UNIVERSIDAD DE CUENCA

Facultad de Ingeniería

Carrera de
Electrónica y Telecomunicaciones

Diseño y simulación de una línea de producción empleando protocolos de comunicación industrial (IIoT) para el control y monitoreo de sistemas de automatización heterogéneos

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones.

Autores:

Pablo Adrián Barriga León C.I: 010672952-8
pbarriga6@gmail.com

Marcos Lenín Villarreal Esquivel C.I: 030259745-5
vmarcoslenin@gmail.com

Director:

Ing. Luis Ismael Minchala Ávila, PhD C.I: 030145348-6

Cuenca - Ecuador
28 de julio de 2021

Resumen

El presente trabajo de titulación desarrolla e implementa una metodología que permite la integración e interconexión de procesos heterogéneos empleando un enfoque de comunicación de internet industrial de las cosas (IIoT, por sus siglas en inglés), en un ambiente de simulación bajo el concepto de Industria 4.0. Los protocolos que se usan, principalmente, son MQQT y HTTP y operan bajo la arquitectura cliente-servidor, además de filtrar la información mediante eventos *Publisher/Suscriber* y *Request/Response*, respectivamente. Se diseña un proceso industrial que emula a través del software Factory I/O una línea de producción de zumo de naranja, controlada mediante un controlador lógico programable (PLC). Un aspecto importante es el diseño de un controlador PID para la automatización del proceso de forma eficiente.

El proceso de comunicación es implementado con ayuda del servidor OPC, el mismo que hace referencia a un protocolo de comunicación en el campo de control y supervisión de procesos industriales. OPC actúa como *middleware* y permite la conexión entre cada uno de los niveles de la pirámide de automatización. De esta manera, se habilita la opción de envío de información de la fábrica hacia la nube, con el objetivo de monitorizar el estado actual de cada una de las variables que intervienen en el proceso de producción. Para el manejo de información de la planta se empleó el software Node-RED, de esta forma, se logra establecer un enlace con la base de datos de cada uno de los entornos de desarrollo IIoT, así mismo, almacenando los datos de la fábrica en tiempo real.

Con el objetivo de realizar un análisis comparativo en función a la eficiencia del manejo de información se empleó la herramienta Wireshark, obteniendo de esta manera distintos resultados en función al porcentaje de paquetes perdidos para cada protocolo de comunicación. Para el desarrollo del proyecto se utilizó una plataforma del tipo *open source* conocida como Thinger.io y una de modelo de pago que es AWS. De igual manera, se emplea un bloque de MySQL para capturar paquetes de forma local. Con propósitos de visualización, se implementa un sistema HMI en LabVIEW y se crea su respectivo gemelo digital en el entorno de Node-RED con el propósito de virtualizar los servicios de la capa MES y brindar acceso a diferentes operarios de la planta.

Como resultado, se programa un agente detector de fallas en el sistema multitanque y se enlaza su funcionamiento a un bloque que emite una notificación a través de un correo electrónico y la interacción entre un *bot* y operarios de la fábrica para que tomen las medidas correctivas.

Palabras clave : Automatización. IIoT. HTTP. Industria 4.0. Middleware. MQTT. Node-RED. PLC. Protocolo de comunicación. PID

Abstract

This paper towards obtaining an academic degree, develops and implements a methodology allowing the integration and interconnection of heterogeneous processes using an industrial internet of things (IIoT) communication approach in a simulation environment under the concept of Industry 4.0. The protocols used are ,mainly, MQQT and HTTP and operate under the client-server architecture, in addition to filtering information through ,respectively , Publisher / Subscriber and Request / Response events.

An industrial-like process is designed through Factory I / O software . An orange juice production line, controlled by a programmable logic controller (PLC). An important aspect is the design of a PID controller to automate the process efficiently.

The communication process is implemented with the aid of the OPC server, which refers to a communication protocol in the field of control and supervision of industrial processes. OPC interacts as middleware and allows the connection between each of the levels of the automation pyramid. In such a way, the option of sending information from the factory to the cloud is enabled, in order to monitor the current state of each of the variables involved in the production process. The Node-RED software was used to manage the plant information, in this way, it is possible to establish a link with the database of each of the IIoT development environments, likewise, storing the factory data in real time.

In order to carry out a comparative análisis, based on the efficiency of information management, the Wireshark tool was used, thus obtaining different results based on the percentage of lost packets for each communication protocol. For the development of the project, an open source platform known as Thinger.io and a payment model AWS, was utilized. Similarly, a MySQL block is used to capture packets locally. For visualization purposes, an HMI system is implemented in LabVIEW and its respective digital-twin is created in the Node-RED environment with the purpose of: virtualizing the services of the MES layer and providing access to different plant operators.

As a result, a fault detection agent is programmed in the multitank system and its operation is linked to a block that issues a notification via email and the interaction between a bot and factory operators to take corrective measures.

Keywords : Automation. IIoT. HTTP. Industry 4.0. Middleware. MQTT. Node-RED. PLC. Communication protocol. PID

Índice general

Índice de figuras

Índice de tablas

Cláusula de Propiedad Intelectual

Yo, Pablo Adrián Barriga León, autor de la tesis "Diseño y simulación de una línea de producción empleando protocolos de comunicación industrial (IIoT) para el control y monitoreo de sistemas de automatización heterogéneos", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 27 de julio de 2021

Pablo Adrián Barriga León
010672952-8

Cláusula de Propiedad Intelectual

Yo, Marcos Lenín Villarreal Esquivel, autor de la tesis "Diseño y simulación de una línea de producción empleando protocolos de comunicación industrial (IIoT) para el control y monitoreo de sistemas de automatización heterogéneos", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 27 de julio de 2021

Marcos Lenín Villarreal Esquivel
030259745-5

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Pablo Adrián Barriga León en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Diseño y simulación de una línea de producción empleando protocolos de comunicación industrial (IIoT) para el control y monitoreo de sistemas de automatización heterogéneos", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 27 de julio de 2021

Pablo Adrián Barriga León
010672952-8

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

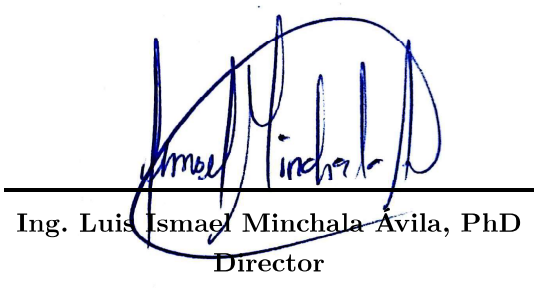
Yo, Marcos Lenín Villarreal Esquivel en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Diseño y simulación de una línea de producción empleando protocolos de comunicación industrial (IIoT) para el control y monitoreo de sistemas de automatización heterogéneos", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 27 de julio de 2021

Marcos Lenín Villarreal Esquivel
030259745-5

Certifico

Que el presente proyecto de tesis: Diseño y simulación de una línea de producción empleando protocolos de comunicación industrial (IIoT) para el control y monitoreo de sistemas de automatización heterogéneos, fue dirigido y revisado por mi persona.



Ing. Luis Ismael Minchala Ávila, PhD
Director

Dedicatoria

Dentro de mi recorrido por la universidad me puede percatar de que existen muchas cosas para lo que tengo ciertas destrezas. Por lo general, soy una persona que prefiere trabajar de manera individual. No obstante, siempre se tendrá el mejor resultado si las metas las realizas con el apoyo y ayuda de las personas correctas. Este aprendizaje nació en mi, cuando me di cuenta de que la ayuda idónea siempre llega en el momento preciso.

Por esta y más razones, deseo dedicar mi trabajo de titulación, en primer lugar a Dios por llenarme de sabiduría para no darme por vencido en momentos en donde casi todo estaba perdido. A mis padre, Diego Fabián Barriga Morales y Doris Cecilia León León, por ser mi motor y pilar en cada etapa de mi formación académica y sobre todo por brindarme su apoyo incondicional en cada decisión que tomaba en el camino. A mis hermanos Juan, Pedro y Domenica por nunca dudar de mis capacidades y afrontar los problemas conmigo. Al ingeniero Ismael Minchala, por ser un guía y un maestro, cuyos consejos permitieron que el presente trabajo culminará de la mejor manera. A mi amigo, Marcos Villarreal, por permitirme ser su compañero en este grato viaje de aprendizaje y la paciencia prestada. A familiares, maestros y amigos más cercanos que hicieron posible este trabajo.

Pablo Adrián Barriga León

Dedicatoria

A mis ángeles en el cielo, nada de esto fuese posible sin su amor constante que lo siento día a día. Les extraño mucho Papa Yoyito y Mamita Maruja.

A mis padres, por haber batallado y no dejarse vencer por la vida. A Nohemí Cristina, le amo mami y lo voy a hacer hasta el final de los días. Soy el mismo niño al cual usted regalo sus alas y su todo y el día de hoy empiezo a saldar una deuda que espero Dios me permita pagarla con creces.

Eh aquí el esclavo del Señor hágase en mi según tu palabra.

Marcos Lenin Villarreal Esquivel

Agradecimientos

En primera instancia, agradezco a Dios por brindarme su infinita sabiduría y permitir cumplir cada uno de mis logros, pese a las pruebas que se han presentado en el camino. A mis padres Diego y Doris, que han sido una bendición en todo el sentido de la palabra, por ello no cesan mis ganas de decir que gracias a ustedes he logrado cumplir una meta más en mi vida. Gracias por estar presentes en cada etapa de mi vida ofreciendo y buscando lo mejor para mi persona. A mis hermanos, por el apoyo incondicional y los consejos brindados. Al ingeniero Ismael Minchala por su sabiduría y esfuerzo dedicado en cada etapa de este proyecto. A mi gran amigo Marcos Villarreal por brindarme la oportunidad de compartir el desarrollo del presente trabajo de titulación y ayudarme a culminar un meta más. El camino no ha sido fácil, pero gracias maestros, familiares y amigos por el conocimiento impartido, sin el cual no hubiera sido posible cumplir con cada uno de los objetivos planteados en la vida universitaria y sobre todo, de esta manera permitirme obtener un afable título profesional.

Pablo Adrián Barriga León

Agradecimientos

A Dios, por luego de tanto tiempo demostrarme que su gracia es infinita y que sus designios son perfectos.

A mi Padre, por enseñarme que nada en la vida es difícil que solo se necesita un "poquito"de esfuerzo.

A mi Madre, por haber batallado y ganado de la mano de Dios y no dejar solos a sus hijos.

A mi Tío Diego, por enseñarme que hay que siempre tener fe en que las cosas van a mejorar y sobre todo que a pesar de los golpes que nos de la vida hay que mantenernos con el corazón puro y limpio como el de un niño.

A los Gonzalez, por ser mi segunda familia con los que compartí muchos momentos felices de mi vida cuando de niño.

A Ismael, por haber sido profesor, guía y mentor en todo este largo caminar de la Universidad y enseñarnos a todos y cada uno de sus alumnos lo que significa en realidad ser un amigo mas.

A Pablo, por creer en mi y haberse arriesgado a las ideas que dia a día llevaba a su hogar para que la tesis siga avanzando, sin tu ayuda y tu apoyo nada de esto hubiese llegado a su fin.

Por ultimo y mas importante, a ese amor constante y prudente, que me permitió crecer y verla crecer a mi lado. Mónica, te amo y siempre lo voy a hacer, se que el caminar no ah sido fácil pero poco a poco vamos llegando a todas las cosas que algún día como adolescentes soñamos. Recuerda siempre que cada una de las cosas que hago es con todo el amor para nuestra pequeña familia. Nuestros sueños están a la vuelta de la esquina. Te amo.

Marcos Lenin Villarreal Esquivel

Abreviaciones y Acrónimos

AWS Amazon Web Service. 26, 39, 40, 48–50, 61

CIM manufactura integrada por computador. 5, 9, 16
CRC Cyclic Redundancy Check. 13

DTI diagrama de tubería e instrumentación. 20

FVC Flow Valve Control. 25

HMI interfáz Hombre-Máquina. 5, 6, 36, 37, 44, 45, 56, 62
HTTP Hypertext Transfer Protocol. 17, 18, 39, 47, 56

IIoT Industrial Internet of Things. 1, 2, 4–6, 8, 9, 17, 37, 45, 46, 62
IoT Internet of Things. 8, 9, 17–19, 47, 50, 61, 62
IT Tecnología de información estándar. 4, 12

JSON JavaScript Object Notation. 18

LC Level Control. 26
LG Level Viewing Device. 25
LK Control Station. 25

M2M Machine to Machine. 4, 18
MQTT Message Queuing Telemetry Transport. 4, 6, 17, 18, 20, 39, 47, 53, 56

ODBC Open DataBase Connectivity. 4
OPC Open Protocol Communication. 3, 4, 14, 15, 20, 34–37, 50, 62
OT Tecnología operativa. 4

PB Pressure Burner. 25
PLC Programmable Logic Controller. 3, 4, 12, 16, 20, 27–29, 34, 35, 37, 38, 56, 57, 59, 62
PROFINET Process Field Network. 4, 12

REST Representational State Transfer. 4

SCADA Supervisión, Control y Adquisición de Datos. 3, 5, 6, 16, 36, 37, 50
SDN Redes definidas por software. 5
SNMP Simple Network Management Protocol. 4

TIC Tecnologías de la información y comunicación. 2

YE State Sensor. 25

Capítulo 1

Introducción

El Internet industrial de las cosas o Industrial Internet of Things (IIoT) consiste en maquinaria conectada a Internet y en avanzadas plataformas de análisis que procesan los datos que se producen. Los dispositivos IIoT van desde diminutos sensores ambientales hasta complejos robots industriales. Si bien la palabra «industrial» puede referirse a almacenes, astilleros y fábricas, las tecnologías IIoT son prometedoras para una amplia gama de sectores industriales, como la agricultura, la sanidad, los servicios financieros, el comercio minorista y la publicidad [1].

Actualmente existe una gran variedad de plantas industriales encargadas de la fabricación de productos necesarios para satisfacer las necesidades del desarrollo del mercado. Según [2], la implementación de nuevas tecnologías mediante el IIoT permite un avance tecnológico, ocasionando que los campos de monitoreo y control jueguen un papel fundamental en los procesos de manufacturación.

1.1. Antecedentes

La automatización es la única forma de incrementar la productividad de un país en pleno desarrollo. Numerosos autores discuten acerca de distintas formas de cómo poner en práctica ese concepto. No obstante, cada uno de ellos converge en la misma resolución, la "falta de interés por parte del ser humano". Los autores de [3] consideran que el principal valor de un hombre es el tiempo libre y la cantidad que disponga de él, se encuentra determinada por la productividad del trabajo en la sociedad. En este contexto, propone numerosos factores acerca de cómo incrementar la productividad laboral, tales como: recursos laborales baratos, fortalecimiento de la explotación humana, capacitación de los trabajadores y organización de la producción. Sin embargo, cada uno de ellos no traen consigo buenos resultados debido a la falta de modernización por parte de las industrias.

Anteriormente, la automatización de un proceso industrial se consideraba una utopía debido al alto grado de complejidad. No fue hasta la invención del Internet, cuando todo esto se creía posible. La influencia del Internet logró un creciente número de sistemas de automatización habilitados por el

IIoT [4]. De esta forma se posibilita la creación de nuevos modelos de negocios que parten desde el control remoto de procesos de manufacturación, hasta despliegues de sistemas de diagnóstico. Por esta razón, los autores de [5] aseguran que la nueva trayectoria que tomen las comunicaciones industriales se vuelve indispensable para el desarrollo de la nueva era de procesos de producción. Efectivamente, las Tecnologías de la información y comunicación (TIC) juegan un papel fundamental dentro del marco de estudio que se está analizando, puesto que, facilitan un nuevo proceso de organización, y en consecuencia una mayor respuesta por parte de las empresas ante las necesidades de los clientes.

Existe una gran variedad de problemas que dificultan el avance tecnológico en el campo de la industria. Según [6], el error más común recae en el manejo pobre de la información al momento de la toma de decisiones sobre cualquier proceso que se vea involucrado dentro de la línea de producción. Una solución ante esta problemática son las llamadas fábricas inteligentes, cuya función es la interconexión entre cada uno de sus sistemas de producción mediante redes. De esta manera, se genera una enorme cantidad de información la cual puede ser manejada a través de distintos protocolos de IIoT. Por esta razón, [7] aborda una visión en el desarrollo e implementación de estándares que involucren la conectividad de las tecnologías IIoT en distintas plataformas con el objetivo de promover la industria 4.0, enfatizando principalmente en la seguridad del sistema. Finalmente, en [8] se destacan numerosos estudios en donde se detalla que la tecnología IIoT ha adquirido aceptación por parte de las industrias, debido a que permite la creación de sistemas potentes que benefician la línea de producción para cualquier campo. No obstante, se debe considerar que conforme se aumenta la complejidad del sistema, el nivel de seguridad de la información que se maneja se debe incrementar radicalmente.

1.2. Identificación del problema

Actualmente, una de las áreas que se encuentra en constante desarrollo es el mejoramiento de sistemas productivos y la optimización de costos a través de tecnologías de automatización aplicadas a la industria. La escasa implementación de tecnologías de última generación aplicadas a sistemas de manufactura demuestra la necesidad de mejorar el manejo, supervisión y control de procesos heterogéneos, ampliamente desplegados en la industria moderna, que no contemplan interconexión directa por restricciones de protocolos de comunicación, entre otras cosas.

Algunos países de Latinoamérica como Ecuador, cuentan con la oportunidad de aprovechar las posibilidades que el Internet brinda para evolucionar en campos de la industria. Sin embargo, la implementación de escenarios reales se encuentra limitada debido a la falta de equipamiento para el manejo y envió de datos a la nube. En este contexto, el principal problema al que se enfrenta un ambiente de automatización simulado es el manejo de la enorme cantidad de protocolos de comunicación industrial y como interconectar cada uno de ellos en función a los distintos niveles de la pirámide de automatización. Las empresas hoy en día emplean distintos sistemas de hardware para entrelazar procesos heterogéneos y comunicarlos con un servidor en la nube, de esta forma se establece un nivel de seguridad respecto a la información [9]. Los sistemas que involucran protocolos IIoT plantean desafíos para la conexión entre procesos, uno de ellos hace referencia a la entrada dinámica de componentes durante el tiempo de ejecución y la heterogeneidad de los mismos. Los autores de la referencia [10] aseguran que el desafío se encuentra en cómo satisfacer la heterogeneidad de cada uno de los componentes involucrados en el

proceso de automatización y de esta manera permitir la integración durante el tiempo de simulación. Efectivamente, el reto de enlazar cada proceso trae consigo problemas de seguridad. La alta conectividad entre todos los objetos obliga a realizar una auditoría ante cualquier falla que puede presentarse [11]. Otro aspecto que se debe considerar engloba el mercado de las comunicaciones industriales, ya que, se encuentra dominado por sistemas de bus de datos. Como resultado, las empresas se enfrentan a un ecosistema con una abundante cantidad de tecnologías que deben producirse, ejecutarse, almacenarse y sobre todo interconectarse [12].

De acuerdo a lo descrito anteriormente, y con el objetivo de recalcar las deficiencias que presenta la etapa de automatización de un proceso industrial, en la figura 1.1 se observa una descripción gráfica acerca del problema central de la investigación.

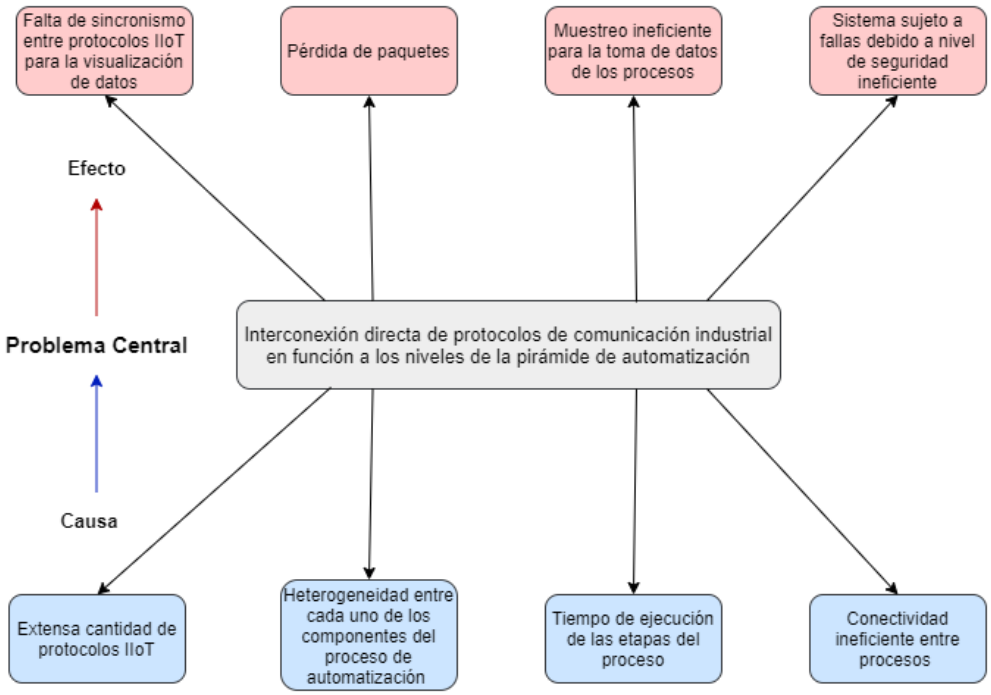


Figura 1.1: Árbol del problema.

Un proceso de automatización se fundamenta en una estructura jerárquica o pirámide de cinco niveles en donde se integran todas las tecnologías involucradas en el proceso de la línea de producción [13]. Estos niveles son: campo, control, supervisión, planeación y gestión y se explicarán a detalle en el capítulo 2. Cada uno de esos niveles presentan distintas tecnologías que deben emplear para cumplir una tarea específica. Por ejemplo, en el nivel de campo se puede usar cualquier herramienta de simulación de procesos industriales como Factory I/O [14] o FlexSim [15]. Entre estas opciones en este trabajo se usa el software Factory I/O debido a su semejanza con un ambiente industrial real y sobre todo las opciones de interconectividad con dispositivos Programmable Logic Controller (PLC), así como también distintos tipos de *middleware*. Cada uno de los procesos didácticos industriales que se definen en primer nivel emplean Open Protocol Communication (OPC) con dispositivos Siemens que son controlados y monitoreados a través de un sistema de Supervisión, Control y Adquisición de

Datos (SCADA).

La comunicación en la nube permite un manejo flexible y dinámico de la información, pero presenta una desventaja frente a sistemas que no son en tiempo real. Según [16], esto se debe principalmente a la falta de control al momento de crear respaldos o bases de datos para la información de etapas de producción. De esta manera se evita cualquier tipo de inconveniente cuando se produzca un enlace roto con la nube. La transferencia de datos entre cada uno de los niveles resulta crucial al momento de implementar cualquier tipo de proceso industrial. Actualmente, existen algunos tipos de *middleware* del tipo *open source*, que se emplean para establecer un enlace de comunicación entre los niveles de la pirámide y la web, uno de ellos es Modbus RTU. Mobdus es un protocolo estándar para la industria basado en la arquitectura cliente/servidor, permitiendo el control de una red de dispositivos que pueden comunicar los resultados a un ordenador [17].

Adicionalmente [18], describe un protocolo denominado Process Field Network (PROFINET). Básicamente es un estándar de comunicación Ethernet industrial basado en estándares TCP/IP y desarrollado como un mecanismo para intercambiar datos entre controladores y actuadores. Sin embargo, la tecnología de comunicación industrial más empleado por las empresas es OPC, debido a que maneja una gran cantidad de funciones utilizadas por sistemas de comunicación industriales, lo que implica la posibilidad de habilitar un enlace adecuado entre las capas de Tecnología operativa (OT) y Tecnología de información estándar (IT) [19].

Con referencia a OPC, [20] señala que es una solución óptima a la problemática de interconectividad entre los niveles de la estructura jerárquica de automatización. Además, de permitir el intercambio de información entre múltiplos dispositivos PLC's, presenta la información en distintos formatos como:

- Simple Network Management Protocol (SNMP).
- Open DataBase Connectivity (ODBC).
- Representational State Transfer (REST).
- Message Queuing Telemetry Transport (MQTT).

En este trabajo se utiliza MQTT debido a que se considera el protocolo de comunicación IIoT más empleado en procesos de manufacturación, ya que permite la comunicación Machine to Machine (M2M).

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar una metodología de integración de procesos heterogéneos de automatización empleando protocolos de comunicación industrial IIoT.

1.3.2. Objetivos específicos

El presente trabajo tiene los siguientes objetivos específicos:

- Diseñar y simular un sistema SCADA para una línea de producción de zumo de naranja.
- Investigar nuevas alternativas para el desarrollo de aplicaciones en el campo de la automatización.
- Simular procesos de manufacturación y enviar la información de los procesos de entrada de la línea de producción hacia un servidor en la nube.
- Implementar un interfáz Hombre-Máquina (HMI) para el control de línea de producción.
- Diseñar los sistemas de producción usando el diseño de manufactura integrada por computador (CIM) y virtualizar los servicios de la capa 4 (sistema de ejecución de manufactura).

1.4. Revisión del estado del arte

El *boom* de la industria ha ocasionado que numerosos países se adapten a los cambios que trae consigo la revolución industrial. Evidentemente, dentro de esos cambios existe una amplia gama de campos que se pueden mejorar, no obstante, el de mayor interés en la actualidad recae en la automatización industrial de una línea de producción. Con la Industria 4.0, el cliente podrá comunicarse directamente con el fabricante, es decir, se disminuiría el tiempo de espera a las solicitudes. Estos cambios, efectivamente, requieren un mayor control por parte de las empresas [21]. La Industria 4.0 trajo consigo el desarrollo de técnicas de detección de fallas, análisis de datos, y sobre todo la comunicación entre las distintas etapas que engloba un proceso de manufactura. Según [22], cada uno estos procesos convergen en un solo punto, el surgimiento y desarrollo del IIoT. De acuerdo a varios autores, el IIoT supone una revolución en la monitorización y mantenimiento de sistemas automatizados, por ende, resulta preciso establecer un entorno de desarrollo en donde los usuarios interactúen en cada etapa de la línea de producción a través de servidores web.

El IIoT presenta ciertas desventajas en cada uno de los campos de aplicación. En este contexto, se puede mencionar las redes industriales, cuya característica principal recae en la heterogeneidad. De acuerdo a [23], las redes industriales son heterogéneas debido a la enorme cantidad de dispositivos que se encuentran interconectados mediante protocolos de comunicación industrial. Sin embargo, dicha conectividad presenta deficiencias en sistemas de control automatizados. Por lo tanto, lo que se busca es desacoplar los planos de control y datos empleando Redes definidas por software (SDN), de esta manera se logran cumplir los requisitos que exige la Industria 4.0, los mismos que son: confiabilidad, escalabilidad y baja latencia. Actualmente, la *Fog Computing* es la tecnología que permite brindar un soporte de procesamiento local con una latencia aceptable para dispositivos IIoT [24].

Un trabajo interesante se puede mencionar en [25], en él se describe los principales problemas acerca de una línea de manufacturación. Básicamente, considera que el alto nivel de variabilidad se debe a la heterogeneidad de cada uno de los factores que se ven involucrados en un proceso industrial. En consecuencia, ello implica que el nivel de producción disminuya y por ende el tiempo de latencia aumente. En la actualidad, las industrias encargadas de la producción de semiconductores se ven más afectadas por dicha desventaja. Por otro lado, los autores de la referencia [26] describen un flujo de una red IIoT orientado a la industria. Básicamente plantean un flujograma desde la recopilación de datos hasta la toma de decisiones, donde el procesamiento de los mismos juega un papel fundamental. De esta manera, con la ayuda de un conjunto de protocolos de comunicación se interconecta cada etapa en función a las fases de la pirámide de automatización.

De acuerdo a algunos autores, Japón encabeza la lista de países con el mayor índice de empresas automatizadas, lo que implica un elevado crecimiento económico. Según el diario "El Español", Japón ocupa la cuarta economía más automatizada a nivel mundial debido al elevado índice de fabricación de robots, aproximadamente un 56 % [27]. En Ecuador aún se ve distante el ingreso a la Industria 4.0, por lo que, actualmente se encuentra desarrollando procesos bajo la idea de un sistema más eficiente en diferentes áreas de aplicación. A continuación, se detalla algunos de estos trabajos.

En [5], se describe el diseño de un sistema distribuido que emplea protocolos IIoT, pero monitorizado en tiempo real. Es decir, implementa cualquier tipo de comunicación industrial y envía la información a través de Internet para el manejo y motorización desde algún servidor web que almacena Node-RED en la nube. El autor de [1] detalla un proceso de parametrización automática de variables de control en un proceso de mezclado de compuestos termoplásticos. El proceso consiste en la lectura de un código de barras para configurar de manera inmediata cada una de las variables que participan en la etapa de mezclado, de esta manera se elimina cada uno de los residuos para un posterior envío hacia un sistema SCADA y manejo de información en la nube empleando el protocolo MQTT. En [9] se describe la misma problemática que los otros autores, con la diferencia de un controlador PID para el manejo del caudal de un río analizando un gran conjunto de métricas como: medición de un sensor ultrasónico, valor de un caudalímetro, válvula de globo que permita interrumpir el paso del caudal ante una falla del sistema, entre otros.

De acuerdo a lo descrito anteriormente, resulta sencillo concluir que la investigación de redes IIoT se centra en la transmisión a gran escala para adaptarse al campo de la industria. Es decir, los procesos industriales se deben ajustar necesariamente a las variantes que presentan los sistemas de comunicación dentro de la línea de manufactura [28]. En este contexto, vale la pena mencionar que numerosos autores proponen que las redes IIoT deben considerar tres métricas incluidas, el tamaño, área y gestión. No obstante, [29] indica que esta última se convierte en una gran dificultad cuando la cantidad de datos a ser tratados es sumamente grande. En consecuencia, la computación en la nube anticuada resulta ineficiente al momento de satisfacer las nuevas necesidades de la comunicación IIoT. Finalmente, [30] describe un marco referente a una fábrica inteligente en función a la combinación de una red industrial, la nube y terminales de control como transportador, producto y máquina.

1.5. Contribuciones

Actualmente, la mayoría de procesos industriales orientados al campo de la automatización presentan paneles de control para el monitoreo y manejo total de cada etapa de la línea de producción. En consecuencia, el sistema HMI implementado para ello presenta una restricción respecto al acceso y manejo del mismo por parte de operarios que no tiene acometida al mismo. Se sabe que escenarios industriales físicos, la posibilidad de que una etapa presente una falla es elevada, en consecuencia, surge la necesidad de diseñar una solución en el beneficio de la industria.

El presente proyecto brinda la posibilidad de virtualización de servicios de la capa MES mediante la implementación de un agente detector de fallas a nivel local de la red empleando la herramienta de

Node-RED. Es decir, se habilita la posibilidad de informar de manera general el estado de los procesos de la industria, sin la necesidad de tener que acceder al panel de control del sistema. De igual manera, con el objetivo de diseñar la solución de manera más dinámica, el sistema incluye un *bot* con el cual cada operario de la fábrica podrá interactuar, para así conocer cada uno de los aspectos más relevantes de la producción.

Capítulo 2

Fundamentos teóricos

En la siguiente investigación es fundamental describir algunos conceptos fundamentales. Este capítulo presenta los siguientes fundamentos teóricos: la industria 4.0, redes industriales, protocolos de comunicación industrial, modelo de manufactura integrado por computador, IIoT, entornos de desarrollo IIoT, concepto de gemelo digital y la normativa ANSI/ISA S 5.1.

2.1. La industria 4.0

2.1.1. Industria 4.0 y manufactura inteligente

El término industria 4.0 ha tomado fuerza en los últimos años, el cual presenta un modelo novedoso de organización y control de las cadenas de producción, a partir del ciclo de vida del producto que se está elaborando. La industria 4.0 no es más que la aplicación del Internet de las cosas o Internet of Things (IoT) en la industria, lo que ha provocado la llamada cuarta revolución industrial, debido a su potencial y los beneficios de la integración de procesos de producción, con un alto nivel de innovación y autonomía por parte de los eslabones de cada uno de los procesos productivos. La industria 4.0 ha sido factible a partir de las tecnologías de la información, el intercambio de datos a gran velocidad, sistemas cibernéticos y físicos relacionados con el IoT [31] [32]. La figura 2.1 muestra un diagrama de interrelación entre cada uno de los componentes que conforman la industria 4.0.

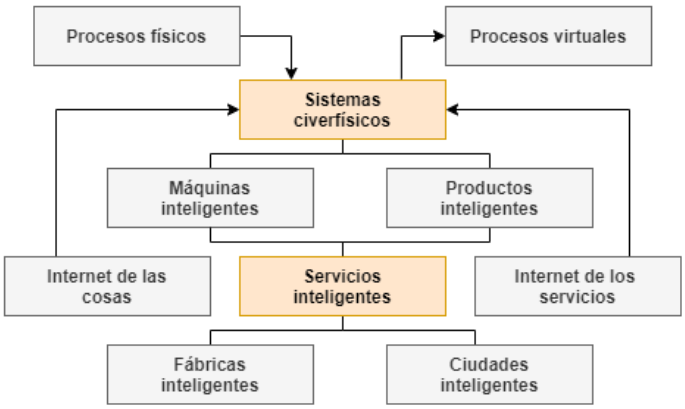


Figura 2.1: Diagrama de interconexión entre componentes de la industria 4.0

En este mismo orden de ideas, la manufactura inteligente es considerada como la capacidad de representar de manera digital cada aspecto de la manufactura convencional, desde el diseño hasta el proceso de de producción, con la ayuda de distintas herramientas de software como el CIM. Este tipo de manufactura se especializa el uso de tecnologías para la planeación y validación de las etapas de fabricación. De esta manera, se busca flexibilizar el proceso obsoleto, al mejorar la calidad del producto y acelerar el tiempo de respuesta al mercado [33] [34]. Desde este punto de vista, los cambios que se están dando en el proceso de manufactura son resultado de tecnologías desarrolladas para la digitalización de producción y la automatización.

2.1.2. Tecnologías asociadas a la industria 4.0 y la manufactura inteligente

Dentro de las tecnologías que sustentan la industria 4.0 y la manufactura inteligente, se refieren la simulación, fabricación aditiva, los sistemas de integración horizontal y vertical, la ciberseguridad, la realidad aumentada, el cómputo en la nube, los robots autónomos, el IIoT y el *big data* [35]. La industria 4.0 posee una serie de componentes fundamentales para su desarrollo, estos componentes giran en torno a los sistemas ciber físicos, máquinas y productos inteligentes, pues como se mencionó anteriormente, uno de los pilares es el ciclo de vida que posee el producto fabricado para la optimización del proceso productivo. El IoT y el internet de los servicios permiten que tanto los procesos físicos como virtuales integren cada eslabón de la cadena de producción en las empresas [31] [32].

Mientras que, el *big data,* el cómputo en la nube y la inteligencia artificial sean facilitadores de la industria 4.0, junto con la automatización industrial están cambiando la forma en la que los productos se fabrican al contribuir al mejoramiento de la manufactura y a que las empresas cuenten con procesos totalmente automatizados e interconectados, que faciliten el flujo de información, la descentralización de la manufactura, la creación de nuevos procesos, la toma de decisiones y un enfoque al desarrollo de competencia [36].

2.2. Redes industriales y automatización

Se conoce como automatización al proceso de utilizar maquinaria para llevar a cabo determinadas tareas que comúnmente han sido ejecutadas por el ser humano, con la finalidad de controlar la

secuencia de operaciones sin intervención humana. El término automatización describe también a aquellos sistemas no destinados a la fabricación que son programados o pueden funcionar de forma independiente, sin la necesidad de un operario. Existe un innumerable rango de áreas en las que se pueden aplicar procesos de automatización, tales como: las comunicaciones, aviación, astronáutica y dispositivos como los equipos de conmutación telefónica. Con la implementación de autómatas, pilotos automáticos y sistemas automatizados de guía, se pretende controlar diversas tareas con una mayor velocidad y precisión en comparación con la intervención humana [37].

Uno de los campos que se ha visto beneficiado por la automatización de procesos es la industria, ya que, al reemplazar la mano de obra convencional por autómatas, se ha logrado obtener una mejora considerable en la línea de producción. La automatización industrial (automatización; proviene del griego antiguo auto: guiado por uno mismo) es la aplicación de sistemas o elementos computarizados que permiten un control de maquinaria y/o procesos que se llevan a cabo día a día en la industria. El alcance y visión de los procesos automatizados va más allá de una simple mecanización, puesto que provee a los operadores una variedad de mecanismos para asistirlos en el trabajo, facilitando así cierto tipo de procesos que para el ser humano son complicados de realizar, como lo son los trabajos de fuerza y movilización de materiales ocupados en la industria. La automatización ha reducido de gran manera la intervención de la habilidad sensorial y mental del ser humano [37].

El gran progreso de la automatización a través de los años se ha visto reflejado en su máxima expresión en el campo de la robótica industrial, puesto que los robots con sus múltiples funcionalidades, han facilitado muchos procesos tediosos y con gran margen de error en la línea de producción industrial. Algunas de las ventajas de la automatización industrial son: la repetitividad, el control de calidad, mayor eficiencia, integración con sistemas empresariales, incremento de productividad y reducción de trabajo. Por otra parte, las desventajas que presentan son requerimientos de un gran capital e inversión, decremento severo en la flexibilidad, y un incremento del mantenimiento y reparación.

Pero no se puede negar que, dentro de la industria, la automatización se ha convertido en la herramienta indispensable a la hora de mejorar la producción, reduciendo el tiempo de fabricación con una simplificación de tareas complejas y tratando de desperdiciar la menor cantidad de recursos posibles [37].

Dependiendo de las necesidades y sobre todo del capital de inversión que posea la empresa se podrán ejecutar distintos tipos de automatización, conocidos como niveles de automatización:

- **Manual:** Proceso en el cual el ser humano realiza todas las operaciones mediante el uso de herramientas especificas dependiendo del ámbito en el que se encuentre.
- **Mecanizado:** Implementación de maquinaria para realizar la operación, pero controlada por el ser humano.
- **Automatización parcial:** Es el proceso en el cual la máquina realiza diversas operaciones en secuencia y autónomamente, pero que requiere de cierta intervención del ser humano para completar el proceso.
- **Automatización total:** En este nivel la máquina es la que se encarga de realizar el proceso de principio a fin, sin la necesidad de la intervención del ser humano. Únicamente el operador intervendrá en procesos de supervisión y mantenimiento preventivo [38].

Si bien la maquinaria es un pilar fundamental de la Automatización Industrial, otro de ellos es la implementación de redes de comunicación para el intercambio de información y la correcta sincronía entre cada eslabón del proceso de producción, así como el registro y control que permita la corrección de errores y evitar el deterioro de la maquinaria con el tiempo. En los últimos años, el aumento de la demanda en el campo de la automatización ha dado paso a la implementación de diversos tipos de comunicaciones, conexión de sensores, actuadores y equipos de control, para garantizar el correcto funcionamiento entre los instrumentos de campo, el registro de control y los operadores.

Para la implementación de una red de comunicación industrial se necesitan equipos y herramientas previamente programadas para trabajar de forma correcta, esto con la finalidad de llevar un registro y control de los actuadores del sistema, lo cual servirá para mejorar la calidad de los productos.

2.3. Protocolos de comunicación industrial

En las redes de comunicación industrial se deben seguir un conjunto de reglas, las cuales permiten el intercambio de información entre los diferentes componentes del proceso de automatización. Este conjunto de reglas se lo conoce como protocolo de comunicación industrial, que permite enlazar diferentes dispositivos que componen la red de comunicación. Con el pasar de los años y el avance tecnológico de la automatización los protocolos han evolucionado, con la finalidad de tener una respuesta a las necesidades en la industria. La figura 2.2 muestra un breve esquema acerca de los protocolos de comunicación industrial más empleados en el mercado.

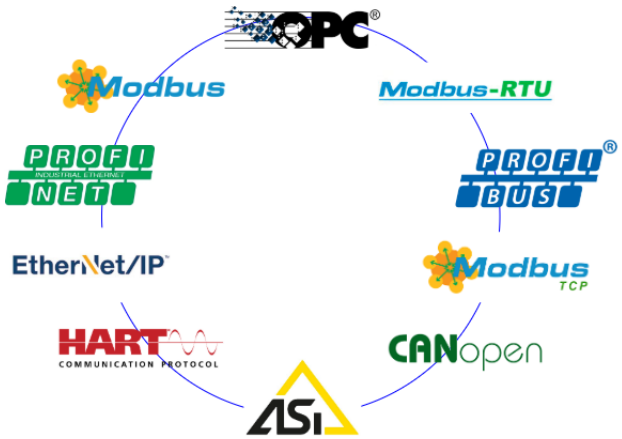


Figura 2.2: Protocolos de comunicación industrial empleados para proceso de automatización.

A continuación se describe de manera breve los protocolos de comunicación industrial más aceptados en el mercado.

- **Ethernet/IP:** Es un protocolo basado en el estándar TCP/IP para la implementación de aplicaciones de Automatización Industrial, mediante la clasificación de nodos en base a los dispositivos preestablecidos. Ethernet/IP aplica tecnología tradicional mediante protocolos de transporte

TCP, el Internet y la tecnología de acceso y señalización en la interfaz de las tarjetas Ethernet.

- **PROFIBUS**: Se caracteriza por ser un protocolo abierto e independiente, por lo cual genera una amplia gama de aplicaciones en procesos, fabricación y automatización, debido a que no representa a ningún proveedor. PROFIBUS posee tres versiones en cuanto a comunicaciones se refiere, los cuales son: PROFIBUS-DB, posee una alta velocidad de transmisión al establecer la comunicación entre el controlador y los dispositivos de campo. El segundo protocolo es PROFIBUS-PA, el cual establece una comunicación de alta velocidad y sobre todo confiable al momento de trabajar en ambientes expuestos a riesgos de explosión. El tercer y último estándar es PROFIBUS-FMS, el cual establece una comunicación a nivel de célula, cuyo objetivo principal es el manejo de volúmenes de información y el tiempo de respuesta [39] [40].

- **PROFINET:** Es un protocolo que combina ciertos estándares de comunicación relacionados a IT y el Ethernet industrial TCP/IP. Una de sus principales características es que trabaja en tiempo real, a partir de la comunicación por el bus de campo procesando solicitudes que se realizan dentro del bus. En PROFINET también existen una serie de sub protocolos con funciones específicas:

  1. **PROFINET/CBA**: Aplicado para entornos industriales mediante automatización distribuida.
  2. **PROFINET/DCP**: Se lo utiliza para la configuración de nombres de dispositivos y asignación de direcciones IP, es decir, es un protocolo basado en la capa de enlace.
  3. **PROFINET/PTCP**: Es otro de los estándares basado en la capa de enlace, diseñado para sincronizar señales de reloj/tiempo en conjuntos de PLC's.
  4. **DeviceNet**: Protocolo que permite la interconexión de dispositivos de control para el intercambio de información. DeviceNet tiene la capacidad de conectar dispositivos individuales con el controlador de la red [39] [40].

2.3.1. Mobdus

Este protocolo ha sido diseñado para poder controlar una red de dispositivos, mediante una estructura de mensajería, con la finalidad de establecer una conexión entre cliente y servidor de los dispositivos. Dentro de Modbus existen dos tipos de protocolos:

2.3.1.1. Modbus TCP/IP:

Es una variante de la familia Modbus de protocolos de comunicación simples y neutrales para la supervisión y el control de equipos de automatización. Específicamente, cubre el uso de la mensajería Modbus en un entorno 'Intranet' o 'Internet' utilizando los protocolos TCP/IP. El uso más común de los protocolos en este momento es para la conexión Ethernet de PLC's, módulos de E/S y "puertas de enlace" a otros buses de campo o redes de E/S simples. El protocolo está publicando como un estándar de automatización. Sin embargo, se ha intentado aclarar qué funciones dentro de Modbus tienen valor para la interoperabilidad de los equipos de automatización generales y qué partes son "equipaje" del uso alternativo como protocolo de programación para PLC [40].

En Modbus, las transacciones de datos son tradicionalmente sin estado, lo que las hace altamente resistentes a las interrupciones del ruido y, sin embargo, requieren que se mantenga una mínima información de recuperación en cada extremo. Las operaciones de programación, por otro lado, esperan un enfoque orientado a la conexión. Modbus TCP/IP maneja ambas situaciones. Una conexión es fácilmente reconocible a nivel de protocolo, y una sola conexión puede llevar múltiples transacciones independientes. Además, TCP/IP permite un gran número de conexiones simultáneas, por lo que en la mayoría de los casos es la elección del iniciador si se vuelve a conectar según sea necesario o se reutiliza una conexión de larga duración [40]. La principal razón por la que se usa el protocolo TCP/IP es para mantener el control de una 'transacción' individual encerrándola en una conexión que pueda ser identificada, supervisada y cancelada sin requerir una acción específica por parte de las aplicaciones del cliente y del servidor. Esto le da al mecanismo una amplia tolerancia a los cambios en el rendimiento de la red, y permite que se agreguen fácilmente características de seguridad como cortafuegos y servidores *proxy* [40].

2.3.1.2. Modbus RTU

Es un protocolo de comunicación punto a punto del tipo *open source*. Se utiliza para desarrollar la comunicación *Multi-Master Slave / Server Client* entre dispositivos inteligentes, además de utilizar distintas topologías según la especificación de la capa física; como RS-232, RS-422 o RS-485. RS-232 es una topología dúplex que transmite y recibe datos al mismo tiempo, mientras que RS-485 es una topología semidúplex en la que los procesos de transmisión y recepción se llevan a cabo uno tras otro [41].

En Modbus RTU, la capa física es la responsable del direccionamiento de dispositivos esclavo, el bit de inicio y de parada, el tiempo de espera y la detección de errores en la trama de transmisión. Por otro lado, la capa de enlace se encarga del reconocimiento o rechazo del código Cyclic Redundancy Check (CRC). El código de función para este protocolo es del tipo *float* con una longitud de 32 bits. La figura 2.3 muestra el formato de envió de mensajes en Modbus RTU. Básicamente, los dispositivos maestros envían consultas a los esclavos y estos envían una respuesta a la consulta en función a un código específico.

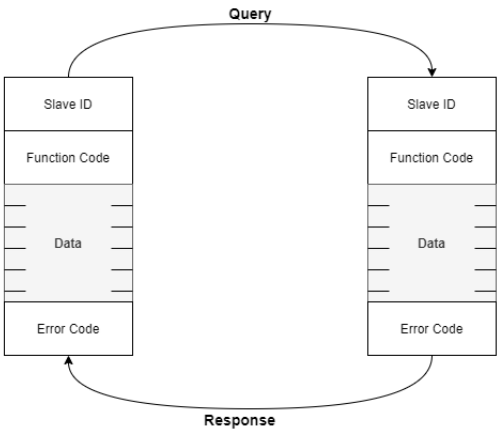


Figura 2.3: Formato de mensajes en Modbus RTU.

Donde:

- **Slave ID:** Es un campo de dirección capaz de conectar hasta 256 dispositivos a la red. El ID de esclavo 0 se utiliza para retransmisiones o como maestro. Por otro lado, los ID en el rango 1 a 247 se utilizan para dispositivos esclavos y las 248 a 255 para puertas de enlace.
- **Function Code**: Es un campo de dirección que indica el tipo de acción a ejecutarse por un dispositivo esclavo.
- **Data:** Es un campo destinado al almacenamiento de información de cada uno de los dispositivos que se encuentran conectados a la red.

2.3.2. OPC

Es un estándar que se implementa como un servidor de enlace entre diferentes softwares, para facilitar la transmisión de datos entre sí. Este protocolo está basado en la arquitectura cliente/servidor, donde el servidor OPC es la fuente de los datos a la cual, cualquier aplicación que esté basada en este protocolo puede acceder para leer o escribir en cualquier variable del servidor. Dentro de las principales características de OPC está la capacidad resolutiva a problemas de interacción entre los softwares y autómatas que son parte de la red [42].

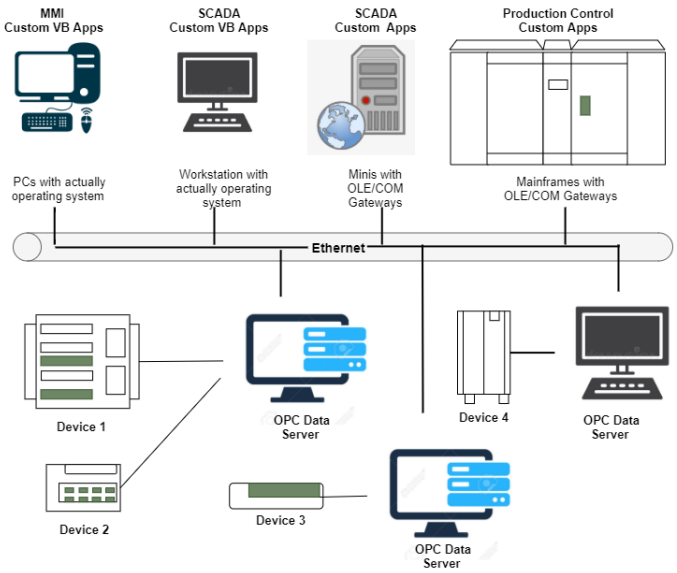


Figura 2.4: Red de comunicación OPC.

La figura 2.4 describe la comunicación OPC como una herramienta en la cual diferentes equipos con protocolos de comunicación distintos pueden realizar una transmisión de datos.

Existen cuatro tipos de servidores OPC definidos por la *OPC Foundation*, los mismos que son:

- **Servidor OPC DA:** Diseñado específicamente para la transmsión de datos en tiempo real.
- **Servidor OPC HDA:** Basado en la especificación de acceso a datos historizados que provee al cliente OPC HDA.

- **Servidor OPC A&E Server:** Basado en la especificación de Alarmas y Eventos. Por lo general, transfiere alarmas y eventos desde el dispositivo hacia el cliente OPC A&E.
- **Servidor OPC UA:** Basado en la especificación de Arquitectura Unificada y en el *set* más nuevo y avanzado de la *OPC Foundation*, permite a los servidores OPC trabajar con cualquier tipo de datos.

En conjunto, los tres primeros tipos de servidores OPC se conocen como "Clásicos" para distinguirlos de OPC UA que se convertirá en la base de las futuras arquitecturas OPC.

2.3.2.1. OPC UA

Es un estándar de interoperabilidad para el intercambio de información sumamente empleado en la automatización industrial. OPC UA consta de 14 especificaciones principales que define la interfaz entre clientes y servidores, incluido el acceso a datos en tiempo real. Actualmente, se ha incluido un nuevo modelo del tipo *publisher/subscriber* con el propósito de permitir la transmisión multidifusión, y de esta manera mejorar la capacidad de adquisición de datos en tiempo real [xx]. Básicamente, la tecnología OPC UA consta de los siguientes elementos:

- Un metamodelo para definir modelos de información específicos.
- Un conjunto de especificaciones del protoloco de transporte, para el intercambio de información entre dispositivos.
- Servidores OPC UA, que contienen el modelo de información relacionado con la instalación en tiempo real. El modelo de información comprende una estructura jerárquica compuesta por un conjunto de nodos.
- Clientes OPC UA, encargados de la transmisión y recepción de mensajes para el acceso a los datos de los nodos en el modelo de información de los servidores.

La figura 2.5 muestra una diagrama básico de la arquitectura OPC UA y como se interconecta cada uno de los componentes descritos anteriormente.

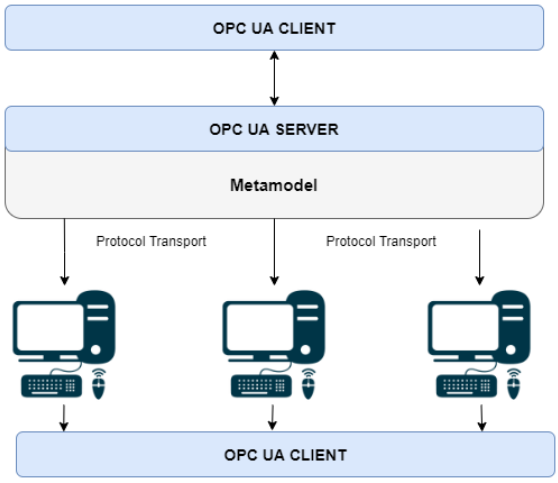


Figura 2.5: Interconexión de elementos del protocolo OPC UA.

2.4. Modelo de manufactura integrada por computador

En los procesos de automatización industriales, parte importante del éxito que tengan las empresas depende de un flujo efectivo y continuo de la información dentro de las mismas, con la finalidad de integrar a cada uno de los actores involucrados, por lo tanto, para lograr este objetivo se ha implementado el modelo de pirámide de automatización. La figura 2.6 muestra la estructura jerárquica del modelo de manufactura, en donde se describe de manera breve los cinco niveles de automatización .

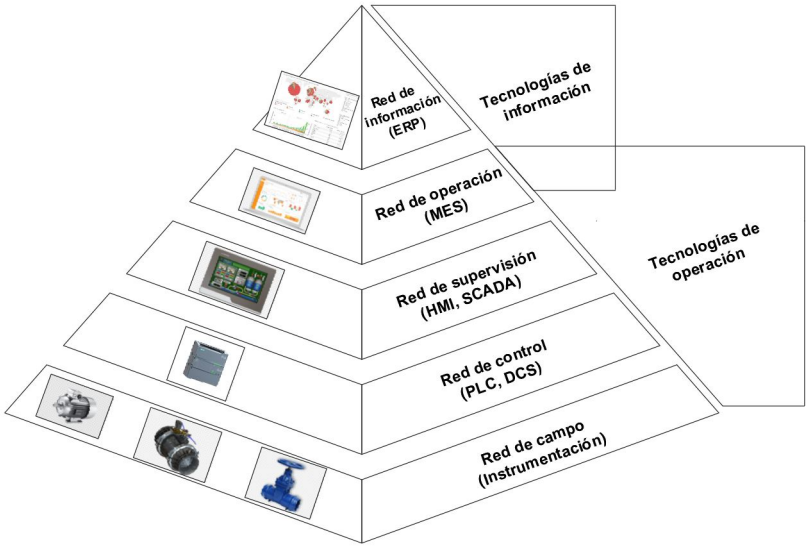


Figura 2.6: Pirámide de automatización industrial

La pirámide de automatización (CIM) es una estructura conformada por sistemas de control distribuidos que trabajan en un entorno de producción. Esta pirámide muestra la integración de los procesos de fabricación con los sistemas de gestión y la comunicación que existe entre cada uno de los eslabones de la cadena de producción. Si bien existen diversas variantes de la pirámide de automatización, generalmente todos coinciden en mostrar una arquitectura basada en cinco niveles que se distinguen por sus funciones y sus segmentos de red. Para lograr una correcta integración en la automatización de los procesos los niveles de la pirámide se comunican entre sí [39] [43] [44]. Los niveles de automatización son:

- **Nivel 0: Nivel de campo.** En el primer nivel de la pirámide se incluye la adquisición de datos de campo o instrumentos, que se encuentran distribuidos por todo el proceso y que tienen la función de controlar las máquinas y equipos que forman parte del proceso productivo o industrial.
- **Nivel 1: Nivel de control.** En este nivel se agrupan los controladores de equipos productivos o industriales tales como: ordenadores, PLC's, controladores PID, etc. Estos controladores utilizan la información proporcionada por los instrumentos del nivel de campo para controlar los actuadores de ese mismo nivel.
- **Nivel 2: Nivel de supervisión.** Para este nivel se cuenta con equipos que permiten controlar la secuencia de fabricación y/o producción tales como el sistema SCADA. Otra función importante de los equipos que pertenecen a este nivel es recoger y analizar los datos de la secuencia de

producción.

- **Nivel 3: Nivel de operación.** Conocido también como nivel de planificación. En este nivel se gestiona el flujo de trabajo y la ejecución para la producción y optimización de los productos fabricados.
- **Nivel 4: Nivel de gestión.** En esta etapa de la pirámide se abarca todas las actividades relacionadas con el negocio y la gestión de la empresa, comunicando a las diferentes plantas de producción y manteniendo una correcta relación con los proveedores y clientes [39] [43] [44].

Un punto importante a tener en cuenta al momento de implementar un modelo de pirámide de automatización es que hasta el nivel 3, se debe gestionar en tiempo real la información de la cadena de producción. El modelo de pirámide de automatización tradicional que se utilizaba antiguamente se presentaba de manera muy rígida a nivel de instrumentos, por lo que no permitía la comunicación fluida entre cada nivel de automatización. En la actualidad y con la nueva industria 4.0 se precisa de redes que permitan integrar todos los recursos de la planta de producción, con un nivel de comunicación alto y procesos con mayor precisión y velocidad [39] [43] [44].

2.5. Internet industrial de las cosas

2.5.1. IIoT

El IIoT es la aplicación del IoT en el sector industrial, que está ligado directamente con el concepto de Industria 4.0 que se centra en la optimización de los procesos industriales. IIoT es un sistema conformado por componentes inteligentes, tecnologías de la información y plataformas de cómputo en la nube, lo que permite un almacenamiento de gran información, el acceso y recopilación de datos e intercambio de información en tiempo real, de forma inteligente y autónoma [45] [46]. El IIoT permite que los dispositivos tecnológicos interactúen entre ellos utilizando protocolos de Internet y el análisis *big data* para la prevención de errores, los cuales se pueden configurar de forma autónoma y que se pueden adaptar a cambios según las necesidades de la empresa. Es fundamental la vinculación del mundo físico con el mundo digital, puesto que la conexión de dispositivos, sistemas y operador requiere una colaboración total para el éxito del proceso de producción, generando ciclos de vida óptimos para los productos fabricados [45] [46].

Para el despliegue de sistemas y servicios IIoT, se requiere un diseño de arquitectura fiable que permita realizar operaciones de forma eficiente y efectiva, así como la función de interoperabilidad tomando en cuenta los servicios y la cantidad de partes involucradas, como lo son dispositivos inteligentes, sistemas de comunicación, proveedores de servicios y desarrolladores de negocios. Es por eso que se trabaja con énfasis en el desarrollo de estándares de referencia que puedan ser adaptadas por los diferentes interesados en este servicio [45] [46].

2.5.2. Protocolos de comunicación en IIoT

Para poner en marcha la automatización aplicando el IIoT se debe seguir un conjunto de reglas o protocolos preestablecidos para una correcta comunicación entre cada uno de los actores de las cadenas de producción. Para el presente proyecto se han aplicado dos protocolos para la implementación de IIoT, los cuales son Hypertext Transfer Protocol (HTTP) y MQTT.

2.5.2.1. HTTP

Es un protocolo cliente/servidor que sigue el esquema de petición y respuesta. El cliente es el encargado de realizar la petición al enviar un mensaje con determinado formato, posteriormente el servidor envía un mensaje de respuesta. En el entorno del IoT este protocolo se implementa con un solo cliente en el dispositivo, por lo que solamente tiene la posibilidad de realizar peticiones a un servidor web sin recibir peticiones de conexión ya que no posee un servidor implementado. Con este proceso se pretende aumentar la seguridad de la red, ya que las máquinas externas no pueden conectar con los dispositivos de la instalación IoT. El protocolo HTTP es perfecto para ser implementado situaciones en las que es necesario transmitir grandes cantidades de datos. La actual industria ya conoce y tiene experiencia trabajando con el protocolo HTTP en cuanto a la configuración de productos y dispositivos, pero no para acceder a los datos, por lo que muchas de las empresas utilizan este protocolo para proveer y recibir información. Se recomienda usar HTTP para enviar gran cantidad de información en lapsos de tiempo cortos [47] [48].

2.5.2.2. MQTT

Cola de mensajes telemetría y transporte, consiste en un protocolo basado en el método publicar/-suscribir donde el cliente se comunica de forma directa con el punto final, es decir está basado en la comunicación M2M, el método publicar/suscribir envía un mensaje desde el cliente hacia otro o varios suscriptores que reciben el mensaje. Es un protocolo simple y fácil de implementar, el cual requiere de pocos recursos en cuanto a procesamiento y ancho de banda se refiere. Así como HTTP, MQTT es específica para su aplicación, donde la mayoría de las implementaciones usan un formato JavaScript Object Notation (JSON) personalizado o binario.

Como MQTT sigue el modelo publicar/suscribir los clientes no se conocen unos a otros, por lo que cada uno de ellos solamente puede conocer la dirección y el puerto bróker, lo que permite que la comunicación entre los clientes sea asíncrona, dando lugar a una comunicación denominada *near-realtime*. El bróker cumple la función de redirigir los diferentes mensajes a los clientes según la suscripción de los tópicos [47] [48]. En la figura 2.7 se puede observar un esquema simple que describe el funcionamiento de este protocolo.

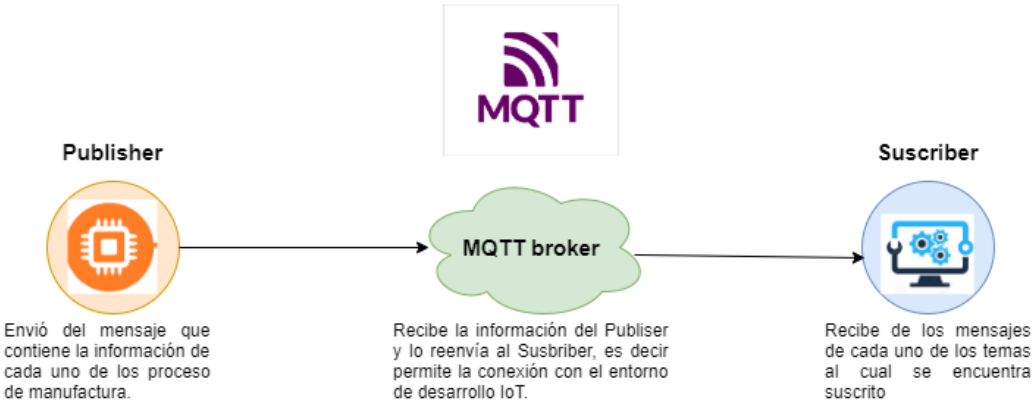


Figura 2.7: Diagrama de funcionamiento del protocolo MQTT.

2.6. Entornos de desarrollo IIoT

2.6.0.1. Thinger.io

Thinger.io es una plataforma *open source* para su implementación en la IoT, en vista de que permite gestionar diferentes tipos de dispositivos los cuales van a estar conectados a un servidor en donde se almacenan los datos que son enviados mediante la interconexión digital. Entre sus funciones principales se puede mencionar:

- **Almacenar y visualizar información recibida desde sensores:** Se puede almacenar y gestionar los datos de manera gráfica para que de esta manera sea mas sencillo interpretar estos y poder generar informes.
- **Enviar información o instrucciones a dispositivos:** Permite enviar información desde la plataforma hacia los dispositivos para poder conectarlos de manera remota.
- **Interactuar con aplicaciones como IFTTT:** Existen extensiones o *endpoints* que facilitan las conexiones con servicios de almacenamiento como Google, Dropbox y demás [49].

2.6.0.2. AWS

Amazon Web Services es un conjunto de herramientas de *cloud computing* que engloba una gran cantidad de servicios que permiten gestionar almacenamiento de datos, instancias virtuales, desarrollo de aplicaciones móviles y demás. Amazon ha sido una de las plataformas de más fidelidad debido a que sus servicios son usados por grandes empresas como Netflix, NASA, CIA, etc, todo esto debido al gran abanico de herramientas disponibles [50]. Algunas de las herramientas que ofrece Amazon Web Services son:

- ***Cloud computing:*** Creación de instancias para el almacenamiento y escalamiento de estas.
- Bases de datos: Diferentes tipos de bases de datos para el almacenamiento de la información, bases de datos como MySQL, DynamoDB, SQL Server, etc.
- **Aplicaciones empresariales:** Amazon web services ofrece un servicio de correo empresarial, en el cual se pueden trabajar conjuntamente con otros servicios como son Amazon Workdocs y Amazon Workspaces.
- **Internet de las cosas:** Permite establecer conexiones con diferentes tipos de dispositivos conectados a internet para a posterior poder analizar los datos que envían estos.
- **Herramienta de desarrolladores:** Es una herramienta que permite a los desarrolladores y programadores almacenar código para poder implementarlo de manera automática y poder luego de esto publicarlo [50].

2.7. Gemelo Digital

Un gemelo digital puede definirse como la representación digital de un producto, servicio o proceso. Su finalidad consiste en conseguir un entorno digital que represente una realidad física y permita la toma de decisiones para mejorar la eficiencia, productividad y competitividad de una fábrica. En este contexto, resulta importante mencionar que los gemelos digitales no deben ser entendidos como modelos aislados. Una de sus principales ventajas es la predicción de un entorno virtual, de esta manera

se logra un ahorro de costes y tiempo, lo que se traduce a una mayor eficiencia, productividad, calidad y beneficios económicos. Por otro lado, si se introducen cambios o fallas relevantes en el sistemas, el gemelo logra adelantarse a posibles accidentes.

Un gemelo digital puede ser aplicado en una planta, creando una copia perfecta de lo que pueda suceder en la realidad, de esta manera, es posible analizar casos y retornar los resultados al mundo físico. Independientemente del caso que se desee representar, el gemelo siempre se implementa a partir de cuatro niveles: nivel de sensor, nivel de dispositivo, nivel de control y nivel de aplicación empresarial. Para su implementación es necesario una plataforma tecnológica, en la que se pueda identificar las siguientes características [51]:

- Almacenamiento basado en tecnologías *big data*.
- Soporte OPC-UA / MQTT.
- Integrable a plataformas de capturas de datos.
- Almacenamiento de toda la información en base a modelos normalizados basados en estándar de la ndustria (ISA95).
- Permitir la conexión con sistemas MES y ERP para integrar procesos productivos.

2.8. Normativa ANSI/ISA S5.1

El estándar ANSI/ISA S5.1 es uno de los estándares de la ISA más utilizado durante la ingeniería de diseño de plantas químicas en la realización de planos y documentos; por ejemplo, en diagrama de tubería e instrumentación (DTI); en índice de instrumentos; en diagramas de lazo; en el diseño de gráficos dinámicos para el monitoreo y control digital de los sistemas distribuidos, y sobre todo en sistemas de seguridad (PLC), que va de la mano con Ciberseguridad, etc. En él se establecen los lineamientos para representar e identificar los instrumentos o dispositivos y sus funciones inherentes, sistemas y funciones de instrumentación, así como su representación gráfica. [52].

2.8.1. Identificación de la instrumentación

La identificación de un instrumento es una combinación de literales y números en donde a cada instrumento o función se le designa un código alfanumérico o número de identificación, de tal manera que de izquierda a derecha se tienen las literales y enseguida la numeración designada. La figura 2.8 muestra la identificación de un proceso de un indicador de visualización de nivel.

Figura 2.8: Identificación de un indicador de visualización de nivel .

El estándar presenta una tabla con distintos códigos para la identificación de algunos procesos. Sin embargo, este punto no pertenece a los objetivos del presenta trabajo, por lo que se deja a disposición del lector un referencia en la cual podrá informase más al respecto.

2.8.2. Numeración de los instrumentos

El estándar específica que para la identificación numérica de los instrumentos pertenecientes a un lazo de control es conveniente que la instrumentación asociada a ese lazo (abierto o cerrado) tenga esa misma numeración, es decir si en el lazo existe una válvula de control de presión con identificación PV-105, la instrumentación asociada deberá tener la numeración 105, por ejemplo el transmisor, el controlador y el indicador de presión local [52].

Capítulo 3

Diseño de planta productora de zumo de naranja e implementación de proceso de automatización y protocolos de comunicación heterogéneos

Este capítulo describe la implementación del modelo de la pirámide de automatización, la misma que será utilizada para el desarrollo de la planta. Para esto se especifican los procesos y subprocesos que se desarrollan en cada una de las capas por las cuales está conformado toda la línea de producción de la planta. Adicionalmente se presenta la manera en la que fueron implementados los protocolos de comunicación usados para el envío de información a la nube como para la comunicación interna de la planta y *middleware*.

3.1. Descripción del modelado de la planta de producción

Existen varios procesos que han sido industrializados con el paso de los años, todo esto debido al gran desarrollo de la tecnología. La mayoría de productos para consumo humano en la actualidad son producidos con la ayuda de una fábrica que ha sido automatizada con el objetivo de aumentar sus niveles de producción para así poder cubrir las necesidades del mercado. No solamente las grandes empresas han visto la necesidad de implementar una línea de producción, sino que también pequeñas empresas ahora emplean este tipo de tecnologías para así poder estar en un nivel competitivo con las grandes empresas.

Para el caso de estudio presente se analizó la línea de producción de zumo de naranja, este proceso comienza desde la siembra y cosecha del producto. Además del proceso de transporte en furgones refrigerados que van a permitir que se mantenga la temperatura del producto para así evitar que se dañe. Lo que se emula dentro de la planta es el proceso posterior a la llegada de la naranja, es así que los procesos realizados constan de cuatro etapas fundamentales que son:

- **1.- Llegado y limpieza primaria:** El producto será trasladado mediante bandas transportadoras hacia una zona de lavado de rodillos giratorios y mangueras.

- **2.- Selección y limpieza secundaria:** Una vez culminado el proceso de lavado, se procede a seleccionar cada una de las naranjas en función a su tamaño (grande, mediana y pequeña) con ayuda de un sensor óptico.

- **3.- Exprimido y mezclado:** Una vez finalizado el proceso de selección, cada uno de los productos primarios se dirigen hacia trituradores donde las naranjas son exprimidas y su jugo es vertido en tanques de mezclado. Un controlador accionará distintas válvulas de llenado y vaciado, en función a un nivel instrumentado por un sensor.

- **4.- Embotellado y empaquetado:** En esta ultima sección se embotella y empaqueta el producto y de manera similar a la fase 2, se clasifican cada una de las cajas de acuerdo a su tamaño.

  La figura 3.1 muestra el modelo de la planta en 3D realizado dentro de Factory I/O, en donde se aprecia cada uno de los procesos descritos.

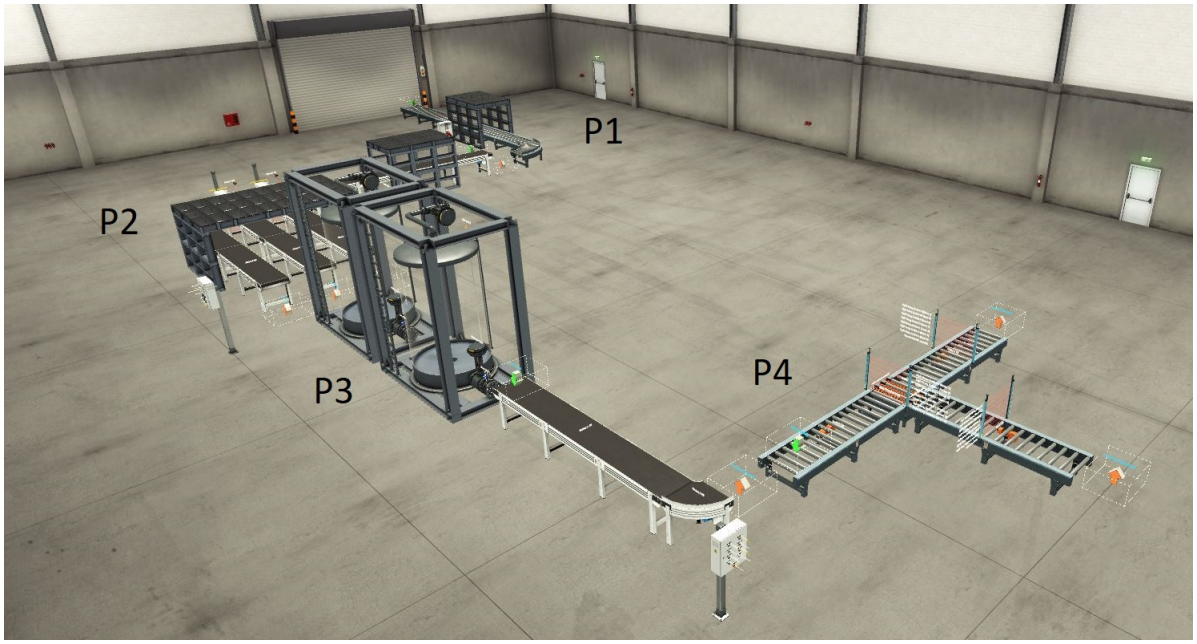


Figura 3.1: Modelado de la planta de producción de zumo de naranja

3.2. Diagrama de instrumentación

Todo proceso industrial necesita ser esquematizado mediante un diagrama de instrumentación. De esta forma, facilita al lector acerca de la distribución y conexión de cada uno de los componentes que se ven involucrados en el proceso de automatización industrial. Actualmente, existen numerosos estándares para nombrar e identificar los instrumentos que interviene en la línea de manufacturación. En este trabajo se usa la norma ISA/ANSI 5.1, en vista de que presenta un gran número de herramientas para el entendimiento e identificación de la instrumentación a cualquier nivel. La figura 3.2 muestra el diagrama de instrumentación de la planta de producción de zumo de naranja.

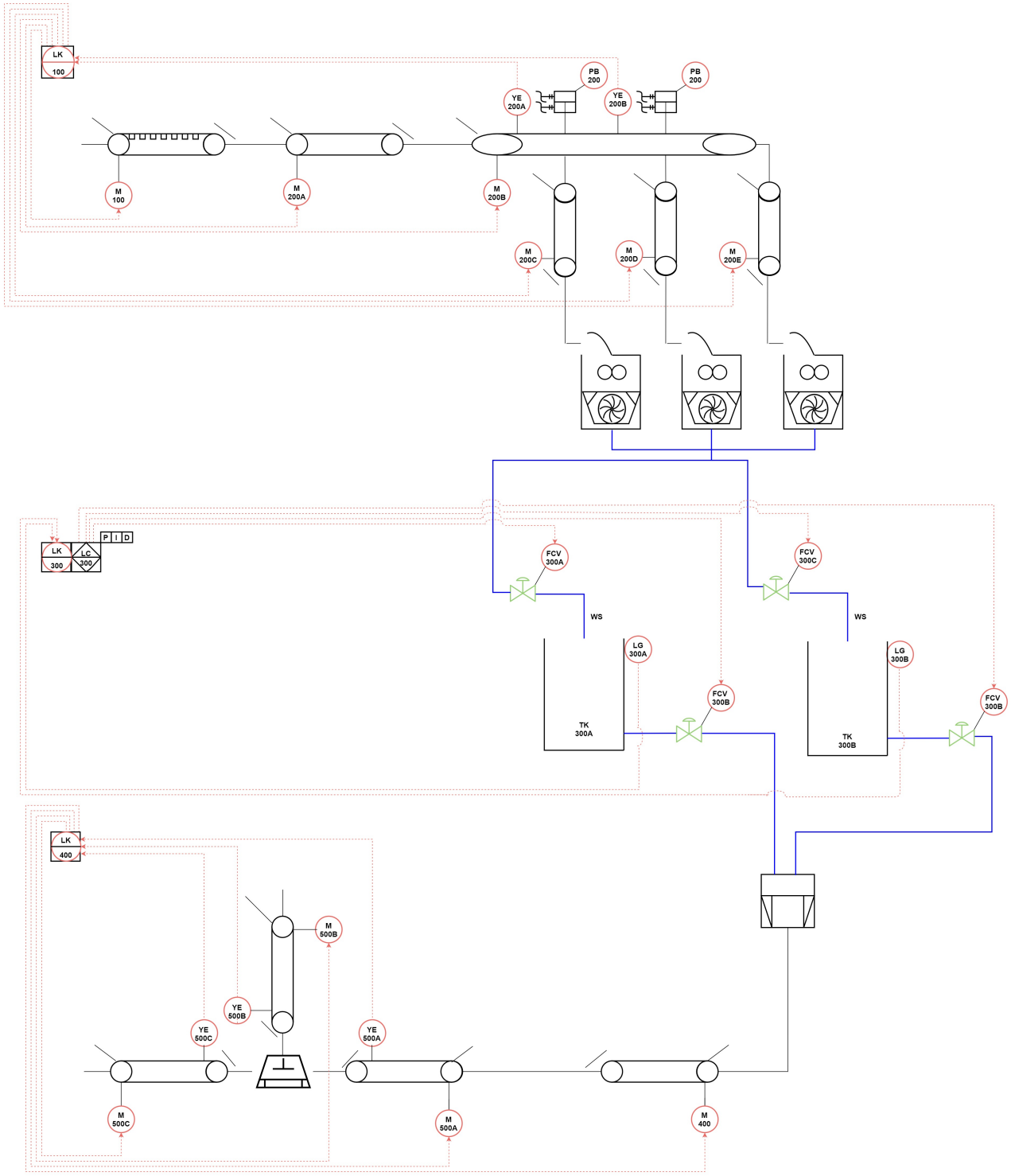


Figura 3.2: Diagrama de instrumentación del proceso de manufactura

El proceso industrial que se describirá más adelante, emplea una gran cantidad de maquinaria para cada una de las etapas. En consecuencia, en las tablas 3.1 y 3.2 se detalla la categoría, descripción y simbología de cada uno de ellos.

Tabla 3.1: Simbología de instrumentación - Parte I.

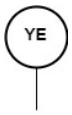
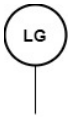
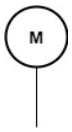
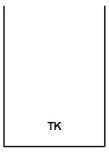
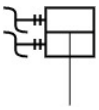
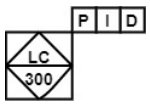
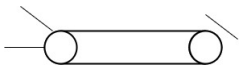
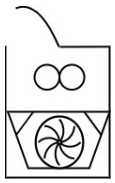
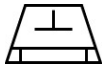
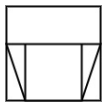
Simbología de instrumentación		
Categoría	**Descripción**	**Simbología**
Elementos primarios	State Sensor (YE) montando en campo	
	Level Viewing Device (LG) montando en campo	
	Motor montando en campo	
Diagramas de tuberías e instrumentación	Flow Valve Control (FVC)	
	Tanque de almacenamiento	
Actuadores	Pressure Burner (PB)	
Control compartido	Control Station (LK)-Ubicación accesible normalmente al operador.	

Tabla 3.2: Simbología de instrumentación - Parte II.

Simbología de instrumentación		
Categoría	**Descripción**	**Simbología**
Controlador lógico programable	Level Control (LC)-Ubicación accesible normalmente al operador.	
Componentes adicionales	Banda transportadora con rodillos giratorios.	
	Banda transportadora lisa	
	Exprimidor de naranjas industrial	
	Clasificador a presión	
	Empaquetador de jugo de naranja	

3.3. Descripción de la arquitectura

El sistema está integrado por cuatro capas mencionadas anteriormente. Cada una de éstas realiza un proceso para la estructuración de la planta de producción de jugo de naranja. La capa base es la que se encarga de la simulación, la segunda se hará cargo del control, la tercera capa de la supervisión y la última capa es en donde se va a tener los datos de la planta almacenados en dos nubes de información como es Thinger.io y Amazon Web Service (AWS). La figura 3.3 muestra el modelo de la pirámide y las herramientas usadas para su implementación, así como los protocolos de comunicación implementados.

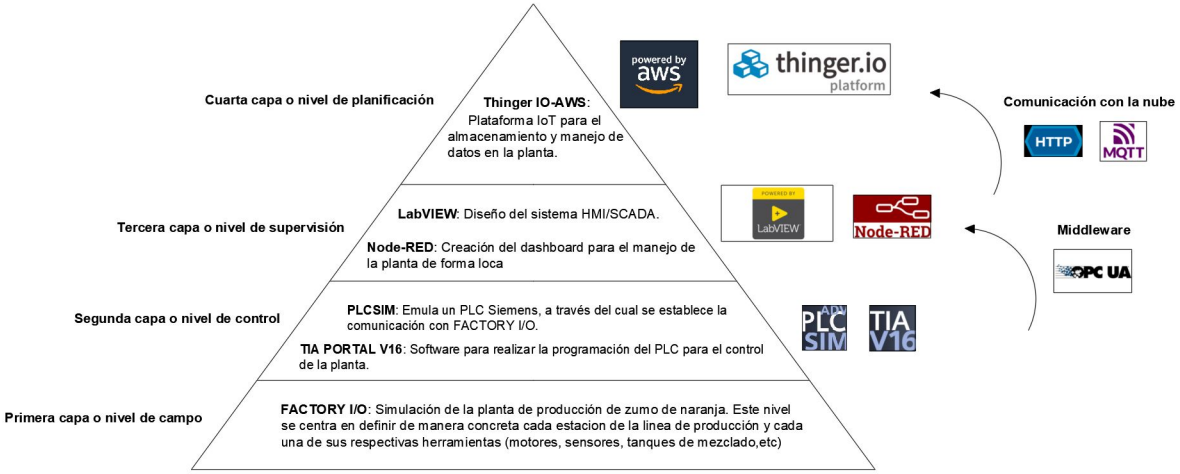


Figura 3.3: Pirámide de automatización para el proceso de manufactura de zumo de naranja.

3.3.1. Primera capa

La primera capa o nivel de campo, permite la interacción entre los dispositivos físicos de la fábrica como motores, bandas, sensores y actuadores. Para la simulación de este nivel se usó el software Factory I/O.

3.3.2. Segunda capa

La segunda capa o nivel de control, permite el uso de dispositivos encargados del control de la línea de manufactura, es decir el PLC. Como se trata de un ambiente simulado, se necesita de una herramienta que emule el dispositivo dentro del ordenador, para ello, se empleó el software PLCSIM. Posteriormente, se procede a enlazar los respectivos puertos a través de los drivers de Factory I/O, con el objetivo de automatizar la factoría virtual a través del PLC del software PLCSIM. La programación del PLC se desarrolla utilizando programación en escaleras (*ladder diagram*) en combinación con programación estructura (ST) en el ambiente TIA Portal version 16.

3.3.2.1. Configuración de drivers dentro de Factory I/O

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 obtiene la automatización de todos los procesos.

Al momento de escoger el driver en Factory I/O se debe seleccionar el driver *Siemens S7-PLCSIM*, que se enlaza con el emulador del PLC. La figura 3.4 muestra la configuración realizada para el driver dentro de Factory I/O. El PLC seleccionado corresponde al modelo Siemens S7-1200, el mismo que gracias a sus características de procesamiento es adecuado para la implementación de los procesos involucrados en la planta.

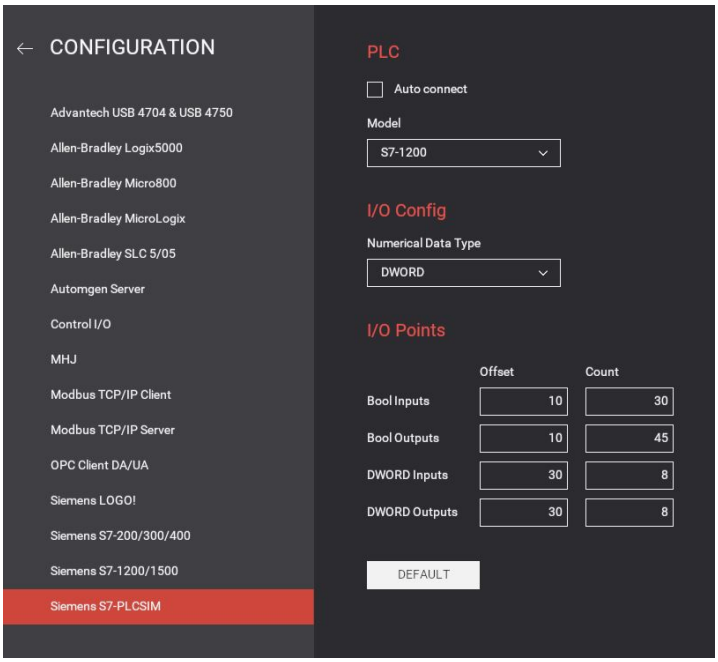


Figura 3.4: Configuración del driver *Siemens S7-PLCSIM*

3.3.2.2. Criterios de automatización de la planta dentro de TIA PORTAL V16.

La automatización y control de procesos emulados dentro de Factory I/O son programados dentro de TIA PORTAL V16. En las etapas de transporte, selección y lavado están involucrados motores, bandas, actuadores y sensores los cuales envían información hacia el PLC para su control. En la etapa de exprimido y mezclado se implementa un sistema multitanque controlado por un PID logrando la estabilización del llenado de tanques.

Proceso 1: Llegado y limpieza primaria

El primer proceso está conformado por tres bandas lineales y una banda tipo codo, la figura 3.5 muestra la configuración de bandas descrita. Para la programación de este proceso únicamente se accionaron los motores de cada una de las bandas y fueron controlados con pulsantes, los mismos que se encuentran ubicados dentro del panel de control de la planta.

Figura 3.5: Diseño del primer proceso dentro de Factory I/O.

La programación del accionamiento de los motores del primer proceso dentro de TIA PORTAL se muestra en la figura 3.6, TIA PORTAL V16 permite la programación en un ambiente gráfico de tipo escalera.

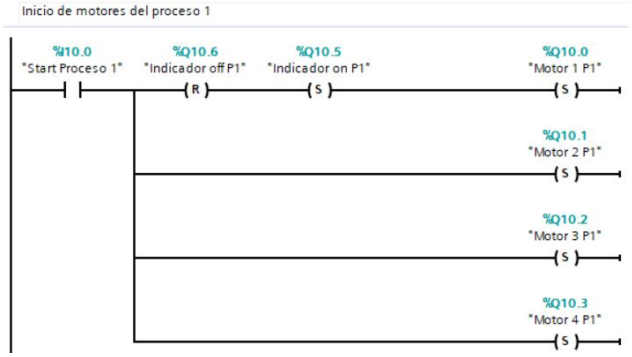


Figura 3.6: Programación de *start* del primer proceso dentro de TIA PORTAL.

Cada uno de los actuadores están etiquetados con una variable de entrada y una de salida. Este etiquetado se realiza para todos los actuadores dentro de la pestaña "Variables PLC". La figura 3.7 muestra las variables que han sido etiquetadas para todo el proceso de automatización de la planta. Sean espacios de memoria o actuadores deben de estar correctamente etiquetados para que se pueda trabajar con cada una de ellas. Los sensores y actuadores deben de estar marcados con las mismas direcciones que en el software Factory I/O, para que estos actúen con normalidad y se pueda controlar a través del PLC, caso contrario, los actuadores y sensores no van a recibir ningún tipo de datos desde la plataforma. La figura 3.8 muestra la asignación de variables dentro de Factory I/O.

Variables PLC

	Nombre	Tabla de variables	Tipo de datos	Dirección	Rema...	Acces...	Escrib...	Visibl...
1	Start Proceso 1	Standard-Variablen...	Bool	%I10.0		☑	☑	☑
2	Start Proceso 2	Standard-Variablen...	Bool	%I10.1		☑	☑	☑
3	Stop Proceso 1	Standard-Variablen...	Bool	%I10.2		☑	☑	☑
4	Stop Proceso 2	Standard-Variablen...	Bool	%I10.3		☑	☑	☑
5	Emergency Stop 1	Standard-Variablen...	Bool	%I10.4		☑	☑	☑
6	Emergency Stop 2	Standard-Variablen...	Bool	%I10.5		☑	☑	☑
7	Emergency Stop 3	Standard-Variablen...	Bool	%I10.6		☑	☑	☑
8	Motor 1 P1	Standard-Variablen...	Bool	%Q10.0		☑	☑	☑
9	Motor 2 P1	Standard-Variablen...	Bool	%Q10.1		☑	☑	☑
10	Motor 3 P1	Standard-Variablen...	Bool	%Q10.2		☑	☑	☑
11	Motor 4 P1	Standard-Variablen...	Bool	%Q10.3		☑	☑	☑
12	Sirena	Standard-Variablen...	Bool	%Q10.4		☑	☑	☑
13	Memoria Sirenas	Standard-Variablen...	Bool	%M10.0		☑	☑	☑
14	Indicador on P1	Standard-Variablen...	Bool	%Q10.5		☑	☑	☑
15	Indicador off P1	Standard-Variablen...	Bool	%Q10.6		☑	☑	☑
16	Sensor 1 P2	Standard-Variablen...	Bool	%I10.7		☑	☑	☑
17	Sensor 2 P2	Standard-Variablen...	Bool	%I11.0		☑	☑	☑
18	Motor 1 P2	Standard-Variab...	Bool	%Q10.7		☑	☑	☑
19	Motor 2 P2	Standard-Variablen...	Bool	%Q11.0		☑	☑	☑
20	Motor 3 P2	Standard-Variablen...	Bool	%Q11.1		☑	☑	☑
21	Motor 4 P2	Standard-Variablen...	Bool	%Q11.2		☑	☑	☑
22	Motor 5 P2	Standard-Variablen...	Bool	%Q11.3		☑	☑	☑
23	Motor 6 P2	Standard-Variablen...	Bool	%Q11.4		☑	☑	☑
24	Motor 7 P2	Standard-Variablen...	Bool	%Q11.5		☑	☑	☑
25	Motor 8 P2	Standard-Variablen...	Bool	%Q11.6		☑	☑	☑

Figura 3.7: Tabla de etiquetas de variables en TIA PORTAL

Figura 3.8: Tabla de etiquetas de variables en Factory I/O

Proceso 2: Llegado y limpieza primaria

El segundo proceso de la planta está conformado por dos bandas principales más tres bandas que se encuentran ubicadas de manera perpendicular a la banda principal, la figura 3.9 muestra la configuración de bandas para el segundo proceso . Este proceso se encarga del transporte de las naranjas ya seleccionadas las cuales son transportadas a través de un *pusher* mecánico se acciona de acuerdo a la lógica de un sensor óptico.

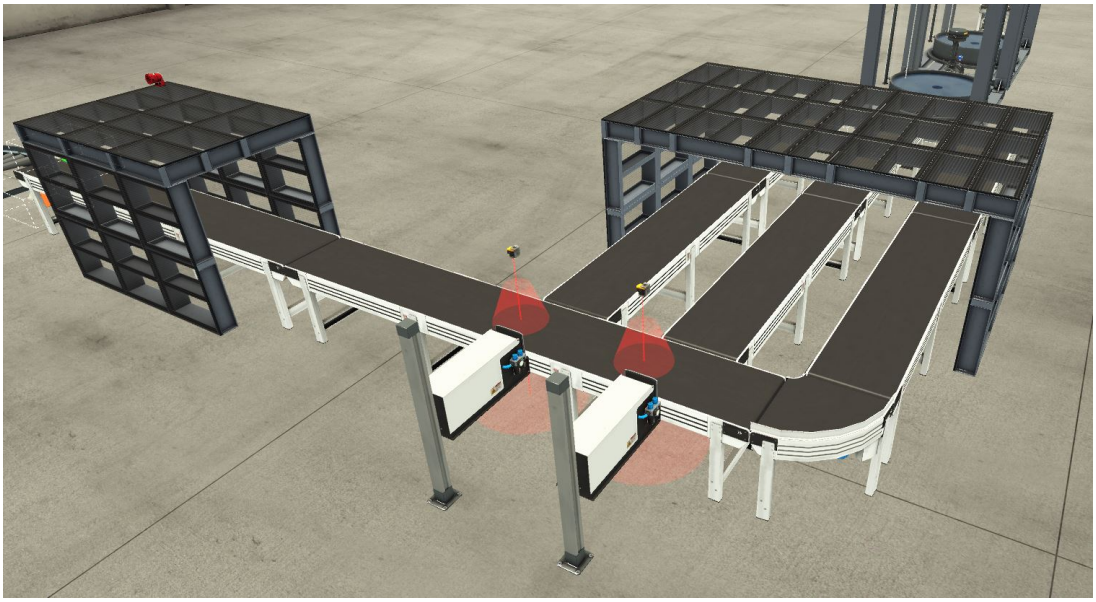


Figura 3.9: Diseño del segundo proceso dentro de Factory I/O.

Proceso 3: Exprimido y mezclado

El tercer proceso está conformado por dos tanques que emulan el proceso de exprimido y mezclado de líquidos para la producción de zumo de naranja. La figura 3.10 muestra el sistema multitanque implementado en la planta el cual tiene dos válvulas de llenado y dos de vaciado. Para controlar los procesos de los tanques se optó por implementar un controlador PID. De esta manera se mantiene bajo control el nivel de referencia asegurando que se encuentre en los límites de funcionamiento.

Figura 3.10: Diseño del tercer proceso dentro de Factory I/O.

Al momento que un *setpoint* es registrado, éste pasa por un proceso de normalización cuya salida es un valor comprendido entre el rango de 0 a 300. Este valor hace referencia a la capacidad máxima del tanque en litros. De igual manera el sensor de nivel dentro de los tanques pasa por un proceso de normalización el cual permite controlar el nivel hasta el valor límite detallado anteriormente. La figura 3.11 muestra un diagrama de flujo el cual detalla el proceso de normalización de las dos variables de control para los tanques.

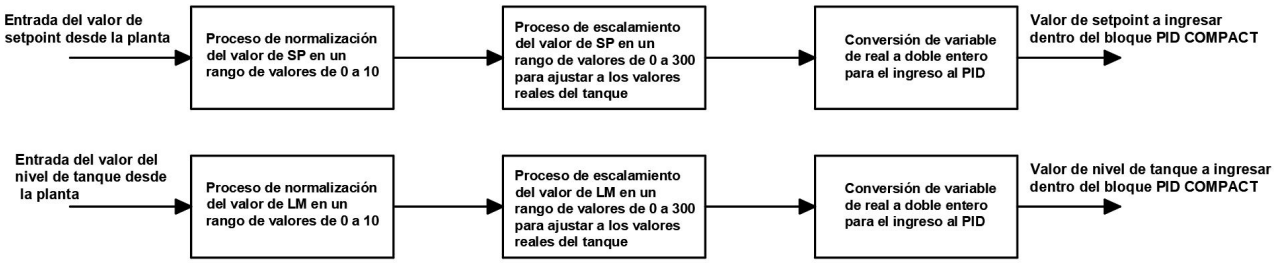


Figura 3.11: Normalización y conversión de variables para el control de los tanques

Luego de que hayan sido normalizadas las dos variables, éstas ingresan dentro del bloque *PID Compact*, el mismo que realiza la funcionalidad del controlador PID. Una vez se tenga la salida del PID se debe pasar esta variable por un proceso de normalización, ya que dentro del Factory I/O el valor máximo de salida de las válvulas de agua es 10. La figura 3.12 muestra un diagrama de bloques en el cual se detalla el ingreso de variables y el proceso de normalización de la salida del PID para el control de los tanques en la planta. Para del controlador se realizó una configuración manual, mediante una metodología de prueba y error, hasta que los valores de salida sean exactos a los valores de la planta. De esta manera, se programa un proceso de control eficiente. La misma metodología se repite para el segundo tanque.

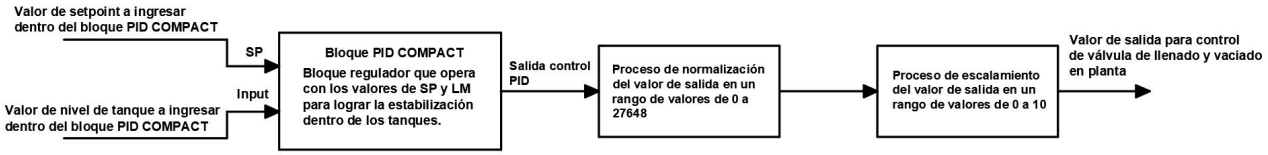


Figura 3.12: Operación del bloque PID COMPACT.

Proceso 4: Embotellado y Empaquetado

Para el proceso 4 se optó por el uso de bandas al igual que la primera etapa. El proceso de embotellado consta de 2 bandas, junto con una banda codo por la cual circulan las piezas de metal que emulan las botellas. La figura 3.13 muestra la configuración de bandas para el proceso de embotellado. Para la programación de este proceso únicamente es necesario la activación de los actuadores para que los motores se accionen.

Figura 3.13: Diseño del proceso de embotellado dentro de Factory I/O.

Para la última parte de la línea de producción se optó por el uso de bandas y sensores para el conteo y clasificación de cajas acuerdo a su tamaño. La figura 3.14 muestra la estructura que se diseñó para el proceso de clasificación de cajas.

Figura 3.14: Diseño del proceso de empaquetado dentro de Factory I/O.

Para la programación de este proceso se optó por el uso de sensores para la clasificación y conteo de las cajas. Una vez que el sensor detecta una caja de tamaño grande se procede a la detención de los motores, a excepción del motor de la banda de transferencia, debido a que de esta manera la caja se traslada a su banda correspondiente. Una vez que la caja pasa por un sensor de detección de movimiento los motores de las bandas se vuelven a accionar para que la siguiente caja sea clasificada. Para el caso de las cajas de tamaño pequeño únicamente las bandas siguen su proceso normal así clasificándolas al final de la banda. La figura 3.15 muestra el diagrama de bloques implementado en TIA PORTAL V16 para el proceso de empaquetado.

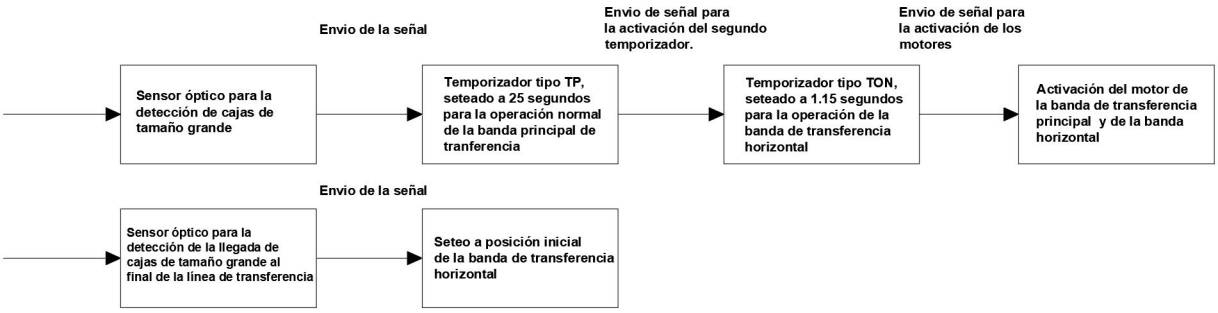


Figura 3.15: Diagrama de bloques de la programación del proceso de empaquetado dentro de Factory I/O.

3.3.3. Tercera capa

La tercera capa corresponde a los sistemas de supervisión, control y adquisición de datos. Para este nivel se optó por el uso de los softwares LabVIEW y Node-RED. Esta capa está enlazada a través del *middleware* de la planta en el cual se ejecuta el protocolo de comunicación OPC con la ayuda de las herramientas NI OPC SERVERS y NettoPLCSim. Estos permiten la ejecución de un servidor, que realiza la lectura y adquisición de las variables y los datos desde el PLC.

3.3.3.1. Creación de servidor OPC y enlace de cliente

La capa que permite la adquisición de los datos es conocida como *middleware*, en esta capa se implementó el protocolo de comunicación OPC, que fue descrito con anterioridad en este documento. Este servidor permite que todos los datos que están en el PLC sean extraídos para realizar su monitorización y manipulación.

Para la creación del servidor se usó la herramienta NetToPLCsim. Dentro del programa se deben asignar direcciones IP para la creación del servidor. La figura 3.16 muestra la configuración del servidor OPC. Existen dos direcciones IP las cuales deben de ser asignadas, la primera es la dirección del PLC y la segunda es donde se ejecuta el servidor.

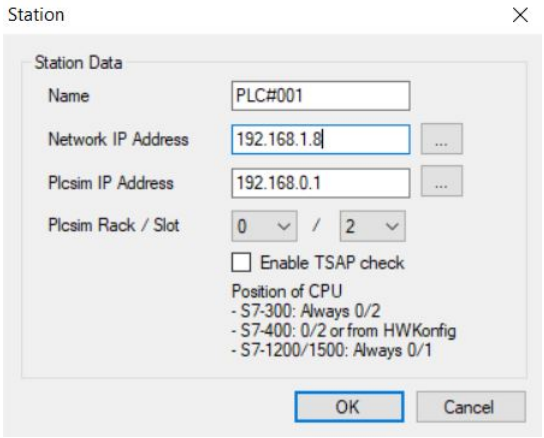


Figura 3.16: Asignación de direcciones para el servidor OPC.

La dirección del PLC es asignada por defecto dentro de TIA PORTAL. En este caso la herramienta arroja la dirección 192.168.0.1, la figura 3.17 muestra la ejecución del PLC dentro del software PLCSIM. La dirección de donde va a estar alojado el servidor OPC varía dependiendo de la red a la que se encuentra conectado el computador.

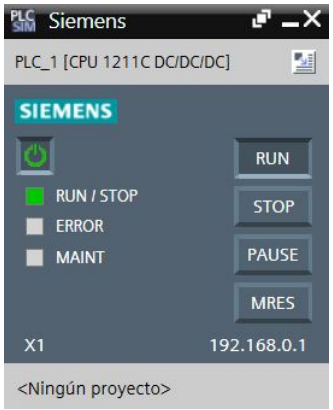


Figura 3.17: Emulación del PLC dentro del servidor

Cuando el servidor se ejecuta con normalidad se establece una conexión entre el cliente y el servidor para obtener los datos que están en el mismo. Para ello, se usó el software NI OPC Servers, y se creó

un nuevo dispositivo con el nombre de "Tesis". En la creación del dispositivo se debe de asignar su respectivo modelo y la dirección a la que se suscribe. La figura 3.18 muestra el proceso de creación y suscripción de un nuevo dispositivo dentro de NI OPC Servers.

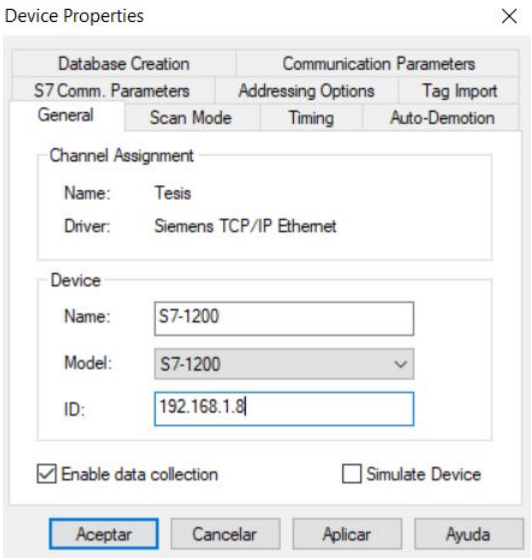


Figura 3.18: Asignación de dirección IP para la suscripción al servidor OPC.

Una vez creado el dispositivo se deben etiquetar las variables para su supervisión. La dirección de las variables son las mismas que fueron asignadas con anterioridad dentro de TIA PORTAL y Factory I/O. Una vez creada cada una de las etiquetas se puede comprobar que la lectura de las variables sea correcta. Para ello, se usa la herramienta OPC Quick Client que permite observar el estado y el valor de las variables que fueron ingresadas para su lectura. La figura 3.19 muestra la lectura en tiempo real de las variables que fueron ingresadas para la supervisión.

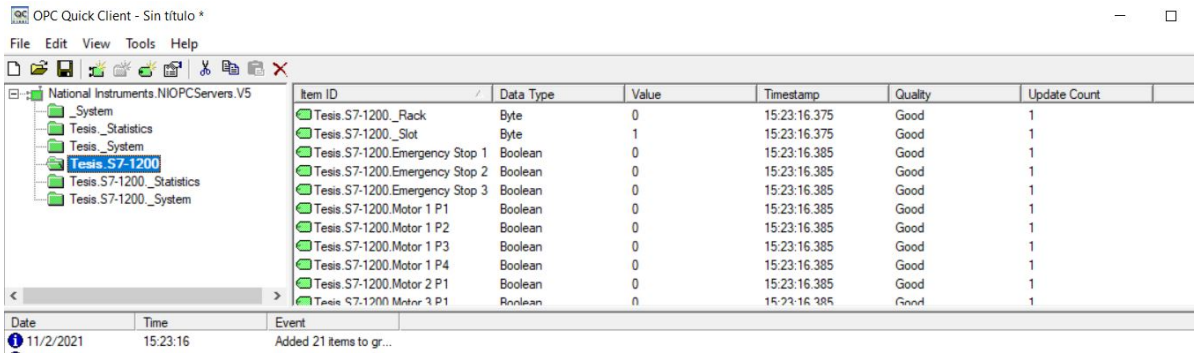


Figura 3.19: Software OPC Quick Client usado para la supervisión de variables.

3.3.3.2. Diseño e implementación del sistema HMI / SCADA

Para la implementación del sistema HMI/SCADA se utilizó el software LabVIEW. Dentro de la herramienta se crea una nueva instancia I/O, en donde se encuentran alojadas todas las variables del servidor. Posteriormente, se ingresa cada una de las variables dentro de LabVIEW para manipularlas e inmediatamente se ingresa al panel *Create Bound Variables*, el cual permite la importación de las

variables que se necesite dentro del sistema.

Luego de que se hayan ingresado todas las variables se procede a realizar el diseño del HMI desde un VI el cual va a permitir el control de la planta, el estado de los tanques y los procesos sin necesidad de ingresar a Factory I/O. La figura 3.20 muestra la implementación de un *layout* dentro del sistema HMI/SCADA con el cual se tiene una visualización completa del sistema y conocimiento del estado de cada una de las etapas de la planta, además dentro del panel se puede observar indicadores de alarmas lo cual es importante para los operarios. Así al momento que se accione una alarma dentro de la planta los operarios van a poder dirigirse hacia la sección en donde ocurrió la falla.

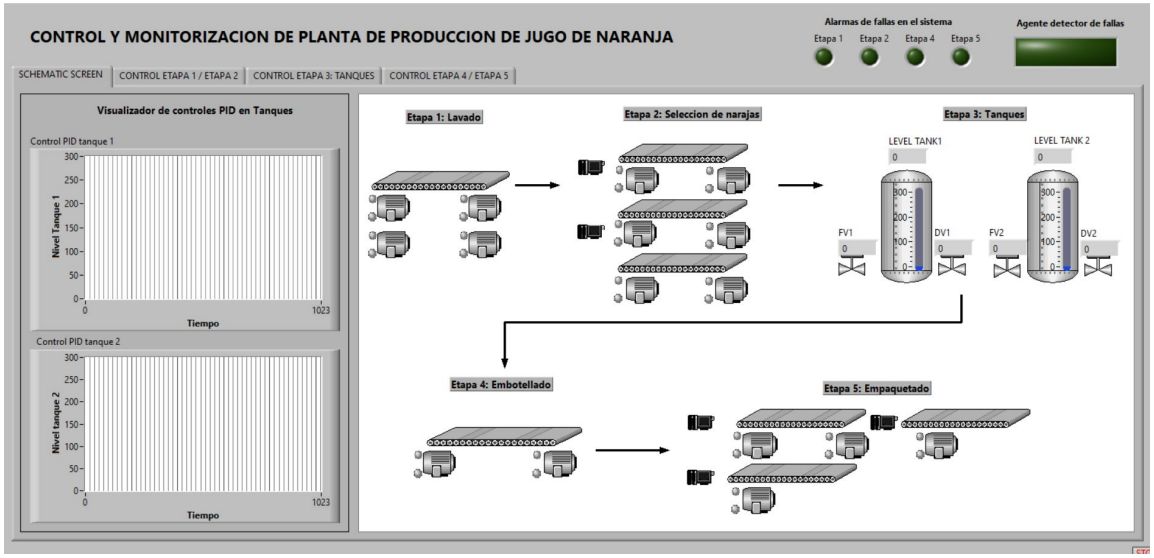


Figura 3.20: Diseño del HMI dentro de LabVIEW.

3.3.3.3. Lectura de variables y creación del *dashboard* en Node-RED

Node-RED es una herramienta de programación visual en donde se pueden añadir nodos de diferentes tipos de extensiones para así poder conectar dispositivos de hardware, API's y servicios de Internet. Todos estos tipos de extensiones y diversos nodos que están a disposición dentro de Node-RED permite que sean una herramienta dedicada para el uso dentro de la IIoT [53]. La funcionalidad de esta herramienta reacae en la facilidad del envió de información a la nube. Además de la implementación de un *dashboard* para el monitoreo local de la planta. Como se mencionó anteriormente, el protocolo de comunicación OPC permite obtener los datos desde el PLC, con el objetivo de controlar el conjunto de variables para la comunicación con la nube. Para ello, se procede con la instalación del nodo Siemens. La figura 3.21 muestra la configuración del nodo de Siemens dentro de Node-RED en donde se asigna la dirección del servidor OPC para la obtención de datos.

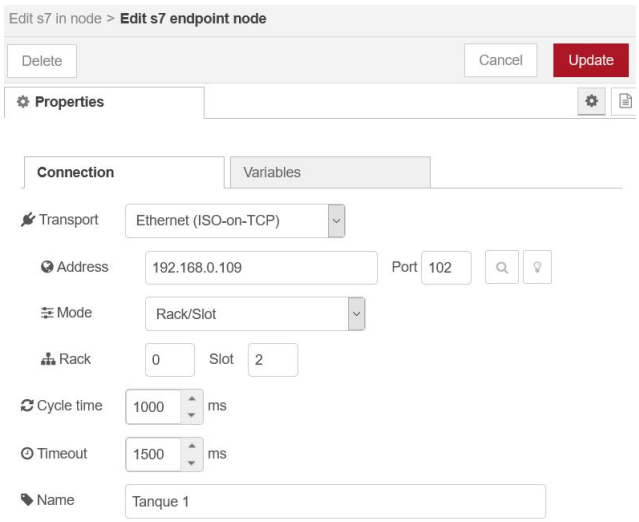


Figura 3.21: Configuración del nodo PLC dentro de Node-RED.

Para la lectura de las variables dentro de Node-RED es necesario la creación de un bloque de datos en TIA PORTAL, ya que, al momento del direccionamiento de las mismas resulta indispensable especificar el bloque de datos y tipo de variable al que pertenece cada uno de los datos que van a ser leídos. Las figuras 3.22 y 3.23 muestran la manera en la que deben de ser etiquetadas las variables tanto en Node-RED como en TIA PORTAL, en caso de que no sean etiquetadas de manera correcta Node-RED va a presentar un mensaje de error de conexión con el PLC.

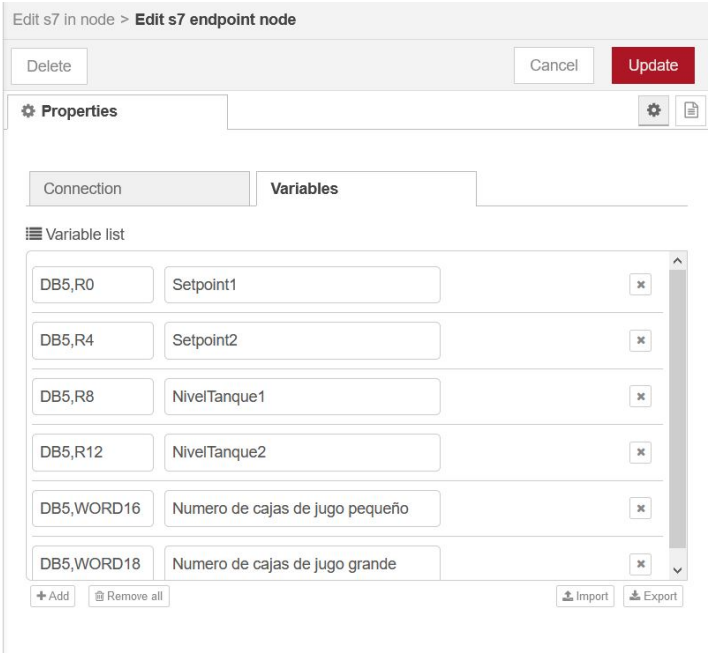


Figura 3.22: Configuración de variables dentro de Node-RED.

Bloque de datos_1

	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ...	Valor de a...
1	Static								
2	Setpoint1	Real	0.0	0.0					
3	Setpoint2	Real	4.0	0.0					
4	NivelTanque1	Real	8.0	0.0					
5	NivelTanque2	Real	12.0	0.0					
6	Caja transferida grande	Int	16.0	0					
7	Caja transferida peque...	Int	18.0	0					

Figura 3.23: Configuración de bloque de datos en TIA PORTAL.

Una vez que se tiene ingresado cada una de las variables, es necesario realizar la programación a través de los bloques disponibles en el *dashboard*. La figura 3.24 muestra el primer diseño del *dashboard*, en el cual se ingresa indicadores visuales con el objetivo de observar los niveles de los tanques. De igual manera, se visualiza la curva de control y niveles de producción de la planta. En este contexto, resulta importante mencionar que dentro del mismo entorno de desarrollo se implementará el concepto *digital twins* para los accionadores de las bandas de la planta y controladores de *setpoints*. De esta manera, se tiene un control desde un punto de vista administrativo.

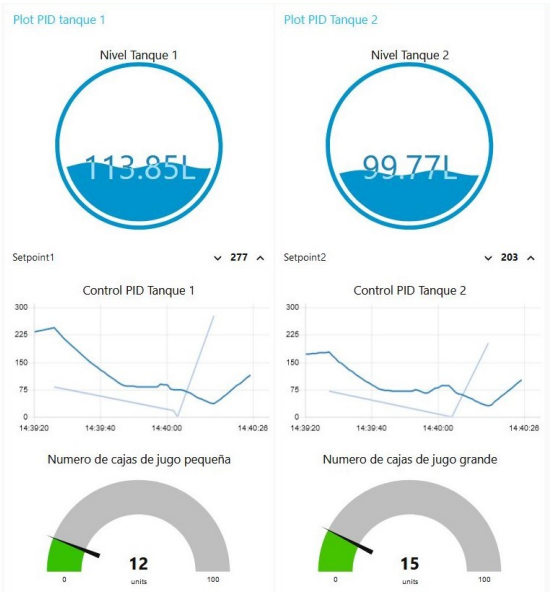


Figura 3.24: Configuración de *dashboard* en Node-RED.

3.3.4. Cuarta capa

La cuarta capa está conformada por nubes de información en donde se encuentran almacenados los datos de la fábrica. Para el envío de la información se usan dos protocolos de comunicación que son HTTP y MQTT. En este contexto resulta importante mencionar que se emplean dos protocolos con el objetivo de realizar un estudio comparativo entre ellos.

- Para HTTP se utilizó Thinger.io
- Para MQTT se utilizó AWS.

Se optó por el uso de Thinger i.o y AWS debido a la facilidad de manejo, almacenamiento y manipulación de datos para analizarlos de una manera gráfica y así poder tener un mayor entendimiento de los datos que están siendo almacenados. Además, un punto clave de selección fue la disposición de *"pluggins"* de acceso para los protocolos antes mencionados. De igual manera, se almaceno la información de manera local usando MySQL. El envío de datos se realizó con la ayuda de Node-RED. Las figuras 3.25 y 3.26 muestra el diagrama de flujo dentro de Node-RED y un diagrama explicativo el cual detalla el proceso y las dirección de envío de información hacia cada uno de los servidores de datos, además se implementa MySQL para el almacenamiento local de datos que va a ser necesario dentro del agente detector de fallas.

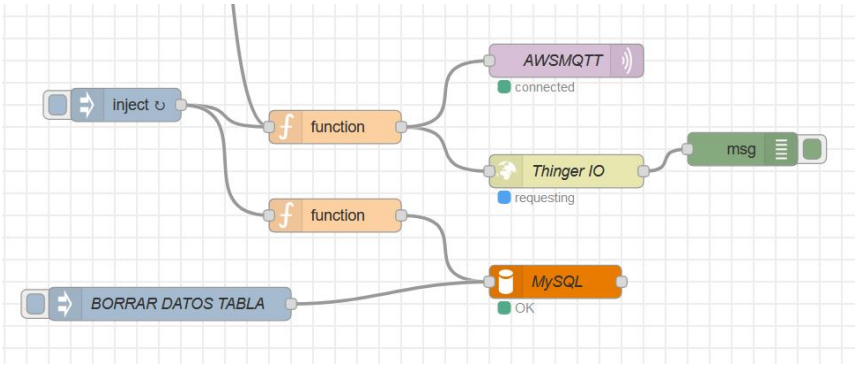


Figura 3.25: Programación en Node-RED para el envió de información.

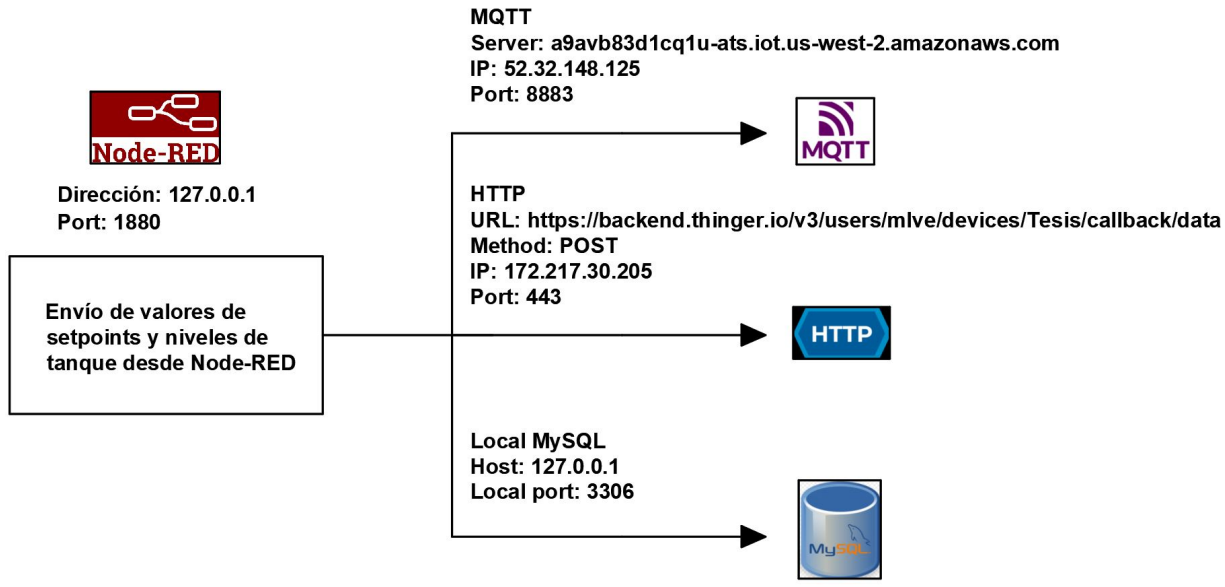


Figura 3.26: Envió de información desde Node-RED a las plataformas de almacenamiento de datos.

3.3.4.1. Agente detector de fallas

Para realizar la virtualización de servicios en la capa cuatro, se optó por la implementación de un agente detector de fallas. Dicho agente va a realizar la monitorización en tiempo real de los tanques de la planta, para así evitar cualquier tipo de error en la operación de los mismos. Las fallas que están

contempladas dentro del agente son fugas, accionamiento involuntario de válvulas y niveles incorrectos en los tanques.

El agente esta implementado en Node-RED, pero para ponerlo en funcionamiento fue necesario trabajar con un conjunto de datos reales de la línea producción, los mismos que fueron almacenados de manera local en una base de datos de MySQL. Una vez identificado los datos, se procede a elaborar un *script* en MATLAB, cuyo funcionamiento se explicará a continuación.

El agente trabaja con dos conjuntos de datos, el primer bloque corresponde a los datos de entrenamiento, mientras que el segundo hace referencia a un *set* de datos con errores. En este contexto, resulta importante mencionar que los datos con errores se los produce de manera intencionada con ayuda de interruptores para la acción involuntaria de las válvulas. Una vez identificado cada bloque de datos, se procede a obtener la media y desviación estándar respectivamente. De esta manera, se define un umbral de trabajo sobre el que se puede asegurar que los tanques funcionan sin ninguna falla. Para obtener el umbral de trabajo se necesita determinar la probabilidad con la cual un falso negativo puede ocurrir. Es decir, aquellos datos que se encuentran dentro del umbral de trabajo, pero presentan cualquier tipo de error. La probabilidad de ocurrencia de estos falsos depende de como se define los límites de la banda de umbral.

Las ecuación 3.1 describe la expresión matemática para el cálculo del límite superior en función a la media (μ) y la desviación estándar (σ). No obstante, se debe considerar un intervalo de confianza ($\alpha/2$), el cual permite que la cantidad de datos erróneos en el limite superior disminuya. La ecuación 3.2 muestra la operación contraría para el límite inferior

$$U_b = \mu + c_{\frac{\alpha}{2}}\sigma \tag{3.1}$$

$$L_b = \mu - c_{\frac{\alpha}{2}}\sigma \tag{3.2}$$

La tabla 3.3 muestra los resultados del error para distintos intervalos de confianza. Después de varias pruebas se optó por definir un intervalo de confianza del 0,00135, lo que implica, que la desviación estándar se multiplica por un valor constante de 3.

Tabla 3.3: Valores estándares para el calculo de error

$\alpha/2$	$C_{\alpha/2}$
0,00135	3,00
0,0025	2,81
0,005	2,58
0,01	2,33
0,025	1,96

3.3.4.2. Estadístico de Hotelling

Una vez identificada la banda de umbral se procede a definir la operación del agente detector de fallas, para ello se implementó el método del estadístico de Hotelling, el mismo que hace referencia a un método de identificación multivariable. La figura 3.27 muestra un diagrama que sigue el estadístico de Hotelling para estudiar los procesos aleatorios en los diferentes casos de las variables multivariantes.

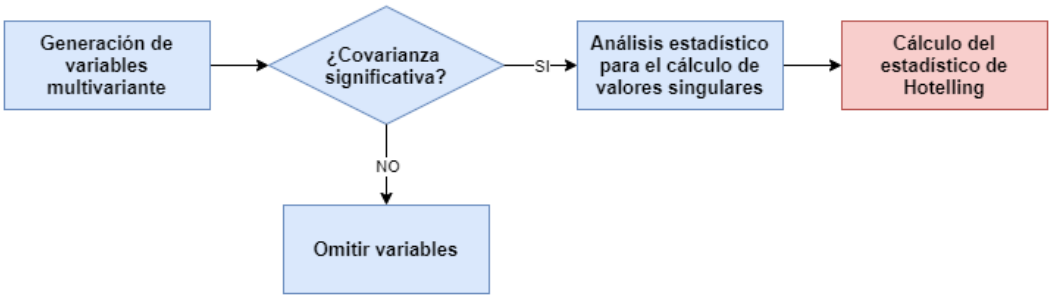


Figura 3.27: Diagrama para el cálculo del estadístico de Hotelling.

En primer lugar se calcula la matriz de convarianza en función al vector de medias μ. La ecuación 3.3 describe la expresión matemática para el cálculo de la matriz de covarianza S, donde n hace referencia al número de muestras y $\mathbf{X}$ corresponde al vector de medias.

$$S = \frac{1}{n-1} * X^T X \tag{3.3}$$

Posteriormente, con ayuda del comando ***svd*** (Singular Value Descomposition, por sus siglas en inglés) propio de MATLAB se procede a obtener cada uno de los valores singulares que conforman la matriz de convarianza, con el objetivo de descomponer el proceso aleatorio en vectores necesarios para el cálculo de la variación. Esta expresión se puede observar en la ecuación 3.4.

$$[U, \lambda, V] = svd(S) \tag{3.4}$$

Finalmente, a partir del cálculo anterior se considera a S como un matriz no singular y se usa la definición presentada en la ecuación 3.5 para así obtener el valor final del estadístico de Hotelling empleando la ecuación 3.6. En este contexto, resulta importante mencionar que el valor T^2, es directamente proporcional a la distancia entre el punto de observación y el valor objetivo de la muestra que se esta analizando.

$$z = \lambda^{-\frac{1}{2}} V^T X \tag{3.5}$$

$$T^2 = z^T z \tag{3.6}$$

El valor de T^2 controla los cambios en el vector de medias suponiendo que la matriz de convarianza del proceso es igual a la matriz de mediciones cuando el proceso esta bajo control, es decir, cuando no se presenta una falla en el sistema.

Capítulo 4

Resultados

En este capítulo se presentan los resultados obtenidos luego de haber desarrollado cada una de las secciones descritas en la metodología. Para ello, cada una de las etapas y capas de la pirámide de automatización se vieron expuestas a experimentación.

4.1. Análisis del diseño y modelado de la planta

Como se mencionó anteriormente, para el diseño de la fábrica se empleó el software Factory I/O. Actualmente, esta plataforma presenta una serie de ventajas respecto a otras herramientas que permiten la simulación de ambientes industriales, tales como son PROMODEL, FREECAD, ARENA,etc. No obstante, también presenta una serie de desventajas y sobre todo limitaciones. El diseño de la planta se basó en la línea de producción de jugo de naranja. Sin embargo, en las etapas de lavado, selección y empaquetado se presentaron ciertos inconvenientes, relacionados principalmente con la maquinaria que la herramienta brinda para la simulación de todo tipo de proceso industrial. Por otro lado, resulta importante recalcar que dicha limitación no presenta inconvenientes con la interconexión de cada uno de los protocolos de comunicación.

La figura 4.1 muestra la limitación que existe en la primera etapa del proceso, en este caso lo que se hizo fue diseñar una estructura metálica con el propósito de emular la sección de lavado del producto primario. Una emulación más aproximada al proceso real implica el uso de mangueras y rociadores, activadas por algun sensor. De igual manera, se presento un problema al momento de iniciar la etapa 3. La figura 4.2 muestra la emulación del proceso de exprimido lo cual se complemento con el diseño de un arco. En las industrias productoras de zumo de naranja lo que se realiza en la sección de exprimido es ingresar las naranjas a engranajes activados por motores los cuales son los encargados de exprimir y separar el restante de naranja en el proceso.

Figura 4.1: Diseño de la etapa de lavado.

Figura 4.2: Diseño de la etapa de exprimido del producto primario.

En este contexto resulta importante mencionar que el diseño de la línea de producción se inició desde cero. A pesar que Factory I/O cuenta con una gran cantidad de escenarios previamente configurados, la mayoría de ellos se consideran complejos debido a la maquinaria empleada. Sin embargo, esta plataforma presenta una gran ventaja, ya que, facilita el uso de un sistema multi-tanque, que por defecto, comprende el bloque esencial para el desarrollo del proyecto. Por todo ello, Factory I/O es considerado un *game changer*, ya que permite realizar una gran cantidad de proyectos orientado a la automatización industrial, sin la presión de poner en riesgo o descomponer cualquier equipo. En consecuencia, se presenta la posibilidad de tener la certeza de ejecutar las mismas pruebas en el campo y garantizar su funcionamiento.

4.2. Análisis del diseño y desarrollo del HMI y de gemelos digitales para operarios

En la sección 3.3.3.2 se describió a detalle el procedimiento realizado para la creación del HMI en la herramienta de LabVIEW, de esta manera, los operarios pueden acceder al manejo de cada una de las variables que intervienen en la línea de producción. El diseño consta principalmente de un *layout*, a

partir del cual se incorporan herramientas como tanques con indicadores de nivel, *leds on/off*, motores y pantallas de visualización del controlador PID sobre cada uno de los tanques, la figura 4.3 muestra el resultado final para el diseño del HMI de la planta dentro de LabVIEW.

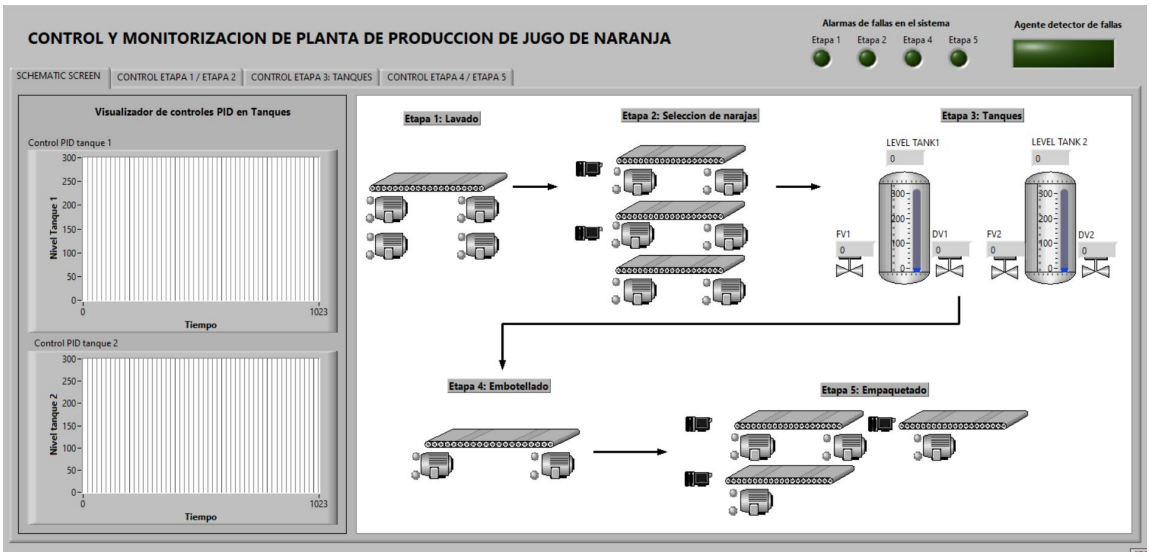


Figura 4.3: Sistema HMI desarrollado en LABVIEW

Como se mencionó anteriormente, el sistema HMI va a ser implementado dentro de la fábrica. Por lo tanto, surge la necesidad de crear un gemelo digital con el objetivo de integrar por completo todas las herramientas IIoT dentro de la misma, para ello se utilizó la herramienta de Node-RED. La figura 4.4 muestra el resultado final del diseño del *dashboard* dentro de Node-RED, de manera conjunta se desarrollo gemelos digitales para los accionadores y controladores del sistema así teniendo el control de la planta desde un nivel administrativo, con la diferencia de que en este caso no se emplea un *layout*. No obstante, se tiene acceso a distintos análisis de datos como el valor del estadístico de Hotelling, el cual va a ser monitorizado mediante una gráfica. Otra funcionalidad del gemelo digital consiste en el envío de alarmas al correo electrónico de una cuenta registrada, cada vez que el agente detecte un comportamiento anormal dentro del sistema. El gemelo digital está destinado para el uso de gerentes y administrativos de la planta, los cuales van a tener a cargo la tarea de llevar el registro e ingreso de niveles de producción de zumo de naranja dependiendo de la demanda de fabricación.

Figura 4.4: Gemelo digital del sistema HMI desarrollado en Node-RED.

A partir del desarrollo del gemelo digital se añadieron dos herramientas para un entorno IIoT, como el envío de correos una vez que el agente detector de fallas se activa para que de esta forma el supervisor de planta analice la falla el momento exacto que se originó y además, un *Bot* con el cual los supervisores y gerentes de planta van a poder interactuar para así tener conocimiento de niveles de producción, el estado de los niveles de tanques y el estado del agente detector de fallas. La figura 4.5 muestra la interacción que se tiene con el bot de planta al momento de solicitar información.

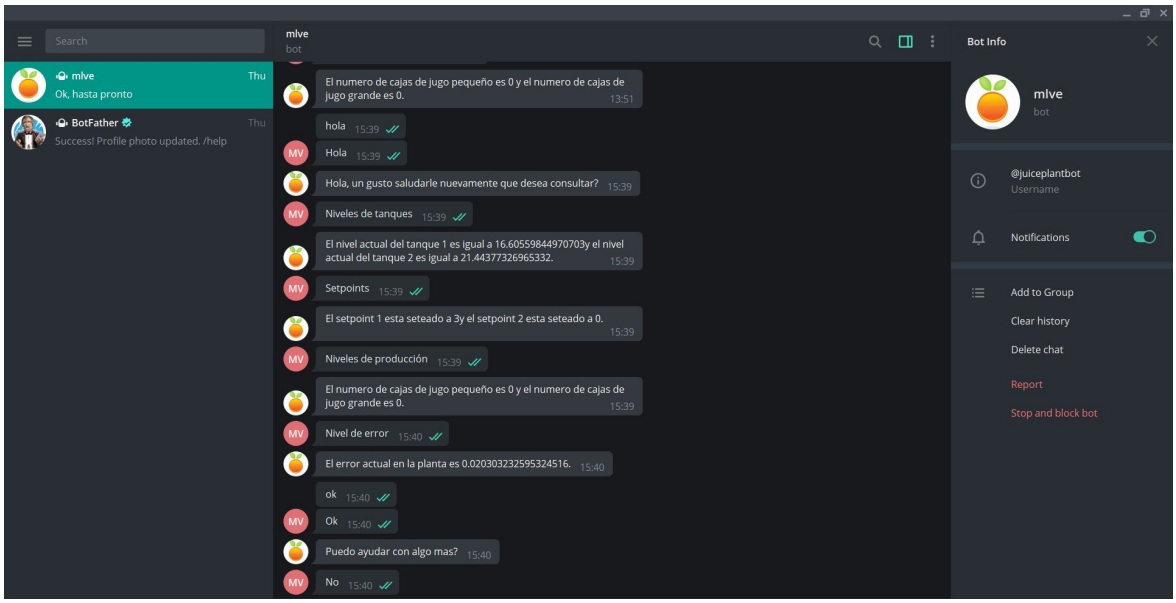


Figura 4.5: *Bot* de comunicación con la planta.

4.3. Análisis del envió de datos a las nubes de información

Para el envió de información de la planta hacia la nube se optó por dos protocolos de comunicación, estos son HTTP y MQTT. La razón por la cual se hace uso de dos protocolos recae en el sentido de compatibilidad, es decir, algunos entornos de desarrollo IoT presentan ciertas falencias respecto al protocolo que se use. Sin embargo, esta desventaja resulta un beneficio, ya que, permite el uso de dos herramientas de desarrollo y de esta manera realizar un estudio comparativo y determinar que protocolo es superior respecto al otro. Los servicios en la nube utilizados son Thinger.io y AWS. Dichos entornos presenta ciertas características y herramientas, las mismas que son explotadas por desarrolladores, con el propósito de realizar distintos análisis a partir del ingreso de datos. En este caso, el envío de la información no solamente se lo realiza hacia las dos plataformas, sino que también a una base de datos local con ayuda de MySQL.

4.3.1. Protocolos de comunicación implementados

Como se mencionó anteriormente, para la transferencia de información hacia la nube se emplearon tres herramientas. No obstante, resulta necesario mencionar que la tasa de muestreo para el envío de datos varía independientemente de como se encuentra configurado en Node-RED. Es decir, la tasa envío desde Node-RED hacia la base de datos de Thinger.io y AWS se lo realiza cada un minuto, aunque a nivel local, dicha métrica se encuentre configurada con un valor de dos segundos. Esto se debe principalmente a como se encuentra configurado a nivel de capa de aplicación cada uno de los entornos de desarrollo. Sin embargo, los resultados que se obtienen presentan una buena tendencia en cuanto a la recepción de datos por parte del sistema.

Teniendo en cuenta lo mencionado, la figura 4.6 muestra un análisis de la eficiencia de tráfico de datos en donde se aprecia al inició un envío de datos con una tasa de recepción estable equivalente a dos segundos. A partir de esto, se reduce el tiempo desde los 2 segundos hasta los 0.1 segundos con un paso de 0.3 segundos. Así se comprueba que el porcentaje de paquetes con error sube de 13.99 % a 79.42 %, es por esto que se mantuvo el tiempo de envío de datos cada dos segundos.

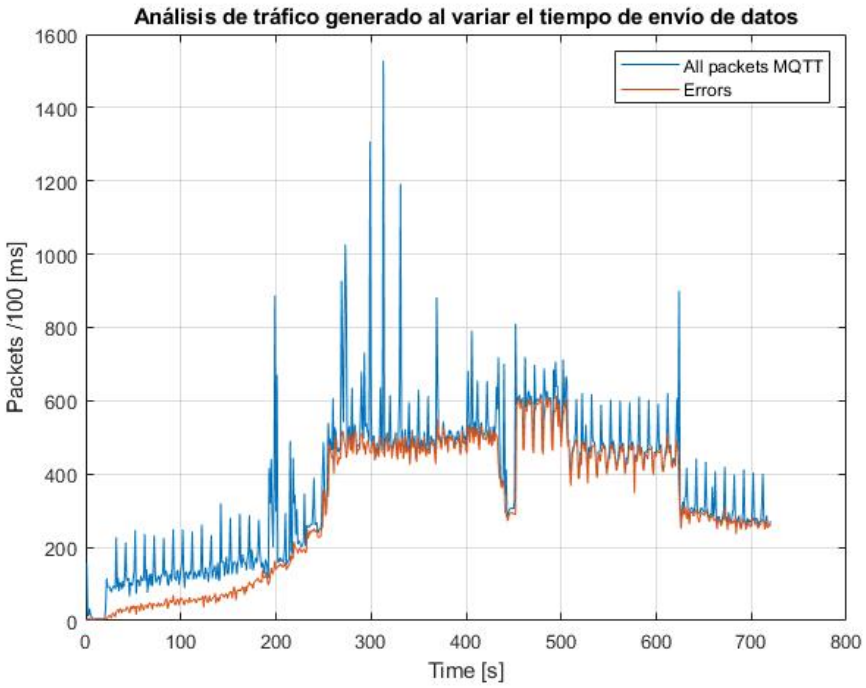


Figura 4.6: Análisis de trafico generado al variar tiempos de envió de datos

Una vez considerada la tasa de muestreo se procede a configurar el almacenamiento de los datos en forma local empleando MySQL. Este punto resulta clave para el agente detector de fallas, debido a que el tiempo de muestreo para la virtualización de datos es de dos segundos. Por lo tanto, resulta evidentemente inferir que el *set* de datos utilizado en el código de MATLAB corresponde a la información almacenada de forma local en MySQL. En este escenario, vale la pena mencionar que el intervalo de envío de datos desde Node-RED debe ser cada 0.1 segundos, para detectar una falla en la planta.

En función a lo detallado anteriormente, se procede a tomar un conjunto de datos de la misma fecha y con la misma tasa de muestreo para cada una de las plataformas de almacenamiento, con estos datos posteriormente se realizaran gráficos de puntos para así poder observar el efecto de la tasa de muestreo sobre el envío de información de cada uno de los tanques.

La figura 4.7 muestra los resultados para el almacenamiento de datos dentro de AWS. AWS únicamente almaceno 26 registros en un espacio de tiempo de una hora. A través de la observación de la gráfica se puede constatar que AWS solamente registro los saltos de niveles dentro de los tanques, más no el proceso de variación del tanque desde un nivel hacia el nuevo nivel solicitado.

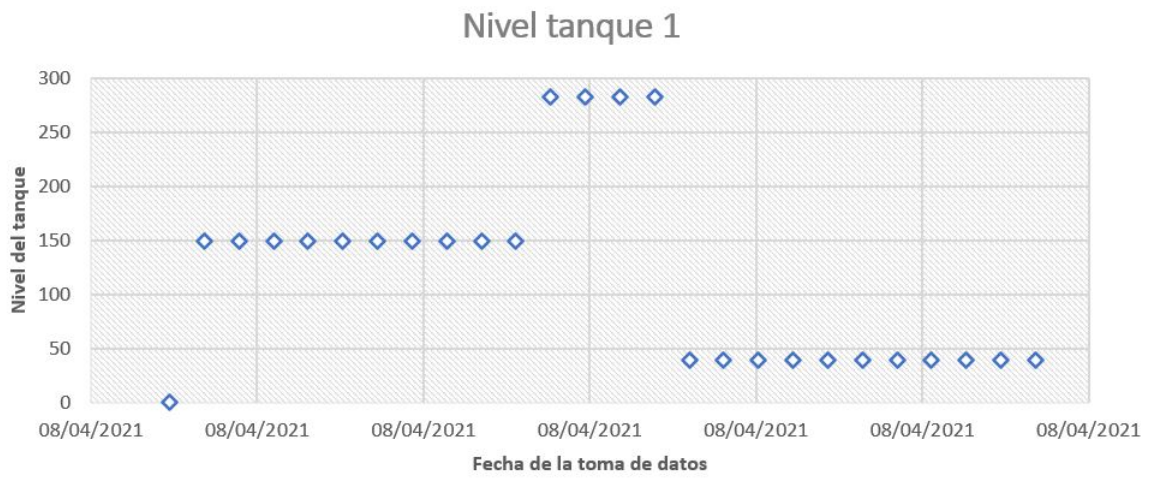


Figura 4.7: Toma de datos de operación del tanque 1 en AWS.

La figura 4.8 muestra los resultados para el almacenamiento de datos dentro de thinger i.o, el mismo que almaceno 60 registros en un espacio de tiempo de una hora. A través de la observación de la gráfica se puede constatar que Thinger i.o de igual manera que AWS registra únicamente los espacios de tiempo en donde el nivel de tanque ya esta estabilizado, lo cual no permite registrar variaciones en el proceso de llenado del tanque.

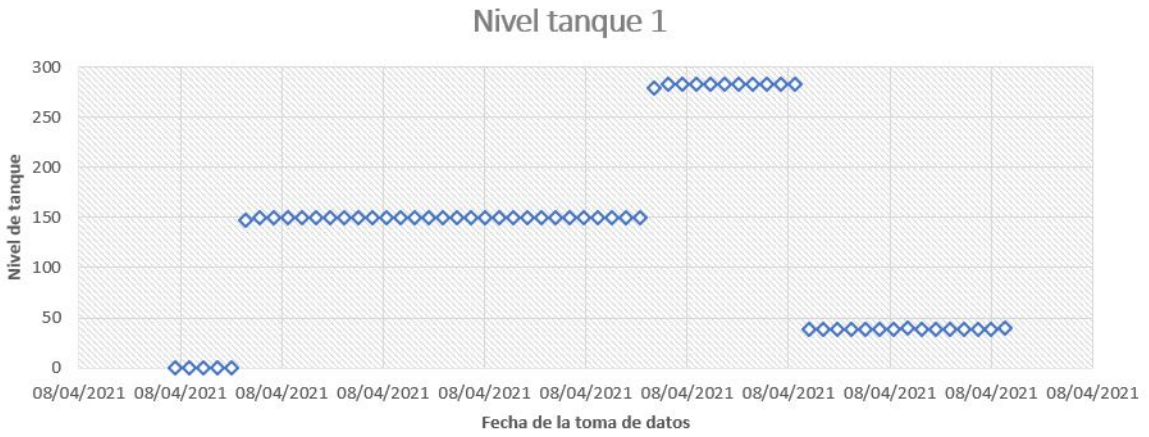


Figura 4.8: Toma de datos de operación del tanque 1 en Thinger i.o.

La figura 4.9 muestra los resultados para el almacenamiento de datos dentro de MySQL. Esta plataforma almaceno 1823 registros en un espacio de tiempo de una hora. A través de la observación de la gráfica se puede constatar que MySQl registra todas las variaciones que se van dando en el proceso de llenado del tanque desde un nivel hacia otro. Lo cual es de vital importancia en caso de que se presente un error en el llenado del tanque y se tenga constancia del mismo.

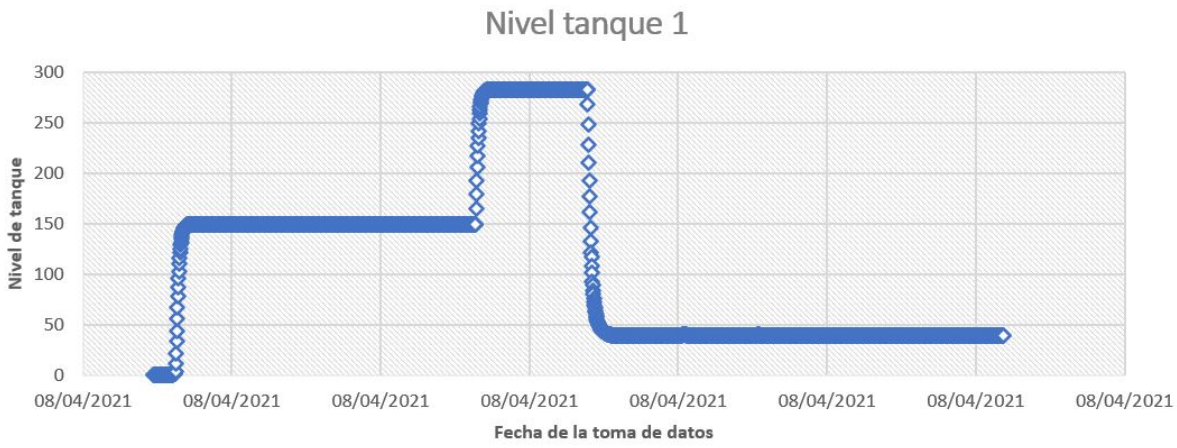


Figura 4.9: Toma de datos de operación del tanque 1 en MySQL.

El tiempo de almacenamiento de información tanto en las plataformas de Thinger.io y AWS es muy amplio y, en consecuencia no es posible visualizar con claridad el funcionamiento del sistema multi-tanque de la planta. Si embargo, este no es el caso de MySQL el cual almacena los datos cada que un nuevo tiempo de muestreo es registrado, con los datos obtenidos dentro de MySQL se entrenó al agente de detección de fallas y su funcionamiento fue óptimo.

4.3.2. Problemática y solución para envío de información a la nube

El envío de datos desde la planta hacia la nube, corresponde al desafío que tuvo que ser solucionado. En aplicaciones prácticas, las industrias emplean un equipo destinado para esta tarea en específico, facilitando así el manejo de datos dentro de la fábrica. El equipo que se implementa para el envío de información es el SIMATIC IoT2040, cuya función es facilitar la comunicación entre varias fuentes de datos. Una ventaja de este dispositivo, es que no extrae solamente los datos, sino, que también los procesa, almacena y los envía a una nube local o externa. Actualmente, corresponde al equipo más empleado en el campo de la industria debido a su compatibilidad con varios sistemas operativos, así como también, con dispositivos *open source* como Arduino.

Para la solución del problema, teniendo en cuenta que el ambiente de prueba es totalmente simulado, se tuvo que optar por el uso de protocolos de comunicación, con la intención de facilitar el envió de información. En este contexto, el punto más importante corresponde al *middleware* (OPC), dado que, cumple la función de columna vertebral de la pirámide de automatización. Es decir, permite la interconexión entre cada uno de los mecanismos de comunicación que emplea cada nivel de la misma. El servidor OPC permite que se establezca la comunicación desde la industria hacia el SCADA, y posteriormente a la nube local. De esta manera, es posible monitorear en tiempo real cada uno de los procesos involucrados desde la nube local.

Node-RED es el pilar para mantener la conexión entre los dos entornos de desarrollo IoT con MySQL. La figura 4.10 muestra el diagrama de bloques implementado para establecer el enlace de envió de datos hacia las plataformas de almacenamiento.

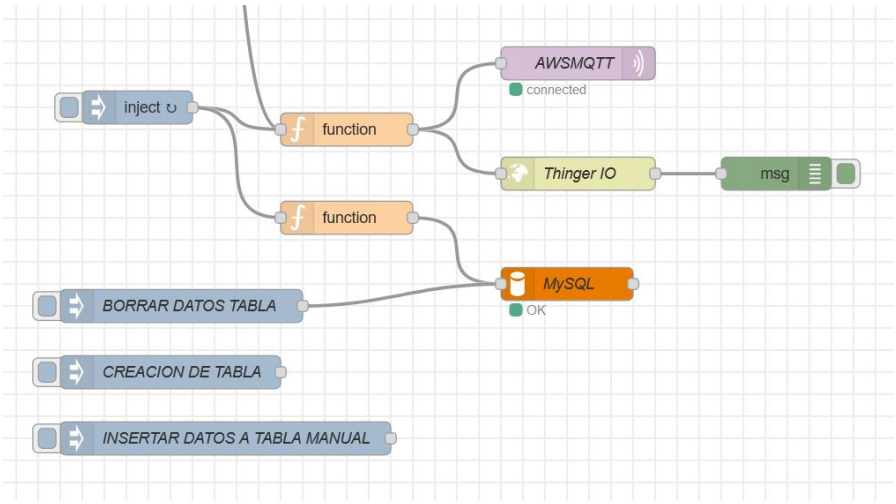


Figura 4.10: Diagrama de flujo para el envió de datos desde Node- RED.

Las figuras 4.11, 4.12 y 4.13 muestra los resultados obtenidos del almacenamiento de datos de cada una de las plataformas.

Elementos devueltos (100)

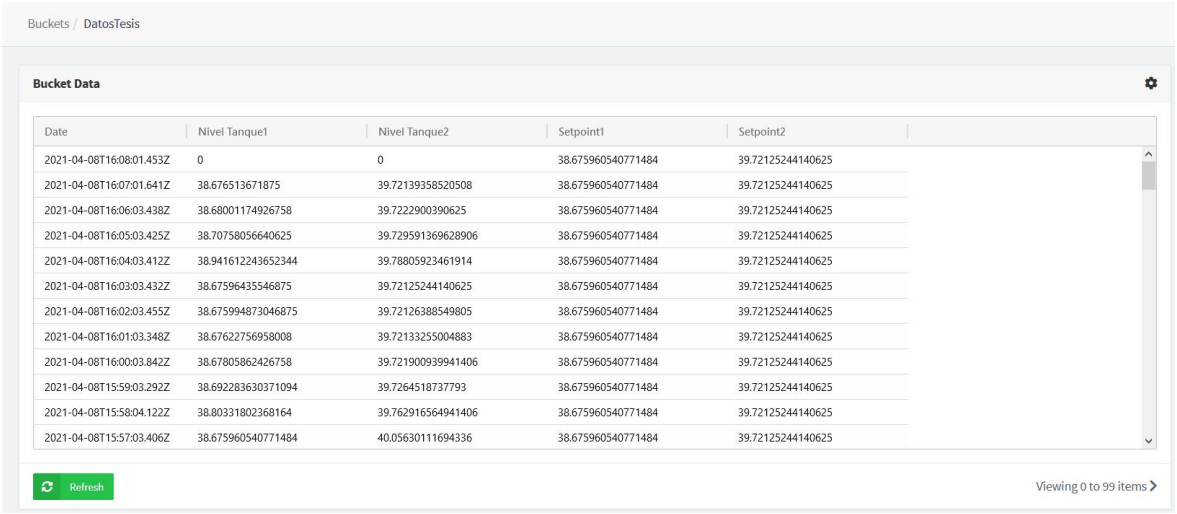
Acciones | Crear elemento

Buscar elementos

1 2 ...

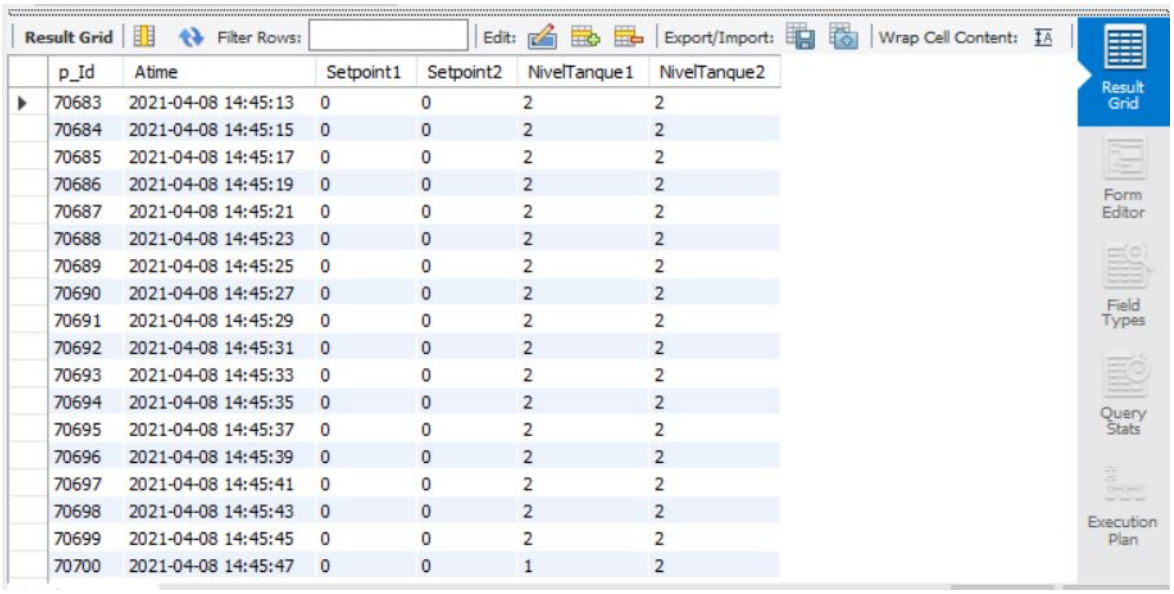
ts	Setpoint1	NivelTan...	NivelTan...	Setpoint2
161791561...	38.675960...	38.677448...	39.721710...	39.72125244140625
161791549...	38.675960...	38.749668...	39.745197...	39.72125244140625
161791542...	38.675960...	38.675960...	40.033950...	39.72125244140625
161791442...	38.675960...	38.676372...	39.721530...	39.72125244140625
161791385...	38.675960...	38.714462...	39.758308...	39.72125244140625
161791379...	38.675960...	39.021827...	40.064094...	39.72125244140625
161791324...	283.27526...	283.27523...	285.36584...	285.3658752441406
161791321...	283.27526...	283.27523...	285.36584...	285.3658752441406
161791237...	149.47734...	149.47732...	164.11148...	164.1114959716797
161791118...	0	1.1450963...	1.1935838...	0
161791073...	15.679443...	15.679440...	17.770032...	17.770034790039062
161707747...	0	0	0	0
161707683...	0	0	0	0

Figura 4.11: Resultado de almacenamiento de información en AWS.



Date	Nivel Tanque1	Nivel Tanque2	Setpoint1	Setpoint2
2021-04-08T16:08:01.453Z	0	0	38.675960540771484	39.72125244140625
2021-04-08T16:07:01.641Z	38.676513671875	39.72139358520508	38.675960540771484	39.72125244140625
2021-04-08T16:06:03.438Z	38.68001174926758	39.7222900390625	38.675960540771484	39.72125244140625
2021-04-08T16:05:03.425Z	38.70758056640625	39.729591369628906	38.675960540771484	39.72125244140625
2021-04-08T16:04:03.412Z	38.941612243652344	39.78805923461914	38.675960540771484	39.72125244140625
2021-04-08T16:03:03.432Z	38.67596435546875	39.72125244140625	38.675960540771484	39.72125244140625
2021-04-08T16:02:03.455Z	38.675994873046875	39.72126388549805	38.675960540771484	39.72125244140625
2021-04-08T16:01:03.348Z	38.67622756958008	39.72133255004883	38.675960540771484	39.72125244140625
2021-04-08T16:00:03.842Z	38.67805862426758	39.721900939941406	38.675960540771484	39.72125244140625
2021-04-08T15:59:03.292Z	38.692283630371094	39.7264518737793	38.675960540771484	39.72125244140625
2021-04-08T15:58:04.122Z	38.80331802368164	39.762916564941406	38.675960540771484	39.72125244140625
2021-04-08T15:57:03.406Z	38.675960540771484	40.05630111694336	38.675960540771484	39.72125244140625

Figura 4.12: Resultado de almacenamiento de información en Thinger.io.



p_Id	Atime	Setpoint1	Setpoint2	NivelTanque1	NivelTanque2
70683	2021-04-08 14:45:13	0	0	2	2
70684	2021-04-08 14:45:15	0	0	2	2
70685	2021-04-08 14:45:17	0	0	2	2
70686	2021-04-08 14:45:19	0	0	2	2
70687	2021-04-08 14:45:21	0	0	2	2
70688	2021-04-08 14:45:23	0	0	2	2
70689	2021-04-08 14:45:25	0	0	2	2
70690	2021-04-08 14:45:27	0	0	2	2
70691	2021-04-08 14:45:29	0	0	2	2
70692	2021-04-08 14:45:31	0	0	2	2
70693	2021-04-08 14:45:33	0	0	2	2
70694	2021-04-08 14:45:35	0	0	2	2
70695	2021-04-08 14:45:37	0	0	2	2
70696	2021-04-08 14:45:39	0	0	2	2
70697	2021-04-08 14:45:41	0	0	2	2
70698	2021-04-08 14:45:43	0	0	2	2
70699	2021-04-08 14:45:45	0	0	2	2
70700	2021-04-08 14:45:47	0	0	1	2

Figura 4.13: Resultado de almacenamiento de información en MySQL

4.4. Análisis de protocolos para el envío de información y tiempos de respuesta

En este apartado se procede a realizar un análisis exhaustivo del tráfico de datos dentro de la red, con el objetivo de realizar una comparativa entre cada uno de los entornos de desarrollo IoT y de esta manera identificar que plataforma es la mejor. En este contexto, resulta importante mencionar que el sistema a analizar es robusto, en consecuencia, la cantidad de datos que se genera es relativamente extenso, por esta razón, es recomendable graficar los resultados con un eje de tiempo mayor a un segundo. Para este proceso se emplea Wireshark, en vista de que cuenta con una gran cantidad de herramientas para el análisis de la red.

4.4.1. Análisis de protocolos de comunicación en la nube

Para el análisis del protocolo MQTT, en primera instancia se captura todos los paquetes recibidos y enviados cuando el sistema se encuentra conectado a la red Wi-Fi. La figura 4.14 muestra la captura de datos en donde se recepta todo el tráfico en la red. La figura 4.15 muestra el filtrado de todos los paquetes que son enviados al puerto **8883** con el objetivo de centrarse únicamente en ese protocolo, en vista de que dicho puerto pertenece al servidor de AWS y, en consecuencia, corresponde a la plataforma con la que trabaja MQTT. La figura 4.16 muestra el intercambio de información entre las direcciones IP del ordenador y AWS.

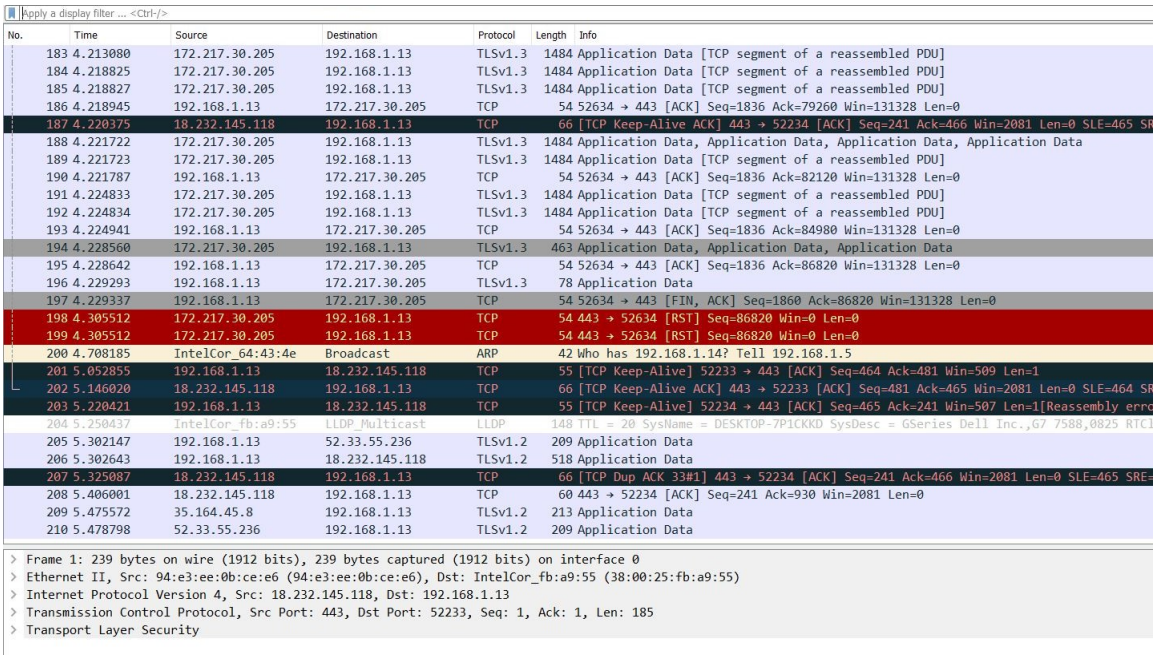


Figura 4.14: Captura de datos con el analizador de red Wireshark.

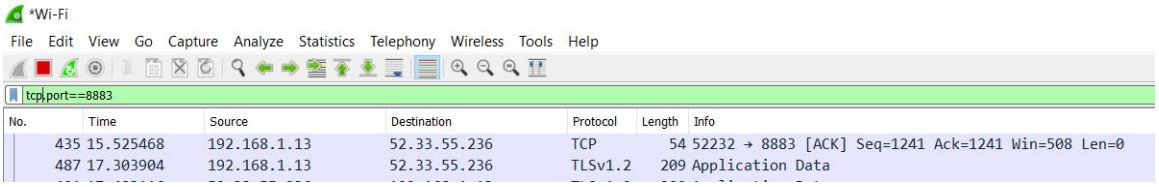


Figura 4.15: Filtrado de paquetes a través del puerto 8883

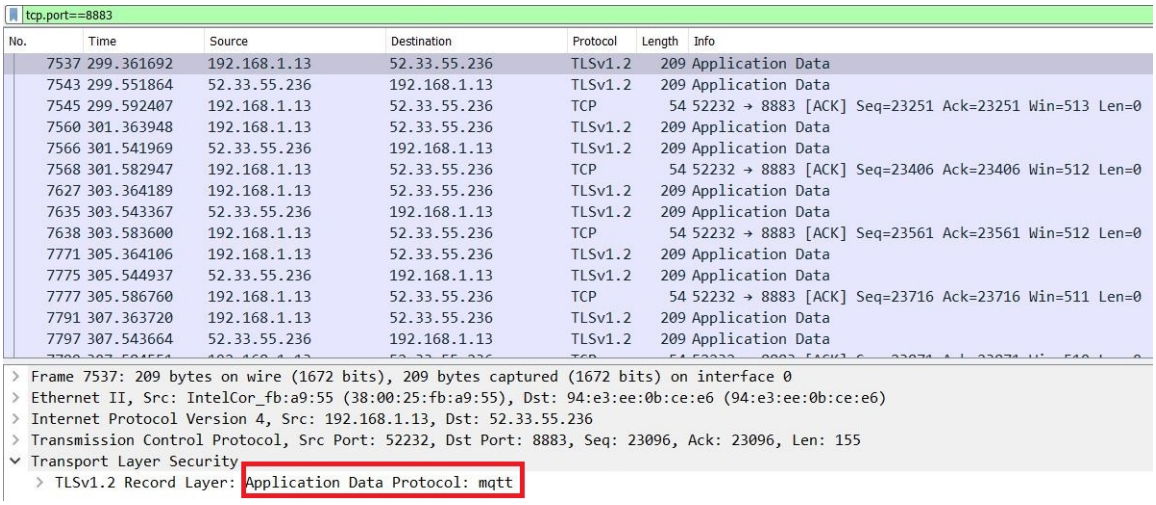


Figura 4.16: Análisis de un paquete MQTT.

Una vez capturado los paquetes, el siguiente paso consiste en determinar la eficiencia de envió y recepción, la figura 4.17 muestra la implementación de la herramienta *I/O Graph* propia de Wireshark en donde se analiza la cantidad de paquetes enviados, así como también el número de paquetes perdidos. Para MQTT, se obtuvieron muy buenos resultados, debido a que, la media de paquetes perdidos era relativamente baja, aproximadamente 34. De igual manera, se tiene un promedio de 209 paquetes enviados y 207 paquetes recibidos, sin errores.

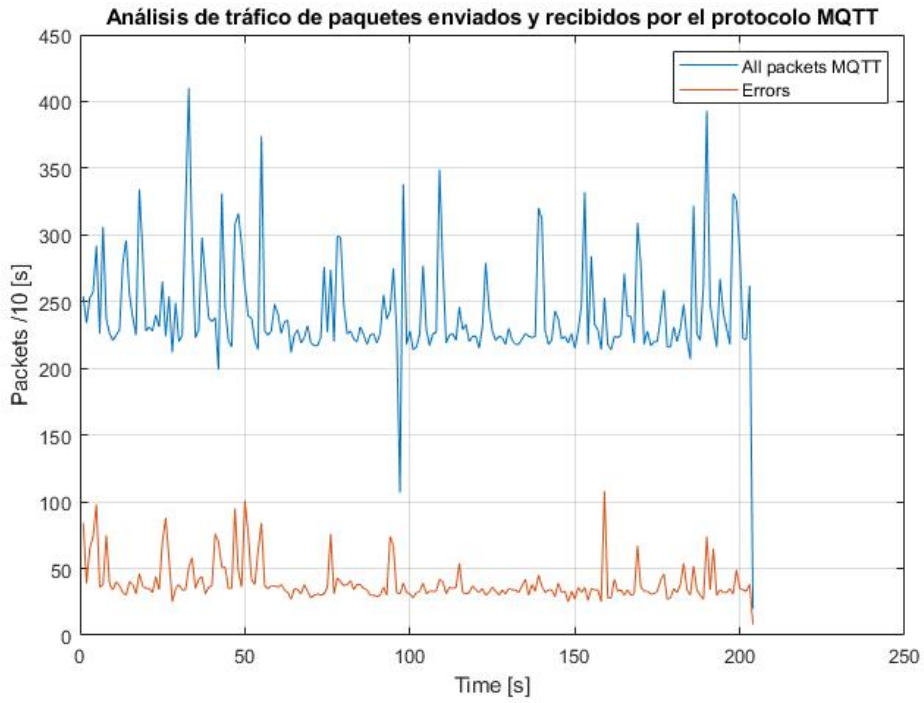


Figura 4.17: Análisis de trafico de paquetes enviados y recibidos por el protocolo MQTT.

Para el análisis de los otros protocolos, se procede a realizar el mismo procedimiento detallado

anteriormente. Las figuras 4.18 y 4.19 muestra los resultados de la captura de paquetes para HTTP y MySQL respectivamente.

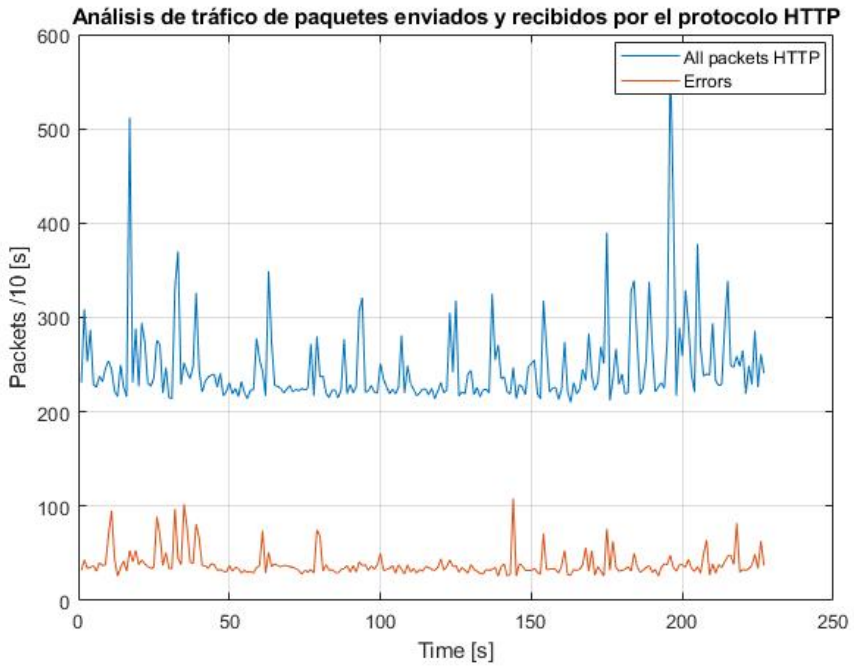


Figura 4.18: Análisis de trafico de paquetes enviados y recibidos por el protocolo HTTP.

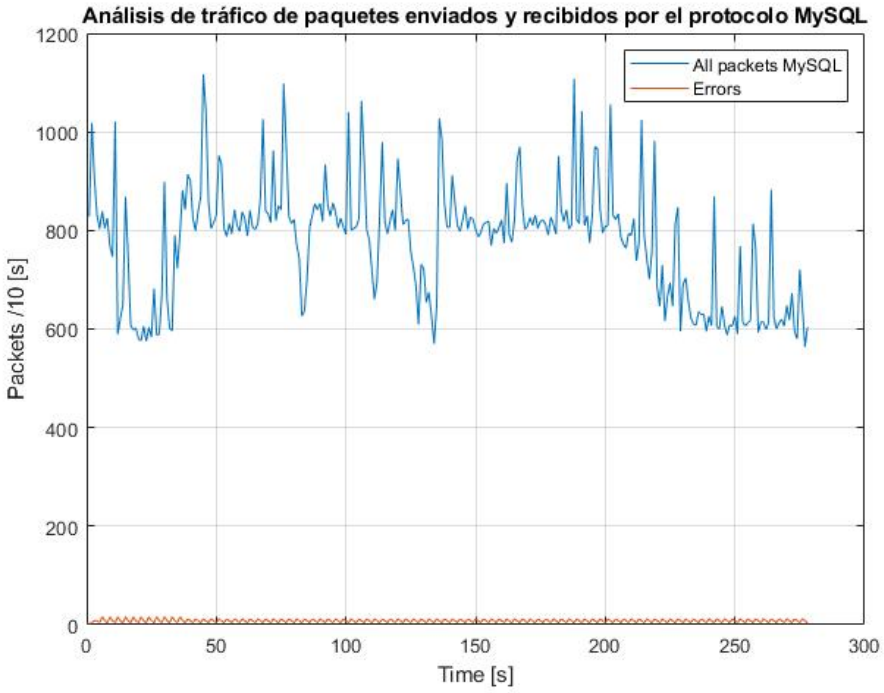


Figura 4.19: Análisis de trafico de paquetes enviados y recibidos por el protocolo de manera local.

La tabla 4.1 muestra un resumen del análisis de cada uno de los protocolos de comunicación. Como se puede observar, MQTT y HTTP presentan un nivel de efectividad relativamente igual, esto se puede comprobar al comparar la cantidad de paquetes perdidos (PP). En consecuencia, se puede concluir que ambos entornos de desarrollo responden de manera eficiente. De igual manera se obtiene un buen resultado para MySQL, en vista de que el promedio de paquetes perdidos equivale 8, lo que implica un mejor desempeño respecto a los otros. No obstante, se debe recordar que la captura paquetes de manera local, en la mayoría de casos es más eficiente, puesto que, se filtra información localizada en la misma nube.

Tabla 4.1: Resultado del envío y recepción de paquetes MQTT, HTTP y local

N^o	**Protocolo**	**Origen**	**Destino**	PE	PR_{se}	**PP**
1	MQTT	**IP:** 192.168.1.13 **Puerto:** 50412	**IP**: 52.32.148.125 **Puerto:** 8883	243	209	34
2	MQTT	**IP**: 52.32.148.125 **Puerto:** 8883	**IP:** 192.168.1.13 **Puerto:** 50412	246	207	39
3	HTTP	**IP:** 192.168.1.13 **Puerto:** 50412	**IP:** 172.217.30.205 **Puerto:** 443	247	209	38
4	HTTP	**IP:** 172.217.30.205 **Puerto:** 443	**IP:** 192.168.1.13 **Puerto:** 50412	248	210	38
5	LOCAL	127.0.0.1	127.0.0.1	776	768	8

- **PE:** Paquetes enviados
- PR_{se}**:** Paquetes recibidos, sin error
- **PP:** Paquetes pérdidos.

4.4.2. Tiempo de respuesta de la comunicación con el PLC

El análisis de tiempo de respuesta por parte del PLC constituye un proceso indispensable en el contexto de eficiencia del sistema, en vista de que constituye el punto de enlace entre la nube y la planta. Además, de poner a prueba el protocolo de comunicación MQTT como un sistema de control y monitorización en tiempo real. En vista de que el HMI y el gemelo digital fueron implementados para funcionar en conjunto con el servidor local, implica que los tiempos de respuesta hacia el PLC sean relativamente cortos, en comparación a los que se obtuvieron, cuando el sistema envío información a una nube ajena al servidor local de la planta. La figura 4.20 presenta la captura de trafico de datos generado por el PLC. Finalmente, la tabla 4.2 muestra los resultados obtenidos para los tiempos de respuestas cuando se analiza diferentes tipos de variables del proceso de automatización de la planta.

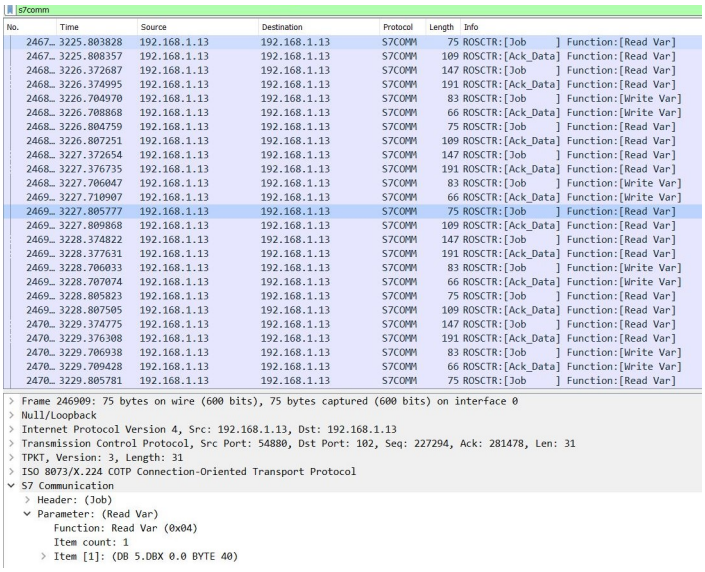


Figura 4.20: Captura de trafico de paquetes enviados al PLC.

Tabla 4.2: Resultado de tiempos de respuesta del PLC

Tipo de Variable	**Descripción**	**Dirección PLC**	**Tiempo Wireshark**	**Tiempo cronometrado**
Bool	Inicio	DB5.DBX0.0	$8ms$	$< 1s$
Bool	Paro	DB5.DBX10.0	$9ms$	$< 1s$
Bool	Emergencia	DB5.DBX10.1	$15ms$	$< 1s$
Real	Error Tanque	DB5.DBD12	$30ms$	$< 1s$
Real	*Setpoint* 1	DB5.DBD2	$37ms$	$< 1s$
Real	Nivel *Tanque* 1	DB5.DBD24	$34ms$	$< 1s$
Int	Caja transferida grande	DB5.DBW32	$31ms$	$< 1s$

4.5. Agente detector de fallas

Como se indicó en la sección 3.3.4.1, para la implementación del agente detector de fallas, fue necesario desarrollar un código en MATLAB (véase anexo A).Con ayuda de dicho *script* se obtiene las matrices D y V, las mismas que fueron calculadas luego de haber aplicado una descomposición en valores singulares de la matriz de covarianza correspondiente a los datos ingresados como bloque de entrenamiento para el agente (véase las ecuaciones 4.1 y 4.2).

$$\lambda = \begin{bmatrix} 56009{,}54683 & 0 \\ 0 & 104{,}03487 \end{bmatrix} \tag{4.1}$$

$$V = \begin{bmatrix} -0{,}77944 & -0{,}62646 \\ -0{,}62646 & 0{,}77944 \end{bmatrix} \tag{4.2}$$

Las figuras 4.21 y 4.22 muestran las gráficas de operación de los tanques una vez implementado el

código. La figura 4.23 muestra el rango de operación normal del bloque de datos y se puede apreciar cuando existen errores en el sistema. Los errores que se tomaron en cuenta en el sistema fueron:

- Accionamiento involuntario de válvulas.
- Fugas debido a rupturas o daño en válvulas.
- Nivel de tanques inadecuado para la producción en planta.

Estas fallas se inducen con la ayuda del control manual de planta que ofrece Factory I/O.

Figura 4.21: Respuesta de h_1 con y sin fallas.

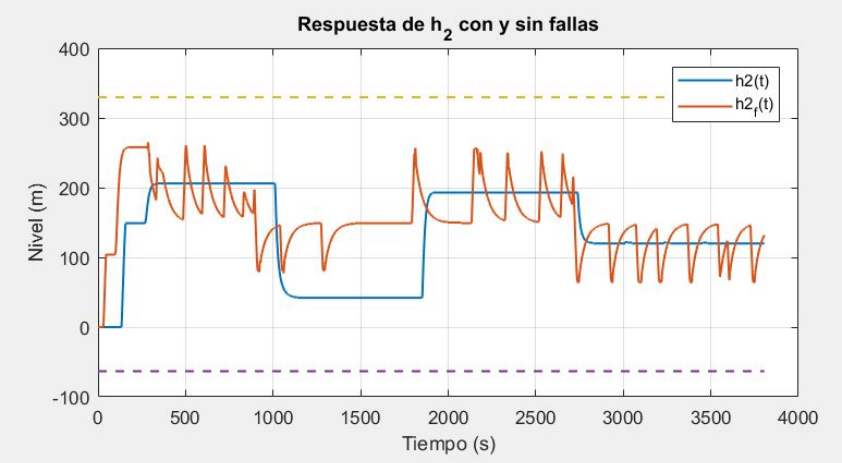


Figura 4.22: Respuesta de h_2 con y sin fallas.

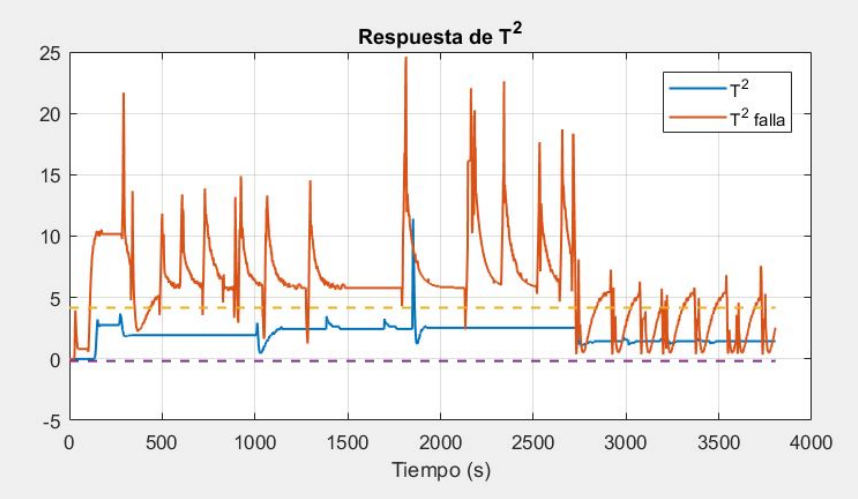


Figura 4.23: Respuesta de T^2.

Una vez calculados los valores resultantes, se procede a implementar en la herramienta Node-RED el método estadístico de Hotelling, con el objetivo de monitorizar la planta en tiempo real, la figura 4.24 muestra un diagrama de flujo en donde se detalla el proceso que se realiza una vez detectada una falla dentro de la planta. Los cálculos para obtener el valor de T^2 se realizan dentro de la nube local de la fábrica que se ejecuta en Node-RED, la figura 4.25 presenta la operación del agente detector de fallas. Cuando se detecte un error Node-RED procede a enviar la información al PLC para que este active una alarma, y de esta manera se suspenda la operación en cada uno de los procesos que conforman la línea de producción . Las figuras 4.26 y 4.27 muestran los resultados obtenidos luego de que se suscite un error dentro de la planta. En este contexto, resulta importante mencionar que el proceso de monitorización del error se esta ejecutando en tiempo real. En consecuencia, el envío de información desde la Node-RED hacia el PLC se realizará cada 0.1 segundos, de esta manera no existe un retardo en la respuesta del sistema. El único escenario que puede presentarse para que exista un retardo en el envió de información desde la nube, es que la conexión a internet de los dispositivos falle.

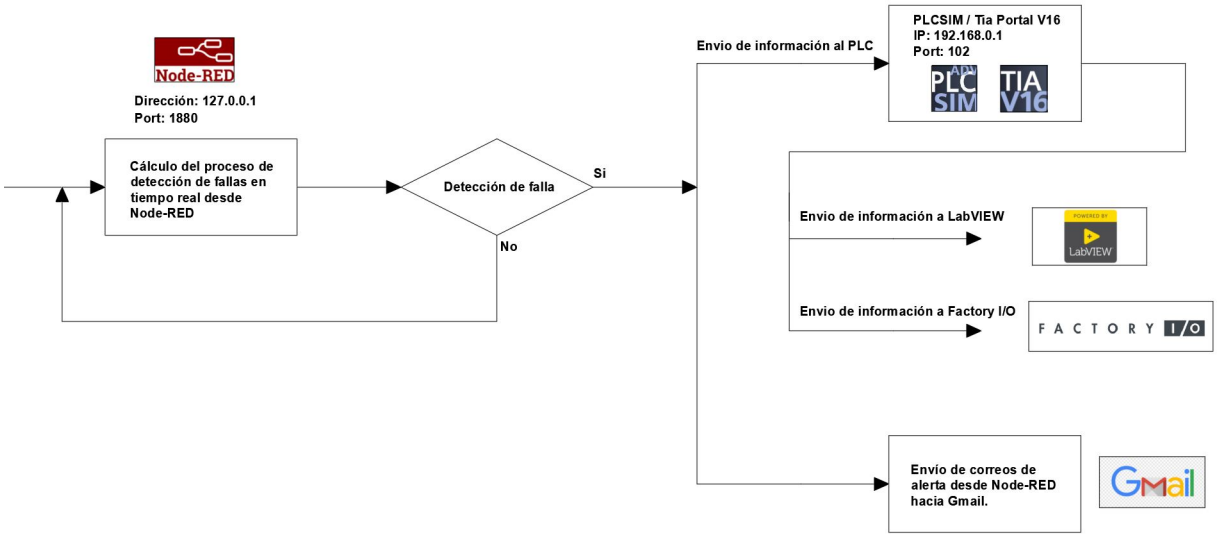


Figura 4.24: Diagrama de bloques que detalla el proceso de detección de una falla dentro de la planta

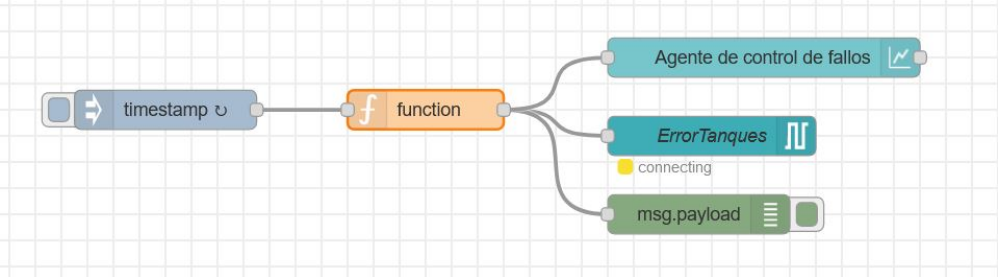


Figura 4.25: Operación del agente detector de fallas en Node-RED.

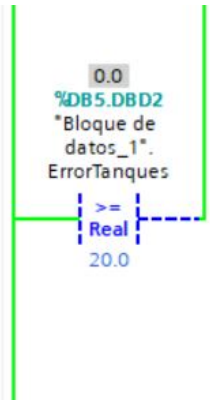


Figura 4.26: Recepción de información del detector desde la nube hacia el PLC.

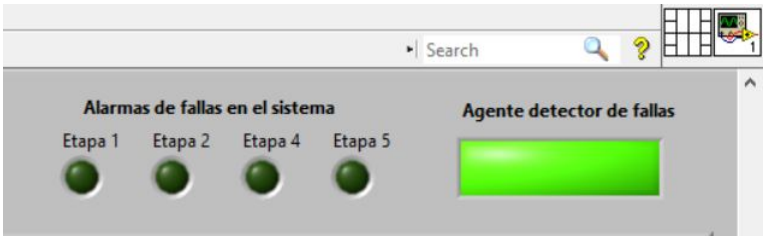


Figura 4.27: Detección de falla dentro del HMI de planta.

Capítulo 5

Conclusiones y Recomendaciones

5.1. Conclusiones

- La evolución de la Industria 4.0 implica avances en cada uno de sus enfoques, la interconectividad, la automatización, el aprendizaje automático y los datos en tiempo real. Mediante la ejecución de este trabajo fue posible corroborar la existencia de varias herramientas para la simulación de procesos industriales, mismas que facilitan el aprendizaje de control y automatización de procesos, y, a la vez, potencian su evolución. El uso de estas herramientas permite adquirir conocimientos con respecto a procesos industriales que se manejan en la actualidad; Factory I/O es una de ellas, esta ha permitido experimentar en el campo sin necesidad de pisar una fábrica o un laboratorio. Personalmente, consideramos que esta herramienta debe ser introducida para el conocimiento de próximas generaciones y así generar nuevos campos de conocimiento.

- Sin el suministro de los servicios informáticos, como los servidores, bases de datos y software, a través del internet no se estaría implementando una tecnología IoT en realidad, por esta razón utilizar nubes de información fue de vital importancia para la ejecución de este proyecto. Como se detalló en secciones anteriores, se usan dos nubes de información, una plataforma libre y una plataforma de pago, que son Thinger.i.o y AWS respectivamente. Thinger.i.o cuenta con un mejor sistema de almacenamiento frente a AWS, esto se determinó con base a las pruebas realizadas, en vista de que recibe un gran tráfico de datos y la nube no logra almacenarlos a en su totalidad debido a que se debe de expandir el tamaño de almacenamiento de las bases de datos.

  AWS es una herramienta para el análisis de datos muy grande y que tiene muchas funcionalidades adicionales, sin embargo, lo que se buscó dentro de las plataformas es que nos permitan visualizar los datos que se reciben sin mayor complicación, de manera que los gerentes de las empresas puedan analizarlos sin problema. Desde el punto de vista de los desarrolladores, se considera que Thinger.i.o es la herramienta que permite alcanzar este tipo de objetivos.

Existe un concepto que se volvió interesante al momento de realizar el agente detector de fallas y este fue el almacenamiento de datos de manera local para la planta, de esta forma se crea un *set* de entrenamiento para el agente. Como se expuso en la sección de resultados, el envío de información se realizó cada dos segundos para cada una de las nubes debido a que si se disminuye se produce una pérdida de paquetes elevada, por esta razón se optó por mantener el mismo. No obstante, en caso de reducir el tiempo de envío, la fluctuación de datos manera local no se ve afectado en gran medida así permitiendo variar el tiempo de envío a cualquier valor deseado.

- La sección de *middleware* es la columna vertebral de este proyecto, sin esta no existe una comunicación entre las nubes de información y la planta simulada. Dentro del *middleware* se implementó un servidor OPC, en el cual están enlazados clientes que tienen acceso a las variables que se manejan de manera local. De esta forma, fue posible enviar todos los datos necesarios desde el PLC hacia el *middleware* para posteriormente enviarlos a las nubes de información. El desarrollo de un servidor local OPC y la implementación en Node-RED fue vital para solventar el vacío que se tuvo al momento del envío de datos, puesto que, como se detalló en la sección de resultados, en las grandes empresas se emplean dispositivos destinados para este uso en específico.

- Un concepto interesante dentro de IoT fue la implementación de un gemelo digital para el sistema de línea de producción que se encuentra dentro de la planta. La virtualización de un sistema de control destinado al manejo de los procesos de la línea de producción es uno de los conceptos que se pudo desarrollar a lo largo del proyecto y que se vio envuelto en el desarrollo de IIoT. Principalmente, lo que se busca con la implementación de este sistema es que no se tenga como punto único de acceso el HMI, sino que sea accesible para el personal destinado al control y direccionamiento de la planta. De esta manera, se conforma un solo grupo de trabajo que puede estudiar los resultados que se tiene en la planta en tiempo real, con el objetivo de analizar la información y tomar decisiones sobre los procesos y/o fallas que se pueden presentar.

- La virtualización de servicios fue el objetivo final para el completo funcionamiento del proyecto. La implementación de un agente detector de fallas fue el punto más interesante del desarrollo, ya que, como su nombre indica va a permitir que la planta funcione con total normalidad evitando errores dentro del sistema multi-tanque. La toma de datos para crear un *set* de entrenamiento fue muy importante, así como también el tiempo de envío de datos. Se debe tener en cuenta que el agente tiene que experimentar diferentes tipos de escenarios para así poder detectar fallas. Para ello, se utilizó el método estadístico multivariable de Hotelling. Una vez que se implementó el sistema, se pudo comprobar que el mismo fue capaz de detectar diversos tipos de fallas dentro de la planta, tales como fugas, activaciones involuntarias de válvulas, mal ingreso de niveles para el momento de producción y niveles muy altos o muy bajos dentro de los tanques.

5.2. Recomendaciones

- Para la etapa de diseño de la planta se recomienda simular ambientes en los cuales las herramientas se puedan explorar en su totalidad. Es decir, considerar un software en el cual se cuenta con una cantidad de maquinaria extensa, y de esta manera evitar limitaciones en la implementación

de etapas en una línea de producción. En la actualidad Factory I/O es la herramienta mas completa para trabajar en ambientes simulado en industria, se espera que en nuevas versiones del *software* ingresen nuevos complementos de maquinaria.

- Al momento del envió de información se recomienda el uso de Thinger.io, pues para esta aplicación resultó acertado debido a su eficiencia de funcionamiento; también es recomendable emplear otros entornos de desarrollo que permitan seleccionar las herramientas adecuadas para potenciar el desarrollo del trabajo.

- Para la simulación de la planta en su totalidad se requiere correr varios programas al mismo tiempo y una conexión estable a internet, por esta razón se recomienda utilizar un ordenador que cumpla con las exigencias de funcionamiento del proyecto, de manera que la simulación se ejecute con normalidad.

5.3. Trabajos futuros

- Implementar un sistema de redes neuronales que permita el control y monitoreo de la planta con el fin de sustituir el agente detector de fallas por un sistema más robusto.

- Implementar la última capa de nivel de gestión de la planta, en donde se podrá tener información de clientes, proveedores, contratos, producción y demás datos para consolidar la información, y de esta manera a través de aplicación de redes neuronales controlar la producción de la planta de manera remota con solo interactuar con un agente.

Anexo A

Agente detector de fallas

A.1. Implementación en MATLAB

```
                                        % Simulation time

                                        % Sampling time for simulat
                                        % # samples per simulation



        'opdata_noFalla1.txt'            % Database no-fault
        'opdata_falla1.txt'              % Database fault

%t(1:ind_t1000) = []; A(1:ind_t1000,:) = []; B(1:ind_t1000,:) = [];

% Mapa de variables de los archivos de la base de datos de operacion
% c1 = h1 || c2 = h2 || c3 = u1 || c4  = u2 || c5 = u3

                                        % Level 1 no-fault
                                        % Level 1 with fault
                                        % Estadisticos de h1









                                        % Level 2 no-fault
                                        % Level 2 with fault
                                        % Estadisticos de h2









% ***************** + T2 + *************************
                                        % # datos de la muestra
                                        % # variables

```

```
39
40
41
42
43 for
44
45
46
47
48 end
49
50                                        % Estadisticos de Hotelling
51
52
53
54
55
56 % ******************************************************
57
58
59
60                                                      ...
61
62
63
64
65
66                              '__'               '__'  'LineWidth'
67       'Tiempo (s)'          'Nivel (m)'
68       'h1(t)'  'h1_f(t)'
69      'Respuesta de h_1 con y sin fallas'
70
71
72                              '__'               '__'  'LineWidth'
73       'Tiempo (s)'          'Nivel (m)'
74       'h2(t)'  'h2_f(t)'
75      'Respuesta de h_2 con y sin fallas'
76
77
78             'o'            'x'
79
80       'h_1'          'h_2'         'Espacio geometrico del problema'
81                       'k'           'k'  'LineWidth'
82       'Sin falla'  'Falla'  'Location'  'NorthWest'
83
84
85                                '__'  ...
86                   '__'  'LineWidth'
87       'T^2'  'T^2 falla'  'Location'  'NorthEast'
88       'Tiempo (s)'         'Respuesta de T^2'
```

Bibliografía

[1] Oscar Vinicio Altamirano Bautista., "Parametrización automática de las variables de control y su influencia en el proceso de mezclado de compuestos termoplásticos aplicando principios de industria 4.0," Ph.D. dissertation, Universidad Técnica de Ambato, 2018.

[2] H. G. Alvarado, "Desarrollo de un SCADA para una planta simulada de producción de vidrio templado," Ph.D. dissertation, Universidad Técnica de Ambato, 2019.

[3] V. M. Chadeev y N. I. Aristova, "Control of industrial automation," *Proceedings of 2017 10th International Conference Management of Large-Scale System Development, MLSD 2017*, 2017.

[4] A. Faul, N. Jazdi, y M. Weyrich, "Approach to interconnect existing industrial automation systems with the Industrial Internet," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, vol. 2016-November, pp. 2–5, 2016.

[5] A. R. M. Abata, "Diseño e implementación de un sistema distribuido empleando protocolo de comunicación industrial enfocado a los objetos (IIoT), para el control monitoreo remoto en tiempo real (RT) a través de la web en el laboratorio de Hidrónica y Neutrónica de la Unive," Ph.D. dissertation, Universidad de las Fuerzas Armadas, 2019.

[6] T. M. Fernandez-Carames y P. Fraga-Lamas, "A Review on Human-Centered IoT-Connected Smart Labels for the Industry 4.0," *IEEE Access*, vol. 6, num. c, pp. 25 939–25 957, 2018.

[7] H. Sasajima, T. Ishikuma, y H. Hayashi, "Future IIOT in process automation-Latest trends of standardization in industrial automation, IEC/TC65," *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2015*, pp. 963–967, 2015.

[8] V. Domova y A. Dagnino, "Towards intelligent alarm management in the Age of IIoT," *GIoTS 2017 - Global Internet of Things Summit, Proceedings*, 2017.

[9] B. B. J. Fernando y G. Q. D. Patricio, "Diseño e implementación de un sistema SCADA con control proporcional – integral – derivativo para caudal mediante un servidor, PLC y software en el laboratorio de automatización - Facultad de Mecánica," Ph.D. dissertation, Escuela Superior Politécnica de Chimborazo, 2018.

[10] T. Jung, N. Jazdi, y M. Weyrich, "A survey on dynamic simulation of automation systems and components in the internet of things," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–4.

[11] J. L. Flores y I. Mugarza, "Runtime vulnerability discovery as a service on industrial internet of things (iiot) systems," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, Sep. 2018, pp. 948–955.

[12] V. Jirkovský, M. Obitko, P. Kadera, y V. Mařík, "Toward plug play cyber-physical system components," *IEEE Transactions on Industrial Informatics*, vol. 14, num. 6, pp. 2803–2811, June 2018.

[13] D. Cortes, J. Ramirez, L. Villagomez, R. Batres, V. Vasquez-Lopez, y A. Molina, "Digital Pyramid: An approach to relate industrial automation and digital twin concepts," *Proceedings - 2020 IEEE International Conference on Engineering, Technology and Innovation, ICE/ITMC 2020*, 2020.

[14] C. Schlette, E. G. Kaigom, D. Losch, G. Grinshpun, M. Emde, R. Waspe, N. Wantia, y J. Robmann, "3D simulation-based user interfaces for a highly-reconfigurable industrial assembly cell," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, vol. 2016-November, 2016.

[15] G. Wu, L. Yao, y S. Yu, "Simulation and optimization of production line based on FlexSim," *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, vol. 116023, num. 2, pp. 3358–3363, 2018.

[16] S. S. Gilani, F. Jungbluth, H. Flatt, V. Wendt, y J. Jasperneite, "Alternative controls for soft real-time industrial control services in case of broken cloud links," *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, vol. 2016-November, pp. 1–4, 2016.

[17] S. Tamboli, M. Rawale, R. Thoraiet, y S. Agashe, "Implementation of modbus rtu and modbus tcp communication using siemens s7-1200 plc for batch process," in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, May 2015, pp. 258–263.

[18] A. L. Dias, G. S. Sestito, A. C. Turcato, y D. Brandão, "Panorama, challenges and opportunities in profinet protocol research," in *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, 2018, pp. 186–193.

[19] D. Bruckner, M. P. Stanica, R. Blair, S. Schriegel, S. Kehrer, M. Seewald, y T. Sauter, "An Introduction to OPC UA TSN for Industrial Communication Systems," *Proceedings of the IEEE*, vol. 107, num. 6, pp. 1121–1131, 2019.

[20] M. Endi, Y. Z. Elhalwagy, y A. Hashad, "Three-layer PLC/SCADA system architecture in process automation and data monitoring," *2010 The 2nd International Conference on Computer and Automation Engineering, ICCAE 2010*, vol. 2, pp. 774–779, 2010.

[21] W. Yang y S. Takakuwa, "Simulation-based dynamic shop floor scheduling for a flexible manufacturing system in the industry 4.0 environment," in *2017 Winter Simulation Conference (WSC)*, Dec 2017, pp. 3908–3916.

[22] P. Zhang, S. Member, Y. Wu, S. Member, y H. Zhu, "Open ecosystem for future industrial Internet of things (IIoT): Architecture and application," *CSEE Journal of Power and Energy Systems*, vol. 6, num. 1, pp. 1–11, 2020.

[23] I. Bedhief, L. Foschini, P. Bellavista, M. Kassar, y T. Aguili, "Toward self-adaptive software defined fog networking architecture for IIoT and industry 4.0," *IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD*, vol. 2019-September, pp. 1–5, 2019.

[24] Z. Liu, X. Yang, Y. Yang, K. Wang, y G. Mao, "Dats: Dispersive stable task scheduling in heterogeneous fog networks," *IEEE Internet of Things Journal*, vol. 6, num. 2, pp. 3423–3436, April 2019.

[25] D. Kean y L. Pierre, "A literature review on variability in semiconductor manufacturing: The next forward leap to Industry 4.0." pp. 3213–3224, 2016.

[26] H. Kathiriya, A. Pandya, V. Dubay, y A. Bavarva, "State of Art: Energy Efficient Protocols for Self-Powered Wireless Sensor Network in IIoT to Support Industry 4.0," *ICRITO 2020 - IEEE 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)*, pp. 1311–1314, 2020.

[27] "Japón, uno de los países más robotizados, teme por el futuro del empleo." [En línea]. Disponible: https://www.elespanol.com/invertia/disruptores-innovadores/innovadores/20181113/japon-paises-robotizados-teme-futuro-empleo/352966324{_}0.html

[28] Y. H. Lai, Y. H. Huang, C. F. Lai, S. Y. Chen, y Y. C. Chang, "Dynamic Adjustment Mechanism based on OPC-UA Architecture for IIoT Applications," *Indo - Taiwan 2nd International Conference on Computing, Analytics and Networks, Indo-Taiwan ICAN 2020 - Proceedings*, pp. 335–338, 2020.

[29] T. Sivakumaran, F. Kohne, y M. Toth, "Identification of critical success factors for emerging market entry planning processes in the automotive industry," *IEEE International Conference on Industrial Engineering and Engineering Management*, vol. 2016-January, pp. 1694–1698, 2016.

[30] C. Nagpal, P. K. Upadhyay, S. Shahzeb Hussain, A. C. Bimal, y S. Jain, "IIoT Based Smart Factory 4.0 over the Cloud," *Proceedings of 2019 International Conference on Computational Intelligence and Knowledge Economy, ICCIKE 2019*, pp. 668–673, 2019.

[31] J. Del Val, "Industria 4.0. La Transformación Digital de la Industria Española," *Coddiinforme*, p. 120, 2012.

[32] M. Ynzunza Cortés, Carmen Berenice; Izar Landeta, Juan Manuel; Bocarando Chacón, Jacqueline Guadalupe; Aguilar Pereyra, Felipe; Larios Osorio, "El Entorno de la Industria 4.0: Implicaciones y Perspectivas Futuras," *Conciencia Tecnológica*, p. 54, 2017.

[33] Y. Carmen, I. Juan, L. Martín, A. Felipe, B. Jacqueline, y U. T. D. Querétaro, "Implicaciones de la industria 4.0 en el trabajo y la competencia del capital humano." vol. 4, num. 10, pp. 5–13, 2017.

[34] M. Christopher y M. Holweg, ""Supply Chain 2.0": Managing supply chains in the era of turbulence," *International Journal of Physical Distribution and Logistics Management*, vol. 41, num. 1, pp. 63–82, 2015.

[35] M. Rüßmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel, y M. Harnisch, "Industry 4.0 The Future of Productivity and Growth in Manufacturing Industries," vol. 2, p. 20, 2015.

[36] J. G. B. Cortés, C. B. Y., Landeta, J. M. I., Chacón, "El entorno de la industria 4.0: implicaciones y perspectivas futuras," *Conciencia tecnológica*, vol. 54, pp. 33–45, 2017. [En línea]. Disponible: https://www.redalyc.org/jatsRepo/944/94454631006/html/index.html

[37] J. Machado, "Automatización de los procesos Productivos en la planta II División Partes y Piezas para la Empresa Indurama S.A." Ph.D. dissertation, Universidad de Cuenca, 2019.

[38] C. Ruedas, "Automatización industrial: áreas de aplicación para ingeniería," *Boletín electrónico, Universidad Rafael landívar*, vol. 2008, num. 10, pp. 1–19, 2008.

[39] J. P. Torres Vásquez y A. R. Vega Soto, "Diseño E Implementación De Un Laboratorio De Redes De Comunicación Industrial Para La Universidad Politécnica Salesiana, Cuenca," Ph.D. dissertation, Universidad Politécnica Salesiana, 2015.

[40] J. A. Estrada Roque, "Protocolos de comunicaciones industriales," vol. 1, num. 33, pp. 1–3, 2015.

[41] S. Tamboli, M. Rawale, R. Thoraiet, y S. Agashe, "Implementation of modbus rtu and modbus tcp communication using siemens s7-1200 plc for batch process," in *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, 2015, pp. 258–263.

[42] H. González-Acevedo y Ó. G. Villamizar-Galvis, "Implementación de un sistema de control para regular la velocidad y posición de motores industriales utilizando el protocolo de comunicación OPC," *Revista UIS Ingenierías*, vol. 18, num. 2, pp. 147–158, 2019.

[43] B. Medina Delgado, S. Castro Casadiego, y L. Camargo Ariza, "Tecnologías de código abierto para la gestión de un proceso industrial," *Revista Gti*, vol. 14, num. 38, pp. 43–58, 2015.

[44] F. J. Mosqueira Sáez, "Tecnologías de comunicación en tiempo real en entornos de automatización industrial. Análisis de la problemática y alternativas," p. 90, 2020.

[45] A. Toro, G. Sánchez, M. Strefezza, y E. Granado, "IIoT y sistemas de control: oportunidades, desafíos y arquitecturas," *Ciencia e Ingeniería*, vol. 38, num. 3, pp. 209–214, 2018.

[46] K. A. Yague Zapata, J. E. Hernández Alvarado, C. F. Trujillo Trujillo, y D. Ricardo Delgado, "Internet industrial de las cosas , evolución y desafíos ." *Universidad Cooperativa de Colombia*, p. 15, 2020.

[47] K. Aron Semle, "Protocolos IoT para considerar," *Aadeca Revista*, p. 34, 2016.

[48] F. Moreno Cerdà, "Demostrador arquitectura publish / subscribe con MQTT," p. 55, 2018.

[49] T. IO. (2021) Overview. [En línea]. Disponible: https://docs.thinger.io/

[50] AMAZON. (2021) What is awss. [En línea]. Disponible: https://aws.amazon.com/es/what-is-aws/

[51] J. Wu, Y. Yang, X. Cheng, H. Zuo, y Z. Cheng, "The development of digital twin technology review," in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 4901–4906.

[52] U. T. Federico, S. Maria, N. A. Isa, P. Ing, y J. S. Lopez, "AMERICA NATIONAL STANDARD Adaptación y traducción Símbolos e Identificaciones," vol. 1984, 2016.

[53] N. RED. (2021) Getting started with node red. [En línea]. Disponible: https://nodered.org/