



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**Aplicación de internet industrial de las cosas al monitoreo y supervisión de un proceso de
manufactura simulado que integra tecnologías de comunicación heterogéneas**

Lema Molina, Henry David y Morales Jordán, Carlos Daniel

Vicerrectorado de Investigación, Innovación y Transferencia Tecnológica

Centro de Estudios de Posgrado

Maestría en Electrónica y Automatización mención Redes Industriales

Trabajo de titulación, previo a la obtención del título de Magíster en Electrónica y

Automatización mención Redes Industriales

Ing. Minchala Ávila, Luis Ismael PhD.

15 de abril de 2022

Latacunga

Reporte de verificación de contenido



PROYECTO TITULACIÓN_LEMA_MORALES_2022_VF.pdf

Scanned on: 17:39 March 15, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	1172
Words with Minor Changes	0
Paraphrased Words	0
Omitted Words	0

A handwritten signature in black ink, appearing to read "Luis Ismael Minchala Avila".

Digitally signed
by LUIS ISMAEL
MINCHALA AVILA
Date: 2022.03.15
13:13:45 -05'00'

Ing. Minchala Ávila, Luis Ismael PhD.

C.C.: 0301453486



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADOS

MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN MENCIÓN REDES INDUSTRIALES

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**Aplicación de internet industrial de las cosas al monitoreo y supervisión de un proceso de manufactura simulado que integra tecnologías de comunicación heterogéneas**” fue realizado por los señores **Lema Molina, Henry David y Morales Jordán, Carlos Daniel** el mismo que ha sido revisado y analizado en su totalidad, por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de las Fuerzas Armadas ESPE, por la cual me permito acreditar y autorizar para que lo sustenten públicamente.


Digitally signed
by LUIS ISMAEL
MINCHALA AVILA
Date: 2022.03.15
13:13:45 -05'00'

Latacunga, 15 de abril del 2022

Dr. Minchala Ávila, Luis Ismael

DIRECTOR

C.C.: 0301453486



ESPE
 UNIVERSIDAD DE LAS FUERZAS ARMADAS
 INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADOS

MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN MENCIÓN REDES INDUSTRIALES

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Lema Molina, Henry David y Morales Jordán, Carlos Daniel** con cédulas de ciudadanía nº 0504266909 y nº 1803673340 respectivamente, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Aplicación de internet industrial de las cosas al monitoreo y supervisión de un proceso de manufactura simulado que integra tecnologías de comunicación heterogéneas.”** es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 15 de abril del 2022



Firmado electrónicamente por:
**HENRY DAVID
 LEMA MOLINA**

Lema Molina, Henry David

C.C.: 0504266909



Firmado electrónicamente por:
**CARLOS DANIEL
 MORALES JORDAN**

Morales Jordán, Carlos Daniel

C.C.: 1803673340



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

VICERRECTORADO DE INVESTIGACIÓN, INNOVACIÓN Y TRANSFERENCIA DE TECNOLOGÍA

CENTRO DE POSGRADOS

MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN MENCIÓN REDES INDUSTRIALES

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, **Lema Molina, Henry David** y **Morales Jordán, Carlos Daniel**; autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Aplicación de internet industrial de las cosas al monitoreo y supervisión de un proceso de manufactura simulado que integra tecnologías de comunicación heterogéneas.”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 15 de abril del 2022



Firmado electrónicamente por:
**HENRY DAVID
LEMA MOLINA**

.....
Lema Molina, Henry David

C.C.: 0504266909



Firmado electrónicamente por:
**CARLOS DANIEL
MORALES JORDAN**

.....
Morales Jordán, Carlos Daniel

C.C.: 1803673340

Dedicatoria

A mi Dios quien me ha regalado la vida y la oportunidad de poder realizar mis estudios con normalidad. A mi madre quien siempre ha sido el apoyo y la compañía en toda circunstancia. A mi novia que me ha motivado a continuar con la mentalidad de que en un futuro los esfuerzos darán frutos. A mis hermanas y familia que me dan sus palabras de aliento, aconsejándome y con el orgullo de contar conmigo en cada travesía.

¡Gracias a todos!

Henry David Lema Molina

Dedicatoria

“Lo que sabemos es una gota de agua; lo que ignoramos es el océano”

Isaac Newton

Este logro va dedicado de manera especial a mi padre Carlos Humberto Morales y a mi madre Martha Jordán, pues ellos han sido mi principal fortaleza y que han sentado en mis las bases de responsabilidad y deseos de superación.

A mis hermanos, quienes son mi fuente de inspiración con su gran corazón y valentía ante las vicisitudes de la vida, cualidades que me llevan cada día a admirarlos incomparablemente.

Carlos D. Morales J.

Agradecimiento

Agradezco infinitamente a Dios Padre por la sabiduría y la bondad que me brinda.

A mis padres y familia por el apoyo en el transcurso de mi vida profesional

A mi novia por su amor y su apeo en cada paso a seguir

A mi tutor de este trabajo el Dr. Ismael Minchala quien nos supo guiar para llevar a cabo el mismo.

Henry David Lema Molina

Agradecimiento

“Pon tu corazón, mente, intelecto y alma incluso en tus actos más pequeños. Ese es el secreto del éxito”.

Swami Sivananda

Agradezco a Dios por su infinita bondad y misericordia.

A mis padres, hermanos y toda mi familia que me han brindado su apoyo y bendiciones a lo largo de mi carrera profesional.

A la Universidad de las Fuerzas Armadas, especialmente a las autoridades y docentes que conforman la facultad de Eléctrica y Electrónica de la sede Latacunga.

A mi tutor del proyecto de titulación, Dr. Luis Ismael Minchala, que me ha orientado para poder hoy alcanzar este objetivo en mi formación.

Carlos D. Morales J.

ÍNDICE DE CONTENIDOS

Carátula.....	1
Reporte de verificación de contenido	2
Certificación	3
Responsabilidad de autoría	4
Autorización de publicación	5
Dedicatoria.....	6
Dedicatoria.....	7
Agradecimiento	8
Agradecimiento	9
Tabla de contenidos.....	10
Indice de figuras	14
Indice de tablas.....	16
Resumen	17
Summary	18
Capítulo I: Introducción.....	19
Antecedentes	19
Planteamiento del problema	20
Justificación, importancia y alcance del proyecto (preguntas de investigación).....	21
Objetivos.....	22

Objetivo General.....	22
Objetivos específicos.....	23
Hipótesis de investigación.....	23
<i>Categorización de las variables de investigación (variables independiente y dependiente)</i>	<i>23</i>
Capítulo II: Marco Teórico	26
Los sistemas de automatización industrial.....	26
<i>Controlador lógico programable (PLC)</i>	<i>27</i>
<i>Interfaz humano-máquina (HMI)</i>	<i>27</i>
<i>Sistema de control distribuido</i>	<i>27</i>
<i>Sistema de control supervisor y adquisición de datos (SCADA)</i>	<i>27</i>
<i>Redes neuronales artificiales</i>	<i>28</i>
Pirámide de automatización industrial (CIM).....	28
<i>Campo/proceso.....</i>	<i>29</i>
<i>Control</i>	<i>29</i>
<i>Supervisión</i>	<i>29</i>
<i>Operación/ planificación</i>	<i>30</i>
<i>Gestión</i>	<i>30</i>
Redes industriales, protocolos de comunicación y buses de campo	31
<i>DeviceNet</i>	<i>31</i>
<i>Modbus</i>	<i>31</i>
<i>Protocolo de comunicación HART</i>	<i>32</i>
<i>Profibus</i>	<i>32</i>

	12
<i>Bus de campo – Fieldbus Foundation</i>	33
<i>OPC (Open protocol for communication)</i>	34
Factory IO	37
<i>Simulación 3D de fábrica</i>	38
Codesys.....	38
<i>Servidor Codesys OPC UA</i>	39
Plataforma Thinger.io	41
El Internet de las cosas	43
<i>Tecnologías posibles con IoT</i>	43
<i>IoT Industrial</i>	44
Capítulo III: Metodología.....	45
Descripción del funcionamiento de la planta de producción.....	46
Diagrama de instrumentación y procesos	47
Descripción de la arquitectura	49
<i>Primera capa o nivel de campo</i>	50
<i>Segunda capa o nivel de control</i>	50
<i>Tercera capa o nivel de supervisión</i>	57
<i>Virtualización de servicios</i>	62
Capítulo IV: Resultados	68
Análisis para el modelado y diseño de la planta.....	68
Análisis de las plataformas de interacción con la planta.....	69
<i>Sistema de monitoreo y control en planta</i>	70

<i>Sistema de monitoreo externo del proceso</i>	71
<i>Sistema de consulta y control externo por chat</i>	72
<i>Análisis de comunicación y protocolos de los sistemas de interacción con la planta simulada</i>	73
Análisis del envío de información a la base de datos local y a la nube de almacenamiento de información	78
<i>Análisis de la información almacenada en la base de datos local y la nube de almacenamiento de información</i>	79
<i>Problemática y solución para envío de información a la nube</i>	81
Análisis del tiempo de respuesta en la comunicación con el PLC	82
Comprobación de la hipótesis	83
<i>Facilidad de comunicación de los sistemas de automatización industrial con interfaces heterogéneas</i>	84
<i>Implementación de IoT industrial</i>	85
Capítulo V: Conclusiones y recomendaciones	87
Conclusiones	87
Recomendaciones	89
Trabajos futuros	90
Bibliografía	91
Anexos	93

Índice de figuras

Figura 1 Pirámide de automatización CIM	29
Figura 2 Comunicación de diferentes tecnologías a través de OPC UA.	35
Figura 3 Interconexión cliente servidor OPC-UA.	36
Figura 4 Escena con una brazo robótico y máquina CNC – Factory IO	37
Figura 5 Interfaz de funcionamiento de Codesys	39
Figura 6 Funcionalidad del servidor OPC.....	40
Figura 7 Conceptualización de Codesys en los equipos	41
Figura 8 Modelado de la planta de producción de piezas metalmecánicas y embalaje en el programa Factory IO	47
Figura 9 Arquitectura propuesta para el sistema	50
Figura 10 Configuración del driver OPC Client DA/UA	51
Figura 11 Diseño del primer proceso dentro de Factory IO	52
Figura 12 Programación de inicio del primer proceso en el programa Codesys	53
Figura 13 Tabla de variables en Codesys	54
Figura 14 Tabla de variables en Factory IO	55
Figura 15 Diseño del segundo proceso dentro de Factory IO	56
Figura 16 Diseño del tercer proceso dentro de Factory IO	56
Figura 17 Diseño del cuarto proceso dentro de Factory IO	57
Figura 18 Establecimiento de conexión para el servidor OPC-UA	58
Figura 19 Asignación de direcciones para Gateway y PLC	59
Figura 20 Asignación por defecto de la dirección IP para el PLC.....	59
Figura 21 Contenedor de variables implementadas en Codesys.....	60
Figura 22 Configuración de nodo OPC-UA Client para lectura de variables en Node-RED	61

Figura 23 Configuración de nodo OPC-UA Ítem para escritura de variables en Node-RED	62
Figura 24 Configuración del nodo MySQL en Node-RED.....	63
Figura 25 Configuración del nodo http request en Node-RED	65
Figura 26 Diagrama de envío de información desde Node-RED a las plataformas de almacenamiento de datos.....	66
Figura 27 Configuración del nodo telegram receiver en Node-RED	67
Figura 28 Disposición de los elementos “resbaladilla” en el diseño de la planta en Factory IO.....	69
Figura 29 Diseño por defecto realizado por la máquina CNC en las piezas en Factory IO.....	70
Figura 30 Diseño final del dashboard diseñado en Node-RED.....	71
Figura 31 Diseño final del dashboard diseñado en Thinger IO	72
Figura 32 Consulta de información y control desde el chatbot implementado en Telegram.....	73
Figura 33 Captura de envío de paquete del servidor a Node-RED en Wireshark.....	74
Figura 34 Captura de envío de paquete de Node-RED al chatbot en Wireshark	75
Figura 35 Captura de envío de paquete de Node-RED a MySQL en Wireshark	76
Figura 36 Tabla de la base de datos local en MySQL con datos almacenados.....	76
Figura 37 Captura de envío de paquete de Node-RED a Thinger.io en Wireshark.....	77
Figura 38 Contenedor de datos en Thinger.io con datos almacenados.....	78
Figura 39 Gráfica de crecimiento de piezas mecanizadas y cajas de acuerdo con los datos almacenados en MySQL	80
Figura 40 Gráfica de crecimiento de piezas mecanizadas y cajas de acuerdo con los datos almacenados en Thinger.IO.....	81

Índice de tablas

Tabla 1 Operacionalización de las Variables.....	24
Tabla 2 Simbología de instrumentación	48
Tabla 3 Resultados de tiempo de respuesta del PLC.....	83
Tabla 4 Costos de licencia por software.....	85

RESUMEN

Este proyecto de titulación implementa un conjunto de procedimientos para la interrelación de tecnologías heterogéneas de hardware y software con enfoque a la comunicación de internet industrial de las cosas (IIoT, por sus siglas en inglés). Bajo un ambiente de simulación se diseña un proceso industrial en el software Factory I/O que emula una línea de producción de piezas mecanizadas a través de máquinas de control numérico. Este proceso es controlado por una Raspberry Pi4 que hace las veces de un controlador lógico programado (PLC) gracias a las librerías contenidas en el entorno de desarrollo Codesys utilizado para el diseño del programa de control del proceso. La comunicación del PLC hacia la nube es implementada con la ayuda de un servidor del protocolo open protocol for communication (OPC) del programa Codesys. Este servidor actúa como *middleware* y permite la conexión entre cada uno de los niveles de la pirámide de automatización y el proceso simulado en Factory IO. Para el tráfico de información desde el servidor OPC hacia la nube se empleó el software Node-RED que permite establecer enlaces con la base de datos de los entornos de desarrollo IIoT y gestionar datos del proceso en tiempo real. Además, con el uso de los diferentes módulos de Node-RED se desarrolló un panel de visualización a través de *Dashboard* que permiten el monitoreo y control de la planta simulada de forma local y a través de HTTP se establece un enlace a la plataforma Thinger IO para crear un panel de visualización para la gestión por medio de Internet. Con el propósito de realizar el análisis comparativo de la eficiencia del manejo de información se utilizó la herramienta Wireshark, que permitió obtener resultados del envío de paquete de datos entre la plataforma de integración Node-RED y los protocolos de comunicación OPC UA, HTTP, MySQL y Python utilizado en la creación del *Chatbot* de Telegram para la gestión del proceso por mensajería móvil. El resultado en el análisis de la gestión de datos confirma la hipótesis planteada en que la comunicación entre las tecnologías heterogéneas es factible con ciertas limitantes en el tiempo de control y almacenamiento de datos que son propias de las plataformas utilizadas.

Palabras clave: IIoT, Codesys, *Middleware*, Node-RED.

SUMMARY

This degree project implements a set of procedures for the interrelation of heterogeneous hardware and software technologies with a focus on the communication of the Industrial Internet of Things (IIoT). Under a simulation environment, an industrial process is designed in the Factory I/O software that emulates a production line of machined parts through numerical control machines. This process is controlled by a Raspberry Pi4 that acts as a programmed logic controller (PLC) thanks to the libraries contained in the Codesys development environment used to design the process control program. Communication from the PLC to the cloud is implemented with the help of an open protocol for communication (OPC) server of the Codesys program. This server acts as *middleware* and allows the connection between each of the levels of the automation pyramid and the simulated process in Factory IO. For the traffic of information from the OPC server to the cloud, the Node-RED software was used, which allows establishing links with the database of the IIoT development environments and managing process data in real time. In addition, with the use of the different Node-RED modules, a visualization panel was developed through *Dashboard* that allows the monitoring and control of the simulated plant locally and through HTTP a link is established to the Thinger IO platform to create a display panel for management via the Internet. With the purpose of carrying out the comparative analysis of the efficiency of information management, the Wireshark tool was used, which allowed obtaining results from the sending of data packets between the Node-RED integration platform and the OPC UA, HTTP, MySQL communication protocols. and Python used in the creation of the Telegram *Chatbot* for the management of the mobile messaging process. The result in the analysis of data management confirms the hypothesis that communication between heterogeneous technologies is feasible with certain limitations in the time of control and storage of data that are specific to the platforms used.

Key words: IIoT, Codesys, *Middleware*, Node-RED.

Capítulo I

Introducción

Antecedentes

La automatización de un proceso es la transformación de una tarea ya sea del ámbito industrial, administrativa o científica; haciéndolo más ágil y efectivo, todo con la finalidad de hacer la tarea más sencilla para el ser humano (Alcuri, 2013).

La simulación se ha vuelto un instrumento necesario en cualquier propósito industrial e investigativo, por lo que hemos enmarcado su importancia en los siguientes aspectos:

- Tiene un menor costo y es más rápido que construir físicamente el sistema real.
- Ayuda a descubrir errores de diseño en el modelo en lugar de hacerlo en el sistema real.
- Sirve como instrumento de estimación y pronóstico.

La simulación permite optimizar el control y la toma de decisiones en el transcurso del tiempo que es un fenómeno que altera el proceso (Álvarez et al., 2013).

Se puede establecer mediante la investigación elaborada que existen pocos trabajos desarrollados en el país tanto por los estudiantes universitarios como del sector empresarial referentes al tema del proyecto propuesto. A continuación, citamos algunos de ellos.

En el repositorio de la Universidad Técnica de Ambato, facultad de Ingeniería en sistemas, Electrónica e Industrial se archiva el trabajo investigativo que evalúa la propuesta para el diseño de un HMI con el uso del conjunto de herramientas provistas por una plataforma web de acceso libre para los PLC s7-1200 y su incidencia en la reducción de costos de monitoreo frente a los sistemas propietarios de pago. Dicha investigación concluye que por medio del protocolo OPC UA se integran procesos industriales con el internet, aplicando las técnicas y lenguajes de programación web tales como: HTML, JQuery, CSS y AJAX (Nuñez, 2019).

El trabajo del autor (Lopez, 2019), presenta una propuesta de solución para el monitoreo de un proceso industrial basado en plataforma Industrial 4.0 con protocolo de comunicación MQTT, el proceso industrial es simulado en Factory IO y controlado Mediante PLC.

De igual manera tenemos el trabajo que se encuentra en el repositorio de la Universidad de las Américas, de los autores (Changuaro Vásconez et al., 2020) con enfoque en la industria 4.0; desarrollando una aplicación multiplataforma para supervisión y control eléctrico en industria 4.0 con equipos PLC siemens S7-1200". Dicho trabajo igualmente tiene enfoque en la Industria 4.0 desarrollando una aplicación multiplataforma para el control de equipos industriales desde el computador o una aplicación móvil con la herramienta de LABVIEW llamada Web Publishing Tool.

Finalmente, a nivel mundial podemos encontrar distintos trabajos investigativos y artículos que crean un precedente para el desarrollo del proyecto. Con la documentación adquirida se investiga y elige las tecnologías, protocolos de comunicación y entornos de desarrollo que permitan una implementación óptima del monitoreo y control por medio de una plataforma IoT del proceso simulado.

Planteamiento del problema

Existe una diversidad de plantas industriales a nivel mundial que realizan diferentes procesos en la elaboración de sus productos para que integren las necesidades del actual estilo de vida, por consiguiente, la automatización de estas plantas constituye un aspecto prioritario (Lopez de Ayala, 2017).

En el ambiente presente de competitividad global, adelanto tecnológico e innovación; las compañías sobre todo de producción se ven comprometidas a innovar sus métodos. La Industria 4.0 y la manufactura inteligente son parte de esta innovación, en la que tecnologías de fabricación e información se han integrado creando sistemas de manufactura modernizadas con procedimientos de gestión y convenciones que permiten mejorar los conocimientos de manufactura, obtener una

mayor lenidad, validez y concebir una propuesta para sus clientes, así como responder adecuadamente a las necesidades de su clientela (Aguilar et al.,2017).

Sin embargo, el estudio de la automatización de plantas industriales se tropieza con la dificultad de su aplicación en modelos físicos mediante los cuales se pueda probar los sistemas diseñados, por ende; la simulación se ha constituido una alternativa eficiente para la implementación a través de un entorno virtual, con el cual se comprueba el adecuado funcionamiento del sistema (Morales et al., 2013).

El internet industrial de las cosas y/o industria 4.0 es una tecnología que aún se encuentra en desarrollo y más aún en Latinoamérica y en el Ecuador; donde la industria necesita de profesionales altamente capacitados en este entorno de innovación tecnológica. La maestría en electrónica y automatización busca soluciones a este avance tecnológico de la industria en nuestro país, por lo que es necesario realizar trabajos investigativos con la exploración de entornos industriales simulados automatizados para la adquisición de conocimientos de la nueva tecnología.

Justificación, importancia y alcance del proyecto (preguntas de investigación)

La Industria 4.0 exigirá incorporar elementos y tecnologías de evolución dentro de la industria como Internet de las Cosas (IoT), la ciberseguridad o la realidad virtual, entre otras. La implementación de todo ello supone un desafío muy importante para las empresas. Por un lado, tendrán que integrar todo ello en sus procesos productivos y deberán preparar y formar a las personas para adaptarse al cambio y así permitirles trabajar en un entorno digital y totalmente conectado. Por lo tanto, la Industria 4.0 viene claramente incitada por el *software* y es primordial para la conversión digital.

Las empresas destinadas a conllevar procesos industriales tienen problemas con sus mantenimientos y correcciones tecnológicas; lo que es altamente costoso suspender una línea de producción. La industria 4.0 tiene una gran virtud de conocer el proceso o de otra manera

monitorearlo para anticiparse a un fallo, o lo que es igual simularlo, para poder modificar algunas falencias del sistema que impiden su óptimo desempeño.

La ayuda de una plataforma IoT es muy favorable al proceso industrial ya que obtiene información de sensores, variables de proceso y actuadores; para el técnico es una forma muy versátil para conocer a profundidad la planta. La base de datos donde se almacena dicha información puede tener acceso la gerencia y grupo administrativo.

En el Ecuador, poco o nada se aplica esta tecnología en la pequeña, mediana y grandes empresas, es ahí que surge la necesidad de investigar el tema propuesto. El proyecto de investigación a desarrollar es de gran importancia, ya que permitirá la adquisición de conocimientos para desarrollar soluciones tecnológicas de bajo costo para la empresas pequeñas y medianas del sector industrial del centro del país; dándoles competitividad en el mercado local.

Este proyecto servirá para futuras investigaciones sobre el manejo de procesos a través de una plataforma web, lo cual es importante para minimizar costos de operación y mantenimiento.

El alcance del proyecto se delimita en llevar a cabo la simulación de un proceso y su control mediante un PLC; el estado de variables de proceso, sensores y actuadores serán llevados a una aplicación web para el monitoreo del proceso. De igual manera el control de la planta será evidenciado con un diseño de pantalla HMI para corroborar el correcto desempeño ante variaciones externas a la planta. El manejo de la planta con la ayuda una plataforma de internet Thinger IO es la finalidad del proyecto.

Objetivos

Objetivo General

Implementar una solución de internet industrial de las cosas en el contexto de monitoreo y supervisión de un proceso de manufactura simulado que integra tecnologías de comunicación heterogéneas.

Objetivos específicos

- Investigar y seleccionar el *software* de simulación apropiado para el desarrollo del proceso industrial simulado.
- Desarrollar un proceso industrial simulado, integrando características de un proceso real.
- Diseñar una interfaz hombre máquina (HMI, por sus siglas en inglés), a través de *dashboards*.
- Realizar la comunicación del PLC con la plataforma IoT mediante el protocolo de comunicación OPC-UA.
- Implementar IoT con la plataforma de internet Thinger.io para manejo de la planta desde un móvil.
- Evaluar el control de la planta simulada y la adquisición de datos para su gestión desde la aplicación móvil.

Hipótesis de investigación

La implementación del IOT Industrial en los procesos industriales facilita la comunicación de sistemas de automatización industrial con interfaces de comunicación heterogéneos.

Categorización de las variables de investigación (variables independiente y dependiente)

En el planteamiento de la hipótesis se identifican dos variables:

- **Variable independiente:** la implementación de IOT Industrial.
- **Variable Dependiente:** Facilidad de comunicación de los sistemas de automatización industrial con interfaces heterogéneas.

La operacionalización de las variables se muestra en la siguiente tabla:

Tabla 1*Operacionalización de las Variables.*

VARIABLES	TIPO	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
La implementación de IOT Industrial	Independiente	<p>Es un sistema de dispositivos de informática conectados, aparatos mecánicos y electrónicos, objetos únicos con la capacidad de transferir información a través de una red.</p> <p>Reúne el aprendizaje de máquina y la tecnología de grandes volúmenes de datos (big data), valiéndose de los datos de sensores, comunicación M2M y las tecnologías de la automatización industrial.</p>	<p>La implementación de IOT industrial en el proceso simulado permitirá la optimización del control de las operaciones recayendo en la disminución de costos de fabricación.</p>	<p>- Costo de implementación.</p> <p>- Tiempo de implementación.</p>	<p>- Costo mínimo de implementación del sistema.</p> <p>- Tiempos cortos de implementación.</p>

VARIABLES	TIPO	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
Facilidad de comunicación de los sistemas de automatización industrial con interfaces heterogéneas.	Dependiente	Factibilidad en el diseño de sistemas HMI/SCADA con la integración de interfaces de comunicación como OPC UA	Fácil desarrollo del proyecto de automatización del proceso simulado con la integración de las tecnologías de comunicación.	<ul style="list-style-type: none"> - Heterogeneidad en la integración de <i>software</i> para control de la planta. - Tiempo de respuesta de la planta desde control remoto y dentro de ella. 	<ul style="list-style-type: none"> - Uso de <i>software</i> libre para control de la planta. - Control de tiempos de respuesta de la planta.

Capítulo II

Marco Teórico

Este capítulo describe el fundamento teórico del proyecto de investigación, Las principales definiciones teóricas se asocian con: la automatización industrial, las redes industriales y sus protocolos de comunicación, la industria 4.0, el internet industrial de las cosas (IIoT por sus siglas en ingles) y la simulación de procesos.

Los sistemas de automatización industrial

Los sistemas de automatización industrial fueron desarrollados para controlar y monitorear un proceso, máquina o dispositivo de manera computarizada que por lo regular cumplen funciones repetitivas. Tienen la razón de automatizar, supliendo el rol que desempeña una persona y las decisiones que esta toma en los procesos industriales para así mejorar el trabajo y esfuerzo de una persona en la manufactura. Mediante los diferentes comandos de programación y lógica se obtiene una potencialidad de las máquinas (21, 2017).

La automatización prescinde el trabajo manual, pero aún existe la necesidad de operarios que inspeccionen y lleven a cabo las actividades de mantenimiento de las máquinas. En cambio, una estructura de producción automatizada ya no requiere una persona que inspeccione el proceso; de tal forma, que cuando un sistema de dichas características es implementado en una empresa el proceso de producción se ejecuta en su totalidad sin asistencia humana. (Exsol, 2017)

La automatización facilita la respuesta a cambios diversos con necesidades inmediatas de intervención; como explotar los datos generados por servicios, maquinarias, plantas de producción y sistemas colectivos. Los datos son gestionados en tiempo real; lo que permite generar una información vital para tomar decisiones dentro de la gerencia de una empresa (Ripipsa, 2019).

Diferentes entornos de automatización pueden ser utilizados, tomando en cuenta las especificaciones que se quieran implementar en la automatización de los existentes procesos industriales, tales como:

Controlador lógico programable (PLC)

Son pequeños computadores programables que controlan máquinas en una línea de producción industrial y administran procesos de forma automatizada y en tiempo real. Están diseñados para controlar diferentes señales de entrada y salida. Estos dispositivos tienen distintas aplicaciones y, por lo tanto, tiene distintos usos como controlar la lógica de funcionamiento de máquinas, plantas y procesos industriales, etc. (GSL Industrias, 2015).

Interfaz humano-máquina (HMI)

Dependen de la intervención humana para funcionar. Los ordenadores y los cajeros automáticos son una muestra de sistemas de interfaz-humano-máquina.

Sistema de control distribuido

En esta disposición los controles no están centralizados y se encuentran expandidos por el sistema. Estos controles se propagan en función de la parte del sistema que se tenga que monitorizar. Se utilizan en el control de semáforos, refinerías petrolíferas y estaciones eléctricas (Exsol, 2017).

Sistema de control supervisor y adquisición de datos (SCADA)

Pueden estar compuestos por otros sistemas o subsistemas; como interfaces humano-máquina o acoplados a unidades terminales remotas (RTU), que convierten señales de sensores en datos comprensibles para el SCADA. Este sistema es aplicado para controlar, vigilar y registrar por completo los eventos que se producen en el lugar de manufacturación e inclusive de diversas plantas de producción. Tienen homogeneidad con el control distribuido, con la principal diferencia

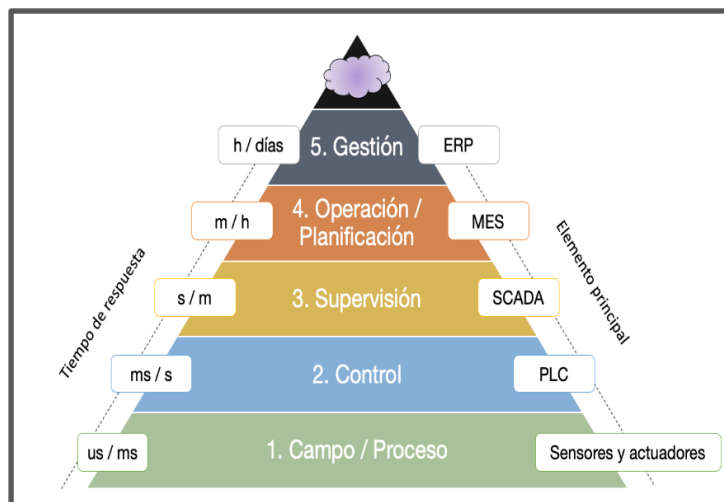
que los SCADA el control no se realiza en tiempo real. No obstante, la línea que difiere entre estos es muy difusa y a menudo se usan en aplicaciones similares (Corrales, 2007).

Redes neuronales artificiales

Es un modelo matemático o computacional que emula los enlaces neuronales y, como sucede en estas redes biológicas, la estructura de la red se adapta a los cambios dependiendo de la información. Se usan para identificar patrones en algoritmos de datos. Entre otras de sus aplicaciones están como filtros de spam en email, reconocimiento de esquemas en sistemas (radares), sistemas automatizados de finanzas y de control (García, 2010).

Pirámide de automatización industrial (CIM)

El nivel de automatización de un proceso viene determinado primordialmente por factores de tipo económico y tecnológico, por ello podemos encontrar una gama muy amplia y variada, dependiendo de los objetivos a alcanzar. Con el fin de optimizar la calidad, producción, automatización de operaciones y la gestión eficiente de las empresas; surge la necesidad de integrar los procesos de producción con los de gestión; lo cual implica una distribución de los sistemas de automatización en cinco niveles; representados pictóricamente en la pirámide de la automatización CIM (manufactura integrada por computador por sus siglas en ingles). En la figura 1 se muestran los dispositivos más característicos y los tiempos de respuesta aproximados para cada nivel. (Brunete et al.,2020)

Figura 1*Pirámide de automatización CIM*

Nota. Tomado de (Brunete et al.,2020)

Campo/proceso

Este es el nivel más bajo de la jerarquía de automatización. Esto incluye actuadores, sensores y diferentes elementos de hardware que componen máquinas y/o procesos. Estos sensores y actuadores que interactúan directamente con el proceso de fabricación se conocen comúnmente como dispositivos de campo. (Field devices). Dichos dispositivos permiten adquirir los datos del proceso (Tabares, 2007).

Control

Este nivel está formado por las máquinas individuales que intervienen en el proceso de manufactura. Por tanto, aquí se encuentran los PLCs que son aquellos que controlan los sistemas numéricos de las máquinas, brazos robóticos, ente otros. Es parte fundamental para el manejo de los procesos de una manera automatizada.

Supervisión

En este nivel es donde encontramos los sistemas SCADA, como también las interfaces humano-máquina (HMI). Mediante las interfaces de usuario se puede supervisar los diferentes

procesos y adquirir información en bases de datos. Un sistema SCADA asiste en el control de múltiples procesos complejos y ubicados en diferentes sitios.

Se marca una diferencia con los niveles 1 y 2 en tanto que un sistema SCADA se usa para depurar o restablecer valores en el nivel de control (Alcuri, 2013).

Operación/ planificación

El cuarto nivel es el encargado de monitorizar todo el proceso de fabricación en una planta o fábrica desde las materias primas hasta el producto terminado. Por lo tanto, permite a la administración conocer exactamente lo que está sucediendo, pudiendo tomar decisiones con esta información, como, reducir la cantidad de pedidos de materia prima o disminuir el desperdicio en un proceso de maquinado. En este nivel se utiliza la gestión informática literalmente conocido como MES (Manufacturing Execution System) (Brunete, et al., 2020).

Gestión

El nivel de la planificación de recursos empresariales conocido como ERP. La alta gerencia puede tomar decisiones y controlar sus operaciones. El ERP con la ayuda de aplicaciones informáticas tiene el manejo de lo que sucede en una fábrica. Esto permite a la empresa ser capaz de monitorear todos los niveles del negocio desde la producción, las ventas, las compras, las finanzas y la nómina, entre muchos otros. Mientras que el MES controla y vigila una empresa, el ERP supervisa, controla y mantiene informado de sus procesos a corporaciones.

Actualmente se podría señalar otro nivel por encima del ERP; siendo este la nube, la cual desarrolla la parte de transformación digital de la empresa; donde los datos de cualquiera de los niveles anteriores pueden ser directamente alimentados en un *pool* de datos u otras aplicaciones. Alcanzando la eficiencia y operatividad de la denominada Industria 4.0 (European Commission, 2018).

Redes industriales, protocolos de comunicación y buses de campo

Los diferentes tipos de redes de comunicación que están diseñados para interactuar con los dispositivos en el campo con los módulos de entrada y salida, basados en los protocolos de comunicación. El protocolo es un conjunto de normas para la comunicación entre dos o más dispositivos.

Basándose en estos protocolos, las redes de comunicación se clasifican en muchos tipos de estándares de comunicación comunes y populares de la industria que a continuación se describen:

DeviceNet

Se trata de una red de protocolo abierto basada en tecnología CAN. Permite la interconexión de los dispositivos de nivel de campo como: sensores, lectores de códigos de barras, interruptores, HMI, etc., con los controladores con un nivel jerárquico superior como los PLCs. El protocolo CAN en su forma básica puede soportar 64 nodos y 2048 punto de conexión.

Disminuye la carga de información en la red cuando integra los dispositivos a un cable de 4 hilos. Reduce los puntos de conexión físicos mediante la alimentación de los dispositivos directamente.

Dicha red es conocida en el sector automotriz y en los semiconductores para circuitos electrónicos (Hurtado, 2016).

Modbus

Es un protocolo abierto que puede funcionar en una variedad de capas físicas y es el más común en el sector industrial. Utiliza la comunicación en serie maestro-esclavo para así obtener información entre los dispositivos de dicha red. Compatible con cualquier medio de transmisión es comúnmente utilizado con RS232 y RS485.

Como capa física el modbus serial RS232 o RS485 facilita la interconectividad de dispositivos Modbus a un bus de datos estructurado. Puede comunicarse entre un maestro y varios esclavos, hasta 247 con una velocidad de transmisión de datos de 19,2 kbits/s.

Modbus TCP/IP es una versión reciente que utiliza Ethernet para la transmisión de datos entre un PLC en diferentes tipos de redes. Con independencia del tipo de red física, facilita un método de acceso y control de un dispositivo por otro (Padilla, 2019).

Protocolo de comunicación HART

Es la única red de comunicación industrial que hace posible tanto la comunicación analógica como la digital en sentido bidireccional y al mismo tiempo por cableado único, por ello se denomina una red híbrida. El diagnóstico del equipo, Los parámetros de configuración del dispositivo, la calibración y otras mediciones adicionales del proceso es información que puede estar contenida en esta señal digital.

HART funciona, tanto como red punto a punto como en multipunto. En modalidad punto a punto la señal de corriente de 4-20 mA es utilizada para controlar el proceso mientras que la señal digital permanece invariable. La estructura multipunto es aprovechada cuando los dispositivos están ampliamente esparcidos. Los instrumentos de campo inteligentes que incorporan HART en sus comunicaciones son utilizados a menudo en muchas industrias. Esta red de comunicación se utiliza principalmente en aplicaciones de Sistemas SCADA (Plant Web University, 2012).

Profibus

Se utiliza principalmente en los campos de la automatización de procesos y de la automatización de fábricas; en tareas de comunicación complejas y aplicaciones en las que el tiempo es un factor imperante.

Existen tres variantes, que son: Profibus DP (Periferia descentralizada), Profibus PA (Automatización de procesos) y Profibus FMS (Especificación de mensajes de bus de campo).

Profibus DP es un bus de campo abierto que maneja la comunicación maestro/ esclavo entre sus dispositivos de red. Como medio de transmisión emplea RS485 o fibra óptica en la capa física y su principal uso es proporcionar comunicación entre controladores y módulos de E/S distribuidas.

Profibus PA, desarrollado primordialmente para la automatización de procesos. Es recomendado para áreas intrínsecamente seguras. Permite que sensores, actuadores y controladores se enlacen a través de un bus común, que proporciona comunicación de datos y alimentación. Utiliza la capa física Manchester Bus Powered (MBP), basada en la norma internacional IEC 61158-2.

Profibus FMS es un formato de mensajería multimaster o peer-to-peer, el cual faculta la comunicación de las unidades maestras entre sí. Establece tareas de transmisión de información en el nivel de control, esencialmente en el subnivel de celda para facilitar el enlace entre PC's maestros.

El uso de FMS y DP simultáneamente en el modo COMBI será requerido en redes en las que se utiliza un PLC junto con un PC. En estas condiciones, el maestro primario se conecta con el maestro secundario mediante FMS, en tanto que los datos de control de la misma red a los dispositivos de E/S es transmitida por intermedio de Profibus DP (Universidad de Valencia, 2015).

Bus de campo – Fieldbus Foundation

Provee una infraestructura de comunicación de campo abierto, perfeccionado para satisfacer las exigencias de los entornos intrínsecamente seguros. Es un modelo de red LAN para instrumentos y controladores provistos con Fieldbus Foundation que es el órgano rector que garantiza la interoperabilidad de los dispositivos bajo la norma IEC 61158-2 (para FF H1) y compatible con equipos Ethernet (21, 2017).

OPC (Open protocol for communication)

Basado bajo la arquitectura de cliente- servidor; un servidor OPC puede estar comunicándose de manera continua con los controladores de campo, RTUs, dispositivos HMI u otras aplicaciones, aun cuando los dispositivos físicos y el *software* sean suministrados por diferentes marcas comerciales. Su implementación viabiliza la transferencia de información de manera continua en tiempo real. Es de lejos la tecnología estándar de comunicación industrial y cuenta con las arquitecturas OPC clásico y OPC UA (Lemos, 2006).

OPC clásico. Fue el resultado de la investigación conjunta de diferentes fabricantes en cooperación con Microsoft. De ello resultó una tecnología basada en el COM / DCOM (Distributed Component Object Model) de los sistemas operativos de Microsoft. Las especificaciones que rigen esta arquitectura son:

Acceso a datos OPC (OPC DA). Establece la tipología de transmisión de datos, incluido valores, el tiempo y la calidad de la información.

Alarmas y eventos OPC (OPC AE). Define el intercambio de información de mensajes de tipo de evento y alarma, así como estados variables y gestión de estados.

Acceso a datos históricos OPC (OPC HDA). Determina metodologías de consulta y diagnóstico que pueden aplicarse a datos históricos con marca de tiempo.

Las especificaciones del OPC Clásico han servido bien a la comunidad OPC, pero tienen algunas limitaciones tales como:

- Dependencia de la plataforma en Microsoft
- Modelos de datos insuficientes
- Seguridad inadecuada

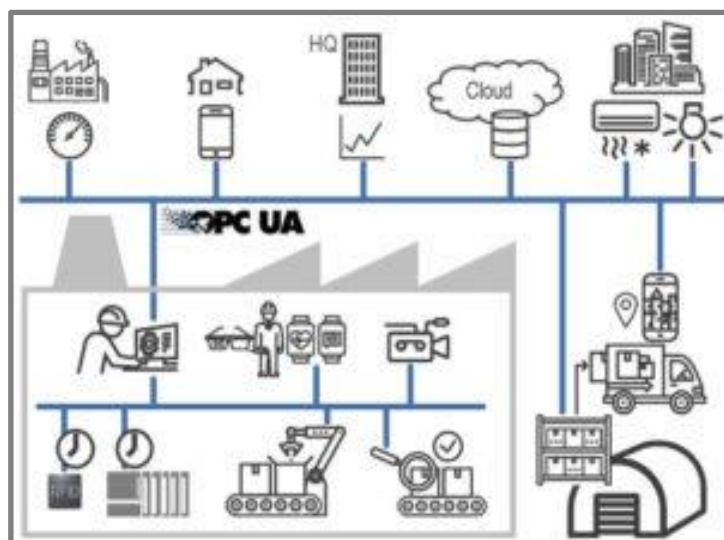
A medida que la tecnología evolucionó, también lo hizo la necesidad de las especificaciones OPC y debido a sus limitaciones fue imposible su escalabilidad.

OPC Unified Architecture (OPC UA). Lanzada en el 2008, es una arquitectura totalmente nueva, orientada a servicios independientes de la plataforma; integra toda la funcionalidad de las especificaciones del OPC Clásico existentes. Es más segura, abierta y confiable para transferir información entre servidores y clientes; proporcionando mejor seguridad y un modelo de información más completo.

Está diseñada para conectar bases de datos, herramientas analíticas, sistemas de planificación de recursos empresariales (ERP) y otros sistemas empresariales (figura 2) con datos reales de controladores de gama baja, sensores, actuadores y dispositivos de monitoreo que interactúan con procesos reales y que controlan y generan datos reales.

Figura 2

Comunicación de diferentes tecnologías a través de OPC UA.



Nota. Tomado de (OPC Foundation, 2019)

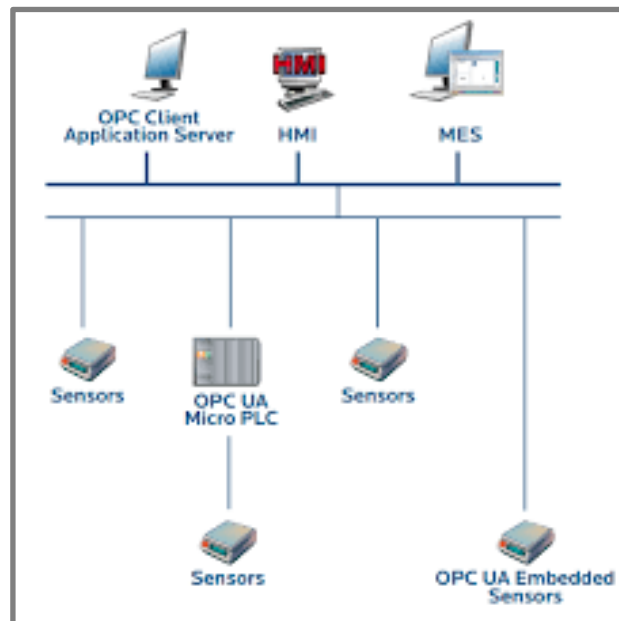
Es escalable, proporciona múltiples modelos de seguridad, varias capas de transporte y un avanzado diseño de información para que cualquier controlador de bajas prestaciones consiga interactuar con aplicaciones sofisticadas de los servidores de alto nivel. OPC UA hace posible la transmisión de cualquier consulta y notificación, desde un simple estado de inactividad hasta casi

ilimitadas sumas de información complejas de toda la planta. Las características de esta tecnología única incluyen escalabilidad, seguridad, capacidad de Internet, Interoperabilidad certificada, sofisticada gestión de alarmas y eventos. Entre otros.

Los servidores OPC UA son gestores de datos que compilan información y la muestran en formas que son útiles para diversos tipos de clientes OPC UA, dispositivos ubicados en la periferia de una fábrica como HMI's o RTU's, controladores patentados; programas gestores de recetas, de base de datos o programas de análisis sofisticado, ubicados en servidores empresariales (figura3).

Figura 3

Interconexión cliente servidor OPC-UA.



Nota. Tomado de (OPC Foundation, 2019)

De manera notable, los datos no están restringidos a un solo nodo físico. Los objetos pueden estar relacionados a otros objetos, a variables de diferentes tipos de datos que coexistan en nodos externos de la subred o en otros lugares de la arquitectura o inclusive en algún espacio dentro de Internet.

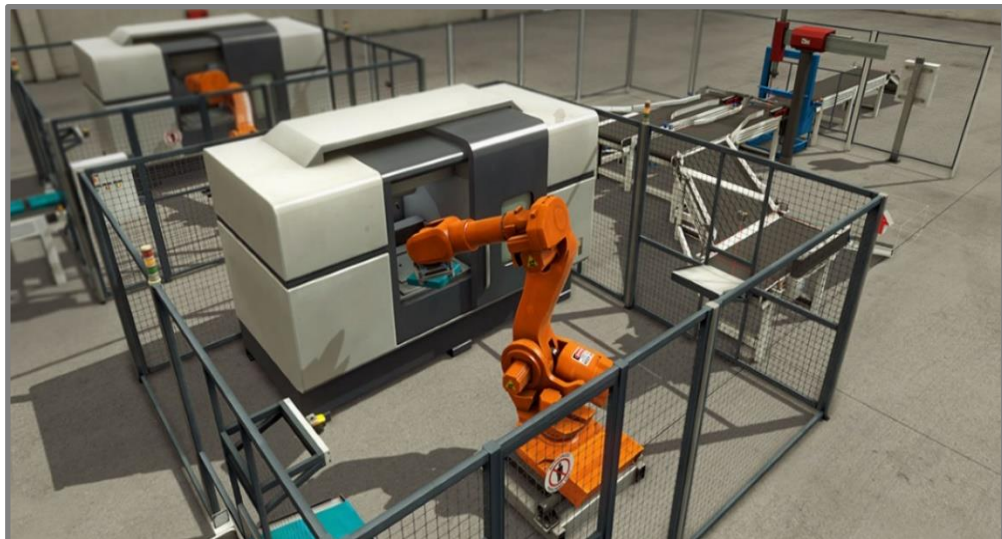
OPC UA dispone los procesos, sistemas, datos e información de tal manera que sean únicas para la experiencia de la automatización. Es una arquitectura inmejorable que resuelve paradigmas de comunicación diferentes a las solucionadas por los protocolos EtherNet / IP, Modbus TCP y Profinet IO Ethernet. OPC UA es una herramienta de entrega y modelado de información que brinda acceso a esa información a los clientes en toda la empresa.

Factory IO

Se comprende como el concepto de Factory IO a la simulación de un proceso 3D con el fin de un mejor aprendizaje de nuevas tecnologías. Las piezas que contiene este *software* son las que se manejan en una industria y sirven para construir un prototipo lo más cercano a la realidad. Además, existen escenas con fundamentos a los procesos de manufactura más comunes con diferentes niveles de dificultad para el programador (Sánchez, 2018).

Figura 4

Escena con una brazo robótico y máquina CNC – Factory IO



Nota. Tomado de (Sánchez , 2018)

Dicho de otra manera, Factory IO se concede como una plataforma virtual para la capacitación del PLC como un órgano vital en la automatización, utilizando también microcontroladores, Modbus, SoftPLC, Modbus, entre otras tecnologías. En la figura 4 se puede visualizar una escena.

Simulación 3D de fábrica

Este tipo de simulación es totalmente interactiva e incluye gráficos y sonido de alta calidad, lo que proporciona un entorno industrial realista. Al mismo tiempo, utiliza una tecnología avanzada, que permite la creación fácil y rápida de sistemas de automatización 3D, controlables en tiempo real conectando Factory IO y un dispositivo externo (PLC o microprocesador).

Las características clave que hacen de Factory IO un gran *software* para aprender tareas de control reales, a través de API, son: las escenas inspiradas en los procesos industriales más populares, utilizando una biblioteca de más de 80 mecanismos industriales, con los que además se pueden diseñar una ilimitada cantidad de fábricas virtuales; la capacidad de crear estrategias para el análisis de fallas, tanto para enchufes abiertos como en cortocircuito; la seguridad y efectividad del *software* al vincularse con aplicaciones y protocolos de comunicación industriales comunes, evitando los problemas de costo, daños a personas y equipos (Sánchez , 2018).

Codesys

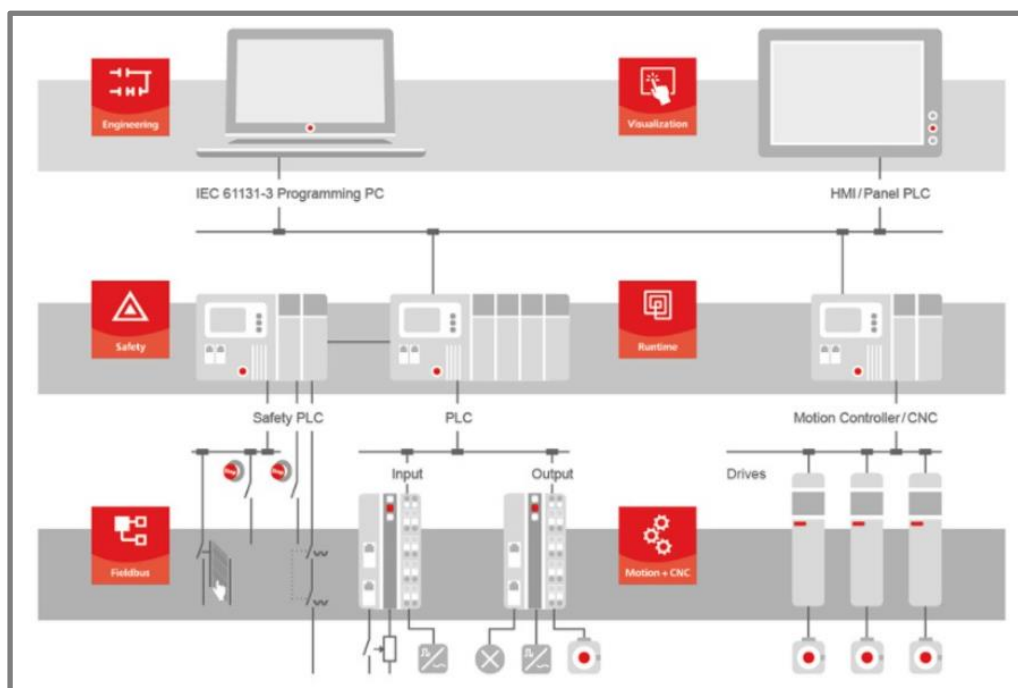
Es un *software* orientado a las diferentes tecnologías de la pirámide de la automatización industrial. Se basa en un protocolo de desarrollo Codesys mediante la herramienta IEC 61131-3 lo que proporciona a los usuarios la implementación de un soporte ante los entornos de automatización. Codesys es el programa ideal para la Industria 4.0, maneja controladores niebla y nube (“fog” or “cloud”) y por medio del intercambio de datos para redes IoT son las facilidades para el usuario.

El escalamiento de autómatas programables es posible con las diversas funciones de Codesys donde se integra los niveles de la automatización cumpliendo con éxito las exigencias del

proyecto. Estas herramientas cubren ampliamente los objetivos, la programación y la ingeniería que es la parte vital de cada estación de trabajo (SUPRESS, 2017).

Figura 5

Interfaz de funcionamiento de Codesys



Nota. Tomado de (SUPRESS, 2017)

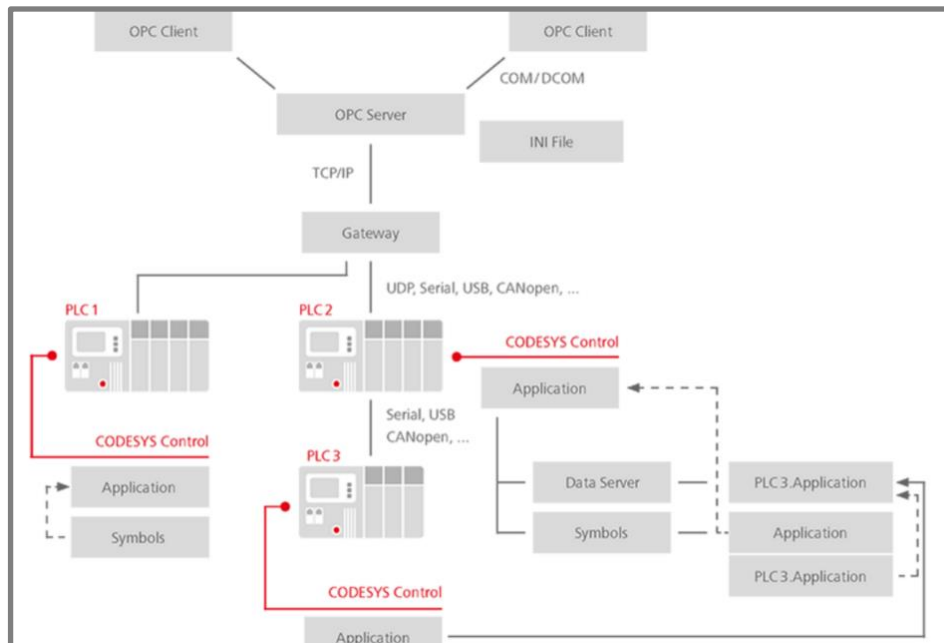
La figura 5 muestra la comunicación integrada entre diferentes equipos a través de un nivel jerárquico de funciones que permite obtener la base de datos que contiene la información necesaria para dar soluciones a problemas suscitados en los procesos de manufactura, ensamble, etc.

Servidor Codesys OPC UA

El intercambio de datos es la tarea principal del servidor OPC UA lo que corresponde a escritura y lectura, parte fundamental para la visualización y registro de los datos en el proceso. Es un *software* adicional de Windows que se configura adicional del sistema IEC 61131-3 pero necesita una licencia llamada SoftContainer para ejecutarla.

Figura 6

Funcionalidad del servidor OPC



Nota. Tomado de (SUPRESS, 2017)

Como se muestra en la figura 6, la funcionalidad del servidor OPC establece de manera automática la comunicación con el cliente e indica cuando se genera un cambio del estado de datos (elementos OPC); es válida la administración de los diferentes tipos de elementos de los grupos (privados o públicos). Funciones integradas en diferentes dispositivos en la rama de automatización:

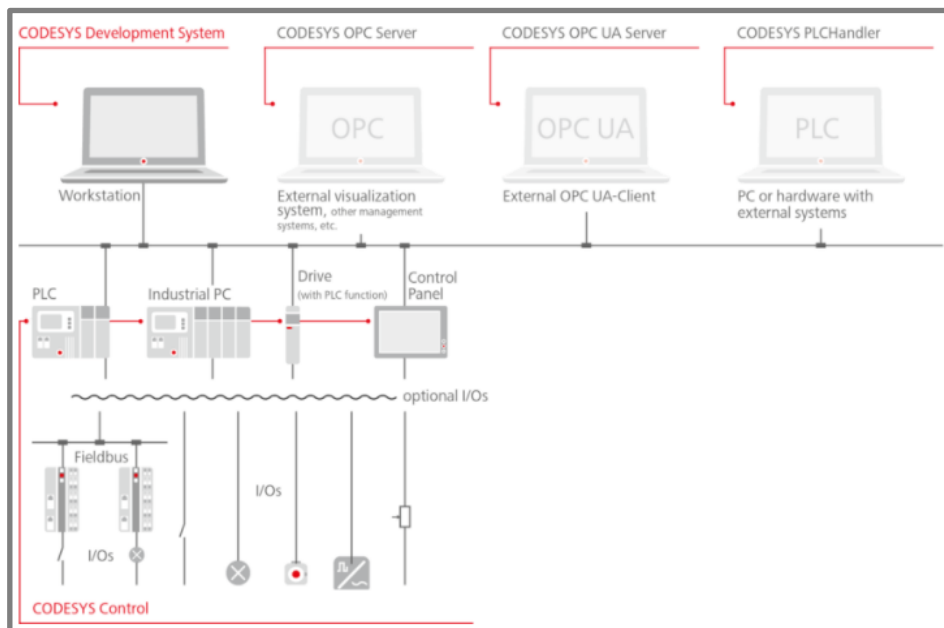
- Configuraciones adaptadas a los buses de campo más habituales E/S (CANopen, EtherCAT y PROFINET).
- Opciones de productos adicionales para diseñar y visualizar interfaces de usuario en acción (por ejemplo, a través de la pantalla de los dispositivos, una pantalla externa conectada o mediante un navegador web).

- Muchos productos opcionales para extender rápidamente la gama de funciones de los dispositivos como: controladores de movimiento, paneles de control o puertos de comunicación)
- Configurar y diseñar protocolos de comunicación específicos de la industria (por ejemplo, para construir un sistema de control de servicios con BACnet).

A continuación, se observa en la figura 7 la conceptualización de los dispositivos.

Figura 7

Conceptualización de Codesys en los equipos



Nota. Tomado de (SUPRESS, 2017)

Plataforma Thinger.io

Contiene librerías de código abierto para la gestión de multitud de dispositivos, entre los que se destacan ESP8266, Arduino, Raspberry Pi, etc. a través de Internet, respondiendo al significado de Internet de las cosas.

Esta plataforma permite recopilar y visualizar información generada por sensores y dispositivos de control; además, es posible graficar los datos para interpretarlos de una manera simple. La aplicación también permite exportar los datos con el propósito de tratarlos con programas de análisis.

Se puede enviar datos o instrucciones de control a distintos dispositivos, parametrizar dispositivos mediante Internet, tal como fijar la temperatura a la que una caldera se encienda o hacer que se abra o cierre una compuerta al requerirse.

Permite a través de aplicaciones como IFTT, crear conexiones con varios *servicios en línea*: servicios de almacenamiento como Google Drive, Dropbox y SkyDrive; distintas redes sociales como Twitter, Facebook, YouTube e Instagram entre otras; canales RSS de blogs y páginas web, correos electrónicos, mensajería de texto, climatológicos y muchos más. La plataforma cuenta con su versión demo con las siguientes limitaciones:

- Conexión de un máximo de 3 dispositivos.
- No existe limitación en cuanto a los recursos de cada dispositivo, es decir el número de parámetros a medir por los sensores conectados o los enviados a través de la plataforma a los dispositivos.
- Se pueden almacenar hasta en un máximo de 10 campos (*Data Buckets*) diferentes. Cada *Data Bucket* puede acumular datos de varios sensores, con una periodicidad máxima de un minuto. Estos datos son accesibles y actualmente no hay limitaciones ni relacionado al tiempo ni al tamaño de los datos.
- Los valores de los equipos e instrumentos de campo pueden ser monitoreados hasta en un máximo de 10 pantallas gráficas (*Dashboards*). Los datos pueden provenir de *Data Buckets* o ser lecturas que se originan al momento en que se reinicia los *Dashboard*.

- Se pueden establecer hasta 10 Endpoints que son las “*marcas*” para interactuar con las aplicaciones de los *servicios en línea*.

El Internet de las cosas

El Internet de las cosas describe una red de objetos físicos que integran sensores, software y otras tecnologías con el fin de conectar e intercambiar datos con otros dispositivos y sistemas en Internet. Estos dispositivos van desde artículos domésticos comunes hasta herramientas industriales sofisticadas.

Tecnologías posibles con IoT

La idea del IoT se concibe desde hace mucho tiempo, pero una serie de avances recientes en tecnologías diferentes la han hecho posible.

Accesibilidad a tecnología de detección de bajo costo y consumo. Los sensores confiables y asequibles hacen que la tecnología IoT sea accesible para más fabricantes.

Conectividad. Muchos protocolos de comunicación diseñados para Internet han hecho posible el enlace de sensores a la nube para una transmisión de datos conveniente incluso a nivel industrial.

Plataforma de computación en la nube. La creciente disponibilidad de plataformas en la nube brinda a las empresas y los consumidores acceso a la infraestructura que necesitan para escalar sin tener que administrarlo todo.

Aprendizaje automático y análisis. Con los avances en el aprendizaje automático y el análisis, así como el acceso a grandes y diversas cantidades de datos almacenados en la nube, las empresas pueden recopilar información de forma más rápida y sencilla. Por un lado, el surgimiento de estas tecnologías aliadas continúa ampliando los límites de IoT; por otro lado, los datos generados por IoT también potencian estas tecnologías.

Inteligencia Artificial (IA). La evolución de las redes neuronales ha traído habilidades para el procesamiento de lenguaje natural (NLP por sus siglas en inglés) a los equipos IoT como Alexa, Cortana y el asistente digital personal Siri entre otros; haciéndolos atractivos, asequibles y factibles para el uso doméstico.

IoT Industrial

Se refiere a la aplicación de la tecnología IoT en entornos industriales, especialmente relacionada con la instrumentación y el control de sensores y dispositivos que utilizan tecnología en la nube. En tiempos recientes, la industria global ha utilizado la comunicación M2M para acceder al control y automatización inalámbrica de procesos. Pero con la aparición de la nube y las tecnologías relacionadas (como el análisis y el aprendizaje automático), las industrias pueden obtener una nueva capa de automatización y, con ella, generar ingresos y un nuevo modelo de negocio. IIoT se considera a menudo como la cuarta ola de la revolución industrial. Estos son algunos usos comunes de IIoT:

- Fabricación Smart.
- Mantenimientos preventivos y correctivos.
- Redes de energía inteligentes.
- Ciudades inteligentes.
- Cadenas de manufactura con gran versatilidad y funcionamiento con bajo porcentaje erróneo.

Capítulo III

Metodología

Este capítulo describe de manera técnica la metodología empleada para la implementación de las capas correspondientes a la pirámide de automatización CIM. Se especifican los procesos desarrollados para el diseño y ejecución de la línea de producción de la planta. Se presenta el funcionamiento de la planta de producción, diagrama de instrumentación y arquitectura por capas.

Para la implementación del trabajo de investigación se toma como modelo la pirámide de automatización CIM truncada. La arquitectura CIM truncada se conforma únicamente de los primeros 3 niveles de la arquitectura total; la interconexión de éstas comprende el uso de redes industriales, en este caso mediante simulación, los otros 2 niveles se pueden concebir intrínsecamente como la virtualización de servicios.

Los niveles de planificación y gestión, conocidas como MES y ERP, de la pirámide CIM del modelo truncado se virtualizarán de forma particular a través de una solución IIoT. Esta solución involucra la integración de al menos un *middleware* capaz de gestionar el protocolo OPC UA, para solucionar la heterogeneidad de comunicación, además de incorporar el flujo de información hacia la nube.

Se parte desde el nivel de campo en donde se desarrolla una planta de producción simulada en el programa Factory IO. El funcionamiento de esta planta se automatiza mediante un controlador o PLC.

En el nivel de control, se utiliza la Raspberry Pi4. Ésta hace las veces de PLC gracias a la plataforma de programación Codesys, que contiene librerías útiles y necesarias para este propósito.

En el nivel de supervisión se utiliza Node-RED, herramienta de control y supervisión de medios IoT. Se implementa un servidor OPC-UA en Codesys que cumple la funcionalidad del *middleware* antes mencionado.

Para la virtualización de las capas superiores el monitoreo de la planta se lleva hacia otras plataformas, con el fin de visualización remota a través de *dashboards*, almacenamiento y procesamiento de información.

Descripción del funcionamiento de la planta de producción

Se desarrolla una planta de producción que emula una línea de fabricación de piezas metalmecánicas y embalaje. La planta diseñada tiene dos líneas de fabricación y una de embalaje y empaquetado, divididas en cuatro procesos y seccionadas en dos plantas: baja y alta.

La línea de fabricación de la pieza “A” ejecuta el primer y tercer proceso que se encuentran en las plantas baja y alta mediante la utilización de un elevador. La línea de fabricación de la pieza “B” desarrolla el segundo proceso que se encuentra únicamente en la planta baja. Cada una de estas líneas cuenta con una máquina CNC que maquina las piezas según las formas, dimensiones y materiales requeridos.

Finalmente, la línea de embalaje y empaquetado desarrolla el cuarto proceso que se encuentra igualmente en la planta baja. Ésta es la línea final de la planta y se encarga de ordenar y empaquetar las piezas para su respectivo almacenamiento y posterior distribución.

La figura 8 muestra la planta de producción diseñada en la plataforma Factory IO con los procesos claramente diferenciados y numerados. Estos procesos se profundizan en la sección 3.3.2.2.

Figura 8

Modelado de la planta de producción de piezas metalmecánicas y embalaje en el programa Factory

IO



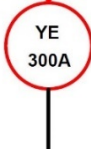
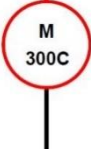
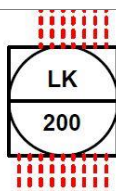
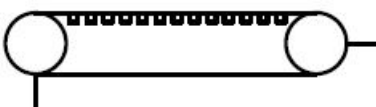
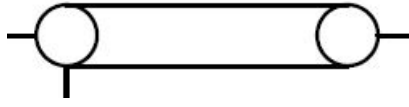
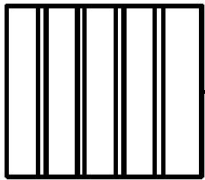
Diagrama de instrumentación y procesos

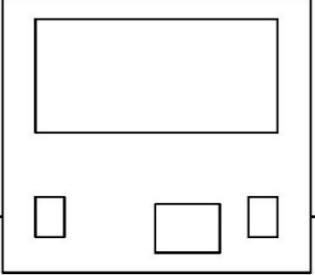
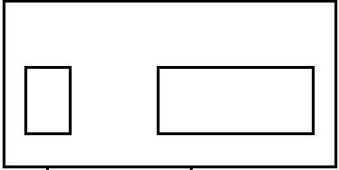
El diagrama de instrumentación y procesos está diseñado de acuerdo con la norma ISA/ANSI 5.1. Este diagrama facilita al lector la comprensión de la distribución y conexión de cada uno de los componentes involucrados en el proceso de automatización industrial. En la sección de Anexos se presenta con mayor claridad este diagrama; tanto para la planta baja como la planta alta.

El proceso industrial que se describirá más adelante emplea diferente maquinaria para cada una de las etapas. En consecuencia, en la Tabla 2 se detalla la categoría, descripción y simbología de cada uno de ellos.

Tabla 2

Simbología de instrumentación

Categoría	Descripción	Simbología
Elementos primarios	State Sensor (YE) montado en campo	
	Motor montado en campo	
Control compartido	Control Station (LK) Ubicación accesible normalmente al operario	
Componentes adicionales	Banda transportadora con rodillos giratorios	
	Banda transportadora lisa	
	Ascensor	

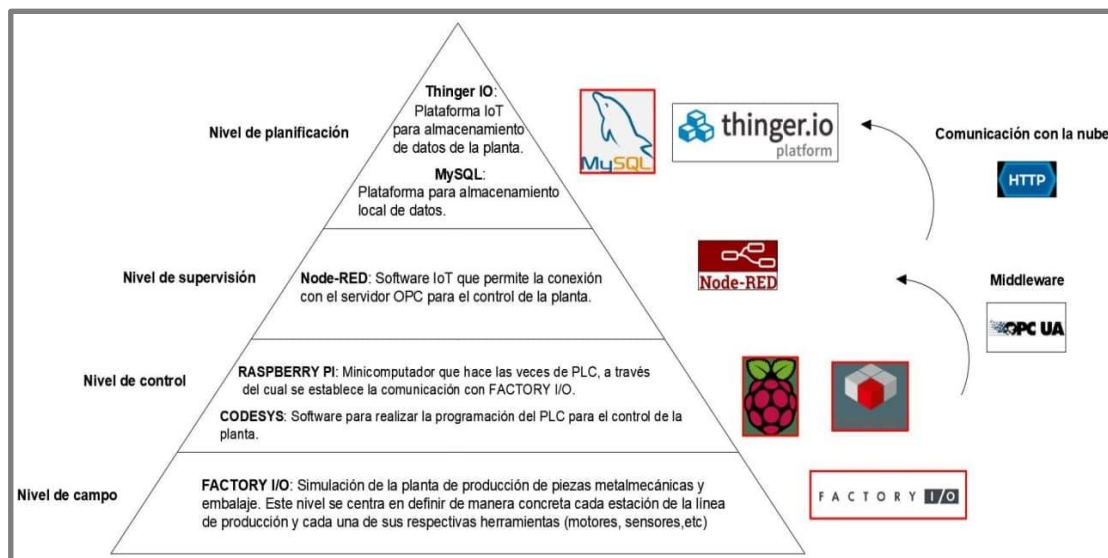
Descripción	Simbología
CNC	
Brazo robótico	

Descripción de la arquitectura

El sistema utiliza la arquitectura CIM truncada de tres capas y se añade una cuarta capa que corresponde a la virtualización como se presenta en la figura 9, de manera general la primera capa se encarga de la simulación de la planta de producción, la segunda se encarga del control, la tercera capa de la supervisión y la última capa se encarga de la virtualización de la información correspondiente a la planta.

Figura 9

Arquitectura propuesta para el sistema

**Primera capa o nivel de campo**

La primera capa corresponde a la simulación de la planta de producción diseñada en el programa Factory IO. En ésta se disponen diferentes equipos y dispositivos físicos como sensores y actuadores, que en conjunto con bandas y motores permiten la interacción entre ellos. Esta capa se encarga de recolectar información y comunicar a la capa superior.

Segunda capa o nivel de control

La segunda capa permite el uso de dispositivos encargados del control de la línea de manufactura, introduciendo así el PLC. Como se observa en la figura 9, la Raspberry Pi4 es el dispositivo físico que hace las veces de PLC. Para este propósito, se utiliza el *software* Codesys que es un entorno de desarrollo para la programación de controladores.

Para la conexión con Factory IO, se enlazan los respectivos puertos a través de los drivers de este programa, con el objetivo de automatizar la factoría virtual a través del PLC implementado en la

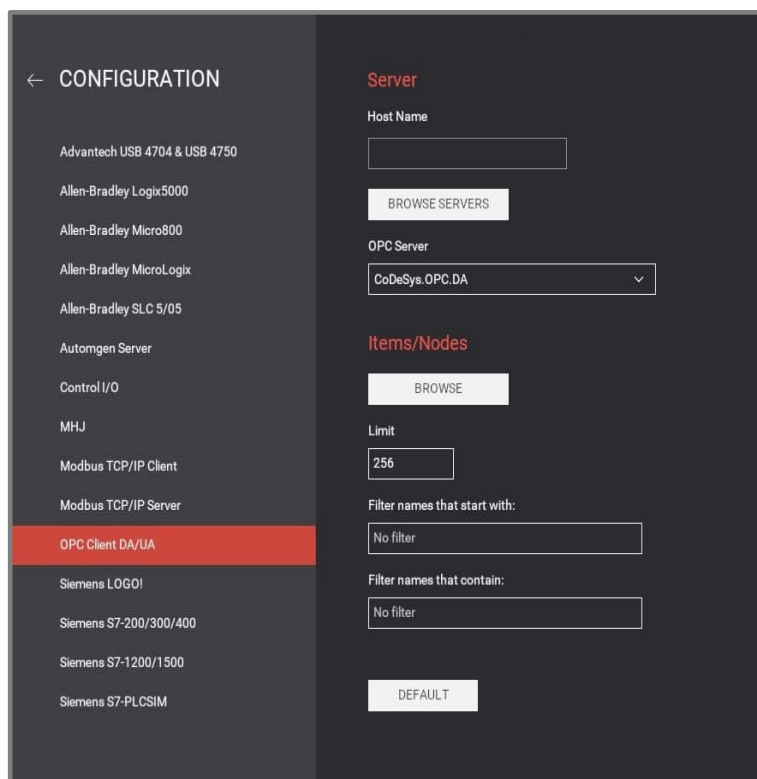
Raspberry Pi4. La programación del PLC se desarrolla utilizando programación en escaleras (*ladder diagram*) en combinación con programación estructurada en el ambiente Codesys.

Configuración de drivers en el programa Factory IO. Existen cuatro etapas de proceso dentro de la planta, que se gestionan de manera independiente en bloques de programación y luego se integran a un solo programa. De esta manera se ejecuta la automatización de todos los procesos.

En el programa Factory IO se selecciona el driver *OPC Client DA/UA*, éste se enlaza con el servidor OPC-UA. La figura 10 presenta la configuración establecida para este driver. El dispositivo físico Raspberry Pi4 hace las veces de PLC, esto mantiene el principio de heterogeneidad de este trabajo; y para los objetivos que se requieren cumplir, sus características de procesamiento son suficientes para la implementación de los procesos que involucra el funcionamiento de la planta.

Figura 10

Configuración del driver OPC Client DA/UA

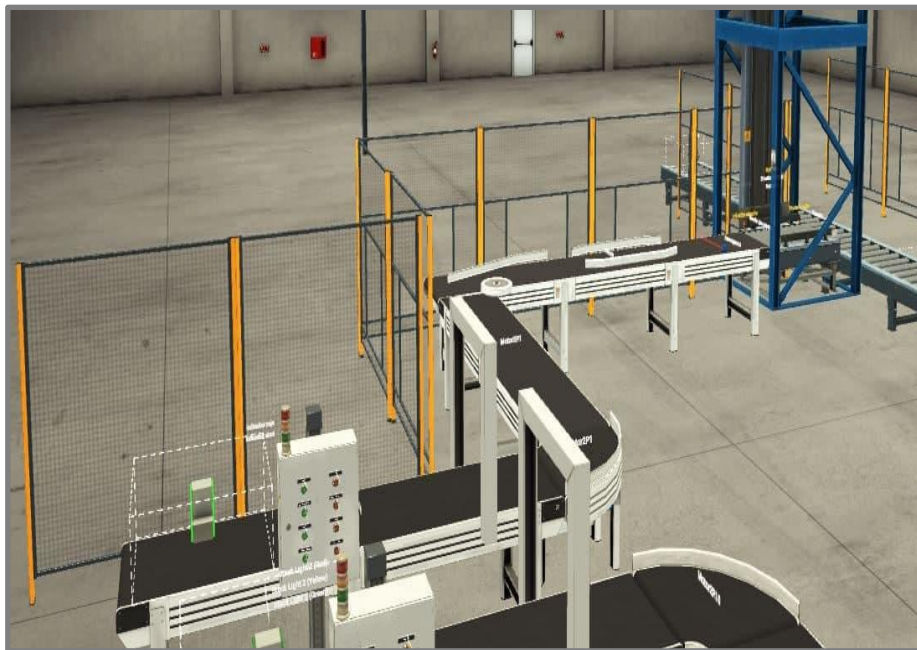


Criterios de automatización de la planta en Codesys. El control de procesos y su automatización emulados en Factory IO son programados en la plataforma Codesys. En todos los procesos están involucrados motores, bandas, actuadores y sensores los cuales envían información hacia el PLC para su control. Los procesos 2 y 3 incluyen una máquina CNC cada uno. A continuación, se describen cada uno de los procesos que ejecuta la planta de producción de piezas metalmecánicas.

Primer proceso: Fabricación de pieza "A". El primer proceso está formado por tres bandas lineales y una tipo codo que permiten el transporte de la pieza "A". Para la programación de este proceso únicamente se accionan o inician los motores de cada una de las bandas y su control se realiza mediante pulsantes ubicados en el panel de control de la planta. Al final de la línea se encuentra un elevador que transporta la pieza "A" a la planta alta, éste comunica al primer proceso con el tercero. La figura 11 presenta el diseño de esta sección de la planta.

Figura 11

Diseño del primer proceso dentro de Factory IO



La programación del accionamiento o inicio de los motores del primer proceso dentro de Codesys se presenta en la figura 12, Codesys permite la programación en un ambiente gráfico de tipo escalera.

Figura 12

Programación de inicio del primer proceso en el programa Codesys



Cada uno de los actuadores tiene asignada una variable de lectura o escritura dentro del proceso. Esta asignación se realiza para todos los actuadores dentro de la pestaña “Variables_Globales” que se encuentra dentro “Configuración de símbolos”. Ésta última es la encargada de dotar de atributos y derechos de acceso a todas las variables involucradas en el proceso. La figura 13 presenta las variables que han sido asignadas para todo el proceso de automatización de la planta en Codesys.

Figura 13

Tabla de variables en Codesys

Device.Application.Variables_Globales		
Expresión	Tipo de datos	Valor
STARTP1	BOOL	FALSE
STOPP1	BOOL	TRUE
MOTOR1P1	BOOL	FALSE
MOTOR2P1	BOOL	FALSE
MOTOR3P1	BOOL	FALSE
MOTOR4P1	BOOL	FALSE
MOTOR5P1	BOOL	FALSE
PROCES01	BOOL	FALSE
LIMITELEVADOR	BOOL	TRUE
MOTORASPOS	BOOL	TRUE
MOTORASNEG	BOOL	FALSE
PROCES02	BOOL	FALSE
SENSORP2	BOOL	FALSE
PROCES03	BOOL	FALSE
PROCES04	BOOL	FALSE
PROCES05	BOOL	FALSE
PROCES06	BOOL	FALSE
PROCES07	BOOL	FALSE
SENSORTOPEAS	BOOL	FALSE
STARTP2	BOOL	FALSE
STOPP2	BOOL	TRUE
MOTOR1P2	BOOL	FALSE
MOTOR2P2	BOOL	FALSE

A diferencia de programar variables etiquetadas para un PLC físico, dado que se ejecuta un servidor OPC-UA dentro del dispositivo Raspberry Pi4, únicamente se tiene que asignar nombres a cada una de las variables. Esto para que Factory IO desde su instancia de cliente OPC-UA reconozca el servidor y establezca la conexión de variables en Codesys y actuadores en este programa. La figura 14 presenta la asignación de variables en de Factory IO.

Figura 14

Tabla de variables en Factory IO

Server: CoDeSys.OPC.DA (79)		
	BRAZOUNOPOS	Brazo1pos
	EMISORPAQUETES	EmisorPaquetes
FinCarreraAscensor	...RRERAASCENSOR	
	INICIOCNC	InicioCNC
Limite Elevador	LIMITELEVADOR	
	MOTOR10P10	Motor10P10
	MOTOR10P2	Motor10P2
	MOTOR11P10	Motor11P10
	MOTOR11P2	Motor11P2
	MOTOR12P10	Motor12P10
	MOTOR12P2	Motor12P2
	MOTOR13P10	Motor13P10
	MOTOR13P2	Motor13P2
	MOTOR14P10	Motor14P10
	MOTOR1P1	Motor1P1
	MOTOR1P10	Motor1P10
	MOTOR1P2	Motor1P2
	MOTOR1P5	Motor1P5
	MOTOR1PEMPA	Motor1P4
	MOTOR2P1	Motor2P1
	MOTOR2P10	Motor2P10
	MOTOR2P2	Motor2P2
	MOTOR2P5	Motor2P5
	MOTOR2PEMPA	Motor2P4
	MOTOR3P1	Motor3P1
	MOTOR3P10	Motor3P10
	MOTOR3P2	Motor3P2
	MOTOR3P5	Motor3P5
	MOTOR4P1	Motor4P1
	MOTOR4P10	Motor4P10
	MOTOR4P2	Motor4P2
	MOTOR4P5	Motor4P5

Segundo proceso: Fabricación de pieza “B”. El segundo proceso está conformado por bandas de transporte para la pieza “B”. Este proceso cuenta con una máquina CNC que se activa mediante la lógica de un sensor óptico. Luego de que la pieza ha sido mecanizada, se transporta mediante un conjunto de bandas hasta el final de la línea, donde mediante el uso de sensores ópticos, se activa un brazo mecánico de succión que traslada esta pieza hacia la línea de transporte principal. La línea de transporte principal lleva las piezas “A” y “B” al cuarto proceso. La figura 15 presenta el diseño de esta sección de la planta.

Figura 15

Diseño del segundo proceso dentro de Factory IO



Tercer proceso: Mecanizado y transporte de pieza "A". El tercer proceso inicia una vez que el elevador entrega la pieza "A" en la planta alta. La pieza se traslada mediante un conjunto de bandas hasta una máquina CNC donde se desarrolla un mecanizado similar al del proceso anterior. La pieza mecanizada ingresa a un nuevo conjunto de bandas que cuenta con una resbaladilla que traslada la pieza hasta la línea de transporte principal. La figura 16 presenta el diseño de esta sección de la planta.

Figura 16

Diseño del tercer proceso dentro de Factory IO



Cuarto proceso: Embalaje y empaquetado. El cuarto proceso corresponde a la última etapa de toda la producción de la planta. Para la programación de este proceso únicamente es necesaria la activación de los actuadores, que a su vez acciona los motores. Al final de la línea de transporte principal se encuentra un sensor óptico, que al detectar la presencia de una pieza envía una caja mediante una banda para que ésta sea empaquetada. La figura 17 presenta el diseño de esta sección de la planta.

Figura 17

Diseño del cuarto proceso dentro de Factory IO



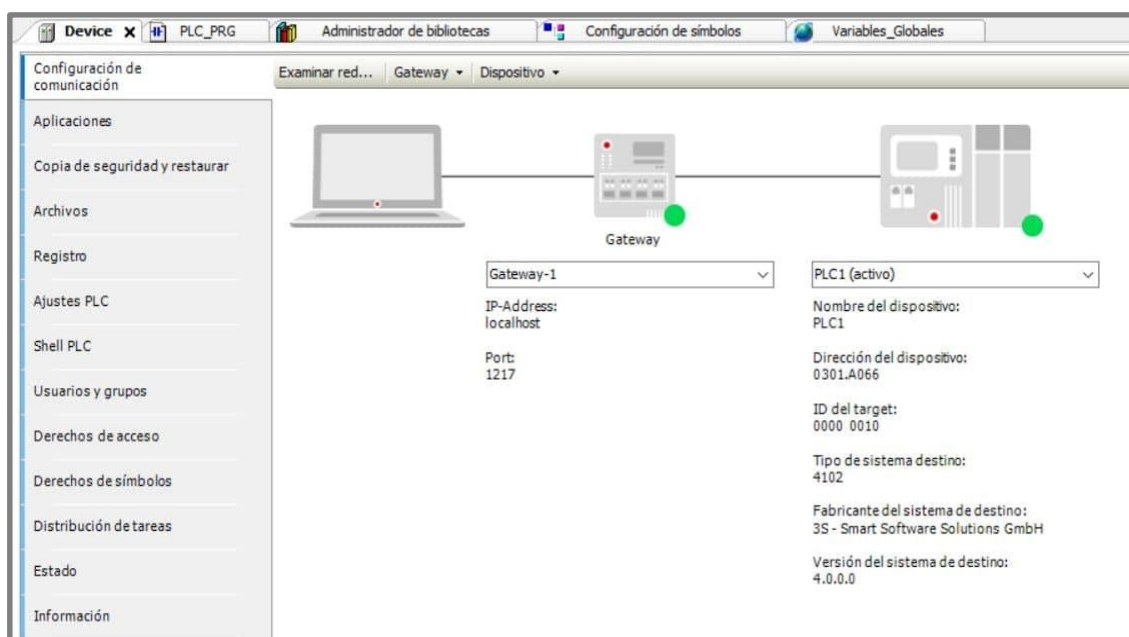
Tercera capa o nivel de supervisión

La tercera capa corresponde a los sistemas de adquisición y supervisión de datos. Se utiliza la herramienta Node-RED que cuenta con funcionalidades IoT. Esta capa está enlazada a través del *middleware* de la planta en el cual se ejecuta el protocolo de comunicación OPC-UA que se establece en la Raspberry Pi4 que hace las veces de PLC. De esta manera, se ejecuta el servidor, que realiza la lectura y adquisición de las variables y los datos desde el PLC.

Creación del servidor OPC-UA y enlace del cliente. La capa que permite la adquisición de los datos es conocida como *middleware*, en esta capa se implementa el protocolo de comunicación OPC-UA, cuyas características fueron descritas en la sección teórica. Este servidor permite que todos los datos que se encuentran en el PLC sean extraídos para realizar su monitoreo y control.

Figura 18

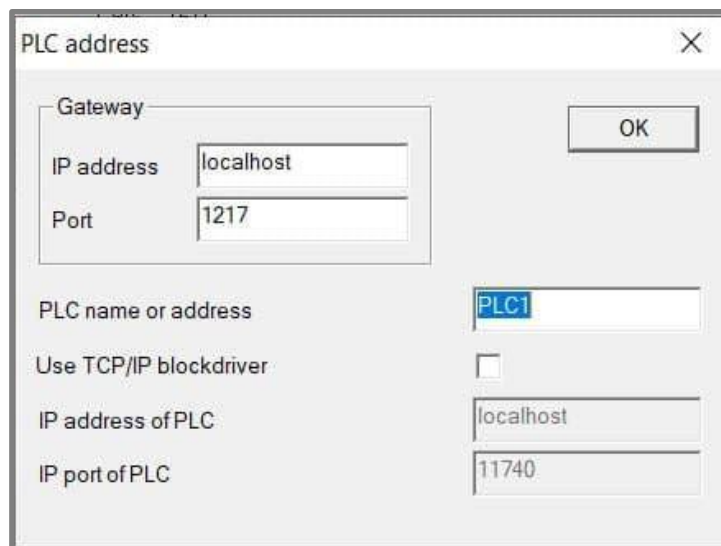
Establecimiento de conexión para el servidor OPC-UA



Para la creación del servidor se utiliza el *software* Codesys con su librería para comunicación OPC-UA. Este servidor se alojará en el dispositivo Raspberry Pi4, que cumple la función de PLC. Dentro del programa se deben asignar las direcciones IP para la creación del servidor. La figura 18 presenta el sistema de conexión del servidor OPC-UA. Como se puede apreciar en la imagen, existen dos instancias a configurar, la del *Gateway* y la del servidor; por lo tanto, se establecen dos direcciones IP, dirección *localhost* para el *Gateway* y dirección bajo el nombre PLC1 para el servidor; esto se evidencia también en la figura 19 donde se presenta la asignación de direcciones.

Figura 19

Asignación de direcciones para Gateway y PLC

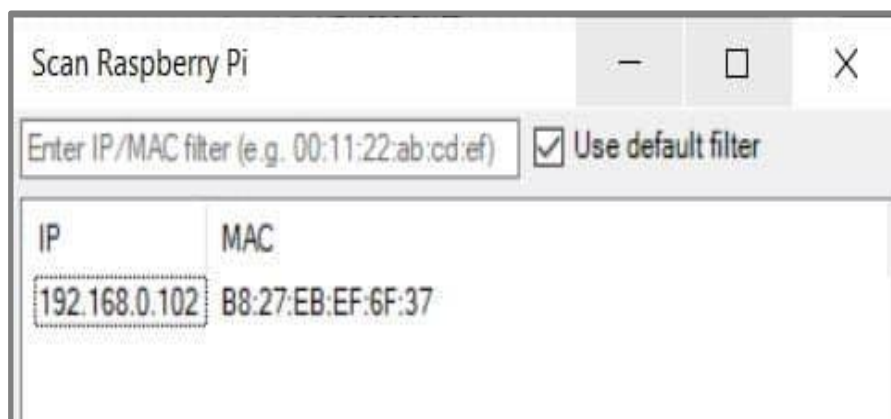


Gateway	
IP address	localhost
Port	1217
PLC name or address	PLC1
Use TCP/IP blockdriver	<input type="checkbox"/>
IP address of PLC	localhost
IP port of PLC	11740

La dirección IP del PLC, bajo el nombre asignado de PLC1, es definida por defecto dentro de Codesys. La herramienta asigna la dirección IP de manera dinámica, para el caso de la captura que se presenta en la figura 20 la IP asignada es 192.168.0.102. La dirección IP para el servidor OPC-UA varía dependiendo de la red a la que se conecta la Raspberry Pi4.

Figura 20

Asignación por defecto de la dirección IP para el PLC



IP	MAC
192.168.0.102	B8:27:EB:EF:6F:37

Para que el servidor se ejecute de manera correcta, todos los actuadores deben asignarse a una variable, con el fin de que todas se alojen dentro de un contenedor y se pueda realizar su supervisión y monitoreo. Si las variables no se encuentran dentro del contenedor, no son adquiridas por los clientes. En la Figura 21 se presenta el contenedor de variables implementado en Codesys.

Figura 21

Contenedor de variables implementadas en Codesys

La configuración modificada de símbolos se transmitirá con la siguiente descarga o cambio en línea.

Símbolos	Derechos de acceso	Máximo	Atributo	Tipo	Variables de miembro	Comentario
<input checked="" type="checkbox"/> BRAZOUNOPOS				BOOL		
<input checked="" type="checkbox"/> CONTADORCAJAS				WORD		
<input checked="" type="checkbox"/> CONTADORCNC1				WORD		
<input checked="" type="checkbox"/> CONTADORCNC2				WORD		
<input checked="" type="checkbox"/> EMISORPAQUETES				BOOL		
<input checked="" type="checkbox"/> FINCARRERAASCENSOR				BOOL		
<input checked="" type="checkbox"/> INICIOCNC				BOOL		
<input checked="" type="checkbox"/> LIMITELEVADOR				BOOL		
<input checked="" type="checkbox"/> MOTOR10P10				BOOL		
<input checked="" type="checkbox"/> MOTOR10P2				BOOL		
<input checked="" type="checkbox"/> MOTOR11P10				BOOL		
<input checked="" type="checkbox"/> MOTOR11P2				BOOL		
<input checked="" type="checkbox"/> MOTOR12P10				BOOL		
<input checked="" type="checkbox"/> MOTOR12P2				BOOL		
<input checked="" type="checkbox"/> MOTOR13P10				BOOL		
<input checked="" type="checkbox"/> MOTOR13P2				BOOL		
<input checked="" type="checkbox"/> MOTOR14P10				BOOL		
<input type="checkbox"/> MOTOR15P10				BOOL		
<input type="checkbox"/> MOTOR16P10				BOOL		
<input checked="" type="checkbox"/> MOTOR1P1				BOOL		
<input checked="" type="checkbox"/> MOTOR1P10				BOOL		
<input checked="" type="checkbox"/> MOTOR1P2				BOOL		

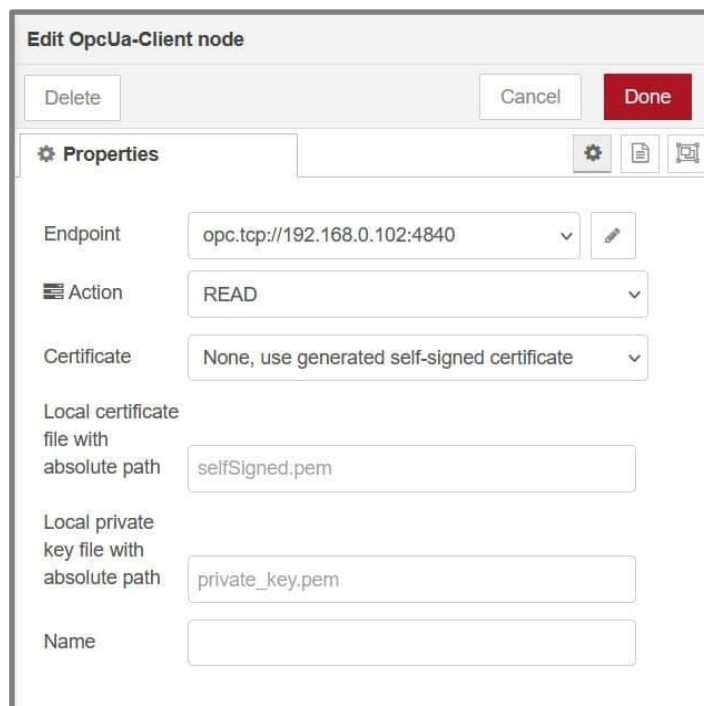
Establecimiento del cliente Node-RED. Node-RED es una herramienta que se destaca por su facilidad de uso, basado en el navegador de internet o web funciona como un editor de flujo donde se puede añadir o eliminar nodos y enlazarlos entre sí con el fin de que se intercomunicen entre ellos. La funcionalidad de esta herramienta recae también en

la facilidad del envío de información a la nube. Además de la implementación de un *dashboard* o panel de visualización para el monitoreo de la planta de producción.

Como se mencionó en la sección anterior, para el desarrollo de este trabajo el protocolo de comunicación OPC-UA permite la obtención los datos desde el PLC. Para ello, se procede con la configuración del nodo *OPC-UA Client*. La figura 22 presenta la configuración de este nodo dentro del entorno web de Node-RED, en éste se asigna la dirección del servidor OPC-UA y su puerto, así como la acción de lectura (*READ*) para la obtención de la información. Para la acción de escritura se utiliza un nodo *OPC-UA Item* en el que se indica el nombre y tipo de variable, como se presenta en la figura 23.

Figura 22

Configuración de nodo OPC-UA Client para lectura de variables en Node-RED



The image shows a configuration window titled "Edit OpcUa-Client node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a "Properties" section with a gear icon and a refresh icon. The configuration fields are as follows:

- Endpoint:** A dropdown menu showing "opc.tcp://192.168.0.102:4840" with a pencil icon to its right.
- Action:** A dropdown menu showing "READ".
- Certificate:** A dropdown menu showing "None, use generated self-signed certificate".
- Local certificate file with absolute path:** A text input field containing "selfSigned.pem".
- Local private key file with absolute path:** A text input field containing "private_key.pem".
- Name:** An empty text input field.

Figura 23

Configuración de nodo OPC-UA Ítem para escritura de variables en Node-RED

The screenshot shows a dialog box titled "Edit OpcUa-Item node". At the top, there are three buttons: "Delete", "Cancel", and "Done". Below the buttons is a "Properties" section with a gear icon and three sub-panels: "Item", "Type", and "Value". The "Item" field contains the text "ns=4;s=|var|CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.STARTP1". The "Type" field is a dropdown menu currently showing "Boolean". The "Value" field contains "msg.payload". Below these fields is a "Name" field containing "StartTrue1".

Virtualización de servicios

Como se describió en la sección 3.1 del trabajo, a partir de una arquitectura CIM truncada, las capas superiores se conciben desde una virtualización de servicios que permite plantear la solución IIoT desde una perspectiva de monitoreo y control de la planta de producción diseñada. Esto es posible gracias a la configuración del *middleware* en el que se ejecuta el servidor OPC UA, servidor donde se alojan todas las variables que contienen la información del funcionamiento de la planta. Los servicios implementados son los siguientes:

Dashboard o panel de visualización. Se implementa un *dashboard* utilizando la plataforma Node-RED. A partir de la programación mediante nodos y flujos, se vuelve sencilla la creación de un panel de visualización con la adición de *widgets* propios de la plataforma. Este *dashboard* se diseña en tres secciones separadas. Éstas se detallan a continuación:

1. Niveles de producción de las máquinas CNC: presentan los conteos de piezas fabricadas por cada máquina, es decir piezas "A" y piezas "B", al igual que el total. Para esto se utilizan *widgets* de nivel tipo brújula.

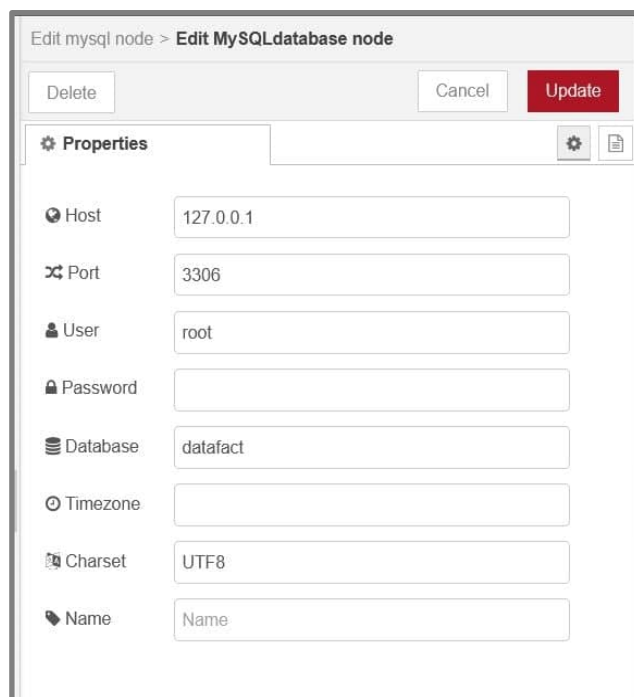
- Control de procesos: presenta la activación y desactivación (ON-OFF) para los procesos 1, 2, 3 y 4. Para esto se utilizan *widjets* de cambio de estado.
- Indicadores de las secciones activas de la planta: para esto se utilizan *widjets* tipo LED que emiten luz verde y roja para indicar la activación o desactivación de secciones como el elevador, máquinas CNC, brazo mecánico y línea de empaquetado, respectivamente.

Más adelante, en la sección de resultados, se presenta la visualización del diseño de este *dashboard*.

Base de datos local. Para el almacenamiento de datos de manera local se utiliza herramientas de *software* libre como el sistema de gestión de bases de datos MySQL junto con phpMyAdmin, herramienta para el manejo de la administración de las bases de datos de MySQL. Ésta última se utiliza únicamente para la visualización de la información que recibe la base de datos.

Figura 24

Configuración del nodo MySQL en Node-RED



The image shows a screenshot of the Node-RED configuration interface for a MySQL database node. The window title is "Edit mysql node > Edit MySQLdatabase node". At the top, there are three buttons: "Delete", "Cancel", and "Update". Below the buttons is a "Properties" section with a gear icon and a document icon. The properties are listed as follows:

Property	Value
Host	127.0.0.1
Port	3306
User	root
Password	
Database	datafact
Timezone	
Charset	UTF8
Name	Name

En MySQL se crea una base de datos de nombre “datafact” y una tabla que almacenará las variables de la planta, que lleva el mismo nombre. El envío de datos se ejecuta desde el cliente implementado en Node-RED, esto debido a que esta plataforma posee la facilidad de conexión directa con una base de datos mediante la instalación del nodo MySQL.

En la figura 24 se presenta la sencilla configuración de este nodo de conexión con el entorno MySQL, para el que se define la dirección de *localhost*, puerto y usuario de la base de datos. Para esto, claramente existe una instancia anterior que obtiene la información de las variables mediante una lectura, la reúne y la comunica a la base de datos en MYSQL para su almacenamiento. El envío de información desde Node-RED se realiza cada 2 segundos, tiempo que ha sido estimado como adecuado de acuerdo con los requerimientos de esta planta de producción.

Envío de datos a la nube. Para el envío de datos a la nube se utiliza Thinger.io que es una plataforma de *software* libre para aplicaciones IoT que proporciona una infraestructura de nube escalable lista para conectar diversos dispositivos o sistemas de automatización. Se requiere la creación de un *data bucket*, que se desempeña como una memoria virtual dentro de la plataforma, con esto se consigue almacenar el comportamiento de la planta dentro de la nube.

El envío de datos se ejecuta desde el cliente implementado en Node-RED, mediante el protocolo de transferencia de hipertexto (HTTP) desde un nodo tipo *http request*. Para lo cual, en Thinger.io se define un *token* de acceso al cual se le asigna el permiso de escritura en el *data bucket*. La solicitud HTTP de Node-RED es configurada con los siguientes parámetros de conexión hacia la plataforma Thinger.io, como se presenta en la figura 25:

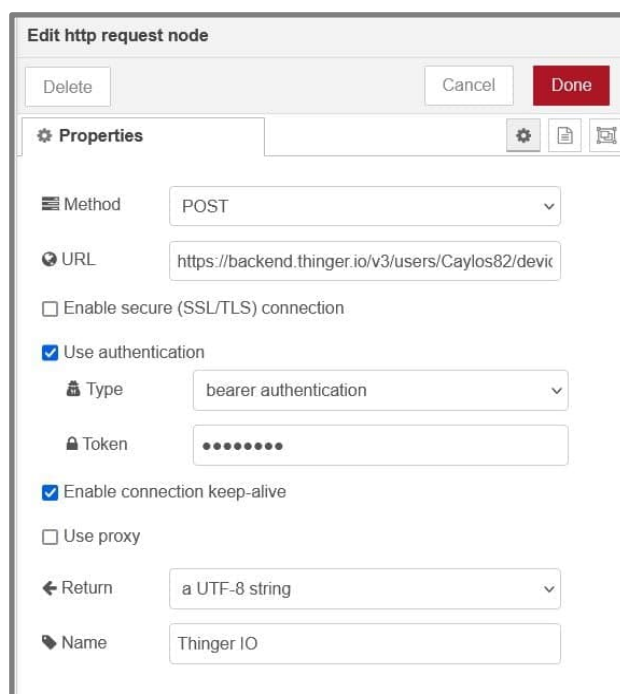
- método POST,
- dirección URL correspondiente a la *data bucket* en Thinger.io,
- autenticación tipo *bearer* y asignación de *token* de acceso definido en Thinger.io,

- habilitación de conexión *keep-alive*, y,
- asignación de cadena UTF-8 como *return* del método asignado.

Al igual que en el caso de la base de datos local, el envío de información desde Node-RED se realiza cada 2 segundos bajo la misma estimación indicada.

Figura 25

Configuración del nodo http request en Node-RED



The image shows the configuration window for an 'http request' node in Node-RED. The window has a title bar 'Edit http request node' and three buttons: 'Delete', 'Cancel', and 'Done'. Below the buttons is a 'Properties' section with several settings:

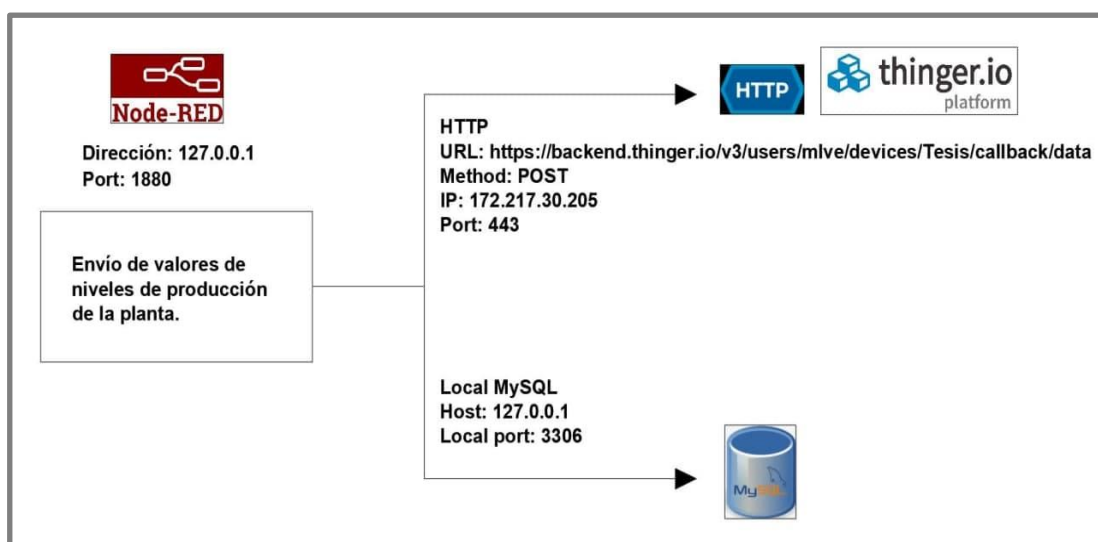
- Method:** A dropdown menu set to 'POST'.
- URL:** A text input field containing 'https://backend.thinger.io/v3/users/Caylos82/devic'.
- Enable secure (SSL/TLS) connection:** An unchecked checkbox.
- Use authentication:** A checked checkbox.
- Type:** A dropdown menu set to 'bearer authentication'.
- Token:** A text input field with ten dots representing a masked token.
- Enable connection keep-alive:** A checked checkbox.
- Use proxy:** An unchecked checkbox.
- Return:** A dropdown menu set to 'a UTF-8 string'.
- Name:** A text input field containing 'Thinger IO'.

Con la finalidad de visualizar los datos, se diseña un *dashboard* en la plataforma Thinger.io. A este *dashboard* se añaden los *widgets* necesarios, a los que se les asigna las variables que recibe el *databucket* de la nube desde Node-RED. En el caso de este panel de visualización, se presentan los totales de piezas fabricadas de cada máquina CNC y el total entre las dos de manera gráfica, con la finalidad de diferenciarla del *dashboard* de Node-RED. Más adelante, en la sección de resultados, se presenta la visualización del diseño de este *dashboard*.

En la figura 26 se presenta un diagrama explicativo que detalla la información relevante sobre el envío de datos desde Node-RED hacia las plataformas de almacenamiento de información, MySQL y Thinger.io. En éste se observa las direcciones IP, puerto y demás información.

Figura 26

Diagrama de envío de información desde Node-RED a las plataformas de almacenamiento de datos



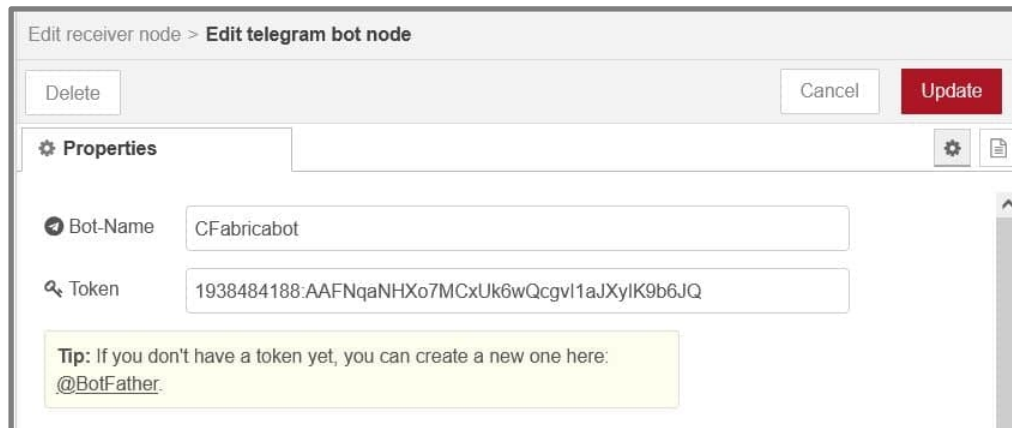
Chatbot. El servicio *chatbot* utiliza la plataforma Telegram para realizar consultas sobre la información de la planta. Para la creación del *bot* se utiliza *BotFather*, que es una herramienta propia de Telegram. Éste, únicamente con información de un nombre y un usuario, crea un *token* que sirve para comunicar el lenguaje de programación Python, bajo el que envía la información de Node-RED hacia Telegram. La conectividad entre Python y Telegram se efectúa a través de una API. En Python el paquete *telegram.ext* es requerido.

Este *chatbot* consulta la información sobre el funcionamiento de la planta al cliente implementado en Node-RED. Para esto se requiere un programa estructurado en un nodo que obtiene la información por medio de una lectura de variables, la reúne y la comunica al *bot*

utilizando el nodo *Telegram receiver*. La figura 27 presenta la configuración de este nodo, con un nombre y el *token* antes mencionado.

Figura 27

Configuración del nodo telegram receiver en Node-RED



Las funcionalidades que el servicio de *chatbot* implementa son:

- Nivel de producción de piezas “A”.
- Nivel de producción de piezas “B”.
- Nivel de producción total de la planta.
- Control de procesos de la planta (activación y desactivación).

Capítulo IV

Resultados

En este capítulo se presentan los resultados obtenidos luego de haber implementado las secciones detalladas en la metodología. Para esto, cada una de las etapas y capas de la pirámide CIM se expusieron a investigación experimental.

Análisis para el modelado y diseño de la planta

La plataforma Factory IO, utilizada para el diseño de la planta de producción, ostenta ventajas significativas con respecto a otros *softwares* de simulación de ambientes industriales, como PROModel, FreeCAD, ARENA simulation, etc. Cuenta con diversos escenarios previamente configurados para utilizarlos en distintas aplicaciones y se caracteriza por ser una plataforma amigable con el usuario. Lo más destacable de Factory IO es que permite realizar diversos proyectos orientados a la automatización industrial, sin la puesta en riesgo de cualquier equipo físico, pues se trata de un entorno simulado.

El diseño concebido de la planta de producción de piezas metalmecánicas y empaquetado para este trabajo se desarrolló desde cero. Es decir, no se utilizaron bloques o escenarios preconfigurados.

Dado que esta planta consiste principalmente en el transporte de las piezas hacia las máquinas, no existió mayor inconveniente. El único contratiempo presentado al momento de implementar el diseño en la plataforma es que ésta no permite redimensionar los elementos o modificar inclinaciones. Esto supone un leve problema ya que, al haber concebido el diseño en dos plantas, con el fin de asemejarla lo más posible a plantas industriales reales que manejan varios niveles, se requiere la utilización de resbaladillas. Y, al no haber sido posible redimensionarlas o reajustar sus ángulos de inclinación, se disponen dos, una a continuación de la otra, para que la pieza pueda llegar a su destino en la banda inferior, como se aprecia en la Figura 28.

Figura 28

Disposición de los elementos “resbaladilla” en el diseño de la planta en Factory IO



Para la emulación del mecanizado de las piezas se utilizan máquinas CNC, como ya se ha mencionado en la sección anterior. Sin embargo, las máquinas CNC disponibles en Factory IO no permiten modificar el trabajo realizado por éstas. Es decir, el diseño realizado en cada pieza no es modificable. Aun así, la utilización de estas máquinas permite cumplir el objetivo de la emulación del mecanizado de una pieza. En la Figura 29 se presenta el diseño por defecto que ejecuta la máquina CNC.

Análisis de las plataformas de interacción con la planta

Los sistemas de interacción con la planta implementados son tres y se listan a continuación.

- Sistema de monitoreo y control en planta: *dashboard* implementado en Node-RED.
- Sistema de monitoreo externo a la planta: *dashboard* implementado de la nube de almacenamiento Thingier.io.
- Sistema de consulta y control externo por *chat*: *chatbot* implementado en Telegram.

Figura 29

Diseño por defecto realizado por la máquina CNC en las piezas en Factory IO



En las siguientes subsecciones se muestran imágenes del diseño final de estos sistemas de interacción con la planta. Para el servicio de inspección o monitoreo, lo valioso es conocer el total de piezas fabricadas y el estado de los procesos, esto para verificar el funcionamiento correcto de la planta. Para el servicio de control, lo indispensable es permitir la activación y desactivación de los diferentes procesos, esto para verificar que responden de manera adecuada.

Los paneles de visualización de la información sobre la planta son dos, el *dashboard* de Node-RED que asume la virtualización del servicio de monitoreo y control; y el *dashboard* de Thinger.io. con la visualización de información en la nube.

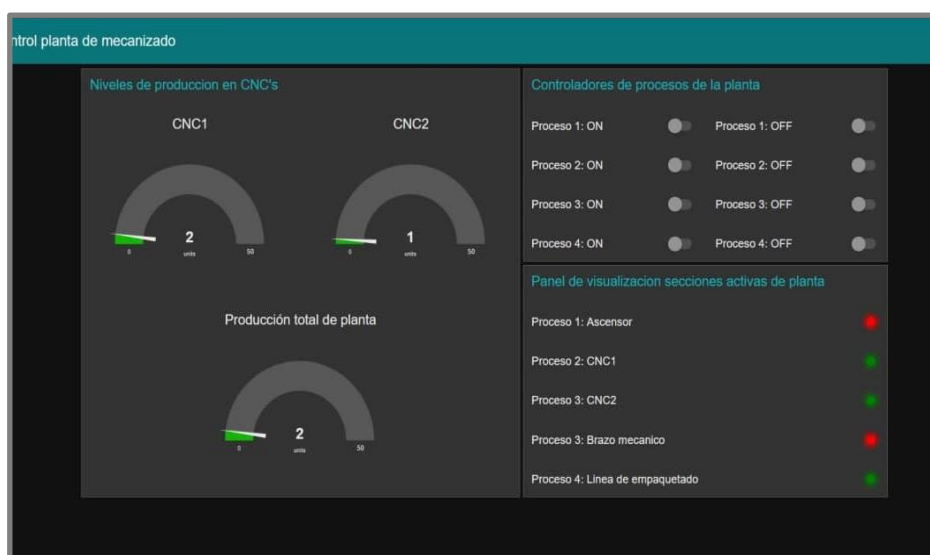
Sistema de monitoreo y control en planta

El sistema de monitoreo y control corresponde al *dashboard* implementado en la plataforma Node-RED. El diseño final de este panel de visualización se presenta en la figura 30 el panel de visualización cuenta con las secciones descritas en la sección 3.3.4.1, de manera que se

tenga por separado el total de fabricación de piezas de los indicadores de estado de los procesos y de su control. Este *dashboard* diseñado en Node-RED está destinado para el uso de personal operativo y administrativo de la planta.

Figura 30

Diseño final del dashboard diseñado en Node-RED



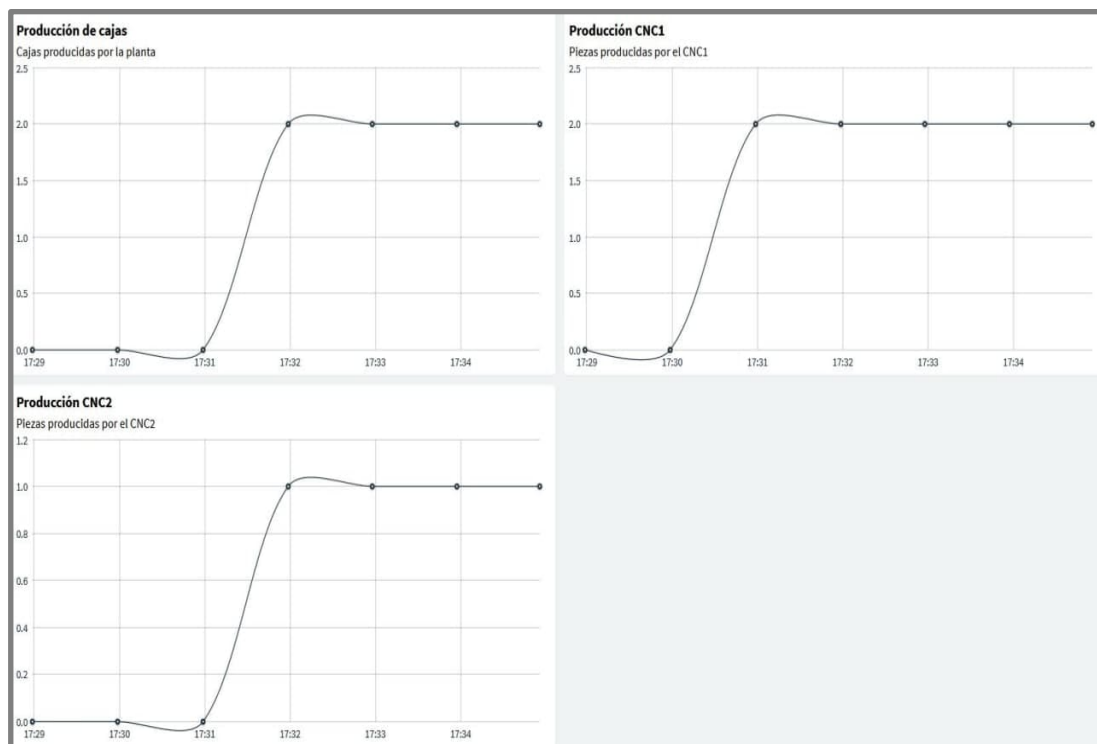
Sistema de monitoreo externo del proceso

El sistema de monitoreo externo a la planta corresponde al *dashboard* implementado en el servidor virtual de información de la plataforma Thinger.io. La figura 31 presenta el diseño del panel de visualización. Como se observa, se presentan los valores de fabricación de manera gráfica. Debido a que no cuenta con la opción de controlar la planta, esta plataforma se concibe como un sistema de monitoreo desde la nube.

A diferencia del *dashboard* de Node-RED, el *dashboard* de Thinger.io, puede ser consultado por cualquier miembro fuera de la planta.

Figura 31

Diseño final del dashboard diseñado en Thinger IO

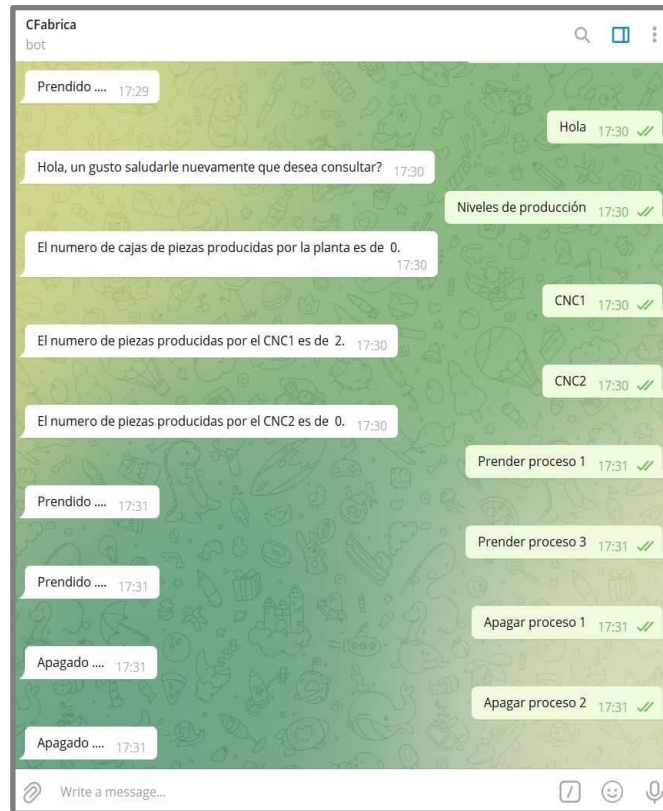


Sistema de consulta y control externo por chat

El sistema de consulta y control por *chat* consiste en el *chatbot* implementado en Telegram. La figura 32 presenta una captura de la conversación en la cual se hace la consulta de varios parámetros que se desean conocer sobre la planta y se dan instrucciones de activación y desactivación de procesos. El *chatbot*, destinado para monitoreo y control externo está disponible para consulta por cualquier personal fuera de planta, con la finalidad de conocer información en el instante de la consulta y controlar el estado de los procesos.

Figura 32

Consulta de información y control desde el chatbot implementado en Telegram



Análisis de comunicación y protocolos de los sistemas de interacción con la planta simulada

Los paquetes de información de la planta son enviados a través de Factory IO, hacia todos los sistemas que interactúan con el proceso, para analizar este tráfico de red en tiempo real se utiliza la herramienta de *software* gratuito Wireshark. Mediante el uso de esta herramienta y la explotación de sus capacidades de análisis de red, se filtran los paquetes de acuerdo con los protocolos de red que demandan ser analizados.

La finalidad de esta prueba es verificar el manejo de los protocolos utilizados por las soluciones IIoT, como son Node-RED, Thinger.io, el *chatbot* de consulta de Telegram y la base de

datos local MySQL. Se capturan los datos del tráfico de red mientras se ejecutan varios procesos de comunicación. Éstos se detallan a continuación:

1. El servidor OPC UA envía la información sobre la planta a la plataforma Node-RED por medio del protocolo OPC UA. La figura 33 captura el envío del paquete desde la dirección IP correspondiente al servidor hacia la dirección IP que aloja al cliente Node-RED. En este caso, 192.168.0.108 es la IP del servidor alojado en la Raspberry Pi4, como se mencionó en la sección 3.3.3.1, esta dirección es dinámica por lo que al momento de realizar esta prueba con Wireshark es diferente. Adicionalmente, la IP para Node-RED, 192.168.0.103, corresponde a la IP del ordenador ya que el cliente se aloja en éste. Se verifica el uso correcto del protocolo OPC UA.

Figura 33

Captura de envío de paquete del servidor a Node-RED en Wireshark

15451	62.168099	192.168.0.103	192.168.0.108	OpcUa	232 UA Secure Conversation Message: ReadRequest
15452	62.168231	192.168.0.103	192.168.0.108	OpcUa	231 UA Secure Conversation Message: ReadRequest
15453	62.168342	192.168.0.103	192.168.0.108	OpcUa	237 UA Secure Conversation Message: ReadRequest
15454	62.168450	192.168.0.103	192.168.0.108	OpcUa	238 UA Secure Conversation Message: ReadRequest
15455	62.168566	192.168.0.103	192.168.0.108	OpcUa	235 UA Secure Conversation Message: ReadRequest
15456	62.168667	192.168.0.103	192.168.0.108	OpcUa	234 UA Secure Conversation Message: ReadRequest
15457	62.168766	192.168.0.103	192.168.0.108	OpcUa	234 UA Secure Conversation Message: ReadRequest
15458	62.168865	192.168.0.103	192.168.0.108	OpcUa	233 UA Secure Conversation Message: ReadRequest
15459	62.168879	192.168.0.108	192.168.0.103	UDP	102 1740 → 1743 Len=60
15460	62.168965	192.168.0.103	192.168.0.108	OpcUa	236 UA Secure Conversation Message: ReadRequest
15461	62.169062	192.168.0.103	192.168.0.108	OpcUa	237 UA Secure Conversation Message: ReadRequest
15462	62.171295	192.168.0.108	192.168.0.103	OpcUa	133 UA Secure Conversation Message: ReadResponse

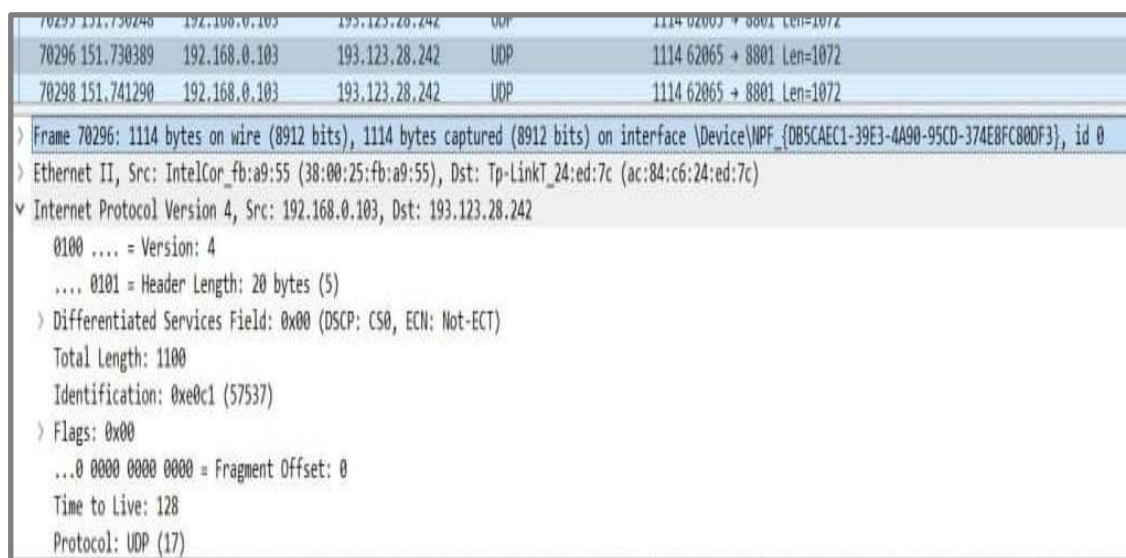
v Frame 1753: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface \Device\NPF_{DB5CAEC1-39E3-4A90-95CD-374E8FC80DF3}, id 0 > Interface id: 0 (\Device\NPF_{DB5CAEC1-39E3-4A90-95CD-374E8FC80DF3}) Encapsulation type: Ethernet (1) Arrival Time: Jan 6, 2022 14:59:50.161164000 Hora est. Pacifico, Sudamérica [Time shift for this packet: 0.000000000 seconds] Epoch Time: 1641499190.161164000 seconds [Time delta from previous captured frame: 0.013200000 seconds] [Time delta from previous displayed frame: 0.013200000 seconds] [Time since reference or first frame: 7.022475000 seconds] Frame Number: 1753 Frame Length: 55 bytes (440 bits)					
---	--	--	--	--	--

2. El cliente implementado en Node-RED envía la información sobre la planta que consulta el *chatbot* implementado en Telegram bajo el protocolo UDP. La figura 34 presenta la captura de

un paquete desde la dirección IP correspondiente al cliente Node-RED hacia la dirección IP que aloja al *chatbot*. En este caso, el *bot* de consulta se aloja en un servidor propio de Telegram, por lo tanto, corresponde a una red externa, con la IP 193.123.28.242. Se verifica el uso correcto del protocolo UDP.

Figura 34

Captura de envío de paquete de Node-RED al chatbot en Wireshark



- El cliente implementado en Node-RED envía la información sobre la planta para su almacenamiento en la base de datos local MySQL. Dado que la comunicación entre estas dos plataformas se realiza de manera local, es esta red la que se analiza mediante Wireshark. La figura 35 presenta la captura de un paquete de comunicación entre el cliente Node-RED y la base de datos MySQL, se verifica la instrucción de insertar los datos en la tabla "datafact" en el campo *Statement*. Adicionalmente, la figura 36 presenta la tabla creada en MySQL con los datos almacenados.

Figura 35

Captura de envío de paquete de Node-RED a MySQL en Wireshark

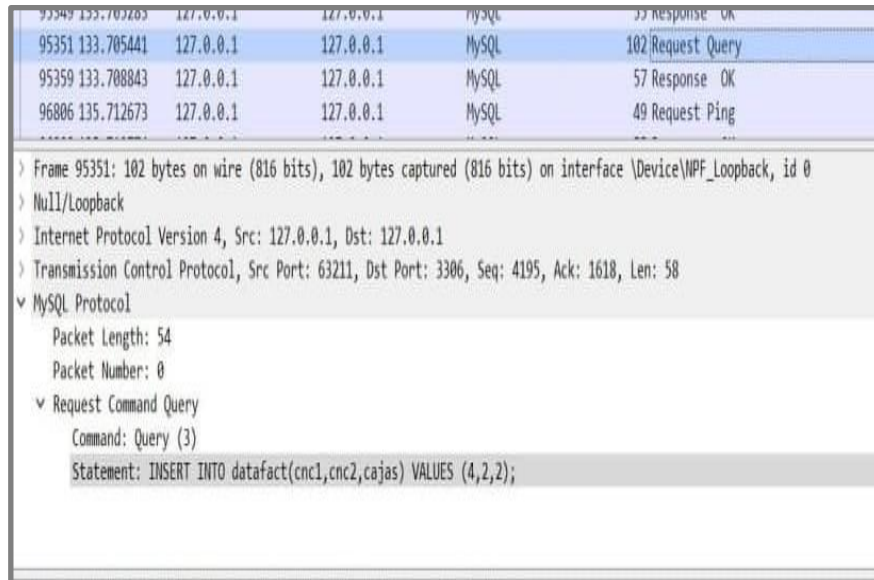


Figura 36

Tabla de la base de datos local en MySQL con datos almacenados

The screenshot shows a web-based MySQL database interface. The table 'datafact' is displayed with the following columns: p_id, Atime, cnc1, cnc2, and cajas. The table contains 15 rows of data.

	p_id	Atime	cnc1	cnc2	cajas
<input type="checkbox"/>	1840	2022-01-06 16:41:58	2	0	0
<input type="checkbox"/>	1841	2022-01-06 16:42:00	2	0	0
<input type="checkbox"/>	1842	2022-01-06 16:42:02	2	0	0
<input type="checkbox"/>	1843	2022-01-06 16:42:04	2	0	0
<input type="checkbox"/>	1844	2022-01-06 16:42:06	2	0	0
<input type="checkbox"/>	1845	2022-01-06 16:42:08	2	0	0
<input type="checkbox"/>	1846	2022-01-06 16:42:10	2	0	0
<input type="checkbox"/>	1847	2022-01-06 16:42:12	2	1	0
<input type="checkbox"/>	1848	2022-01-06 16:42:14	2	1	0
<input type="checkbox"/>	1849	2022-01-06 16:42:16	2	1	0
<input type="checkbox"/>	1850	2022-01-06 16:42:18	2	1	0
<input type="checkbox"/>	1851	2022-01-06 16:42:20	2	1	0
<input type="checkbox"/>	1852	2022-01-06 16:42:22	3	1	0
<input type="checkbox"/>	1853	2022-01-06 16:42:24	3	1	0
<input type="checkbox"/>	1854	2022-01-06 16:42:26	3	1	1
<input type="checkbox"/>	1855	2022-01-06 16:42:28	3	1	1
<input type="checkbox"/>	1856	2022-01-06 16:42:30	3	1	1
<input type="checkbox"/>	1857	2022-01-06 16:42:32	3	1	1
<input type="checkbox"/>	1858	2022-01-06 16:42:34	3	1	1

4. El cliente implementado en Node-RED envía la información sobre la planta para su almacenamiento en la nube de Thinger.io por intermedio del protocolo HTTP. La figura 37 presenta la captura de un paquete desde la dirección IP correspondiente al cliente Node-RED hacia la dirección IP que aloja a la nube de almacenamiento. En este caso, en la nube se aloja en un servidor propio de Thinger.io; por lo tanto, corresponde a una red externa, con la IP 34.107.221.82. Se verifica el uso correcto del protocolo HTTP. Adicionalmente, la figura 38 presenta el contenedor de datos en Thinger.io con los datos almacenados.

Figura 37

Captura de envío de paquete de Node-RED a Thinger.io en Wireshark

No.	Time	Source	Destination	Protocol	Length	Info
14337	21.850127	192.168.0.103	34.107.221.82	HTTP	378	GET /success.txt?ipv4 HTTP/1.1
14363	21.880006	34.107.221.82	192.168.0.103	HTTP	274	HTTP/1.1 200 OK (text/plain)

> Frame 14337: 378 bytes on wire (3024 bits), 378 bytes captured (3024 bits) on interface \Device\NPF_{DB5CAEC1-39E3-4A90-95CD-374E8FC80DF3}, id 0 > Ethernet II, Src: IntelCor_fb:a9:55 (38:00:25:fb:a9:55), Dst: Tp-LinkT_24:ed:7c (ac:84:c6:24:ed:7c) > Internet Protocol Version 4, Src: 192.168.0.103, Dst: 34.107.221.82 > Transmission Control Protocol, Src Port: 63926, Dst Port: 80, Seq: 325, Ack: 221, Len: 324 > Hypertext Transfer Protocol > GET /success.txt?ipv4 HTTP/1.1\r\n Host: detectportal.firefox.com\r\n User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/20100101 Firefox/95.0\r\n Accept: */*\r\n Accept-Language: es-ES;q=0.8,en-US;q=0.5,en;q=0.3\r\n Accept-Encoding: gzip, deflate\r\n Connection: keep-alive\r\n Pragma: no-cache\r\n Cache-Control: no-cache\r\n						
--	--	--	--	--	--	--

Figura 38

Contenedor de datos en Thinger.io con datos almacenados

Date	Cajas	Cnc1	Cnc2
2022-01-06T15:33:27.098Z	18	12	12
2022-01-06T15:32:27.550Z	18	12	12
2022-01-06T15:31:27.158Z	18	12	12
2022-01-06T15:30:28.389Z	18	12	12
2022-01-06T15:29:29.227Z	18	12	12
2022-01-06T15:28:27.987Z	18	12	12
2022-01-06T15:27:28.742Z	18	12	12
2022-01-06T15:13:23.310Z	14	12	8
2022-01-06T15:12:23.362Z	13	12	7
2022-01-06T15:11:23.105Z	12	12	6
2022-01-06T15:10:24.805Z	12	12	5
2022-01-06T15:09:25.095Z	10	12	4
2022-01-06T15:08:26.182Z	9	12	3
2022-01-06T15:07:26.762Z	9	11	3
2022-01-06T15:06:28.079Z	9	10	1
2022-01-06T15:05:27.749Z	4	8	1

Análisis del envío de información a la base de datos local y a la nube de almacenamiento de información

Como se explicó en las secciones 3.3.4.2 y 3.3.4.3, el envío de datos desde el cliente implementado en Node-RED hacia la base de datos local y la nube de almacenamiento, respectivamente, se realiza cada 2 segundos.

Se realiza la captura de los datos en la nube y en la base de datos local mientras la simulación de la planta de producción se ejecuta durante 1 hora. Se encuentra que la base de datos local recibe 1802 paquetes y la nube de información recibe sólo 60, con base en estos resultados, se aprecia que el almacenamiento en Thinger.io es muy reducido en contraste al almacenamiento en

MySQL. Esto se debe a que la nube de información Thinger.io actualiza c los datos que Node-RED envía cada minuto. Lo cual se puede comprobar en la primera columna de la data *bucket* de la plataforma ya que un nuevo dato es almacenado exactamente cada 60 segundos, esto se aprecia en la figura 38 el atributo de Thinger.io de almacenar información cada minuto es predeterminada e inalterable. La solución a este inconveniente estaría directamente en el salto de plataforma.

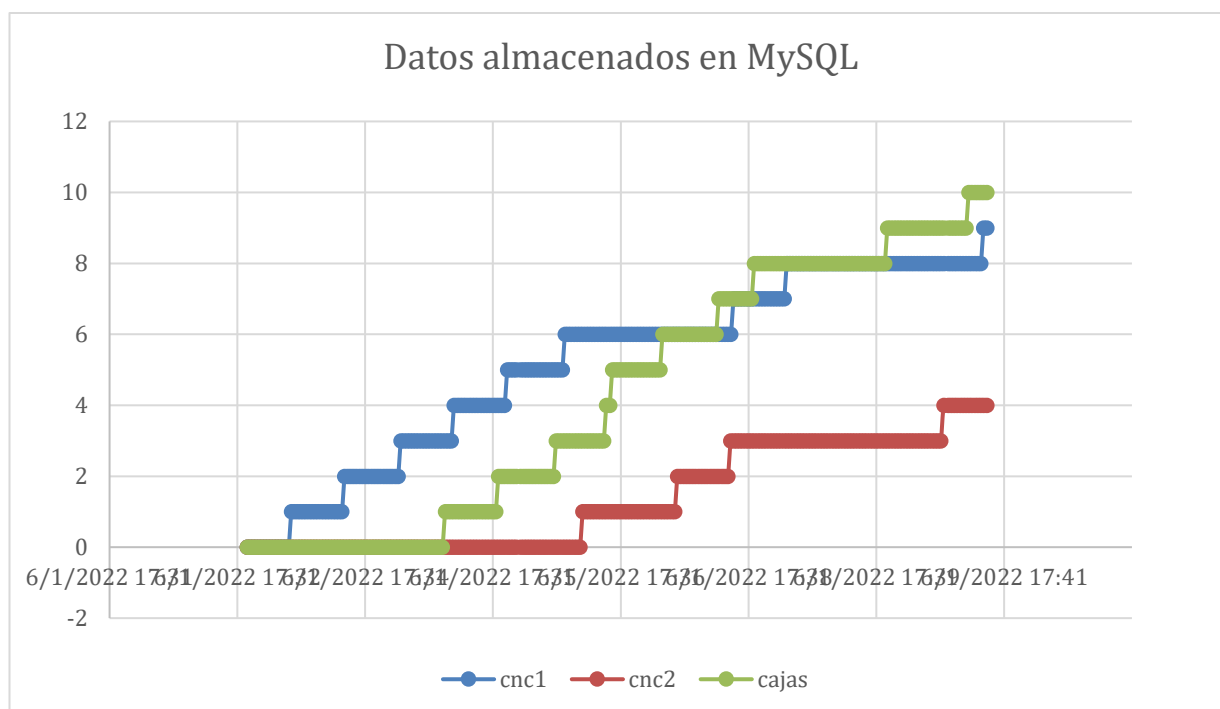
Análisis de la información almacenada en la base de datos local y la nube de almacenamiento de información

Después del conocimiento de esta información, se realiza una gráfica de aumento de la producción de piezas mecanizadas en cada máquina CNC y total de cajas con la información almacenada en MySQL y en Thinger.io, con la finalidad de compararlas de acuerdo con los datos almacenados en las dos plataformas. De los datos capturados durante la prueba de una hora, se toma la información de los primeros 10 minutos con la finalidad de que las gráficas no se vean extensas y se aprecien de mejor manera el crecimiento de totales de piezas y cajas.

La figura 39 presenta la gráfica de acuerdo con los datos almacenados en MySQL. Se aprecia el marcado crecimiento de los totales de piezas y cajas, incluso siendo posible observar claramente los momentos en los que los valores se han mantenido constantes, lo que significa que las piezas están siendo transportadas. La apreciación de estos momentos es clave para analizar el caso de haber un problema en el transporte de las piezas que pudiera traducirse en fallos de las bandas u otros elementos.

Figura 39

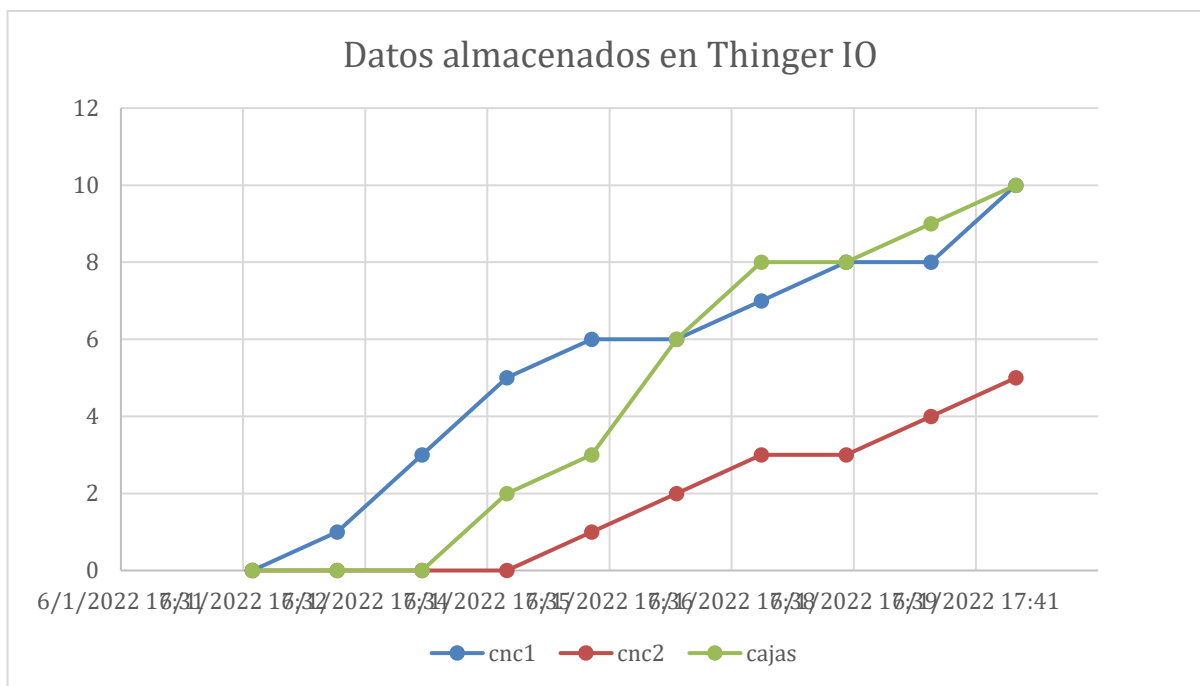
Gráfica de crecimiento de piezas mecanizadas y cajas de acuerdo con los datos almacenados en MySQL



Por otro lado, en la figura 40 se presenta la gráfica de acuerdo con los datos almacenados en Thinger.io. Si se contrasta esta gráfica con la de MySQL, claramente se aprecia que la cantidad de información es reducida. Se tiene una idea general del crecimiento de los totales de piezas y cajas; sin embargo, se pierde información, lo que no permite realizar el mismo análisis a profundidad de la gráfica anterior.

Figura 40

Gráfica de crecimiento de piezas mecanizadas y cajas de acuerdo con los datos almacenados en Thinger.IO



Problemática y solución para envío de información a la nube

El envío de datos sobre el comportamiento de la planta hacia la nube de almacenamiento fue un desafío del desarrollo de este trabajo. En el mundo de la industria se emplean mecanismos destinados para esta tarea en específico, lo que facilita el manejo de datos. El módulo implementado en este caso para el envío de información es la Raspberry Pi4 que hace las veces de PLC, su función es facilitar la comunicación entre las distintas aplicaciones y servicios implementados en este trabajo, siendo uno de ellos la nube de almacenamiento de información.

Para la solución del desafío planteado, teniendo en cuenta que el ambiente de prueba es totalmente simulado, se opta por el uso de protocolos diferentes protocolos de comunicación, con la intención de facilitar el envío de información además de mantener la heterogeneidad. En este

contexto, la implementación más importante corresponde al *middleware* (OPC UA), éste hace las veces de columna vertebral de la pirámide de automatización empleada para este proyecto. En otras palabras, permite la interconexión entre cada uno de los mecanismos de comunicación que emplea cada nivel CIM. El servidor OPC UA permite que se establezca la comunicación desde la planta industrial hacia los sistemas de supervisión, y posteriormente a la nube. De esta manera, es posible monitorear cada uno de los procesos involucrados en el funcionamiento de la planta desde esta nube.

Node-RED cumple también parte fundamental del desarrollo de este proyecto. Mantiene la conexión entre los entornos de desarrollo IoT con la base de datos local implementada en MySQL. Como se presentó anteriormente, en las figuras 36 y 38 se aprecia los resultados del almacenamiento tanto en la base de datos local como en la nube, respectivamente.

Análisis del tiempo de respuesta en la comunicación con el PLC

Analizar los tiempos de respuesta manejados por el PLC implementado en la Raspberry Pi4 es un paso importante para interpretar la eficiencia del sistema desarrollado. El hecho de que el sistema de supervisión implementado en Node-RED, junto con su panel de visualización, se ejecute al mismo tiempo que el servidor OPC UA implementado en la Raspberry Pi4, implica que los tiempos de respuesta en comunicación con el PLC sean cortos.

En la figura 33 presentada en la sección 4.2.4 se observa una captura del tráfico de red entre el servidor implementado en el PLC y el sistema de supervisión implementado en Node-RED. En la tabla 3 se presenta los resultados obtenidos sobre los tiempos de respuesta al analizar diferentes tipos de variables correspondientes al proceso de automatización de la planta. Se aprecia que los tiempos cumplen métricas menores a 1 segundo para todas las variables analizadas. Cabe recalcar que estos tiempos dependen en gran medida del ancho de banda que maneje la implementación del sistema de red diseñado.

Tabla 3

Resultados de tiempo de respuesta del PLC.

Tipo de Variable	Descripción	Dirección servidor OPC	Tiempo Wireshark / cronometrado
Bool	Inicio proceso 1	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.STARTP1	17ms / <1s
Bool	Paro proceso 1	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales.STOPP1	21ms / <1s
Word	Contador CNC1	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales. CONTADORCNC1	8ms / <1s
Word	Contador Cajas	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales. CONTADORCAJAS	9ms / <1s
Bool	Accionamiento de motores	ns=4;s= var CODESYS Control for Raspberry Pi SL.Application.Variables_Globales. MOTOR1P1	19ms / <1s

Comprobación de la hipótesis

Al finalizar las pruebas realizadas con el fin de evaluar el trabajo desarrollado se pretende evidenciar que la implementación de los diferentes servicios IIoT para un sistema de manufactura de piezas facilita la comunicación de información relevante sobre el funcionamiento de la planta

mediante las distintas interfaces de comunicación heterogéneas. Esto se describe más a fondo y con base en las variables de la hipótesis a continuación.

Facilidad de comunicación de los sistemas de automatización industrial con interfaces heterogéneas

La implementación de este trabajo considera la propuesta de la utilización de *software* libre con la finalidad de la facilidad de comunicación multiplataforma, ya que prestan funcionalidades que permiten manejar la heterogeneidad de los sistemas a favor de los diseñadores y desarrolladores de proyectos.

En este sentido, se consigue la implementación de los diversos servicios de IoT industrial para el control de la planta mediante la utilización de *software* libre y multiplataforma, como Node-Red, Thinger.io, Telegram, entre otros. Además, esto permite explotar sus capacidades gracias a la parte central de este trabajo, el servidor OPC UA. De esta manera, el trabajo constituye un conjunto de protocolos y *software* que en su totalidad se comunican entre sí para comunicar a los operarios o usuarios de planta sobre su comportamiento y de sus procesos de producción.

Por otra parte, es indispensable analizar, como parte de la hipótesis de trabajo propuesta, los tiempos de respuesta de la planta. En la sección 4.4, específicamente en la Tabla 3, se presenta los resultados de realizar una prueba en la que se evalúan los tiempos de respuesta de la planta.

Los valores presentados en la tabla mencionada corresponden a un valor medio de haber realizado 3 pruebas, tanto para el tiempo de respuesta capturado por la herramienta “*round trip time*” del analizador de red Wireshark, como para el tiempo de respuesta cronometrado. El escenario de prueba consiste en comandar el estado de activación y desactivación de procesos de la planta en el servidor OPC-UA, alojado en el PLC, desde el servicio de *chatbot* implementado en Telegram, para cada una de las 3 pruebas.

Cabe recalcar que estas pruebas han sido realizadas sin saturación en la red. Esto permite analizar que los resultados son tiempos reducidos para el caso de control remoto a partir de alguno de los servicios implementados. Sin embargo, es correcto analizar que, para el control directo en planta; donde se cuenta con conexiones físicas directas, los tiempos de respuesta claramente se verían aún más reducidos. Esto a nivel de la plataforma de simulación Factory IO no es demostrable, aun así, es claro el razonamiento.

Implementación de IoT industrial

El principal beneficio a partir de la virtualización de los servicios de IoT industrial, que constituye una red de comunicación de datos sobre el funcionamiento de la planta simulada, es que se lleva la etapa de supervisión a otro nivel. Todo esto para optimizar el control de las operaciones de planta; además de obtener ventajas en cuanto a tiempos y costos de implementación.

Con respecto a los costos de implementación, está claro por la ejecución de este trabajo, que se concibe a partir de una simulación, que los costos son reducidos. Sin duda, al compararlos con costos de implementación de una planta real, los costos son notablemente menores, pues no se realizan gastos en infraestructura de planta.

Tabla 4

Costos de licencia por software.

<i>Software</i>	<i>Precio por licencia</i>
Factory IO	\$30,00 / mensual
Servidor OPC-UA Codesys	\$56,63
TOTAL	\$86,63

Este trabajo ha sido desarrollado mayormente utilizando *software* gratuito, sin embargo, en la Tabla 4 se presenta los costos que han sido cubiertos para aquellos programas en los que se ha requerido la compra de una licencia.

Si se continúa con la propuesta de que este trabajo de simulación se realice previo a que las compañías decidan llevar a cabo la implementación física de su planta industrial y todos los sistemas que ésta conlleva, se destaca su reducido costo de implementación. Costo que permitirá a la compañía un análisis y evaluación previa de los equipos, sensores, actuadores y demás materiales necesarios para levantar su planta industrial física. Además, esta implementación virtual en simulación les permitiría en un futuro analizar cambios que se deseen realizar a la planta ya implementada, con el fin de conocer las consecuencias que se presentarían. Cabe recalcar que toda la implementación a nivel de control, supervisión y virtualización de servicios es fácilmente aplicable a una planta física real.

Otro costo que también se reduce en gran medida es el de mano de obra, pues para esta implementación se requiere únicamente de un ingeniero que maneje el *software* utilizado en este trabajo, que para mayor facilidad son amigables al usuario y sencillos de instalar, configurar y programar.

Lo mismo ocurre en términos de tiempos de implementación. La facilidad de uso, recién mencionada, del *software* utilizado en este trabajo permite que las implementaciones y conexiones entre sistemas de comunicación requieran poco tiempo. Lo que para esta implementación supone una extensión de semanas, en una planta física real podría llevar varios meses. Esto apoya aún más la propuesta de realizar este trabajo de simulación previo al levantamiento de la planta física, pues el tiempo que llevaría la adquisición de equipos industriales y distintos dispositivos requeridos se aprovecharía con el análisis de resultados mediante la simulación de planta.

Capítulo V

Conclusiones y recomendaciones

El presente capítulo contiene las conclusiones del trabajo de titulación a partir del análisis de los resultados, teniendo a consideración los objetivos planteados en el capítulo inicial.

Adicionalmente, se presentan las recomendaciones y la identificación de trabajos futuros.

Conclusiones

- La evolución del IoT hacia un nivel empresarial/industrial permite explotar sectores productivos con el objetivo de conseguir mejores resultados en etapas de manufactura y administración. Concebir la automatización de los procesos de manufactura y fabricación implica una comunicación entre distintos equipos, dispositivos, protocolos y servicios diseñados. Esta comunicación no ocurre únicamente a nivel de planta de fabricación sino también a nivel empresarial, pues se requieren sistemas de supervisión y monitoreo que se dispongan a ser utilizados por personal administrativo empresarial. La propuesta de simulación de este trabajo permite evidenciar la comunicación en los niveles mencionados, esto gracias a la implementación de un servidor OPC UA que sirve de base para la unión de los protocolos y servicios IIoT contemplados en el desarrollo de este trabajo.
- El desarrollo de la simulación para la etapa de manufactura y fabricación se concibe en la plataforma Factory IO. El uso de esta herramienta permite un acercamiento real al funcionamiento de plantas existentes en el mercado. Por lo tanto, se adquieren los conocimientos necesarios para manejar equipos y protocolos de gama industrial de gran capacidad. Su simplicidad de uso y su compatibilidad con demás *software* y *hardware* lo convierte en la opción ideal para un despliegue de una planta de manufactura en un entorno simulado.

- La variedad de protocolos que se manejan en este trabajo, especialmente el OPC UA, resuelve el problema de heterogeneidad en ambientes industriales. Lo destacable de la integración de estos protocolos es que permiten que este sistema, a pesar de ser simulado, se puede aplicar a una planta distinta, lo que le da escalabilidad a este proyecto.
- La planta de manufactura propuesta para simulación corresponde a la fabricación de piezas metalmecánicas y embalaje. El diseño de esta planta implica el uso de elementos reales de la industria como bandas de transporte, máquinas CNC, sensores, actuadores, tableros de control y demás. El nivel de realidad en el que se trabaja gracias a la plataforma Factory IO, la vuelve una de las mejores alternativas para simulación y un primer acercamiento al mundo industrial.
- La base en la arquitectura CIM permite un desarrollo lógico y consecuente de este trabajo. Al cumplir con la etapa de simulación, que corresponde a la capa de campo, se reúne la información sobre el funcionamiento de ésta, que debe disponerse para conocimiento de personal operario y administrativo de la planta. Para esto se concibe la capa de supervisión, donde se lleva a cabo el monitoreo del funcionamiento de los procesos que ejecuta la planta. Con la finalidad de una mejor apreciación de la información se implementan paneles de visualización que sirven de interfaz entre el humano y la máquina. El *dashboard* para supervisión y control se diseña en la herramienta Node-RED. La ventaja de esta herramienta es que aplica programación apoyada en nodos y flujos, siendo amigable con el desarrollador, pues no se precisa de conocimientos profundos de programación para desarrollar una aplicación.
- Para este trabajo se implementa gran variedad de *software*, entre ellos Codesys. Esta herramienta permite levantar el servicio base y fundamental de este proyecto, el servidor OPC UA. Para esto, en *hardware* se hace uso del dispositivo Raspberry Pi4, que hace las veces de controlador PLC. Es en éste en el que se ejecuta el servidor mencionado. La utilización de OPC UA permite una comunicación entre todos los niveles de la arquitectura propuesta. La

propuesta de un servidor OPC UA como base fundamental que permite la comunicación de tecnologías heterogéneas, se concibe como un *middleware* que habilita implementar las soluciones IIoT.

- Dentro de las soluciones IIoT de capa superior se implementa el panel de visualización diseñado en Node-RED, que ya se mencionó, el panel de visualización de la plataforma de almacenamiento de información Thinger.io y el bot de consulta implementado en Telegram. Los *dashboards* y el *chatbot* se conciben con enfoques distintos. El *dashboard* de Node-RED y el *chatbot* de Telegram sirven para monitoreo y control, mientras que el *dashboard* de Thinger.io únicamente para monitoreo. La utilización de la herramienta Thinger.io es de fácil accesibilidad al requerir únicamente de un navegador web. Sin embargo, se halló que las bondades de recepción y dispersión de la información son limitadas ya que la actualización de los datos es cada 60 segundos.
- El manejo y gestión de la información del *dashboard* de la nube de almacenamiento desde una aplicación móvil no se realiza debido a que la plataforma Thinger.io no dispone del *kernel* requerido para su despliegue. Por este motivo se utilizan otras herramientas para gestión de los datos como Node-RED, la base de datos local implementada en MySQL y el *bot* de consulta. Sin embargo, para llevar a cabo el monitoreo, tanto en Thinger.io como en Node-RED, únicamente se requiere de una conexión a internet y un navegador web.

Recomendaciones

- La simulación implementada de la planta de fabricación de piezas metalmecánicas y embalaje requiere ejecutar varios programas al mismo tiempo; por tanto, se recomienda contar con una conexión estable a internet y un ordenador que cumpla con las exigencias de funcionamiento del proyecto.

- Adicionalmente, el funcionamiento adecuado de este trabajo depende de que la conexión a internet sea de banda ancha, debido a que la red maneja una consistente cantidad de datos provenientes de las comunicaciones entre varios dispositivos y servicios.
- Con la finalidad de mejorar los resultados de almacenamiento obtenidos con Thinger.io, es recomendable investigar otras plataformas de almacenamiento en nube.

Trabajos futuros

- Implementación a nivel de *hardware* para control físico de un modelo similar a la planta diseñada en el presente trabajo. La variedad de tecnologías heterogéneas permite llevar a la práctica su integración, teniendo las facilidades de acceso a *hardware* con características similares a las simuladas en la planta.
- Seguridad para el envío de información, tanto para el servidor local como para el almacenamiento de datos que se aloja en las nubes de información.
- Integración de nuevos equipos dentro del ambiente simulado, para generar un agente de control de fallos dentro de la planta, de esta manera se puede escalar un nivel más dentro de la pirámide de automatización CIM, llegando así a tener un nivel de supervisión y control de fallos virtualizado.

Bibliografía

- 21, A. (marzo de 2017). *Automatización Industrial: Qué es y cómo funciona*. Universidad de Salamanca, pg. 45-49.
- Alcuri, G. (2013). Control de procesos y caracterización de situaciones industriales por métodos globalizantes. *ALCTRA Recherche & Développement*, 43-58.
- Brunete, A., Pablo, S., & Rebeca, H. (28 de 07 de 2020). *Industria 4.0 redes industriales en manufactura*, pg. 85-99.
- Corrales, L. (2007). *Sistemas Scada*. Quito: EPN.
- European Commission. (2018). *Introducción a los Sistemas de Control Industrial y SCADA*. Madrid: Scada Lab.
- Exsol. (18 de 01 de 2017). *¿Qué es la automatización industrial?* Universidad Politécnica de Lima.
- García, G. (2010). Predicción de fallos en redes IP empleando redes neuronales artificiales. *Polibits*, 3-6.
- GSL Industrias. (06 de 2015). *¿Qué es un PLC y cómo funciona?* Obtenido de <https://www.industriasgsl.com/blog/post/que-es-un-plc-y-como-funciona>
- Hurtado, J. (junio de 2016). *Introducción a las Redes de Comunicación Industrial*. Obtenido de http://www.infoplcn.net/files/documentacion/comunicaciones/infoPLC_net_introduccion-a-las-redes-de-comunicacion-industrial.pdf
- Lemos, J. D. (2006). OPC como alternativa a las tecnologías propietarias de comunicación industrial. *Avances en Sistemas e Informática*, 7-12.
- Lopez de Ayala, G. V. (2017). Desarrollo de Ejercicios Didácticos para Autómatas Programables usando Codesys. *Universidad de Sevilla*, 1-5.
- Lopez, M. (2019). *Industria 4.0 para la monitorización de un proceso industrial. (Tesis de Maestría)*. Universidad Técnica de Ambato, Ambato.

- Nuñez, A. (2019). Diseño de una plataforma web de software libre para la creación de HMIS para los PLCs7-1200 y su incidencia en la reducción de costos de monitoreo. (*Tesis de Maestría*). Universidad Técnica de Ambato, Ambato.
- OPC Foundation. (2019). *OPC Unified Architecture*. Scottsdale: OPC Foundation.
- Pacheco, F. (2006). *Desarrollo de OPC*. Cartagena: Universidad Tecnológica de Bolívar.
- Padilla, J. (marzo de 2019). *Redes Industriales*. Obtenido de http://jpadilla.docentes.upbbga.edu.co/redes_industriales/1-Introduccion.pdf
- Plant Web University. (junio de 2012). *Introduccion a Hart*. Sistemas computacionales en la industria, pg. 66-72.
- Ripipsa. (04 de 2019). *Automatización Industrial ¿Que es la automatización industrial?* Obtenido de <https://ripipsacobots.com/automatizacion-industrial/>
- Sánchez, J. M. (2018). *Diseño de un control distribuido usando Factory IO y Codesys V3*. Sevilla: Universidad de Sevilla.
- Santis, M. (octubre de 2019). *Estudio del protocolo Fieldbus*. Fieldus Foundation, pg. 12-16.
- SUPRESS. (2017). *Introducción al software de programación Codesys*. Madrid: Universidad de León.
- Tabares, H. (2007). Automatizacion de una planta de embotellado, aplicando control borroso para detección de alarmas. *Scientia Et Technica*, 285-290.
- Universidad de Valencia. (septiembre de 2015). *Complementos del protocolo Profibus*, pg. 45-53.

Anexos