfico de funciones continuas (CFC), y diagrama de estado UML (SC).
- Permite la programación en C + +.
- Es posible conectar con Matlab y Simulink.
- Permite crear servidores OPC-UA.
- Posee multitud de bloques funcionales y complementos gratuitos que ayudan no solo a la hora de programar PLC, si no a la hora de simular plantas.
- Posee programación cíclica basada en tareas que resulta bastante útil a la hora de simular plantas.
- Se encuentra disponible en los equipos del centro donde se presenta el trabajo.

Algunas desventajas son:

- Presenta librerías de pago.

A modo de resumen de lo enumerado anteriormente se ha realizado la siguiente tabla comparativa entre los distintos software seleccionado:

	Twincat 3	Step 7	Isp-Soft	CoDeSys V3	Somachin e Basic
Control y simulación en la misma plataforma	Si	No	Si	Si	No
Gratuito	Si	No	Si	Si	Si
Nivel de integración en Windows 10	Muy Alto	Medio (windows XP)	Alto	Alto	Alto
Requerimientos de hardware	Bajos	Bajos	Bajos	Bajos	Bajos
Posee su propio software HMI	Si	Si	Si	Si	No
Disponibilidad en el centro	Si	Si	No	No	No
Desarrollado por una compañía sólida	Si	Si	Si	Si	Si
Lenguajes de programación	Muy alto	Alto	Muy alto	Muy alto	Muy bajos
Software reciente y sujeto a actualizaciones	Si	Si	Si	Si	Si
Posibilidad de establecer protocolo OPC UA	Si	Si	No (solo servidores OPC)	Si	No

Tabla 4.1. Comparativa de las distintas tecnologías de simulación

Por lo tanto, una vez desarrollada la comparativa dos posibles opciones serían Twincat 3 y CoDeSys. Step 7 queda descartado debido a que es un software de pago y no dispone de licencia gratuita para estudiantes, por lo tanto iría en contra de los requisitos que se han fijado. ISPsoft no es una buena opción al no poder crear servidores OPC-UA y Somachine Basic es un programa muy limitado (si queremos contar con un software de este desarrollador deberíamos utilizar el Somachine V4.1, pero no dispone de licencia gratuita. En cuanto a Twincat 3 y CoDeSys poseen entornos de programación muy similares debido a que Twincat 3 está fuertemente inspirado en CodeSys. Sin embargo, presentan algunas diferencias que hacen que Twincat 3 se ajuste mejor a este proyecto: Se encuentra acondicionada un aula en el centro para este programa que cuenta con el software y con autómatas físicos de este fabricante (aula BECKHOFF en el ala izquierda de ETS de informática), presenta un mayor número de librerías que las ofrecidas por CodeSys, presenta la posibilidad de programar en C++ e incluye con su instalación programas como BodePLot, Scope View, etc.

Por tanto, dado todas sus ventajas Twincat 3 es el software seleccionado para realizar este proyecto

4.3 Sistema Scada.

Un sistema de supervisión, control y adquisición de datos (SCADA) permite controlar y supervisar procesos industriales a distancia. Este software accede a los datos que generan los PLCs y sensores creando una plataforma que adapta y moldea los datos para que sean fácilmente interpretables por los usuarios. De esta forma, se puede llevar a cabo un control en tiempo real sobre el funcionamiento de la planta, permitiendo una rápida intervención en caso de que sea necesario.

Para el sistema SCADA se usará el programa TE2000-HMI ofrecido por Beckhoff. Este programa lo hemos elegido debido a:

- La fácil integración que posee con el software seleccionado en el apartado anterior, Twincat 3.
 - Los dos se encuentran implementados en el entorno de Microsoft Visual Studio.
 - Son del mismo desarrollador y la comunicación entre PLC y el sistema SCADA es casi automática.

GIEIA- trabajo de fin de grado

- Es multiplataforma (móvil, Tablet, ordenador o monitor).
- Se permite la conexión a múltiples sistemas run-time.
- Utiliza un lenguaje de programación basado en HTML5 y JavaScript, por lo que se adaptan automáticamente a la resolución, el tamaño y la orientación de la pantalla.
- Permite la conexión con otros sistemas SCADA, PLC o plataformas de simulación vía OPC-UA, por lo que es compatible con múltiples plataformas de programación de PLC, así como programas de simulación de plantas.
- Es un software completamente gratuito.

5. Descripción del entorno de Twincat 3.

5.1 Twincat 3 eXtended Automation Engineering (XAE)

El programa Twincat 3 convierte cualquier PC compatible en un controlador en tiempo real permitiendo la inclusión de múltiples proyectos de PLC en un único sistema run-time. Pone a disposición del usuario múltiples librerías, bloques funcionales y permite programar en C + +, lo que incrementa las posibilidades de simulación, además de que es posible conectar con Matlab y Simulink [5]. Por lo tanto, es el programa idóneo para simular tanto el PLC como la planta. Este programa permite incorporar el protocolo OPC-UA y por tanto nos permite acceder a variables en programas de simulación de terceros y ser capaz de controlarlas desde el software.

Twincat 3 está integrado en el entorno de Microsoft Visual Studio, en los que uno o más proyectos se integran dentro de una solución. Uno de estos proyectos será Twincat XAE Project. Por otro lado, en una misma solución se pueden englobar múltiples proyectos, siendo posible programar el PLC, el sistema SCADA (siempre que utilice la plataforma de Microsoft) y la configuración del servidor OPC-UA de manera conjunta en una misma solución.

Los requisitos mínimos del sistema para que este software funcione correctamente son los siguientes [5]:

- Sistema operativo windows 7 service pack 1, windows Server 2008 R2 SP1, windows Server 20102 (R2) o windows 10.
- Procesador de 1,8 GHz de doble núcleo.
- 4 GB de RAM.
- 10 GB de disco duro.

Se recomienda el uso de un disco duro tipo SSD, pero admite el uso de HDD.

- Tarjeta gráfica de al menos 720p.

GIEIA- trabajo de fin de grado

Por tanto, como vemos los requerimientos del software en base a su instalación son bajos, y puede funcionar en cualquier ordenador de gama media baja sin problemas.

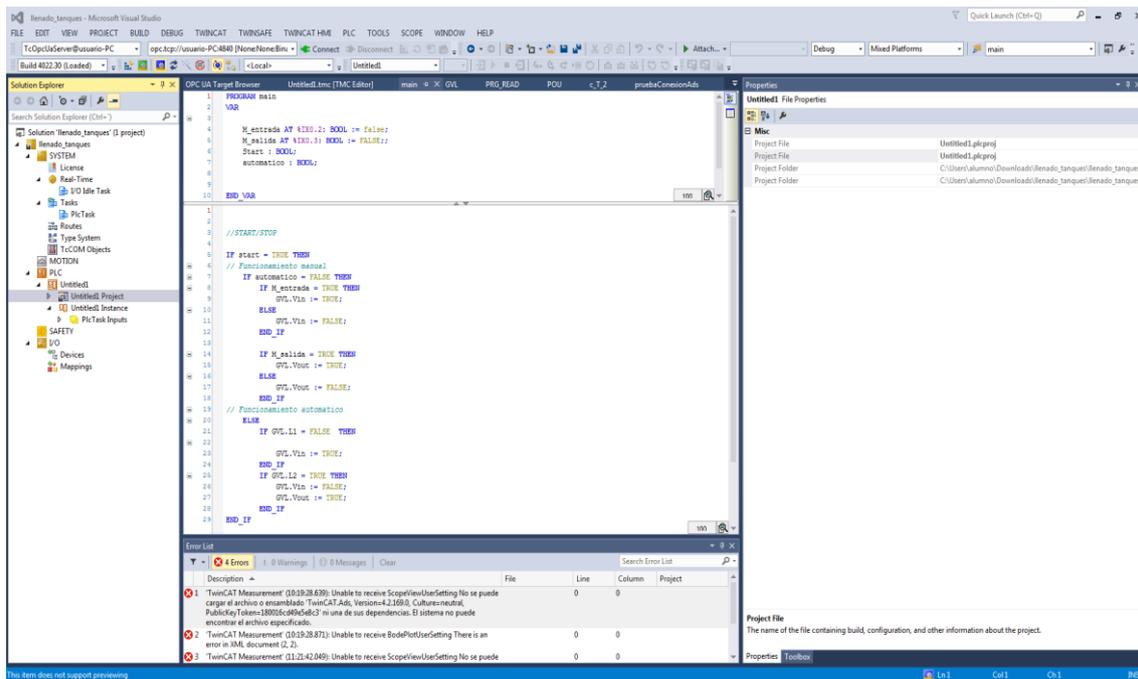


Figura 5.1. Entorno de Twincat 3

Un proyecto de Twincat 3, fijándonos en la barra de soluciones (parte izquierda de la figura) y omitiendo información no relevante para el desarrollo de este proyecto, presenta las siguientes opciones:

1. En la pestaña de sistema (system) se nos muestra información sobre la versión de Twincat 3 que estamos utilizando, así como herramientas de seguridad, también se nos presenta el listado de licencias del que disponemos para llevar a cabo el proyecto. En la pestaña Real-Time se configura los recursos del equipo para que el programa pueda funcionar en tiempo real. Se debe dedicar al menos un núcleo al funcionamiento del programa. En la pestaña de Task (tareas) se establece el tiempo de ciclo de los programas que se ejecuten y estén asociados a esa tarea, siendo el mínimo de 1 ms y el máximo de 65535 ms. En las pestañas de routers se muestran los distintos equipos que están conectados a la misma red.
2. En este apartado, llevaremos a cabo la programación del PLC. Este programa dispone de los siguientes elementos para llevar a cabo esta programación:

- Pou (Unidad de Organización del Programa): Se engloban dentro de este tipo los programas, funciones y bloques de funciones. Este software incluye el estándar internacional IEC 61131-3 y, por tanto, nos permite programar en diagrama de contactos, diagrama de bloques funcionales, texto estructurado, lista de instrucciones y bloques de función secuenciales. Posee lenguajes de programación de gráfico de funciones continuas (CFC), y diagrama de estado UML (SC).
 - DUT: Permite definir al usuario sus propios tipos de datos, como estructuras numéricas y referencias.
 - GVL: Permite al usuario declarar variables que serán accesibles desde cualquier ubicación del programa.
3. Seguridad: Permite editar la lógica de seguridad de los PLC.
 4. I/O (Configuración de las entradas y salidas): Además de conectar salidas físicas permite también la conexión de drivers y módulos adicionales.

5.2 TE-2000 TC3 HMI

TE-2000 TC3 HMI es un programa SCADA+HMI que permite la supervisión y control de procesos de manera remota a través de un servidor vía OPC-UA , entre otros, actuando como cliente. Al igual que Twincat 3 está desarrollado en Microsoft Visual Studio. Está basado en JAVASCRIPT y en HTML5 y permite la integración total desde múltiples dispositivos [6]. Mediante el uso de pulsadores y gráficas de datos nos permite acceder a la información que proporciona el PLC y de esta forma controlar y visualizar la planta, igualmente permite al usuario crear sus propios controladores.

Este programa mantendrá una comunicación vía OPC-UA con el PLC. En este software se podrá además de controlar y supervisar, cambiar las variables de simulación y ver cómo se comporta el sistema ante estos cambios.

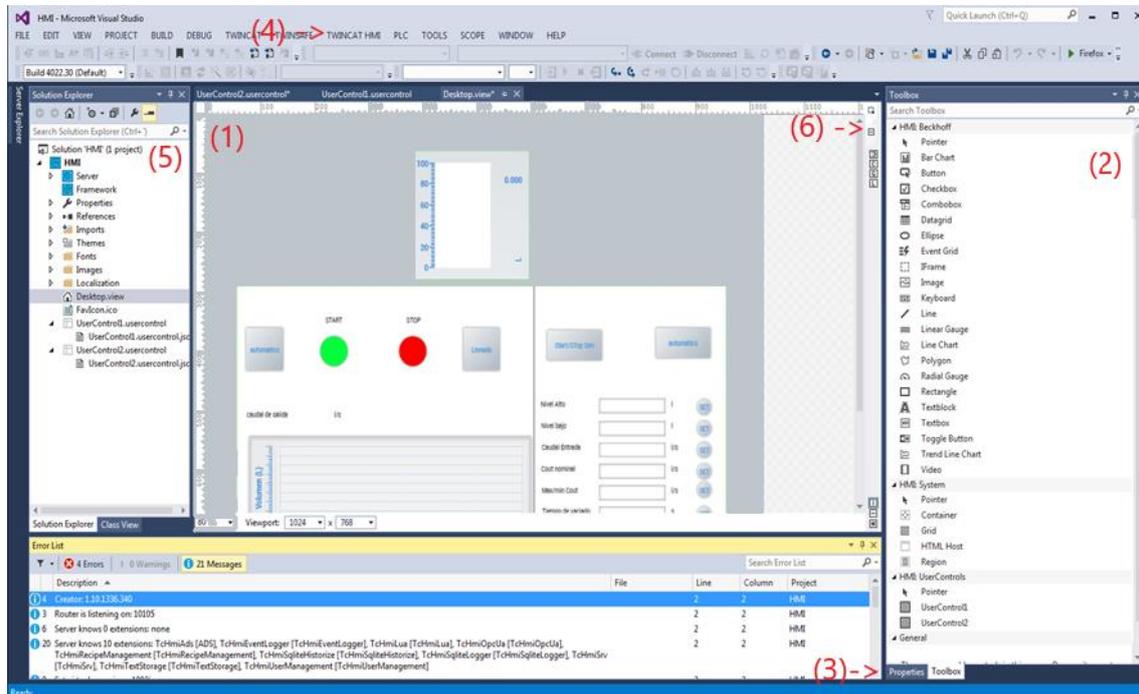


Figura 5.2. Entorno de TE-2000 TC3 HMI

Fijándonos en la Figura 5.2 podemos destacar los elementos más relevantes para llevar a cabo este proyecto [6]:

- (1) Es la pantalla de diseño se permite diseñar tanto gráficamente como con código. En la parte de diseño gráfico se pueden añadir controles almacenados en el “Toolbox”, cambiando los atributos de estos en la ventana de propiedades. El código se generará automáticamente en segundo plano y puede ser editado conjuntamente con el diseño de la parte gráfica. En la esquina superior izquierda del editor se encuentra el botón que permite cambiar esta función.
- (2) La ventana de Toolbox incluye todos los elementos de control del proyecto predeterminados (botones, gráficas, cajas de texto...) así como elementos de control creados por el usuario denominados “User Controls”.
- (3) La ventana de propiedades permite al usuario modificar y configurar los elementos de control.
- (4) La ventana de Twincat HMI Configuration permite acceder a todos los símbolos visibles para el HMI.

- (5) La ventana del explorador de la solución muestra el árbol del proyecto del HMI.
- (6) Twincat HMI LiveView permite probar la visualización en línea, mediante una ventana de Visual Studio o en un navegador web por URL.

En la ventana del explorador de la solución, en el apartado de referencia se pueden activar extensiones. En nuestro caso activamos la extensión TcHMIOpcUa, que nos permite establecer una conexión a un servidor OPC-UA.

5.3 TF-6100

El complemento TF-6100 permite establecer y crear una comunicación vía OPC-UA con otros sistemas. OPC-UA es un protocolo de comunicación industrial que es totalmente abierto y multiplataforma, además múltiples proveedores de PLC implementan complementos para poder utilizar este protocolo. Por lo tanto, dado la versatilidad que posee, permite la integración de múltiples PLC de distintos proveedores en una única red [3]. Por todas estas ventajas resulta más conveniente utilizar este protocolo frente al ADS propio de Beckhoff que solo permite trabajar con tecnologías propias de esta compañía. El protocolo OPC-UA se implementará a todos los niveles de comunicación en el proyecto. Este complemento incorpora las siguientes herramientas:

5.3.1 TwinCAT Connectivity Project

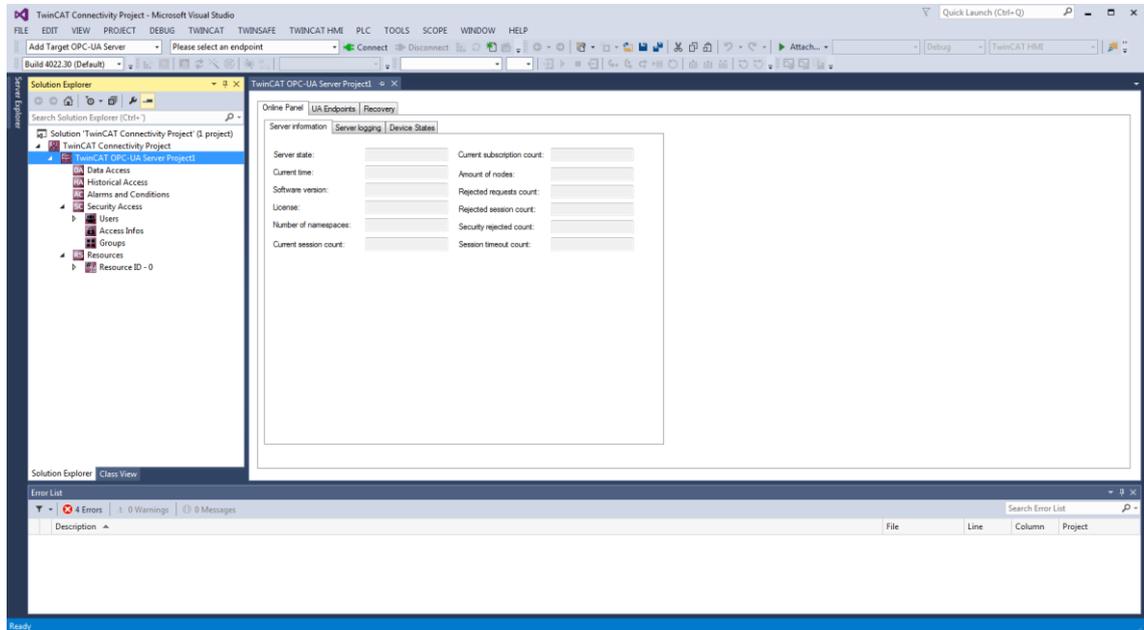


Figura 5.3. Entorno de TwinCAT Conectividad Project.

Utiliza el entorno de Microsoft Visual Studio y se usa para configurar el servidor OPC-UA, permitiendo todas aquellas características que caracterizan a este protocolo de comunicación [7]:

- Establecer Endpoints: Indican que mecanismos de seguridad se aplican durante el establecimiento de la conexión, así como los protocolos de transporte y codificación. Las políticas de seguridad que se pueden implementar son Basic128Rsa15, Basic256 y Basic256sha256.
- Cargar los símbolos que serán visibles en el servidor. Por defecto el programa carga los símbolos del equipo en donde se lanza el servidor.
- Acceso a datos históricos. Los valores de las variables que se configuren se almacenan permanentemente en un dispositivo de almacenamiento de datos. Se puede configurar la forma en la que el servidor almacena estos valores.
- La configuración de alarmas y condiciones. Permite al usuario monitorizar los valores del proceso y establece alarmas y condiciones cuando un símbolo del servidor cambia de estado.

- Establecer certificados de seguridad. Permite al programador del servidor establecer derechos de acceso y modificación según el tipo de usuario.

5.3.2 UAGateway Administration

El UAGateway Administration gestiona la puerta de enlace de las plataformas y equipos entre los que se establece la conexión. Este programa permite conectar aplicaciones OPC COM DA antiguas a aplicaciones OPC-UA. Por otra parte, ofrece una interfaz que hace posible convertir elementos básicos de servidores Twincat OPC-UA a un servidor central OPC-UA [7]. En la figura 5.5 se representa un esquema básico de su funcionamiento.

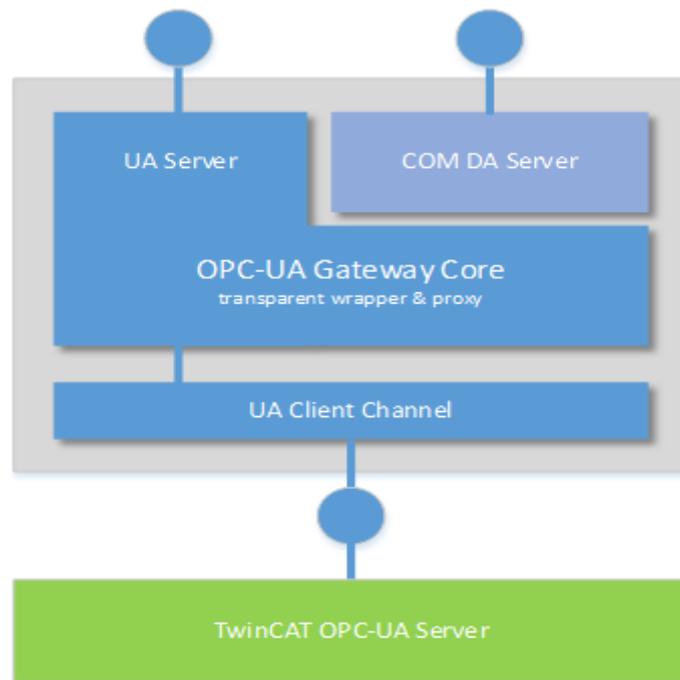


Figura 5.5. Se representa el funcionamiento básico del Gateway (fuente de la imagen: Manual Twincat 3 OPC-UA).

5.3.3 Sample client

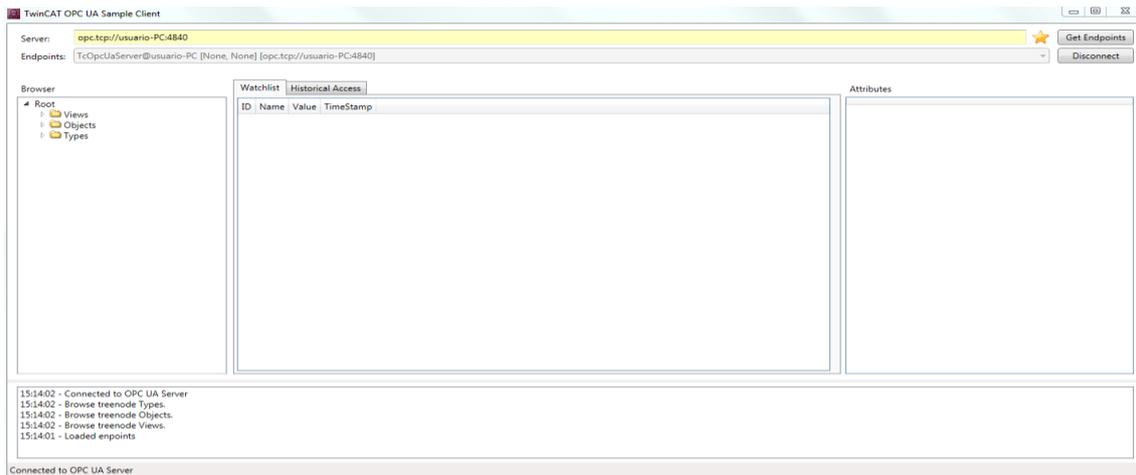


Figura 5.6. Interfaz del Sample Client.

Dispone de una interfaz que permite explorar el espacio de nombres OPC-UA, y agregar un nodo UA desde el espacio de nombres a la lista de observación. Este programa está desarrollado con la intención de que el usuario pueda hacer pruebas iniciales en el servidor OPC-UA.

6. Implementación de un prototipo.

La arquitectura de la solución es la mostrada en el siguiente esquema:

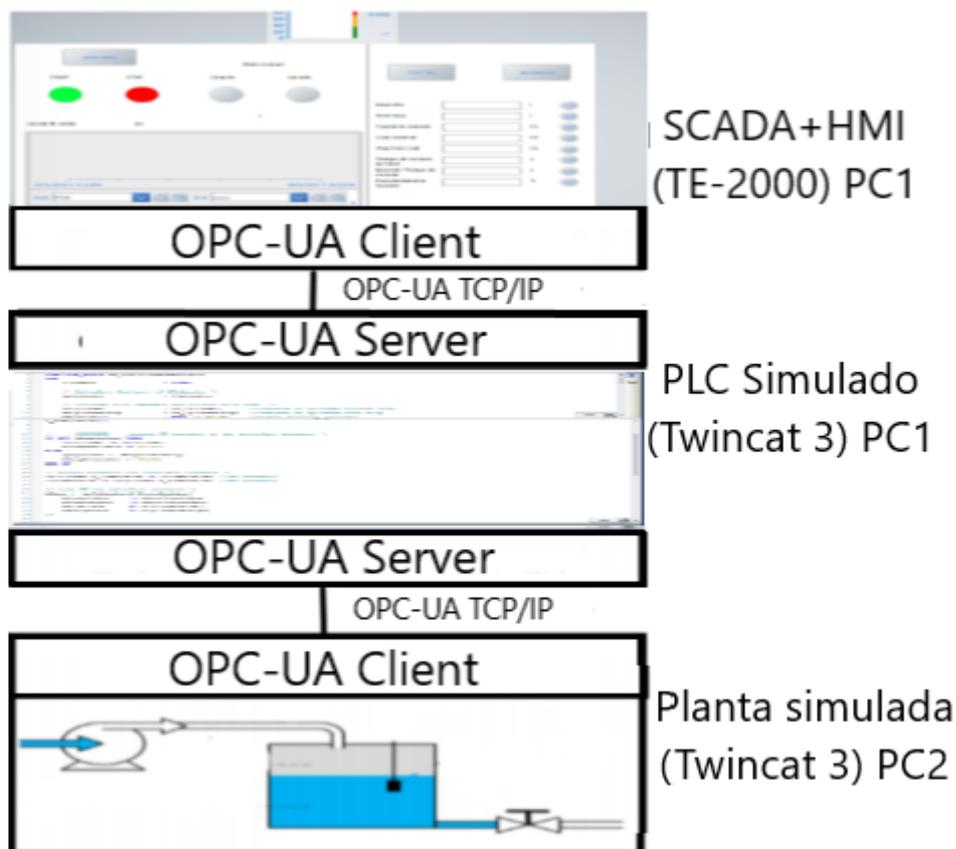


Figura 6.1. Arquitectura de la solución

Como se observa en la figura 6.1, para la realización de este proyecto se requieren al menos dos ordenadores. Uno de estos ordenadores simulará la planta y el otro ordenador simulará el PLC. Estos dos equipos mantendrán una comunicación vía OPC-UA TCP/IP con un protocolo de codificación binario. El equipo que almacena este servidor es el que simula el PLC, mientras que el sistema SCADA+HMI y la planta simulada actúan como clientes. El sistema SCADA+HMI se configura en la misma solución que el programa de PLC. El sistema SCADA+HMI es accesible a cualquier dispositivo de visualización con acceso a internet vía HTTP.

Como se comentó en el capítulo anterior las posibilidades de simulación son muchas y muy diversas. Debido a la versatilidad que posee el software Twincat 3 se pueden realizar simulaciones de procesos en Matlab y Simulink. Además, al implementar el protocolo OPC-UA las posibilidades de simulación no están sujetas a este software en específico, sino que se pueden ampliar o sustituir las distintas herramientas según lo que se quiera llevar a cabo.

6.1 Sistema de control de Nivel.

Como ejemplo de simulación se ha optado por el sistema de llenado de un tanque, cuyo esquema es el que se muestra en la figura 6.2. Consiste en un sistema de llenado básico en el que existen: Una sonda del nivel, una bomba de llenado y una válvula de salida. A partir de estas mediciones se realizará el control.

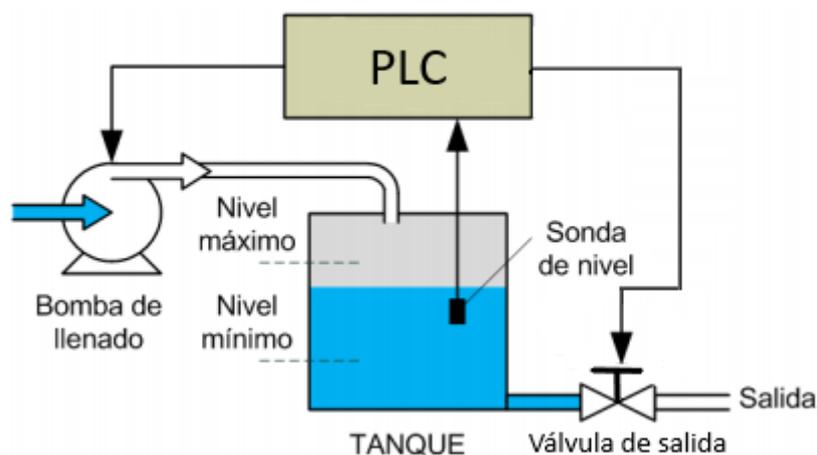


Figura 6.2. Representación del sistema.

6.2 configuración previa de Twincat 3

La simulación del PLC y la planta se realizará mediante el software de Twincat 3 descrito en el apartado 5.1. Dado que Twincat 3 es una herramienta de simulación en tiempo real, y como Windows es un sistema operativo de propósito general, se debe configurar Twincat 3 para que el componente de tiempo de ejecución pueda adelantarse a otras tareas del ordenador mientras se está ejecutando la lógica de control de la máquina. Para que pueda funcionar de esta

manera es necesario reservar al menos un núcleo del ordenador para esta función [5]. En el desarrollo de nuestro proyecto se dedica, debido a las características del equipo, dos núcleos del CPU como se muestra en la tabla 7.1.

CPU	RT-CPU	Base Time	CPU Limit	Latency Warning
0 (Windows)	<input type="checkbox"/>			
1 (Windows)	<input type="checkbox"/>			
2 (Isolated)	<input checked="" type="checkbox"/> Default	1 ms	100 %	(none)
3 (Isolated)	<input checked="" type="checkbox"/>	1 ms	100 %	(none)

Tabla 6.1

Como vemos es posible configurar el límite del CPU de tal forma que si el equipo en el que se trabaja no dispone de la suficiente velocidad se podría modificar este parámetro.

6.3 Simulación de la planta

En cuanto a la simulación de la planta tuvimos en su diseño las siguientes consideraciones generales:

- Los valores de las variables de simulación pueden tomar un valor por defecto (Valores constantes definidos por defecto) o definidas por el usuario.
- La planta se conectará al servidor OPC-UA lanzado por el PLC.
- Las variables de control de la planta serán leídas vía OPC-UA, así mismo las variables de salida de la planta necesarias para realizar el control serán escritas en variables ya definidas por el PLC simulado.

El sistema consta de dos entradas: el caudal de entrada y el caudal de salida, y de una salida: el nivel de llenado. Tanto el caudal de entrada como el caudal de salida son valores constantes fijados por el programa o por el usuario del HMI. El programa almacena, en un primer momento, el tiempo de ciclo de la tarea de la simulación, generando de esta forma una constante de tiempo de simulación. De esta forma los diferenciales de aumento de nivel dependen del tiempo de ciclo del programa. Adaptando así el funcionamiento discreto del software al sistema continuo que se espera que realice la simulación. El modelo matemático que representa el sistema es la variación del nivel del tanque (ΔH) es igual a la diferencia entre el caudal de salida (q_i) y el caudal de entrada (q_o):

$$\Delta H = \frac{dH(t)}{dt} = q_i - q_o$$

6.4 PLC simulado

Para realizar el desarrollo del PLC simulado se tuvieron en cuenta las siguientes disposiciones:

- No se tomó en cuenta la forma de trabajar de ningún modelo de PLC en específico. Por tanto, no se estudió la utilización de ningún driver que simulará el comportamiento de un modelo de PLC real. De esta forma se consigue que la programación sea lo más general posible.
- La declaración de las variables de salida de la planta (necesarias para el control) se realiza en la planta y en el PLC simulado. Estas variables serán visibles en el espacio OPC-UA. Se optó por esta dinámica debido a que es la forma en la que más se asemeja a un sistema real.
- En este apartado sólo se simula el control mediante la modificación del estado de los actuadores, en función de los sensores. Lo que se busca es que la programación del PLC se centre única y exclusivamente a lo que se esperaría en un sistema real.
- El PLC simulado se comunica con el sistema Scada+HMI vía OPC-UA. De esta forma si se opta por la utilización de otro sistema Scada+HMI, no es necesario modificar el protocolo de comunicación.

GIEIA- trabajo de fin de grado

En la simulación del PLC se han declarado las siguientes variables de salida:

- Vin: Señal que activa la bomba de entrada.
- Vout: Señal que activa la válvula de salida.

Las variables de entrada son las siguientes:

- Nivel bajo (L1): La sonda de nivel indica que se ha alcanzado el nivel bajo del depósito.
- Nivel alto (L2): La sonda de nivel indica que se ha alcanzado el nivel bajo del depósito.
- Modo automático (automático): Define si el sistema se comporta en modo automático (ON) o modo manual (off).
- Start: Empieza el modo automático.
- Vaciado (M1): Si se activa comienza el vaciado (en modo manual).
- Llenado (M2): Si se activa comienza el llenado (en modo manual).

Todas las variables de control son del tipo Booleanas (On/Off).

El sistema de control se inicia cuando se selecciona el modo de operación (automático o manual). En el modo automático, el sistema se inicia en modo reposo hasta que se activa la variable Start encendiendo la bomba de entrada (Vin). De esta forma, empieza el llenado hasta alcanzarse el nivel máximo (L2), comenzando así la labor de vaciado activando la válvula de salida (Vout) hasta que se desactive el nivel bajo (L1) volviendo a iniciar al ciclo de llenado. En el segundo el usuario establece cuando se produce el llenado o vaciado, mediante la acción de dos botones de encendido de las salidas (M1), activando el llenado y (M2) activando el vaciado.

En la misma solución donde se programa el PLC se configura el sistema HMI y el servidor OPC-UA.

6.5 Sistema SCADA+HMI

El sistema SCADA+HMI se realiza mediante el software TE-2000 TC3 HMI Engineering. Este programa se lleva a cabo en el mismo árbol de soluciones que la programación del PLC.

Para llevar a cabo este programa se tuvieron en cuenta las siguientes disposiciones:

- Que la comunicación con el PLC y el sistema SCADA utilice la tecnología OPC-UA, de tal forma que se puedan utilizar otras plataformas de simulación de PLC o de procesos, manteniendo el mismo formato del HMI y el SCADA y sin tener que modificar las comunicaciones ni las variables.
- Que se disponga de toda la información posible sobre el funcionamiento de la planta de tal forma que se registren en el SCADA datos históricos sobre el nivel de llenado, caudal de salida, etc. De esta forma se conoce más en detalle el funcionamiento de la planta.
- Presenta dos partes diferenciadas:
 - Una parte de control en las que el usuario tiene acceso a través de un panel de control a las variables de control de proceso y el usuario puede elegir entre un control automático o un control manual.
 - Una parte de simulación en la que el usuario pueda cambiar en todo momento las variables de simulación (nivel alto, nivel bajo, caudal de entrada...) y disponga además de un modo de simulación automática.

El aspecto del HMI es el mostrado en la figura 6.5.



Figura 6.5. Pantalla HMI.

Como se observa en la figura 6.5 la parte de control del PLC se encuentra a la izquierda, y la parte de simulación se encuentra en la parte derecha.

En el lado de control se encuentran los siguientes elementos:

- Un botón que inicia el control automático o el control manual. Según el control que se elija están disponibles uno u otros botones.
- Un gráfico de líneas que representa los distintos valores que ha tomado el nivel del tanque. También se representan en este grafico el nivel alto y el nivel bajo.
- Un gráfico de barras que representa el nivel actual del tanque.

En el lado de la simulación se encuentran los siguientes elementos:

- Un botón que permite empezar o parar la simulación.
- Un botón que inicia el modo de simulación automático.

- Cuadros de texto y botones de set, que permiten cargar datos de manera manual en las variables que conforman la planta.

Esta interfaz es accesible desde cualquier dispositivo con conexión a internet y que se encuentre conectada a la misma red, a través de la dirección IP del equipo que almacena el servidor HMI. El usuario tendrá acceso a toda la información que se presenta en el HMI. Para acceder a este servidor se dispone de un usuario y contraseña.

6.6 Servidor OPC-UA

La comunicación vía OPC-UA, como se ha comentado, tiene como finalidad que cualquier sistema SCADA, programa de simulación, software de procesamiento de datos, equipos o instrumentación, independientemente del desarrollador, mantengan una comunicación directa. Es decir, crear un sistema de comunicación único que no requiera el uso de ningún driver en específico (como se requería antes del uso de esta tecnología) para realizar la labor de comunicación en distintas tecnologías.

El complemento TF-6100, como se comentó en el apartado 5.3, permite establecer protocolos de comunicación OPC-UA. Este software incluye el OPC-UA server, que actúa como capa intermedia de comunicación entre las distintas plataformas. El estándar de comunicación OPC-UA está implementado en la capa de aplicación del modelo TCP/IP. Como se observa en la figura 6.6. El texto de referencia para exponer esta información es el documento “OPC 100000-6: Mappings” de la fundación OPC [3].

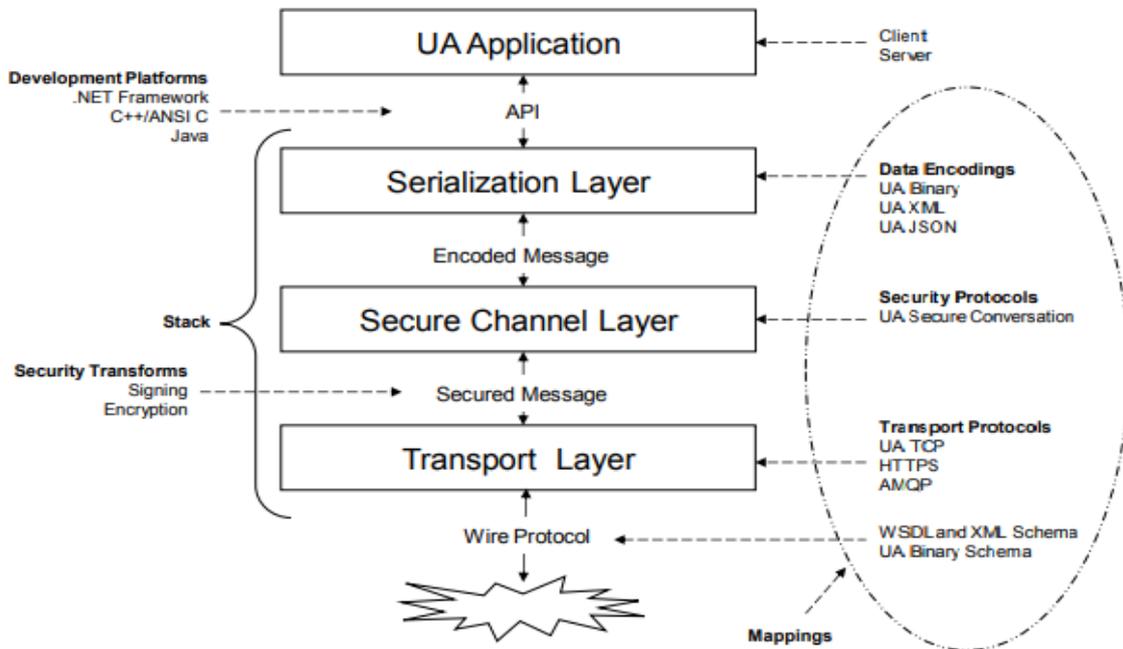


figura 6.6. Descripción general de la pila de OPC-UA.

La comunicación en este servidor se trata de una comunicación Full-Duplex en la que tanto servidores como clientes pueden transmitir y recibir datos al mismo tiempo. Empezando a enumerar de nivel alto hacia nivel bajo encontramos:

- UA application (aplicación UA): Es el cliente del servidor (PLCs, sistemas SCADA, MES, etc.). Este se comunicará con el servidor a través de una API (Interfaz de Programación de Aplicaciones). Este sistema permite a las aplicaciones transmitir información con otros servicios o dispositivos sin necesidad de saber cómo estos están implementados [8]. Esta plataforma puede estar desarrollada en C + +, C, JAVA o .net framework.
- Stack (pila): En este nivel se lleva a cabo el establecimiento y el cierre de la comunicación, se definen los protocolos de comunicación y seguridad y se codifican los datos. En la pila se encuentran los siguientes niveles:
- Serialization Layer (nivel de serialización): En este nivel se establece la codificación de los mensajes que se desean transmitir. Cabe destacar que la información solo podrá ser recibida y enviada, si las dos aplicaciones que se comunican disponen de las herramientas necesarias para decodificar los mensajes. Por ejemplo, si la aplicación del servidor solo admite codificación binaria, la aplicación cliente con sólo codificación XML no puede comunicarse con el servidor. Esta codificación puede ser:

- OPC-UA binary: Está diseñado para la codificación y decodificación rápidas. Para labores que requieren de una rápida respuesta.
 - XML: Está pensado para mover información a grandes bases de datos que no requieran de una actualización constante, o para acceder de manera remota sin necesidad de estar en la misma red.
 - JSON: Pensada para la misma función que la codificación XML. Enfocado sobre todo a aplicaciones que utilicen JAVA.
- Secure Channel Layer (nivel de canal seguro): En este nivel se establece como crear un canal de comunicación seguro entre aplicación y servidor, y como aplica seguridad a los mensajes intercambiados a través de este canal. Los protocolos de seguridad admiten tres modos: None (ninguno), Sign Only (solo registro) y SignAndEncryp (registro y encriptación). Esta información se muestra gráficamente en la figura 6.7.

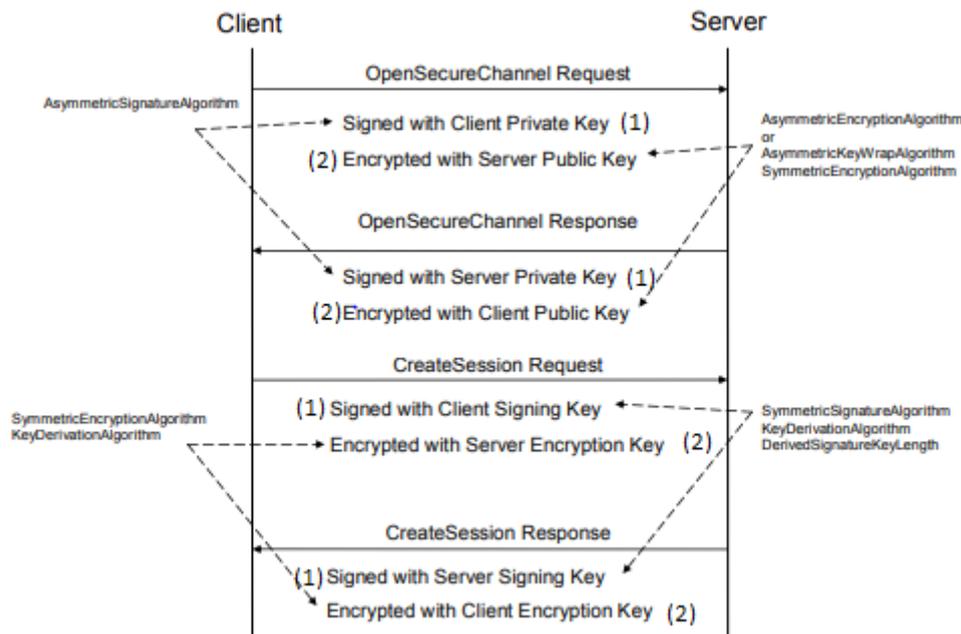


Figura 6.7. Descripción del secure channel layer. (1) y (2) no son usados en modo de seguridad none y (2) si el modo de seguridad es Only Sign

- Transport Layer (nivel de transporte): A este nivel se establecen las comunicaciones. Las comunicaciones pueden ser iniciadas por el cliente o por el servidor. Si esta comunicación es establecida por el

cliente enviará un mensaje al servidor indicando los tamaños de buffer (espacios de memoria) que el cliente soporta. El servidor responde con un mensaje de confirmación. El tamaño de esta memoria intermedia se notifica al nivel superior de seguridad. Si el servidor es el que crea la conexión envía un mensaje al cliente, que de este aceptarlo se inicia el mismo proceso descrito cuando es el cliente el que realiza la conexión. Este proceso se describe en la figura 6.8. La comunicación se termina cuando el cliente envía una solicitud al servidor o cuando existe algún tipo de error. En este nivel se establecen los protocolos de transporte que pueden ser:

- TCP/IP: Es un protocolo ubicuo que proporciona una comunicación full-duplex entre dos aplicaciones.
- HTTPS: se intercambian mensajes a través de una conexión SSL/TLS en lugar de una conexión TCP/IP.

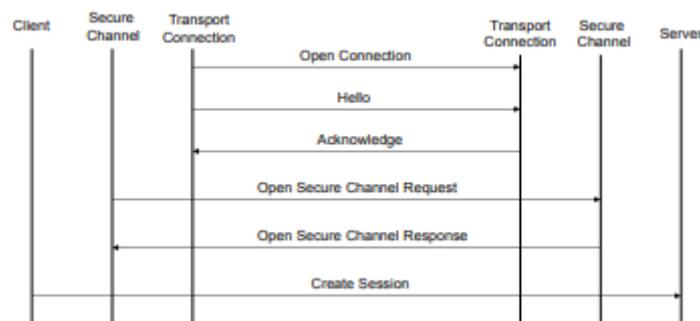


Figura 6.8. Ejemplo de cómo se establece la conexión si esta es empezada por el cliente

- Wire protocol (protocolo del cable).

El administrador del servidor es el que determina cómo se van a realizar las conexiones dentro del servidor OPC-UA. Las combinaciones más utilizadas en las comunicaciones en cuanto a protocolos de transporte y codificación OPC-UA están establecidas en los siguientes Endpoints:

- Endpoints 1: TCP/IP más un protocolo binario. Está pensado para datos que requieran una respuesta rápida y que se actualicen de forma casi inmediata. Se usan preferentemente en entornos a nivel de control.

- Endpoints 2: Usando Web Service y protocolo XML. Permite mover grandes cantidades de datos de forma más lenta que con la otra alternativa, está pensado para mover información a grandes bases de datos que no requieran de una actualización constante, o para acceder de manera remota sin necesidad de estar en la misma red. Poseen la ventaja además de no requerir configurar firewalls.

Sin embargo, los protocolos de transporte y de codificación se pueden combinar para hacer que el sistema se comporte de la manera óptima y eficientemente posible.

6.6.1 Configuración del servidor OPC-UA

El equipo que simula el PLC es el encargado de crear el servidor OPC-UA, comportándose la planta simulada y el sistema SCADA como clientes OPC-UA.

En el presente proyecto se ha optado por utilizar un protocolo TCP/IP junto con un protocolo de codificación binario. Se ha elegido esta configuración ya que resulta la óptima a la hora de realizar la comunicación a nivel de control y supervisión, debido a que se desea obtener un flujo constante de datos. De esta forma se consigue que los datos transferidos a los clientes OPC-UA se actualicen a la misma vez que las variables. Además, el proyecto está pensado para trabajar a nivel local, por lo que en un principio no resulta conveniente utilizar Web Service.

En cuanto al nivel de seguridad no se ha implementado ningún protocolo, ya que este proyecto tiene un fin docente y toda la información que contiene el servidor OPC-UA no tiene ningún tipo de confidencialidad. De esta forma resulta más fácil de implementar y empezar a utilizar este software en otros dispositivos.

6.6.1.1 Conexión de la planta simulada con el servidor OPC-UA

Para conectar la planta y el PLC simulados es necesario la instalación del complemento TF 6100 de Beckhoff. Este complemento nos permite acceder a la librería Tc3_plcOPen_OpcUA. Esta librería está desarrollada por una colaboración conjunta de las asociaciones PLCopen y de OPC. La asociación nace con el objetivo de desarrollar distintas librerías con el fin de que los distintos sistemas industriales utilicen el mismo lenguaje común a la hora de transferir los datos. Por tanto, se consigue que los objetos sean semánticamente idénticos, aunque difieran ligeramente en los puntos de entrada, facilitando significativamente la conexión entre los dispositivos. Esta característica se muestra en la figura 6.9. De esta forma se consigue la comunicación vertical entre PLC simulado o físico y aplicaciones, y horizontal entre los PLC simulados o físicos. Consiguiendo una arquitectura unificada y creando una plataforma de conectividad homogénea, mejorando de esta forma la interconectividad entre sistemas y desarrolladores de distintas compañías. [2]

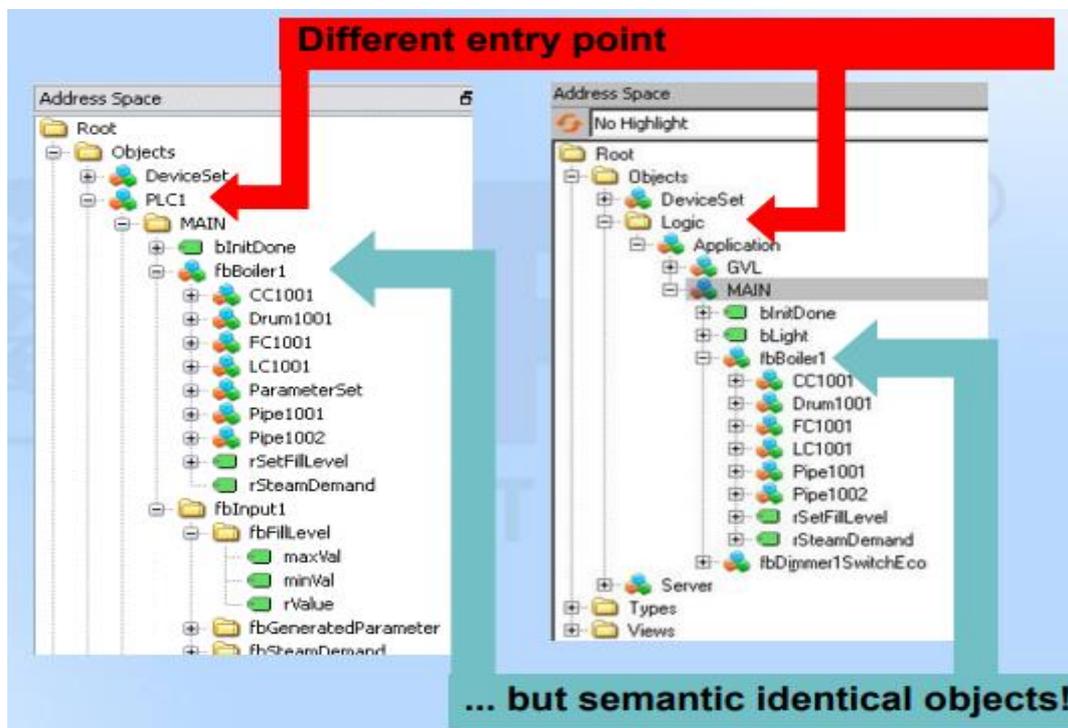


Figura 6.9. Comparativa de objetos visibles en el servidor OPC-UA de distintas compañías. En el lado izquierdo se muestran los objetos visibles en espacio de nombres de un servidor OPC-UA lanzado por la compañía Beckhoff y en lado derecho lanzado por BOSH-Rexroth.

La librería Tc3_plcOPen_OpcUA posee los siguientes bloques funcionales para llevar a cabo la conexión y la lectura de los símbolos del servidor OPC-UA. Estos bloques funcionales son:

- UA connect: Este bloque establece conexión con un servidor OPC-UA remoto mediante su URL e información de conexión: Tipo de seguridad, tipo de transporte y nombre de aplicación. Esta última información se rellena en la estructura "ST_UASessionConnectInfo", devolviendo un identificador de conexión o "connectionHdl" requerido para la utilización de los posteriores bloques funcionales. Si la conexión no se ha realizado correctamente devuelve un mensaje de error. En el ejemplo de aplicación la URL es "OPC.TCP://usuario-PC:4840", el tipo de transporte es TCP y no se establece ningún modelo de seguridad.



Figura 6.10. Bloque funcional UA_connect

- UA_Disconnect Este bloque funcional cierra la conexión OPC-UA.



Figura 6.11. Bloque UA_Disconnect

- UA_GetNamespaceIndex: Este bloque proporciona el índice de espacio de nombres para una URI de espacios de nombres, necesario para identificar símbolos del sistema. El uso de una URI de espacio de nombres permite a varios servidores OPC-UA adjuntos al mismo sistema utilizar el mismo identificador para los objetos. Este programa tiene como entrada el identificador de conexión o "ConnectionHdl" proporcionado por el bloque UA_connect y la URI de espacios de nombres o "NamespaceUri" que en el ejemplo de aplicación es "urn:BeckhoffAutomation:UA:PLC", devolviendo el programa el índice de espacios de nombres o "NameSpaceIndex". Si no se detecta el índice de espacio de nombre devolverá un mensaje de error.



Figura 6.12. Bloque UA_GetNamespaceIndex.

- UA_NodeGetHandleList: Este bloque nos permite conocer los identificadores de nodos (Nodeid) de una lista de símbolos del espacio de nombres UA. Un nodo en el espacio de direcciones OPC-UA es de un tipo determinado (p. ej. objeto, variable o método) y se describe con una lista de atributos. Todos los nodos tienen atributos comunes, como el nombre o la descripción, y atributos específicos, como, por ejemplo, el valor de una variable (6). El bloque funcional tiene como entradas "ConnectionHdl" obtenido del bloque "UA_connect", NodeIDCounts (número de identificadores de nodo que se desea conocer) y NodeIDs que es un array de estructuras que indican un direccionamiento único del nodo UA. El array consta de tipo de identificador, índice de espacio de nombre e identificador. El programa da como salida un array de identificador de nodo o un mensaje de error en caso de fallo.



Figura 6.13. Bloque UA_NodeGetHandleList

- UA_ReadList: Este bloque funcional lee los valores de varios nodos. Este bloque funcional tiene como entradas "ConnectionHdl" (obtenido del bloque "UA_connect"), NodeIDCount (número de variables que se desea leer), NodeHdls (array de identificador de identificador de nodo), stNodeAdddInfo (información adicional sobre qué atributo se lee del espacio de nombres de UA o que rango de índice se utiliza), pVariable (puntero direccionado al almacenamiento donde está alojado el valor del dato que se desea conocer) y cbData (determina el tamaño del dato que se quiere leer). Este bloque funcional devuelve el valor del dato o un mensaje en caso de algún fallo.



Figura 6.14. Bloque UA_ReadList.

- UA_write: Este bloque funcional permite escribir valores en los nodos. Como entradas mantiene las mismas que el bloque funcional anterior, con la diferencia de que este bloque funcional solo permite escribir en un nodo por cada vez que se ejecuta. A diferencia del anterior, en este caso el puntero apunta a la variable que se quiere leer dentro del programa principal, cargándola en una variable previamente declarada en el servidor OPC-UA. Este programa solo presenta una salida.



Figura 6.15. Bloque UA_Write

Cabe destacar que este programa no es único de Twincat 3 para establecer una comunicación OPC-UA, si no que este código al tener estructuras, variables y bloques funcionales propias de la colaboración de PlcOpen y OPC Foundation permiten que este código pueda ser reutilizado si se quiere sustituir la plataforma de Twincat.

Para realizar la conexión y lectura o escritura de las variables se ha llevado a cabo la siguiente máquina de estados mostrada en la figura 6.16.

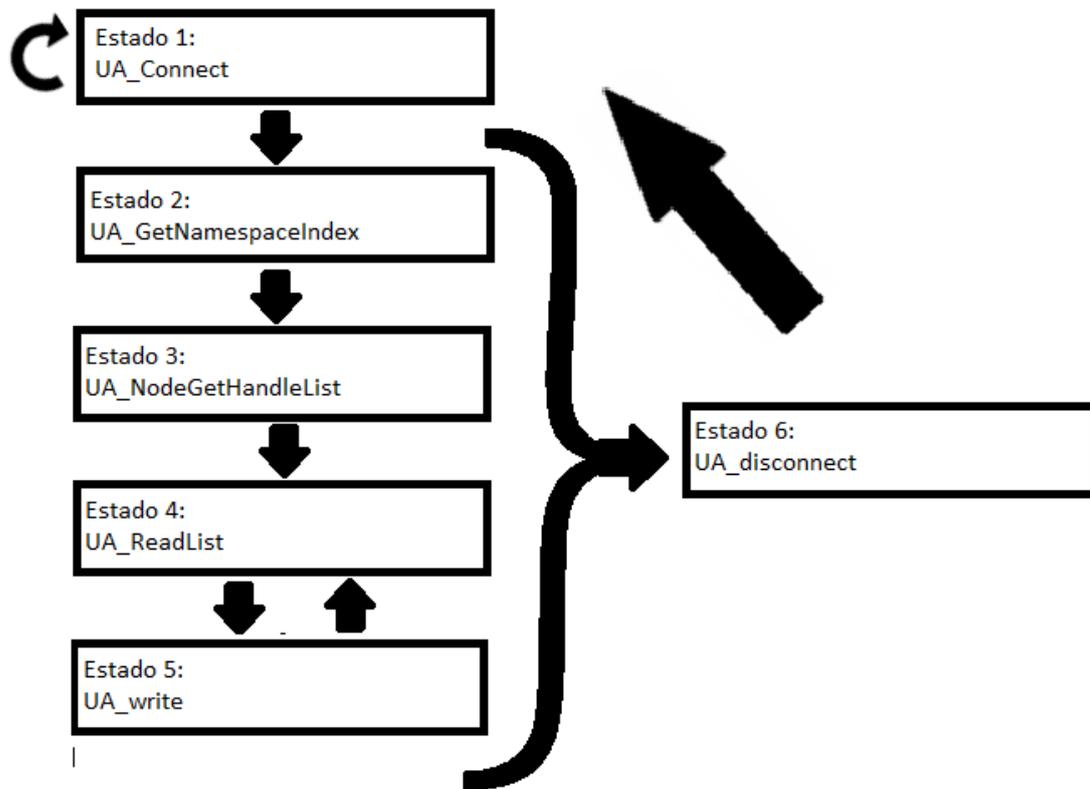


Figura 6.16. Máquina de estados.

Una vez que se ejecutan los estados 1, 2 y 3 el programa entra en bucle en el estado 4 y 5. La única forma de que se salga de este bucle es que se produzca un error en alguno de los estados 2, 3, 4 y 5 o que el sistema deje de funcionar. Si ocurriese un error pasaría al estado 6 de desconexión y seguidamente intentará volver a conectar pasando de nuevo al estado 1. Si el programa no consigue establecer comunicación seguirá ejecutando el bloque 1 hasta que se produzca la conexión.

7. Conclusiones

En el presente proyecto se ha expuesto cómo utilizar el estándar de comunicación OPC-UA para conectar las plataformas de simulación de plantas con una plataforma de programación de autómatas industriales y un sistema SCADA. Con esta solución lo que se pretende es lograr un control y monitorización de una planta industrial, que permita a los alumnos familiarizarse con un entorno industrial. Cabe destacar que lo interesante de este sistema es que se ha realizado siguiendo un estándar de comunicación abierto, por lo que cualquier dispositivo o aplicación adaptada para funcionar con esta tecnología puede conectarse a este servidor, permitiendo de esta forma ampliar el proyecto que aquí se presenta o adaptarlo a las necesidades que vayan surgiendo. Igualmente, se han presentado las características de la adopción por parte de las industrias de los protocolos de comunicación estandarizados, y lo que se entiende que será el futuro de estas.

El proyecto que se presenta satisface los objetivos establecidos en el apartado 3 de este documento, ya que se ha puesto de manifiesto un nuevo modelo formativo en materias relacionadas con la automatización industrial. Más concretamente, se presenta un complemento al aprendizaje que permite tener una mayor versatilidad a la hora de realizar prácticas de estas asignaturas. En contraparte, este proyecto supone un comienzo en lo que se considera que debería ser el modelo final que llegase a los alumnos. Como se explica en el apartado 9 de este documento, este proyecto puede ser ampliado y adaptarlo aún más a las necesidades de los alumnos. Por tanto, es un precedente que sirve para mostrar los beneficios de usar estas tecnologías y lo importante que puede llegar a ser si se amplía aún más en esta idea.

Finalmente, la conclusión más significativa quizás a la que se ha llegado en este proyecto es que las múltiples herramientas de las que disponemos gracias al avance de las tecnologías industriales permiten complementar de una manera relevante la formación del alumnado. Como hemos visto, estas tecnologías pueden ser adaptadas a las necesidades de los alumnos y permiten crear sistemas industriales complejos sin la necesidad de un elevado coste económico.

8. Coclusions

In this project a solution is presented to the growing needs at a training level in terms of process automation. It is a complement to the learning tools by students and professionals in the sector so that they have the necessary resources to carry out their training in automatic industrial processes.

The importance of this project is based mainly on the high cost of acquiring a prototype of a physical plant with all its components for training, in addition to the maintenance costs. Therefore, the need arises to use cheaper and easier alternatives to put into practice in the field of teaching industrial automation. On the other hand, the emergence of industry 4.0 makes the needs increase and it is very difficult to respond with current models. Highlighting the need to seek a training model that is expandable and upgradeable to give a quick response to these changes that occur in the sector.

The project provides a solution to this problem through a simulation of a specific plant and its control trough a simulated PLC. In addition, it allows the visualization of the different parameters generated by the control of the plant trough a SCADA system. Likewise, communication is achieved between the different elements that make up the solution, in such a way that it allows the substitution of some of the components chosen by others without affecting the rest of the elements. Moreover, being able to add extra components without this being a problem when establishing communication. In this way, an expandable project is achieved that adapts to the training needs that arise, adding the elements that are required to the system that is proposed.

8. Líneas futuras

Dada la versatilidad que presenta la implementación del estándar de comunicación abierta OPC-UA, las posibilidades a la hora de ampliar este proyecto son múltiples.

Un futuro proyecto de ampliación sería el de programar el software que simula la planta de tal forma que sea capaz de ejecutar un run-time por cada cliente que se conecta al servidor. Es decir, que cada cliente del servidor pueda modificar la programación del PLC simulado sin que esto afecte a otros clientes del servidor.

Otro futuro proyecto podría ser utilizar una plataforma de simulación de plantas independiente de Twincat 3. Una buena alternativa al programa Twincat 3 es el software llamado Game4automatitiation. Este programa está desarrollado en Unity y permite una simulación visual en 3D de un proceso industrial o mecánico. Presenta complementos que le permiten actuar como cliente de un servidor OPC-UA, por lo tanto, este programa actuaría como simulador de la planta mientras que la programación del controlador la realizaría Twincat 3 y el sistema SCADA por TE-2000. También se podría hacer uso de otros softwares que resulten más convenientes. Además, el simulador presenta una interfaz de diseño de plantas industriales, por lo que las posibilidades a la hora de simular son múltiples. Pudiendo incluso plantear a los alumnos que realicen sus propias simulaciones de plantas para lograr un determinado objetivo. Este proyecto es perfectamente compatible con el proyecto propuesto anteriormente.

Como complemento al proyecto anterior, se podría escalar un nivel más y adquirir algún sistema MES. Así pues, los alumnos se familiarizan con el uso de estos programas. Mejorando de esta forma su desempeño no solo en la programación de autómatas, sino en un verdadero control a todos los niveles de la producción.

Bibliografía

- [1] uvigo. (2015, 29 enero). *Nueva tecnología OPC UA: SDK Servidor OPC-UA embebido(...)* [Vídeo]. YouTube.
<https://www.youtube.com/watch?v=PucO6y2N5i0&t=1021s>
- [2] uvigo. (2013, 13 enero). *OPC-UA: Fluxo seguro e fiable de datos e información(...)* [Vídeo]. YouTube.
https://www.youtube.com/watch?v=QiQXMZf_CTQ
- [3] OPC Foundation. (2017a, noviembre). OPC 10000-6 - Part 6: Mappings.
<https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-6-mappings/>
- [4] Beckhoff Automation GmbH & Co. KG. (2019, noviembre). TwinCAT 3 | eXtended Automation. <https://www.beckhoff.com/>
- [5] TwinCAT 3 Tutorial: Quick Start | Contact and Coil. (2016, junio). contactandcoil. <http://www.contactandcoil.com/twincat-3-tutorial/quick-start/>
- [6] Beckhoff Automation GmbH & Co. KG. (2020, enero). Manual TE2000 TC3 HMI Engineering.
<https://www.beckhoff.com/english.asp?twincat/te2000.htm>
- [7] Beckhoff Automation GmbH & Co. KG. (2020b, abril). Manual TC3 OPC UA. <https://www.beckhoff.com/english.asp?twincat/tf6100.htm>
- [8] Red Hat. (2015). ¿Qué es una API?. RedHat.
<https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces#:~:text=API%20significa%20interfaz%20de%20programaci%C3%B3n,permite%20ahorrar%20tiempo%20y%20dinero.>

Fuente de las figuras

Figura 4.2: recuperado de OPC Foundation, & Hoppe, S. H. (2012, enero). *OPC UA* [Ilustración].

https://tv.uvigo.es/uploads/material/Video/16266/PONENCIA_OPCEUROPE_JAI2012.pdf

Figura 4.3: OPC Foundation, & Hoppe, S. H. (2012, enero). *OPC scalability* [Ilustración].

https://tv.uvigo.es/uploads/material/Video/16266/PONENCIA_OPCEUROPE_JAI2012.pdf

Figura 4.4: Satoshi, S. (2018, 18 junio). [Arquitectura servidor OPC-UA]. OPIRON. <https://www.opiron.com/2017/11/06/opc-ua-6-razones-para-sentirte-mas-seguro/>

Figura 5.5: BECHKOFF. (2020, 21 abril). Gateway [Ilustración]. En *Manual TC 3 OPC UA* (p. 189).

Figura 6.2: Universidad de Alicante. (2012). Figura 1. Esquema de la planta de control de nivel [Ilustración]. En *Práctica 5. Circuitos digitales: control del nivel de un depósito* (p. 1).

Figura 6.6: OPC Foundation. (2017, 22 noviembre). Figure 1 – The OPC UA Stack Overview [Ilustración]. En *OPC 10000-6 - Part 6: Mappings* (1.04 ed., p. 5).

Figura 6.7: OPC Foundation. (2017b, noviembre 22). Figure 10 – Security handshake [Ilustración]. En *OPC 10000-6 - Part 6: Mappings* (1.04 ed., p. 39).

Figura 6.8: OPC Foundation. (2017b, noviembre 22). Figure 13 – Client initiated OPC UA Connection Protocol connection [Ilustración]. En *OPC 10000-6 - Part 6: Mappings* (1.04 ed., p. 56).

Figura 6.9: OPC Foundation. (2012, enero). *OPC-Unified Architecture: From embedded device up to MES/ERP IT Enterprise* [Diapositivas]. Universidad de Vigo.

https://tv.uvigo.es/uploads/material/Video/16266/PONENCIA_OPCEUROPE_JAI2012.pdf

GIEIA- trabajo de fin de grado

Anexos

El acceso al código fuente y documentación asociada a la realización de este proyecto puede ser accedida a través del siguiente enlace:

https://github.com/CarlosPerezHdez/TFG_IEIA_ULL_CarlosPerezHernandez