



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**Implementación de una arquitectura IIoT redundante para el monitoreo en tiempo real de un
proceso de paletizado simulado en un entorno virtual 3D**

Freire Espín, Rodrigo Sebastián

Vicerrectorado de Investigación, Innovación y Transferencia Tecnológica

Centro de Estudios de Posgrado

Maestría en Electrónica y Automatización Mención Redes Industriales

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

Mención Redes Industriales

Ing. Rodríguez Rivera, José María Mgs.

29 de mayo de 2023

Latacunga

Reporte de Verificación de Contenido

Document Information

Analyzed document	TESIS FREIRE RODRIGO.docx (D168890936)
Submitted	2023-05-29 22:05:00
Submitted by	Juan Carlos Altamirano
Submitter email	jc.altamiranoc@uta.edu.ec
Similarity	0%
Analysis address	jc.altamiranoc.uta@analysis.arkund.com

Sources included in the report

W	URL: https://www.innovacion-tecnologia.com/iot/mejores-practicas-de-iiot-disena-como-un-profesional/ Fetched: 2022-08-07 03:34:59	 1
SA	Memoria.pdf Document Memoria.pdf (D109531523)	 2
SA	Trabajo de Grado Andrés Yacelga (2).docx Document Trabajo de Grado Andrés Yacelga (2).docx (D151319323)	 1



.....
Ing. Rodríguez Rivera, José María Mgs.
C.C.: 0501870398
DIRECTOR

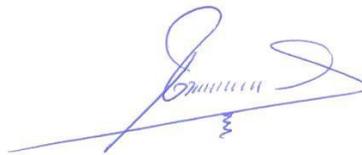
Vicerrectorado de Investigación, Innovación y Transferencia Tecnológica

Centro de Estudios de Posgrado

Certificación

Certifico que el trabajo de titulación, **“Implementación de una arquitectura IIoT redundante para el monitoreo en tiempo real de un proceso de paletizado simulado en un entorno virtual 3D”** fue realizado por el Ing. **Freire Espín, Rodrigo Sebastián**, el mismo que ha sido revisado y analizado en su totalidad, por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Latacunga, 29 de mayo de 2023



.....
Ing. Rodríguez Rivera, José María Mgs.
C.C.: 0501870398
DIRECTOR



Vicerrectorado de Investigación, Innovación y Transferencia Tecnológica

Centro de Estudios de Posgrado

Responsabilidad de Autoría

Yo **Freire Espín, Rodrigo Sebastián**, con cédula de ciudadanía n° **1804335808**, declaro que el contenido, ideas y criterios del trabajo de titulación: **Implementación de una arquitectura IIoT redundante para el monitoreo en tiempo real de un proceso de paletizado simulado en un entorno virtual 3D**, es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 29 de mayo de 2023

.....
Freire Espín, Rodrigo Sebastián
C.C.: 1804335808



Vicerrectorado de Investigación, Innovación y Transferencia Tecnológica

Centro de Estudios de Posgrado

Autorización de Publicación

Yo **Freire Espín, Rodrigo Sebastián**, con cédula de ciudadanía N° **1804335808**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Implementación de una arquitectura IIoT redundante para el monitoreo en tiempo real de un proceso de paletizado simulado en un entorno virtual 3D** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 29 de mayo de 2023

.....
Freire Espín, Rodrigo Sebastián
C.C.: 1804335808

Agradecimiento

Quiero agradecer a Dios por la salud y fortaleza para finalizar esta etapa de estudios, a mis padres y hermanos, agradezco su apoyo y cariño incondicional que me han acompañado durante mi carrera. También quiero agradecer a mi tutor, José María Rodríguez, por su valiosa guía académica durante el desarrollo de este trabajo de titulación.

Finalmente, agradezco a la Universidad de las Fuerzas Armadas ESPE, a los profesores y coordinadores de la Maestría en Electrónica y Automatización Mención Redes Industriales, por su acertada gestión en el desarrollo de este programa de estudios.

Freire Espín, Rodrigo Sebastián

Dedicatoria

Este trabajo dedico de manera especial a mis padres Germán y Wilma, a mi familia querida, y a todas las personas que me apoyaron durante el desarrollo del presente trabajo. Gracias por ser parte de este logro académico.

Freire Espín, Rodrigo Sebastián

ÍNDICE DE CONTENIDO

Carátula.....	1
Reporte de Verificación de Contenido.....	2
Certificación	3
Responsabilidad de Autoría.....	4
Autorización de Publicación.....	5
Agradecimiento	6
Dedicatoria.....	7
Índice de Contenido	8
Índice de Figuras	12
Índice de Tablas	16
Acrónimos	17
Resumen	18
Abstract.....	19
Capítulo I: Características Generales del Proyecto	20
Antecedentes	20
Líneas de Investigación.....	21
Sub Líneas de Investigación	21
Área de Influencia	21
Planteamiento del Problema	21
Justificación, Importancia y Alcance del Proyecto.....	22
Objetivo General.....	24
Objetivos Específicos.....	24
Hipótesis de Investigación	24

Categorización de las Variables de Investigación	24
Capítulo II: Marco Teórico	26
Automatización Industrial	26
Plataforma Factory IO	26
Plataforma SoftLogix.....	27
Sistemas de Control Supervisorio y Adquisición de Datos SCADA.....	28
Sistemas de Ejecución de Manufactura MES	30
Industria 4.0	31
Protocolo MQTT.....	32
IIoT.....	33
Arquitecturas Redundantes IIoT	34
Cloud Computing	34
Plataforma Ignition	35
Capítulo III: Análisis y Diseño:.....	41
Conectividad a Internet.....	41
Hardware y Software	41
Licencias	42
Requisitos de Red	42
Descripción del Proceso de Paletizado	43
Configuración en Plataforma Virtual Factory IO	46
Configuración SoftLogix.....	48
Programación del Software RSLogix 5000 para el Proceso de Paletizado.....	49
Programas y Rutinas.	50
Arquitectura IIoT Redundante Planteada.....	51

Configuración de Ignition Edge en Raspberry PI.	52
Configuración del Broker Primario MQTT.....	57
Configuración del Broker MQTT de Respaldo	61
Configuración del Gateway SCADA-MES	62
Configuración de la Base de Datos.....	63
Configuración de Módulos MES.	65
Diseño de la Aplicación SCADA-MES	66
Evaluación Inicial	67
Lectura de los Tags MQTT.....	69
Desarrollo de la Aplicación SCADA-MES.....	71
Pantallas Ancladas	71
Topología	72
Parámetros.....	72
Vista General	73
Pesaje.....	73
Paletizado.....	74
Almacenamiento.....	74
Alarmas	75
MES.....	75
Capítulo IV: Implementación	77
Prueba de Comunicación entre Publicadores y Suscriptores	77
Prueba de Redundancia y Comunicación de Dispositivos MQTT.....	77
Pruebas de Desempeño de la Aplicación SCADA-MES.....	80
Evaluación de Historización de Variables.	85

Pruebas de Desempeño de la Red en Tiempo Real	86
Validez de la Hipótesis	88
Conclusiones.....	89
Recomendaciones.....	91
Bibliografía	93
Anexos	96

ÍNDICE DE FIGURAS

Figura 1 <i>Interfaz gráfica industrial mediante Factory IO</i>	27
Figura 2 <i>Selección de módulo y tarjeta en Plataforma SoftLogix</i>	28
Figura 3 <i>Pantalla SCADA Ignition</i>	30
Figura 4 <i>Estructura de un sistema MQTT</i>	33
Figura 5 <i>Esquema de funcionamiento del Cloud Computing</i>	35
Figura 6 <i>Módulo Ignition MQTT Transmission</i>	36
Figura 7 <i>Módulo Ignition MQTT Engine</i>	37
Figura 8 <i>Módulo Ignition MQTT Distributor</i>	38
Figura 9 <i>Gestión de parámetros con módulos especiales MES Ignition</i>	40
Figura 10 <i>Detalle de proceso de clasificación de peso</i>	43
Figura 11 <i>Detalle de proceso de paletizado</i>	44
Figura 12 <i>Detalle de bandas de transporte</i>	44
Figura 13 <i>Detalle de sistema de almacenamiento automático</i>	45
Figura 14 <i>Detalle de sistema de sistema de extracción de pallets</i>	46
Figura 15 <i>Componentes de la planta virtual de paletizado</i>	47
Figura 16 <i>Enlace del software Factory IO con el PLC virtual mediante Ethernet</i>	47
Figura 17 <i>Asignación de etiquetas de variables de Factory IO hacia el PLC</i>	48
Figura 18 <i>Configuración de tarjeta de red Ethernet de Softlogix</i>	48
Figura 19 <i>Configuración del PLC virtual</i>	49
Figura 20 <i>Comunicación del PLC virtual con el software RSLogix 5000</i>	50
Figura 21 <i>Detalle de rutinas creadas en el PLC</i>	51
Figura 22 <i>Detalle de componentes de la arquitectura IIoT redundante</i>	52
Figura 23 <i>Esquema de funcionamiento de software Ignition Edge</i>	52

	13
Figura 24 <i>Configuración de driver OPC UA para controladores Logix 5000</i>	53
Figura 25 <i>Verificación de lectura de datos del PLC mediante el protocolo OPC-UA</i>	53
Figura 26 <i>Configuración de módulo de Transmisión MQTT</i>	54
Figura 27 <i>Configuración de protocolo Sparkplug B en el módulo de Transmisión MQTT</i>	55
Figura 28 <i>Selección de rutas de tags a publicar por el módulo de Transmisión MQTT</i>	55
Figura 29 <i>Verificación de comunicación de tags MQTT según Sparkplug B</i>	56
Figura 30 <i>Ubicación de tags del PLC hacia el árbol de tags MQTT</i>	56
Figura 31 <i>Características de la máquina virtual del broker primario MQTT</i>	57
Figura 32 <i>Configuración de apertura de puertos TCP en la máquina virtual</i>	58
Figura 33 <i>Acceso a máquina virtual mediante aplicativo RDP</i>	58
Figura 34 <i>Configuración del broker dentro del módulo MQTT</i>	59
Figura 35 <i>Configuración de usuario autenticado para acceso a los paquetes MQTT</i>	60
Figura 36 <i>Configuración de modo de redundancia en el broker primario</i>	60
Figura 37 <i>Detalle de configuración de broker MQTT de respaldo</i>	61
Figura 38 <i>Detalle de configuración de redundancia en broker de respaldo</i>	62
Figura 39 <i>Configuración de brokers MQTT en el módulo de Recepción</i>	62
Figura 40 <i>Verificación de configuración activa de protocolo Sparkplug B</i>	63
Figura 41 <i>Creación de base de datos a través de MySQL Workbench</i>	63
Figura 42 <i>Detalle de configuración de conexión a plataforma Ignition</i>	64
Figura 43 <i>Verificación de validez de conexión de la base de datos</i>	64
Figura 44 <i>Detalle de instalación de módulos MES en el aplicativo Ignition</i>	65
Figura 45 <i>Detalle de enlace de base de datos al módulo de Producción</i>	65
Figura 46 <i>Asignación del proyecto de edición para enlace del módulo de Producción</i>	66
Figura 47 <i>Detalle de autenticación de usuario en el software de edición</i>	67

	14
Figura 48 <i>Detalle de asignación de la base de datos en el software de edición</i>	67
Figura 49 <i>Verificación del módulo de Producción</i>	68
Figura 50 <i>Verificación de status de brokers MQTT</i>	68
Figura 51 <i>Comunicación entre publicador/suscriptor MQTT a través del broker primario</i>	69
Figura 52 <i>Detalle de revisión de propiedades de Lectura/Escritura (Read/Write)</i>	70
Figura 53 <i>Verificación de tags publicados por el PLC a través de la red IIoT</i>	70
Figura 54 <i>Menú de navegación para acceso a pantallas de control, monitoreo y alarmas</i>	71
Figura 55 <i>Detalle de arquitectura de la red con elementos y protocolos de arquitectura IIoT</i>	72
Figura 56 <i>Detalle de visualización de parámetros de red de los dispositivos IIoT</i>	72
Figura 57 <i>Detalle de interfaz general de proceso de paletizado</i>	73
Figura 58 <i>Detalle de interfaz de proceso de pesaje</i>	73
Figura 59 <i>Detalle de interfaz de proceso de paletizado</i>	74
Figura 60 <i>Detalle de interfaz de almacenamiento</i>	74
Figura 61 <i>Detalle de visualización de alarmas</i>	75
Figura 62 <i>Detalle de interfaz de módulo de Recetas del sistema MES</i>	75
Figura 63 <i>Jerarquización del proceso productivo MES según ISA-95</i>	76
Figura 64 <i>Interfaz de comprobación de redundancia de la red IIoT</i>	77
Figura 65 <i>Interfaz de comprobación de conexión de broker primario</i>	78
Figura 66 <i>Interfaz de respuesta del sistema por desconexión de broker primario</i>	78
Figura 67 <i>Interfaz de detección de pérdida de conexión de broker primario</i>	79
Figura 68 <i>Pruebas de conexión y desconexión de nodos de la red IIoT</i>	79
Figura 69 <i>Verificación de pérdida de comunicaciones con dispositivos IIoT</i>	80
Figura 70 <i>Detalle de comprobación de setpoints en las pantallas del sistema SCADA-MES</i>	81
Figura 71 <i>Pantalla SCADA – comando de arranque de secuencias</i>	82

	15
Figura 72 <i>Factory IO – Líneas de paletizado en ejecución</i>	82
Figura 73 <i>Factory IO – Bandas de transporte en ejecución</i>	83
Figura 74 <i>Factory IO – Proceso de almacenamiento automático en marcha</i>	83
Figura 75 <i>Factory IO – Proceso de almacenamiento automático finalizado</i>	84
Figura 76 <i>Pantalla SCADA – Comando de arranque para despacho según receta</i>	84
Figura 77 <i>Factory IO – Despacho finalizado</i>	85
Figura 78 <i>Verificación de almacenamiento de datos de líneas de producción</i>	85
Figura 79 <i>Verificación de almacenamiento de recetas en base de datos</i>	86
Figura 80 <i>Detalle de análisis de velocidad de comunicación en la arquitectura IIoT</i>	87

ÍNDICE DE TABLAS

Tabla 1 <i>Operalización de las variables de investigación</i>	25
Tabla 2 <i>Requisitos para implementación de arquitectura IIoT</i>	41
Tabla 3 <i>Direccionamiento IP de los dispositivos dentro la red IIoT</i>	42
Tabla 4 <i>Pruebas de comando “ping” para verificar comunicación entre dispositivos IIoT</i>	77
Tabla 5 <i>Detalle de selección de setpoints</i>	81
Tabla 6 <i>Detalle de parámetros de velocidad de transmisión de paquetes MQTT</i>	87

Acrónimos

3D: Tres Dimensiones

MQTT: Message Queuing Telemetry Transport (Transporte de Telemetría de Mensajes en Cola)

PLC: Programmable Logic Controller (Controlador Lógico Programable)

IT: Information Technologies (Tecnologías de Información)

OT: Operation Technologies (Tecnologías de la Operación)

IIoT: Industrial Internet of Things (Internet Industrial de las Cosas)

SCADA: Supervisory Control and Data Acquisition (Supervisión, Control y Adquisición de Datos)

MES: Manufacturing Execution System (Sistemas de Ejecución de Manufactura)

MOM: Manufacturing Operations Management (Gestión de Operaciones de Manufactura)

ERP: Enterprise Resource Planning (Planificación de Recursos Empresariales)

LAN: Local Area Network (Red de Área Local)

OPC: Open Platform Communications (Plataforma de Comunicación Abierta)

OPC UA: OPC Unified Architecture (OPC Arquitectura Unificada)

TCP: Transmission Control Protocol (Protocolo de Control de Transmisión)

IP: Internet Protocol (Protocolo de Internet)

RDP: Remote Desktop Protocol (Protocolo de Escritorio Remoto)

SQL: Structured Query Language (Lenguaje de Consulta Estructurado)

ISA: International Society of Automation (Sociedad Internacional de Automatización)

RFID: Radio Frequency Identification (Identificación por Radio Frecuencia)

Resumen

En el presente trabajo se realizó la implementación de una arquitectura IIoT redundante para monitorear en tiempo real un proceso de paletizado simulado en una plataforma virtual 3D, el cual fue desarrollado con el fin de simular una línea de producción y extraer información para el monitoreo remoto desde un sistema SCADA - MES (Supervisory Control and Data Acquisition – Manufacturing Execution System). Para esto, se utilizó el *Software Factory IO* para armar el proceso, el cual contó con dos líneas de paletizado de cajas. Los *pallets* resultantes fueron transportados y almacenados en un estante. Se utilizaron dos robots cartesianos: uno para el almacenamiento de los *pallets* y otro para el despacho de los mismos, y una vez armado el proceso con todos sus componentes, se programó una secuencia automática en un PLC (*Programmable Logic Controller*) virtual de la marca *Allen Bradley*, específicamente de la familia *SoftLogix*, para comunicarse con la red física y publicar sus datos a las aplicaciones que lo requieran. Una vez probada la secuencia del proceso, se realizó la implementación de una arquitectura IIoT redundante basada en la plataforma *Ignition*, para monitorear el proceso en tiempo real desde un cliente SCADA - MES. La redundancia de la arquitectura está representada por dos servidores (*broker*) MQTT (*Message Queue Telemetry Transport*), uno de los cuales se encuentra alojado en la nube, y otro en la red LAN; de tal forma que, si uno de ellos pierde la comunicación con el PLC, el otro *broker* toma el control de inmediato, permitiendo que el cliente SCADA - MES no pierda ningún dato del proceso, lo que es de gran importancia en aplicaciones industriales. Una vez armada la arquitectura se evaluó su desempeño, y se comprobó la redundancia de los *broker* MQTT.

Palabras clave: IoT, MQTT, SCADA, MES, Industria 4.0.

Abstract

In the present work, the implementation of a redundant IIoT architecture was carried out to monitor in real time a simulated palletizing process in a 3D virtual platform, which was developed in order to simulate a production line and extract information for remote monitoring from a SCADA - MES (Supervisory Control and Data Acquisition – Manufacturing Execution System) system. For this, the Factory IO software was used to build the process, which had two box palletizing lines. The resulting pallets were transported and stored on a rack. Two Cartesian robots were used: one to store the pallets and the other to dispatch them, and once the process was assembled with all its components, an automatic sequence was programmed in a virtual PLC (Programmable Logic Controller) of the brand Allen Bradley, specifically from the SoftLogix family, to communicate with the physical network and publish its data to the applications that require it. Once the process sequence was tested, a redundant IIoT architecture based on the Ignition platform was implemented to monitor the process in real time from a SCADA - MES client. The redundancy of the architecture is represented by two MQTT (Message Queue Telemetry Transport) servers (*broker*), one of which is hosted in the cloud, and another in the LAN; in such a way that, if one of them loses communication with the PLC, the other *broker* takes control immediately, allowing the SCADA - MES client not to lose any data from the process, which is of great importance in industrial applications. Once the architecture was assembled, its performance was evaluated, and the redundancy of the MQTT *brokers* was verified.

Keywords: IoT, MQTT, SCADA, MES, Industry 4.0.

Capítulo I

Características Generales del Proyecto

Antecedentes

Se puede definir al Internet Industrial de las Cosas (IIoT) como la combinación de las Tecnologías de la Información (IT) con las Tecnologías de Operaciones (OT), con el fin de allanar el camino hacia una nueva era tecnológica, que transformará radicalmente las cadenas de valor de la industria y la producción. La nueva era de interconectividad y digitalización ofrece a los líderes empresariales la oportunidad de explorar nuevos medios para lograr sus objetivos. IIoT en realidad, es una evolución que tiene sus orígenes en tecnologías y funcionalidades desarrolladas por proveedores de Automatización visionarios hace más de quince años. Sin embargo, a medida que los estándares globales necesarios maduran, puede llevar otros quince años desarrollar todo el potencial de IIoT (Conway, 2015). Durante este período de tiempo, los cambios en la industria serán de gran alcance. Es por esto que IIoT se presenta como una revolución que está cambiando la faz de la industria de manera profunda.

IIoT se puede caracterizar como una gran cantidad de sistemas industriales conectados, que se comunican y coordinan sus análisis de datos y sus acciones para mejorar el desempeño industrial. La visión de IIoT es un mundo donde los activos inteligentes conectados operan como parte de grandes sistemas que componen la fabricación inteligente dentro de una empresa (Karmakar, Dey, Baral, Chowdhury, & Rehan, 2019). Es por eso que, hoy en día, IIoT está ayudando a mejorar la productividad, reducir los costos operativos y mejorar la seguridad de los trabajadores.

Si se habla del Internet Industrial de las Cosas, es inevitable no pensar en soluciones basadas en la nube. Es decir, tecnologías que permitan acceder remotamente a bases de datos, aplicaciones y sistemas de monitoreo desde cualquier lugar del mundo y en cualquier momento, sin necesidad de conectarse a un ordenador personal o servidor local.

Las soluciones basadas en la nube, han demostrado ser excelentes aliadas en las estrategias IIoT,

y, aunque aún quedan muchos desafíos por recorrer y barreras por superar, éste es el camino por el cual tarde o temprano las empresas deberán recorrer para alcanzar la transformación hacia la cuarta revolución industrial.

Líneas de Investigación

- Automática y Control
- Tecnologías de la Información y Comunicación

Sub Líneas de Investigación

- Protocolos de Comunicación

Área de Influencia

- Gestión de Procesos de Manufactura

Planteamiento del Problema

A nivel funcional, los sistemas de automatización se desarrollaron originalmente para controlar procesos en tiempo real. Sin embargo, la mayor parte del control de procesos implementados hasta la fecha se ha centrado en mejorar la eficiencia de un proceso o máquina, y no en el análisis productivo en conjunto.

Los factores operacionales importantes como la rentabilidad, la fiabilidad y la disponibilidad, que tradicionalmente se han gestionado por el talento humano, ahora necesitan ser analizados en tiempo real. Las decisiones influenciadas por controles manuales o automáticos deben tomarse lo suficientemente rápido como para impactar positivamente la operación de un proceso.

Se estima que, en el futuro, el 72% de las compañías pueden perder participación de mercado si no implementan una estrategia de datos de gran tamaño (Jaidka, Sharma, & Singh, 2020). Ante esta problemática, muchas empresas implementan soluciones de interconectividad en los diferentes niveles de producción, con el fin de obtener datos y procesarlos en tiempo real.

Justificación, Importancia y Alcance del Proyecto

El presente proyecto se basa en una necesidad real de la industria, ya que a medida que el Internet de las cosas Industrial (IIoT) va ganando terreno en los nuevos modelos de integración en los procesos productivos, los profesionales dedicados a la automatización y control de procesos deben conocer cómo funcionan estas plataformas, y, con este conocimiento, ofrecer soluciones reales a las empresas, que les permitan implementar este nuevo tipo de tecnologías.

Muchos empresarios muestran entusiasmo al tratar temas como Industria 4.0, el Internet de las Cosas y Transformación Digital, sin embargo, gran parte no sabe por dónde empezar, ya que hay muchos factores importantes que influyen a la hora de tomar decisiones sobre un tema tan complejo.

Por esta razón, este proyecto demostrará que es completamente posible implementar una arquitectura IIoT redundante que sea sólida, confiable, segura, que se comuniquen con el proceso y lo monitoree en tiempo real.

Por otro lado, este proyecto demuestra que no se necesita operar una línea de producción real para evaluar una arquitectura IIoT, ya que se hará uso de una plataforma virtual 3D, en la cual se armará el proceso de paletizado. Es decir, se instalarán virtualmente las bandas transportadoras, mesas de pesaje, módulos de paletizado, sensores, motores, robots cartesianos y estante de almacenamiento, con lo cual se tiene prácticamente una “línea de producción” con todos sus componentes.

Uno de los puntos más importantes es la plataforma sobre la que se va a desarrollar la arquitectura IIoT. Existen varias plataformas que permiten desarrollar este tipo de aplicaciones. Sin embargo, este proyecto utilizará la plataforma *Ignition*, la cual es basada en web y contiene:

- Módulos de comunicación OPC UA para lectura de datos del PLC.
- Módulo de transformación de datos de OPC UA a MQTT, para enviarlos a la nube.
- Herramientas para la creación de un sistema SCADA de alto desempeño.

- Módulo con funcionalidades de un sistema MES.
- Módulos para conexión rápida y segura a bases de datos.
- Módulos completos de configuración de *broker* MQTT en la nube.

Por lo tanto, no se deberá recurrir a la instalación de diferentes plataformas, fabricantes, licencias o *drivers*, sino que, al trabajar en una única plataforma, la integración se vuelve más fácil.

En resumen, este proyecto beneficiará a profesionales en el área de automatización y control que desean conocer cómo implementar una arquitectura IIoT en una planta real; y también a profesionales del área de gestión de manufactura, ya que podrán analizar las ventajas de integrar sus procesos productivos en una plataforma IIoT.

El alcance del presente proyecto se limitará a realizar una secuencia automática de un proceso de paletizado simulado en una plataforma virtual 3D, mediante la programación de un PLC virtual *SoftLogix*.

Una vez automatizado el proceso, se configurarán dos *broker* MQTT, uno en la nube y otro en la red local, con el fin de implementar redundancia en la red; y se desarrollará una aplicación SCADA - MES, para el monitoreo del proceso. También se enlazará una base de datos al *Gateway* del sistema SCADA – MES.

Respecto a las limitaciones, se contempla la licencia de la plataforma *Ignition*, ya que el tiempo de conexión en modo “*trial*” es de dos horas, con lo cual se necesitará reiniciar la conexión una vez transcurrido este tiempo. Sin embargo, no es un obstáculo para el desarrollo íntegro del proyecto, ya que, al reiniciar las conexiones, se obtendrá acceso nuevamente a toda la información requerida, y el monitoreo y análisis del proceso podrá desarrollarse con total normalidad.

Considerando que MQTT optimiza y usa un ancho de banda limitado, como parte de la evaluación de desempeño se determinará si el tiempo de respuesta de MQTT es suficiente para aplicaciones MES, control y monitoreo en tiempo real de un proceso industrial.

Objetivo General

Implementar una arquitectura IIoT redundante para el monitoreo en tiempo real de un proceso de paletizado simulado en un entorno 3D.

Objetivos Específicos

- Programar una secuencia automática de un proceso de paletizado simulado en un entorno 3D, haciendo uso de un PLC virtual.
- Diseñar una arquitectura IIoT redundante a través del protocolo MQTT.
- Configurar dos *broker* MQTT, uno principal y otro *backup* para gestionar la red y la transmisión de datos.
- Desarrollar una aplicación SCADA - MES y enlazarla a los *broker* redundantes, para monitorear en tiempo real del proceso de paletizado.
- Enlazar una base de datos al *Gateway* SCADA - MES, para historizar y tratar las variables importantes de proceso.
- Evaluar el desempeño de la arquitectura implementada.

Hipótesis de Investigación

La implementación de una arquitectura IIoT redundante permitirá el monitoreo en tiempo real de un proceso de paletizado simulado en un entorno virtual 3D.

Categorización de las Variables de investigación

Variables Independientes:

- Arquitectura IIoT redundante

Variables Dependientes:

- Monitoreo en tiempo real

Tabla 1

Operalización de las variables de investigación

VARIABLES	TIPO	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
Arquitectura IloT redundante	Independiente	Plataforma redundante de comunicación de dispositivos de campo con aplicaciones alojadas en la nube orientadas al control industrial	Sistema de comunicación que enlaza un PLC virtual con un cliente SCADA -MES a través de dos <i>broker</i> MQTT redundantes	-Disponibilidad -Transmisión por diferentes medios -Seguridad	- Mínimo un canal redundante -Al menos dos formas de transmisión a través de TCP/IP -Porcentaje de efectividad contra accesos no autorizados
Monitoreo en tiempo real	Dependiente	Sistema que interactúa activamente con un entorno accediendo a la información tan pronto como ésta es generada	Aplicaciones web que intercambian información con un proceso de paletizado de manera instantánea, controlando o modificando su operación	-Velocidad de transmisión -Confiabilidad de la comunicación -Facilidad de operación	-Actualización de datos en menos de 2 segundos -Porcentaje de variables transmitidas exitosamente

Capítulo II

Marco Teórico

Automatización Industrial

La automatización es el mejoramiento de los procesos de producción a través de la implementación de elementos computarizados, eléctricos, mecánicos, neumáticos e hidráulicos, que interactúan de forma electromecánica, incrementando la eficiencia de los procesos, gestión de suministros y materia prima, eficiencia en procesos de manufactura y producción, incrementando de esta manera la productividad y bienestar de los trabajadores (Escaño, García, & García, 2019).

La importancia de la automatización industrial radica en la capacidad de disminuir costos de producción asociados a los procesos que implican costes laborales por actividades que se realizan de una manera manual. Con este avance, se logra un ahorro en procesos de producción mediante la sustitución de sistemas electromecánicos programados para desarrollar el proceso de una manera automática. De esta manera, se optimizan costos por mano de obra y se dispone del personal para el desarrollo de actividades adicionales que permitan el crecimiento de las organizaciones. Por esta razón, la adopción de proyectos de automatización industrial es de vital importancia para ingresar en el comercio internacional, porque incrementa la eficiencia en la producción de bienes y servicios, fortaleciendo la competitividad de un entorno local (Acharya & Kumar, 2018).

Plataforma Factory IO

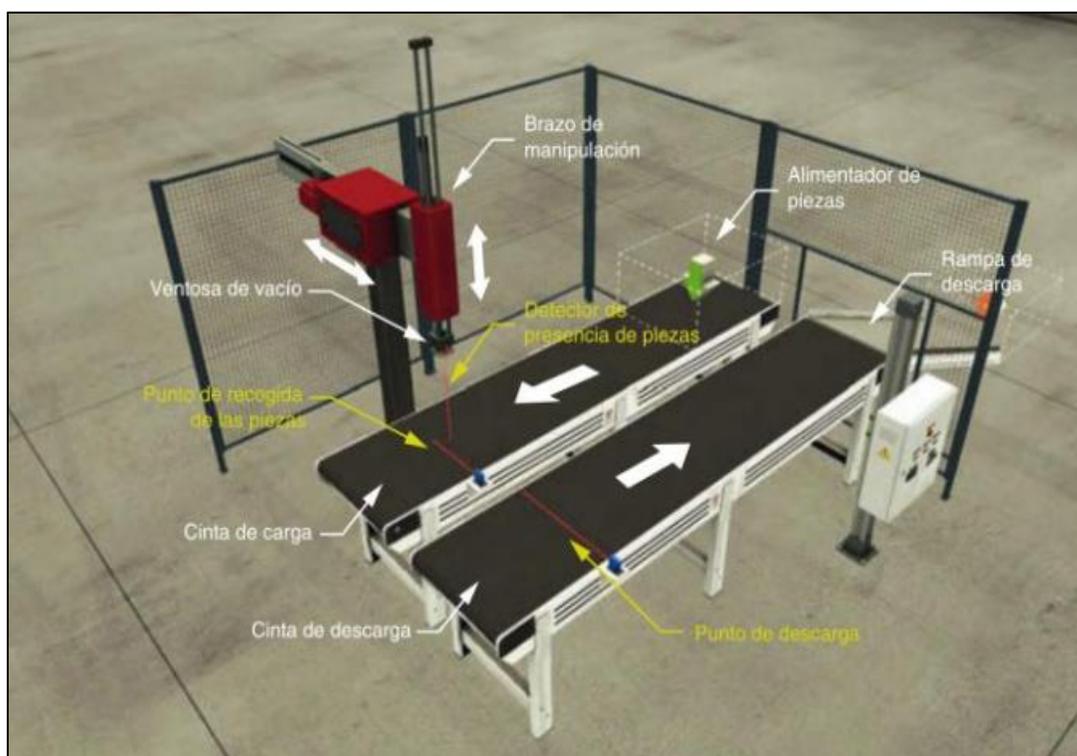
La plataforma virtual de simulación industrial *Factory IO* es un *software* desarrollado por la empresa *Real Games*, que permite crear un proceso industrial virtual con elementos tales como sensores, bandas transportadoras, robots, etc., los mismos que se pueden integrar virtualmente de acuerdo a los escenarios de funcionamiento de una industrial real. Utiliza periféricos de entrada y salida (E/S) de señales digitales para accionamiento y parada de maquinaria, y dispositivos de entrada y salida analógicas, para controles de valores variables como nivel de líquidos o temperatura. Dispone

también de controladores para interactuar con PLC, protocolos como *Modbus*, y demás de una amplia variedad de marcas industriales como *Allen Bradley*, *Siemens*, etc., (Factoryio, 2023).

La capacidad de simulación de procesos industriales *Factory IO* permite el diseño virtual de procesos reales, ya que cuenta con una extensa biblioteca con mecanismos, análisis de fallas, protocolos de comunicación, y la capacidad de integrar elementos de control a un PLC en tiempo real (Sánchez, 2018).

Figura 1

Interfaz gráfica industrial mediante Factory IO



Nota. Tomado de Castillo (2021).

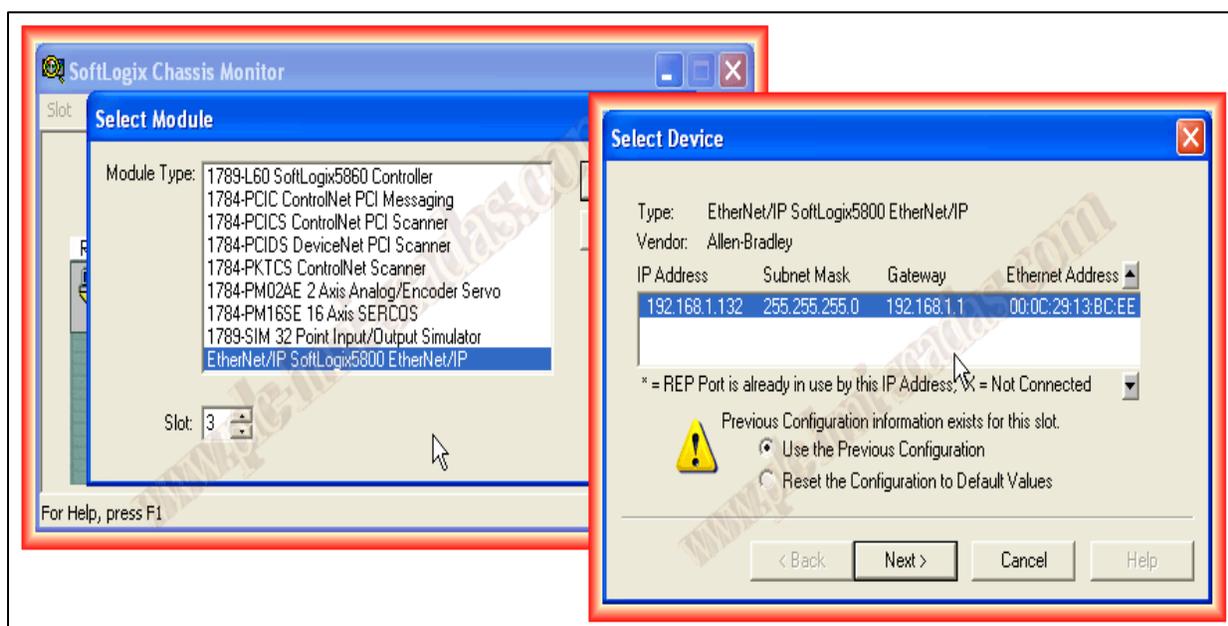
Plataforma SoftLogix

La plataforma *SoftLogix* permite configurar arquitecturas de automatización de un nivel complejo, las cuales utilizan un *CPU*, módulos digitales o análogos, redes computacionales o redes industriales de control (Díaz, Cruz, & Botto-Tobar, 2022).

La utilidad de *SoftLogix* radica en que permite utilizar la computadora como un PLC, es decir, simula módulos de entradas, salidas y comunicación, por lo que las pruebas de programación del PLC resultan más reales, ya que cuenta con alta flexibilidad en su programación y múltiples opciones de comunicación, admiten varias interfaces de entrada y salida, y proporciona conexión a redes abiertas como *EtherNet/IP*, *ControlNet* y *DeviceNet* (Rockwell Automation, 2023).

Figura 2

Selección de módulo y tarjeta en Plataforma SoftLogix



Nota. Tomado de Luque (2023)

Sistemas de Control Supervisorio y Adquisición de Datos SCADA

El Sistema SCADA significa *Supervisory Control and Data Acquisition* (Supervisión, Control y Adquisición de Datos); es un *software* que se ejecuta en una computadora con el objetivo de entablar una comunicación y control con los procesos de una planta industrial, a través de una interfaz gráfica que muestra esquemáticamente los procesos para monitoreo en tiempo real, análisis de datos provenientes de los elementos que forman parte del proceso, generación de alarmas y notificaciones a los operarios y control de equipos de una manera remota mediante el encendido o apagado de

motores, actuadores y válvulas.

Las funciones de un sistema SCADA son la adquisición de datos para su almacenamiento y procesamiento, la supervisión del proceso industrial desde una interfaz que muestra gráficamente el estado y comportamiento en tiempo real de las variables de proceso, y el control de dichas variables mediante la actuación de reguladores y actuadores. Adicionalmente, un sistema SCADA permite la transmisión de información y gestión de bases de datos provenientes del comportamiento del proceso (Gallardo, 2013).

En un sistema SCADA se pueden distinguir cinco elementos estructurales:

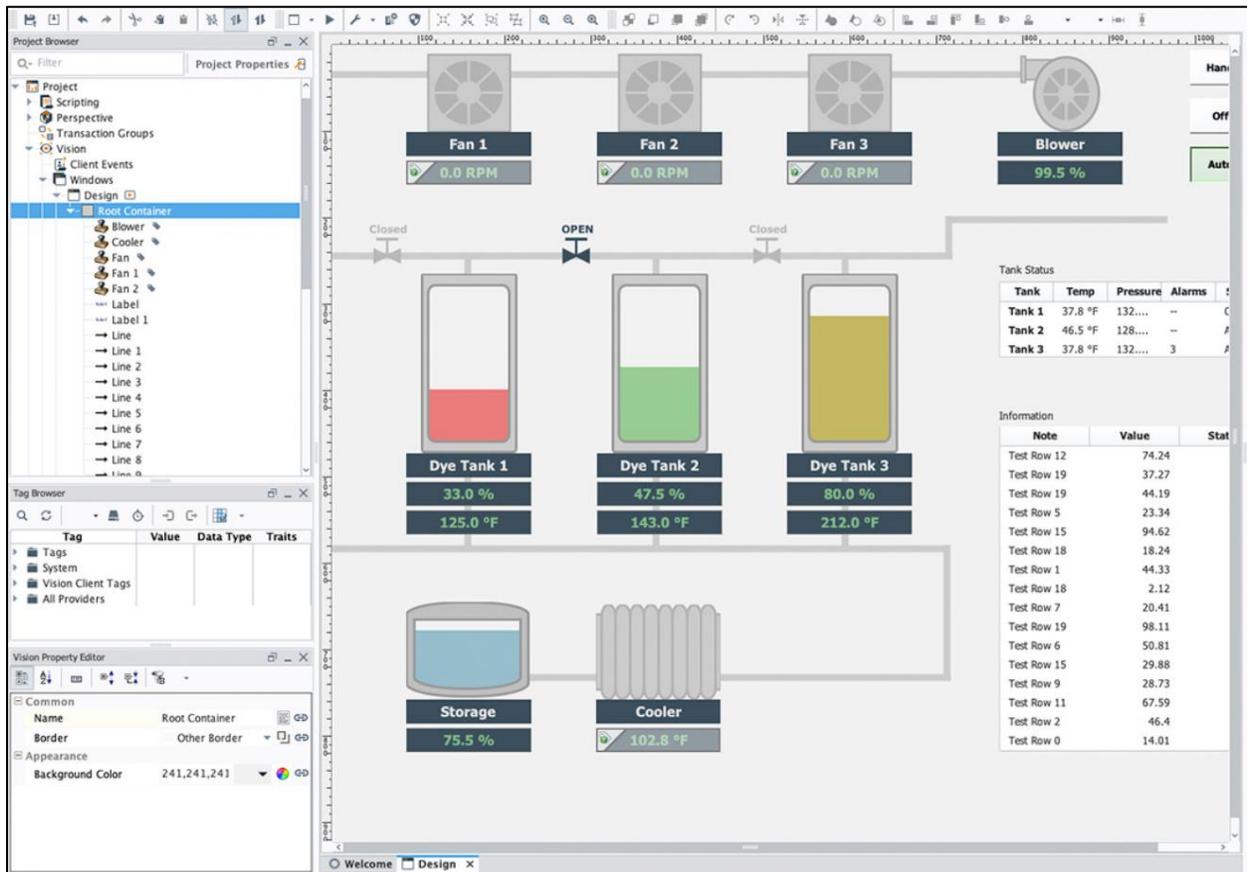
- *Bloque de configuración*, para definir el entorno de trabajo específico del proceso.
- *Interfaz para el operador*, conformada por un diseño esquemático gráfico para poder identificar por parte del operador los elementos y valores del proceso.
- *Bloque del proceso*, que es el conjunto de instrucciones necesarias para ejecutarse a partir de las variables de los elementos que conforman el sistema.
- *Gestión de base de datos*, para almacenar la información generada a partir del proceso.
- *Comunicaciones*, para administrar el intercambio de información entre el Sistema de Control de Adquisición de Datos y los elementos que conforman el proceso.

Finalmente, a nivel de *hardware*, un SCADA se compone de la instrumentación de campo, ubicada físicamente en el sitio del proceso y conformada por elementos tales como sensores, motores, actuadores y válvulas; una red de comunicación para interconectar la instrumentación de campo con el sistema de gestión del proceso; unidades terminales remotas, que forman parte del sistema de automatización del proceso; y finalmente, la unidad terminal máster, que es el equipo central desde el cual se administra todo el sistema SCADA (Gallardo, 2013).

Una pantalla gráfica de un sistema SCADA se muestra en la figura 3:

Figura 3

Pantalla SCADA Ignition



Nota. Tomado de *Inductive Automation*

Sistemas de Ejecución de Manufactura MES

Un Sistema de Ejecución de Manufactura (MES) es una plataforma informática que integra y controla elementos que forman parte de una planta industrial. Es un sistema que incrementa la eficiencia en los procesos de producción de la planta. Tiene a su cargo funciones de monitoreo, seguimiento, documentación y control de procesos desde las etapas iniciales (recepción de materia prima) hasta la entrega del producto final, brinda información al departamento de estrategia de las organizaciones para mejorar la eficiencia de la planta y optimizar recursos de producción (SAP Insights, 2023).

Entre los beneficios de contar con una plataforma industrial MES, se encuentra un mejor control de la transmisión de información en tiempo real; mejor coordinación con departamentos de compras, logística y producción para lograr una reducción de inventarios, mejora el seguimiento de un producto mediante registros actualizados e integra en tiempo real cronogramas de mantenimiento en la producción para mejorar la gestión de productos, insumos, repuestos y servicios suplementarios.

Actualmente, las plataformas MES se rigen por el estándar internacional ISA-95, que permite normalizar las funciones de control estratégicas de la empresa, y permite una comunicación efectiva entre proveedores y fabricantes. Este estándar es muy importante, ya que engloba bajo una norma común diferentes actividades relacionadas a procesos de manufactura en la industria, y su modelo de operación se basa en definir una jerarquía basada en recursos, los cuales se engloban en tres clases: *Personal*, que comprenden individuos o categorías de individuos; *Equipo*, que comprende clases o piezas de un equipo que se utilizan como parte del desempeño del proceso de producción; y *Material*, que comprende lotes y sublotos requeridos como parte de la definición de un producto (Zurawski, 2018).

Finalmente, es posible integrar una plataforma MES con los sistemas ERP, ya que el sistema MES se orienta a la búsqueda de información para mejorar la eficiencia en el manejo de recursos operativos, mientras que un ERP se orienta a la planificación de los programas de la planta (SAP Insights, 2023).

Industria 4.0

La Industria 4.0. es el término que se ha dado a una última revolución industrial que ha logrado un cambio general en el mercado internacional, en los procesos de fabricación de productos, provisión de servicios, mediante la implementación de tecnología disruptiva en la automatización y digitalización de las empresas. Incorpora a los procesos industriales nuevas tecnologías de vanguardia de actualización rápida y permanente como el Internet de las Cosas (IoT), *Cloud Computing*,

Inteligencia Artificial y *Machine Learning* (IBM, 2023). Sus orígenes se remontan a Alemania en el año 2011 como una innovación a presentarse en una feria industrial internacional y ha ido tomando fuerza y relevancia incluso como una estrategia tecnológica internacional (Aguiló, 2019).

La revolución industrial 4.0 logra que una organización industrial sea inteligente en toda su cadena de suministro, e interactúe con un sistema integral que permite una respuesta rápida a los cambios internos y externos que se presenten. Dentro de estas tecnologías sobresale el *Big Data*, que sirve para registrar toda la información de procesos y actividades industriales, y permite potenciar con Inteligencia Artificial para agilizar la toma de decisiones.

Adicionalmente, utiliza recursos tecnológicos como impresión aditiva 3D, robots autónomos para sustitución de procesos manuales tradicionales, ciberseguridad para protección de datos y accesos no autorizados a información digital generada por las empresas. La realidad aumentada es un recurso importante para utilizarse en la industria 4.0, así como la computación en la nube y la integración horizontal y vertical para unir todos los departamentos de la industria y mejorar su comunicación e interacción (SAP, 2023).

Protocolo MQTT

El protocolo MQTT (*Message Queuing Telemetry Transport*) es un protocolo diseñado para la comunicación "*máquina a máquina*" de pequeñas cantidades de información. Se ejecuta sobre TCP/IP y tiene un tamaño de paquete de datos con un mínimo de gastos generales bajos (> 2 bytes).

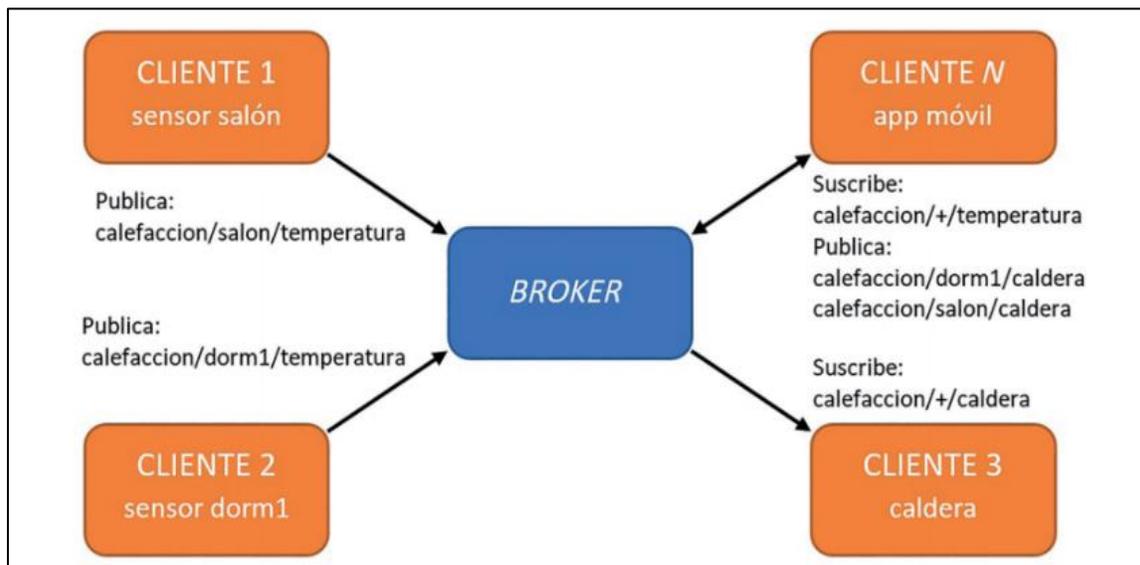
Es un protocolo desarrollado en 1999 y puede transmitir datos en varias formas, como datos binarios, texto, *XML* o *JSON*, y utiliza un modelo de "*publicación/suscripción*" en lugar del modelo tradicional "*cliente/servidor*". La ventaja del modelo de "*publicación/suscripción*", es que no hay interacción directa entre el emisor de datos (publicador) y el receptor de datos (suscriptor), debido a que hay un intermediario (*broker*) que gestiona la comunicación entre ambos participantes.

Un sistema común de MQTT requiere dos componentes principales de *software*:

- El cliente MQTT que se instalará en un dispositivo.
- Un *broker* MQTT que sirve para gestionar los datos de publicación y suscripción.

Figura 4

Estructura de un sistema MQTT



Nota. Tomado de Peláez (2020)

IloT

El Internet Industrial de las Cosas es la aplicación del modelo del Internet de las Cosas (IoT) en procesos industriales. En 2014, el Consorcio de Internet Industrial, creado para reunir organizaciones y tecnologías necesarias para acelerar el crecimiento de Internet Industrial mediante la identificación, ensamblaje, pruebas y la promoción de las mejores prácticas, lanzó el documento "*Arquitectura de referencia*", en el cual proporciona no solo la descripción de la arquitectura general de un sistema IloT (es decir su estructura y la distribución de componentes y la topología por la cual están interconectados), sino que también incluye una descripción de interfaces y protocolos. Se transfieren dos tipos diferentes de información en los sistemas IloT:

- Flujo de datos, si los datos deben procesarse todavía.
- Flujo de control, si los datos son el resultado de algunas elaboraciones.

Debido a la gran cantidad de información generada y cantidad de dispositivos integrados, el IIoT requiere de un centro de datos robusto para una operación óptima. Adicionalmente, se enfrenta a variables externas que afectan su rendimiento tales como la implementación remota y la latencia (Sunitha, Avaniya, & Goundar, 2021).

Arquitecturas Redundantes IIoT

En el Internet Industrial de las Cosas (IIoT), se aprovecha la potencialidad de la red de Internet para mejorar la eficiencia de la información de procesos industriales, por lo que una arquitectura redundante IIoT mejora la precisión, redundancia, y reduce la latencia. Sin embargo, es fundamental para una arquitectura redundante IIoT utilizar diversos canales de comunicación. Por lo tanto, aunque es conveniente contar con una línea dura, se debe apoyar con una copia de seguridad, ya sea satelital o inalámbrica, que permita garantizar que los datos que se generen en la industria cuenten con un sistema que cubra la integridad de los mismos sin riesgo de pérdida de datos. Por otra parte, una arquitectura redundante IIoT posee características determinadas tales como la transmisión de datos mediante la red, lo que obliga a diseñar sistemas robustos de gestión y protección de datos, ya que existen probables vulnerabilidades que pueden afectar e incluso paralizar el flujo de información, y por consiguiente de operación normal de un proceso (Barbara, 2021).

Cloud Computing

La computación en la nube es un servicio basado en suscripción donde se puede obtener un espacio de almacenamiento en red y recursos informáticos. La nube permite acceder a la información desde cualquier lugar del mundo en cualquier momento.

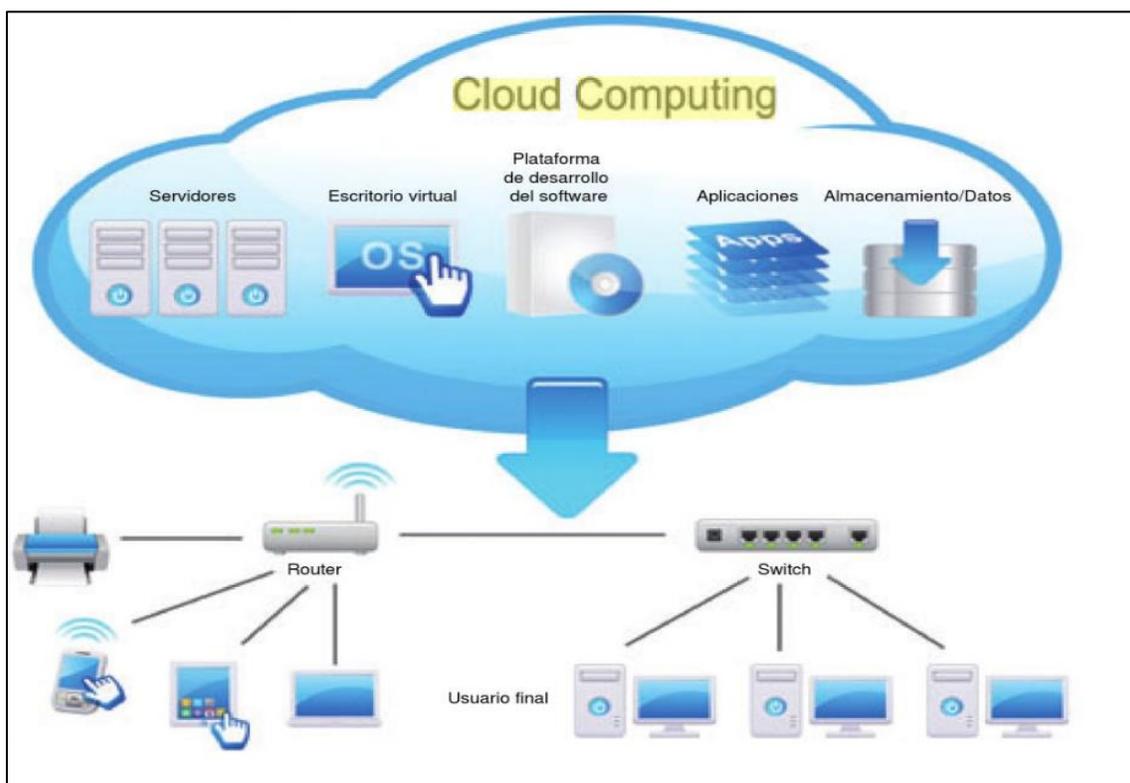
Constituye un modelo para permitir el acceso conveniente a la red de un grupo compartido de recursos informáticos configurables (por ejemplo, redes, servidores, almacenamiento, aplicaciones y servicios), que se pueden aprovisionar y liberar rápidamente con un mínimo esfuerzo de gestión o interacción del proveedor de servicios.

Una de las ventajas consiste en su capacidad de abordar una gran base de clientes con diferentes necesidades computacionales utilizando la misma infraestructura física. Adicionalmente, permite acceder con dispositivos de naturaleza diversa a través de un medio de comunicación como el Internet (Moro, 2013).

La computación en la nube no está orientada a aplicaciones sino a servicios, ya que ofrece recursos virtualizados bajo demanda como utilidades medibles y facturables. Su esquema de funcionamiento se detalla en la siguiente figura:

Figura 5

Esquema de funcionamiento del Cloud Computing



Nota. Tomado de Moro (2013)

Plataforma Ignition

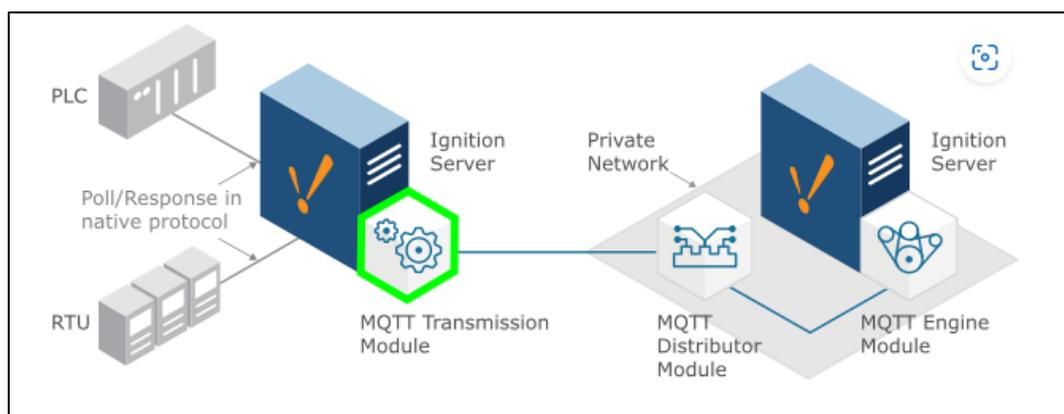
El *software Ignition* permite el diseño de interfaces hombre – máquina (HMI) con alta eficiencia, y permite crear interfaces con tendencias históricas, alarmas para supervisión y control de

procesos, almacenar y analizar datos históricos a través de herramientas de visualización con gráficos y tablas, y gestionar de una manera centralizada los procesos e interfaces automatizados de la industria. Entre los módulos MQTT para la plataforma *Ignition* se encuentran los módulos *MQTT Transmission*, *MQTT Engine* y *MQTT Distributor*.

El objetivo del módulo *MQTT Transmission*, es permitir que cualquier etiqueta en la plataforma *Ignition SCADA* se publique a través de MQTT, transformando *tags* y sus propiedades en el formato *Sparkplug MQTT Topic and Payload* para *Edge Gateways*, de forma que sean accesibles para múltiples plataformas *Ignition* y otras aplicaciones IIoT habilitadas para MQTT. También brinda conectividad para que los datos de las etiquetas se conecten a aplicaciones para *Machine Learning* y *Big Data Analytics*. De esta manera, se proporciona una herramienta de gestión de datos. El módulo *MQTT Transmission* también admite la funcionalidad de almacenamiento y reenvío, lo que significa que cuando se detecta una falla, como un corte en el enlace de comunicación con el servidor MQTT, los datos con marca de tiempo se retendrán localmente hasta que se restablezca el enlace. Una vez que se vuelve a conectar, los datos en vivo se reanudan y los datos históricos se rellenan según lo permita el ancho de banda de comunicación (Cirrus Link, 2023).

Figura 6

Módulo Ignition MQTT Transmission



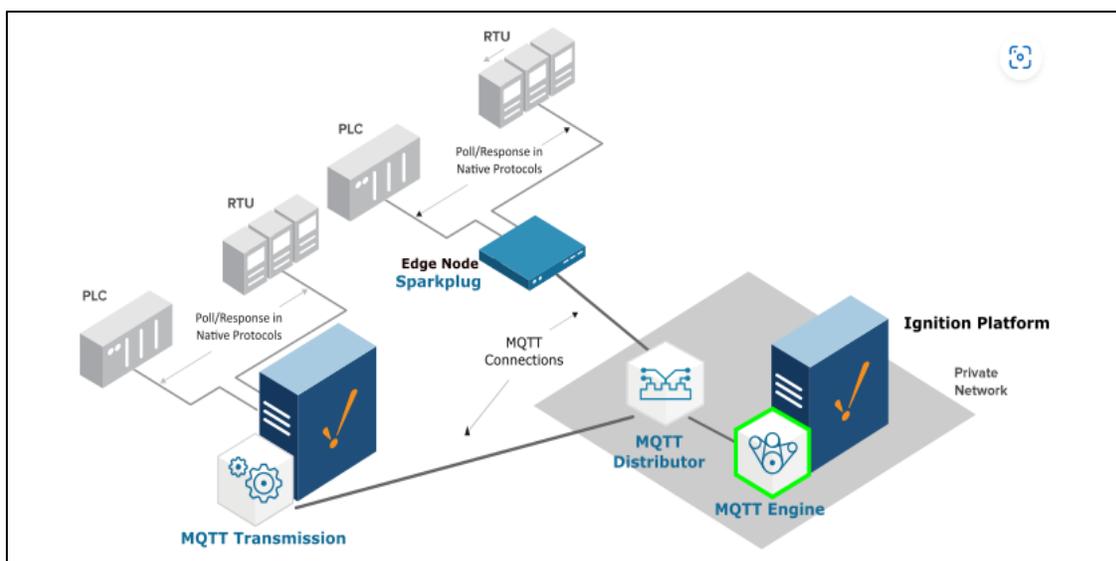
Nota. Tomado de Cirrus Link (2023)

El módulo *MQTT Engine* se utiliza para crear soluciones de IoT Industrial (IIoT) en una infraestructura de *middleware* orientado a mensajes (MOM) de MQTT con el módulo de motor de MQTT, proporcionando una ruta para entregar datos a aplicaciones operativas y comerciales mientras reduce el ancho de banda de datos.

Utiliza las metodologías de publicación y suscripción de MQTT para inyectar datos en las aplicaciones del sistema SCADA industrial al permitir que el módulo de motor MQTT conecte los datos de los servidores MQTT, creando una arquitectura IIoT extremadamente eficiente y robusta con *Ignition*. Con el módulo *MQTT Engine* no es necesario sondear un *host*, ya que utiliza puertas perimetrales que impulsan el sondeo del protocolo propietario al borde de la red del sistema SCADA, aumentando el rendimiento y adquisición de datos de la empresa mediante una canalización de todos los datos. (Cirrus Link, 2023)

Figura 7

Módulo Ignition MQTT Engine



Nota. Tomado de Cirrus Link (2023)

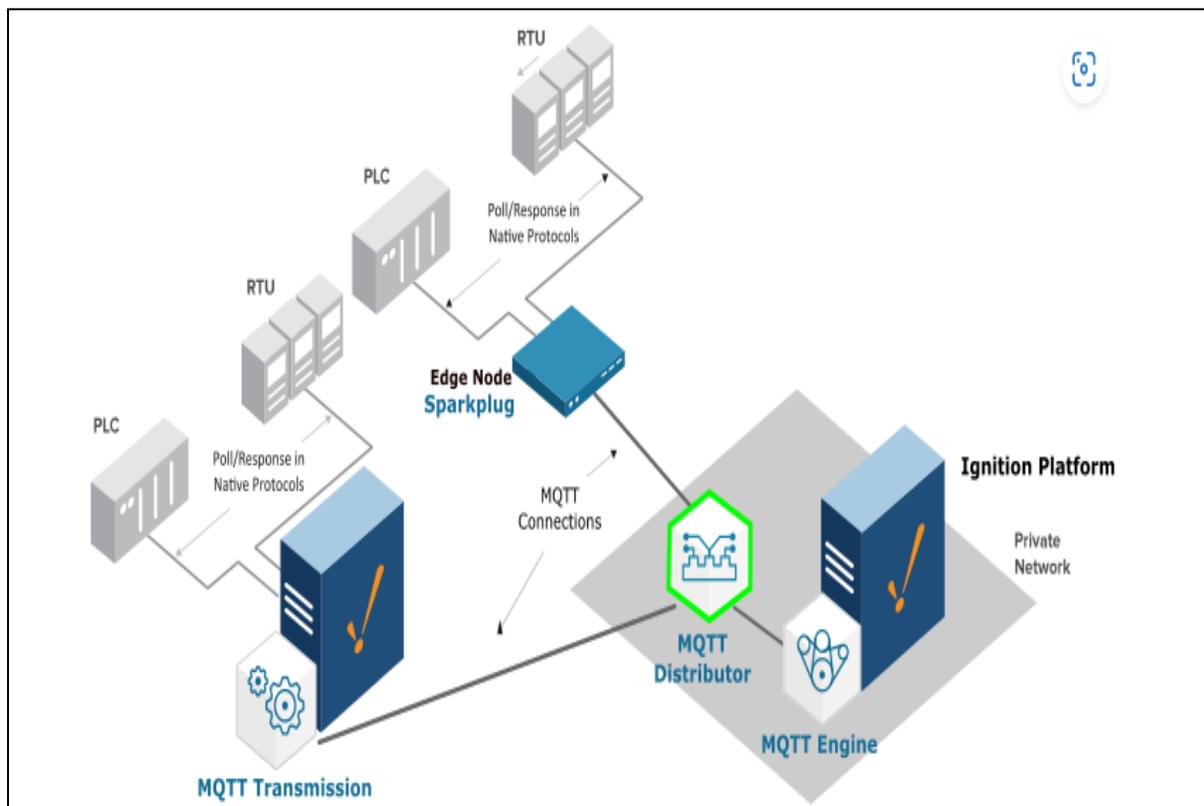
El módulo *MQTT Distributor* es un servidor MQTT que cumple con el estándar OASIS del protocolo 3.1.1 MQTT, y permite que los clientes de MQTT en SCADA se conecten, publiquen y

suscriban datos de forma segura, suministrando datos tanto a aplicaciones operativas como comerciales en toda la empresa. Habilitar *MQTT Distributor* junto con *MQTT Engine Module* proporciona los componentes para una infraestructura MOM autónoma desde una puerta de enlace *Gateway Ignition*.

Esta combinación cumple con los requisitos para soluciones IIoT y aplicaciones SCADA de área amplia, como soluciones de control de tuberías de petróleo y gas. Es ideal para situaciones donde hay comunicaciones restringidas o de alto costo como en VSAT o conectividad celular. Esta solución también es muy eficaz para aumentar el rendimiento de datos para soluciones de planta de alto rendimiento. (Cirrus Link, 2023)

Figura 8

Módulo Ignition MQTT Distributor



Nota. Tomado de Cirrus Link (2023)

En cuanto a módulos especiales MES de la plataforma *Ignition*, es posible contar con servicios para crear, gestionar y supervisar con alto nivel de especialización la configuración de la máquina con módulos de configuración y cambio.

En este caso, el módulo de configuración y cambio es ideal para cambiar de forma rápida y precisa ajustes de la máquina o del proceso, con el cual es posible incrementar la eficiencia, la calidad y el control de producción de las instalaciones de fabricación con una potente gestión de recetas maestras y subrecetas, seguridad de recetas, seguimiento de registro de cambios, seguimiento de variaciones, entre otros (iControls, 2023).

Con los módulos especiales de la plataforma *Ignition*, es posible ampliar el sistema MES con módulos de configuración y cambio para crear, gestionar y supervisar de forma experta la configuración de la máquina. Este potente módulo es ideal para cambiar de forma rápida y precisa la configuración de la máquina o el proceso.

Las recetas ya sean de máquinas, procesos o sistemas industriales se gestionan con un módulo que permite reducir el esfuerzo necesario para administrar numerosas recetas con la funcionalidad de recetas maestras.

Cuando se cambia la configuración de una receta maestra, el módulo se replicará en todas sus recetas secundarias mientras mantiene los valores principales de cada receta secundaria. Con múltiples niveles de recetas maestras, las recetas se pueden organizar de manera jerárquica. Otra ventaja es la facilidad para administrar recetas con el editor visual de recetas que viene incorporado en el módulo, lo que permite crear nuevas recetas, leer valores actuales en una receta, exportar e importar recetas, administrar la seguridad, seleccionar máquinas específicas para recetas y más.

Finalmente, es posible supervisar la configuración de la máquina para determinar las desviaciones en tiempo real lo cual evitará problemas de calidad, tiempo de inactividad u otros problemas asociados que perjudican la producción (iControls, 2023).

Figura 9

Gestión de parámetros con módulos especiales MES Ignition

The screenshot displays the Sepasoft Recipe Overview interface. The top section features the Sepasoft logo and a 'Recipe Overview' title. Below this, a 'Recipe Management' panel shows a tree view of recipes (Recipe C1, C2, C3, C5, C6, C9) and a table for 'Recipe Line X' with columns for Name, Value, Units, and Assigned By. The bottom section, 'Recipe Selection', shows a list of recipes on the left and a grid of parameter settings for various modules: Filler, Sealer, Capper, Labeler, Checkweigher, Casepacker, and Pallettizer. Each module has an 'Enabled' checkbox and a 'Values Count' indicator. The 'Recipe C1' is selected, and its parameters are visible in the grid.

Name	Value	Units	Assigned By
Value 1	500		Recipe Line X - Recipe C1
Value 2	5		Recipe Line X - Recipe C1
Value 3	50k		Recipe Line X - Recipe C1
Value 4	95		Recipe Line X - Recipe C1
Value 5	4		Recipe Line X - Recipe C1

Module	Enabled	Values Count	Parameters
Filler	<input checked="" type="checkbox"/>	Values Count	Line Speed: 10, Fill Weight: 3,000, Max Fill Level: 19, Min Fill Level: 14
Sealer	<input checked="" type="checkbox"/>	Values Count	Line Speed: 70, Temp SP: 90, High Temp: 95, Low Temp: 85
Capper	<input checked="" type="checkbox"/>	Values Count	Line Speed: 100, Force: 14
Labeler	<input checked="" type="checkbox"/>	Values Count	Line Speed: 35, Label ID: ABCD1234, Spacing Delay: 4
Checkweigher	<input checked="" type="checkbox"/>	Values Count	USL: 95, UCL: 90, LCL: 10, LBL: 5
Casepacker	<input checked="" type="checkbox"/>	Values Count	Units - Case: 20, Leading Delay: 5, Trailing Delay: 12, Glue Delay: 10
Pallettizer	<input checked="" type="checkbox"/>	Values Count	Pattern Code: 23,472,834, Row Height: 15

Nota. Tomado de Sepasoft (2023)

Capítulo III

Análisis y Diseño

Conectividad a Internet

Para implementar la arquitectura IIoT se debe disponer de una red Ethernet con salida a Internet de banda ancha. El medio físico puede ser cableado o inalámbrico, y se hará uso de un ruteador para conectar los diferentes dispositivos a la red IIoT.

Hardware y Software

Para implementar la arquitectura planteada, se requieren 3 ordenadores físicos y uno virtual, dentro de los cuales estarán instalados y configurados los paquetes de *software* tanto de control como de monitoreo, tal como se describe en la Tabla 2:

Tabla 2

Requisitos para implementación de arquitectura IIoT

Item	Equipo	RAM (MB)	Software	Aplicación
1	TOSHIBA-PC	16	SoftLogix 58.00 AB Virtual PLC Factory IO v2.4.3 RSLogix 5000 Ignition MQTT Distributor Module MobaXTerm	Programación PLC virtual Planta 3D de paletizado Programación de PLC <i>Broker</i> redundante MQTT Configuración Raspberry PI
2	Raspberry PI	4	Ignition MQTT Transmissor module Ignition OPC UA Ignition Designer	Publicador de datos MQTT Driver PLC Logix5000 Diseño de aplicación SCADA-MES
3	DELL INSPIRON	8	MySQL Ignition MQTT Engine Module Ignition MES Production Module Ignition MES Recipe Module	Base de datos Receptor de datos MQTT Módulo de Producción MES Módulo MES para creación de recetas
4	Servidor Cloud MQTT-1	4	Ignition MQTT Distributor Module	<i>Broker</i> primario MQTT

Licencias

- *SoftLogix / RSLogix 5000*: Licencia personal para la programación y configuración del PLC virtual y uso de la tarjeta *Ethernet* embebida con el *software*.
- *Factory IO*: Licencia personal para diseño de la planta de paletizado 3D y del *driver* de comunicación *Allen-Bradley Logix 5000* con el PLC *SoftLogix*.
- *Raspberry PI*: Licencia no necesaria por ser *software* libre.
- *Ignition*: Para todos los módulos instalados se hace uso de la licencia tipo “*trial*”, la cual consiste en dos horas de uso libre de la plataforma y autenticación con usuario registrado.
- *Servidor Cloud*: Contratación del servicio de uso de máquina virtual con *Windows 10 Pro* en la plataforma de *Microsoft Azure*.

Requisitos de Red

El protocolo MQTT se transmite mediante TCP, por lo cual se hace uso de una red *Ethernet* con acceso a *Internet*, de tal forma que permita la comunicación de los dispositivos LAN con los *broker* MQTT.

Tabla 3

Direccionamiento IP de los dispositivos dentro la red IIoT

Ítem	Equipo	Aplicación	Dirección IP	Máscara de subred
1	TOSHIBA-PC	PLC SoftLogix	192.168.100.230	255.255.255.0
		<i>Broker</i> MQTT respaldo	192.168.100.190	255.255.255.0
2	Raspberry PI	Publicador MQTT	192.168.100.79	255.255.255.0
3	DELL INSPIRON	Receptor MQTT	192.168.100.18	255.255.255.0
4	<i>Servidor Cloud</i> MQTT-1	<i>Broker</i> MQTT primario	20.124.205.206	255.255.255.0

Descripción del Proceso de Paletizado:

El proceso de paletizado consta de dos líneas de producción, cada una dividida en los siguientes subprocesos:

- *Clasificación:* Consta de una alimentadora de cajas hacia la banda transportadora de entrada. Dicha caja es pesada en una balanza electrónica y su peso es comparado con un valor de referencia establecido por el usuario en el sistema MES (tratamiento de recetas). Si la caja cumple con el peso especificado pasará a la paletizadora, caso contrario será desechada del proceso.

Figura 10

Detalle de proceso de clasificación de peso



- *Paletizado:* Las cajas ingresan hacia la máquina paletizadora a través de bandas transportadoras, una vez allí, el sistema de control calcula la cantidad de cajas a paletizar según las filas de cajas necesarias, que se ha programado en el sistema MES (tratamiento de recetas). El proceso es

completamente automático, ya que ha sido programado en el PLC. Una vez completado el paletizado, los *pallets* con las cajas requeridas se depositan en el sistema de bandas de transporte hacia el sistema de almacenamiento automático.

Figura 11

Detalle de proceso de paletizado



- *Bandas de transporte*: Las cajas paletizadas salen de cada una de las líneas de producción hacia las bandas de transporte, las cuales movilizan los *pallets* a los estantes de almacenamiento.

Figura 12

Detalle de bandas de transporte



- *Sistema de almacenamiento automático:* Un robot cartesiano es encargado de almacenar los *pallets* en los estantes asignados para cada una de las líneas de producción. Cada uno de los *pallets* cuentan con sensores RFID, lo que permite al robot identificar a cuál de las líneas pertenece el *pallet*, y lo almacenará en el primer espacio disponible. El estante de almacenamiento consta de 54 estantes; los 27 primeros están asignados a los *pallets* de la línea uno y los otros 27 estantes están asignados a los *pallets* de la línea dos.

Figura 13

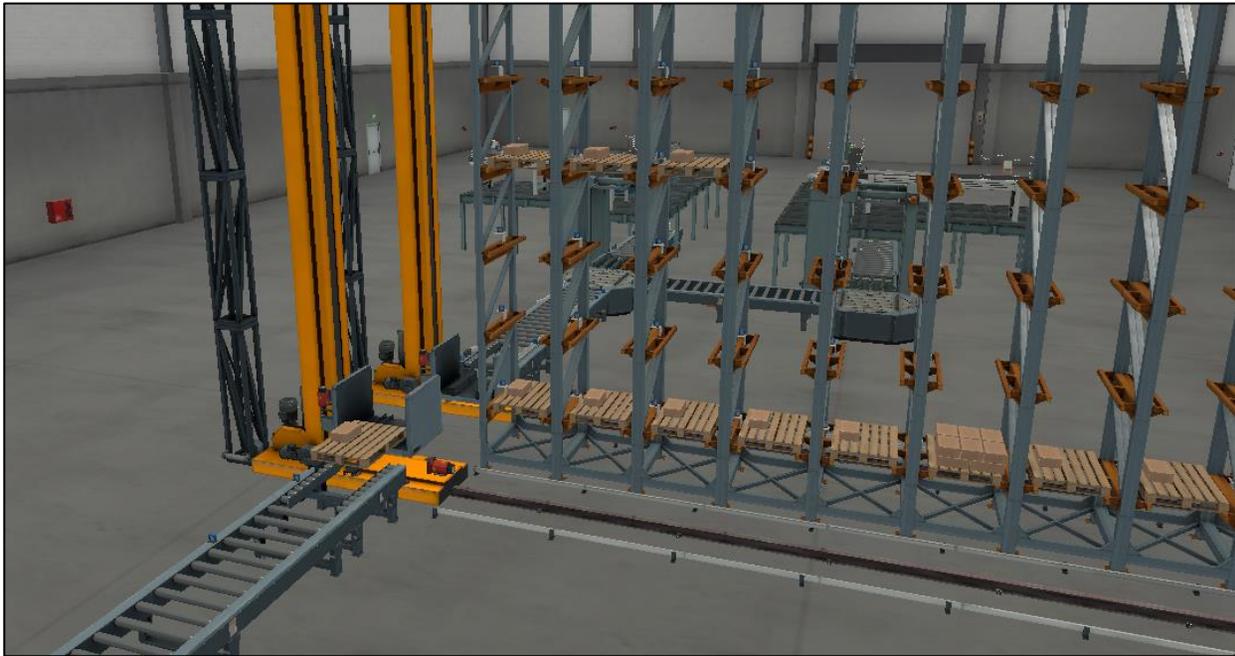
Detalle de sistema de almacenamiento automático



- *Sistema de extracción de pallets:* El usuario puede ingresar en el sistema MES (tratamiento de recetas) la cantidad de *pallets* que desea despachar ya sea de la línea uno o de la línea dos. Para este trabajo, se ha programado un segundo robot cartesiano que ejecuta esta maniobra, el cual se activa al momento de recibir el comando de arranque, tomará el *pallet* de la última ubicación y lo llevará hacia la banda final de despacho, permitiendo así completar todo el proceso de clasificación, paletizado, almacenamiento y despacho automático de las líneas de proceso.

Figura 14

Detalle de sistema de sistema de extracción de pallets



Configuración en Plataforma Virtual Factory IO

Para el diseño de la planta de paletizado, se han utilizado módulos de líneas de fabricación disponibles en el *software* tales como:

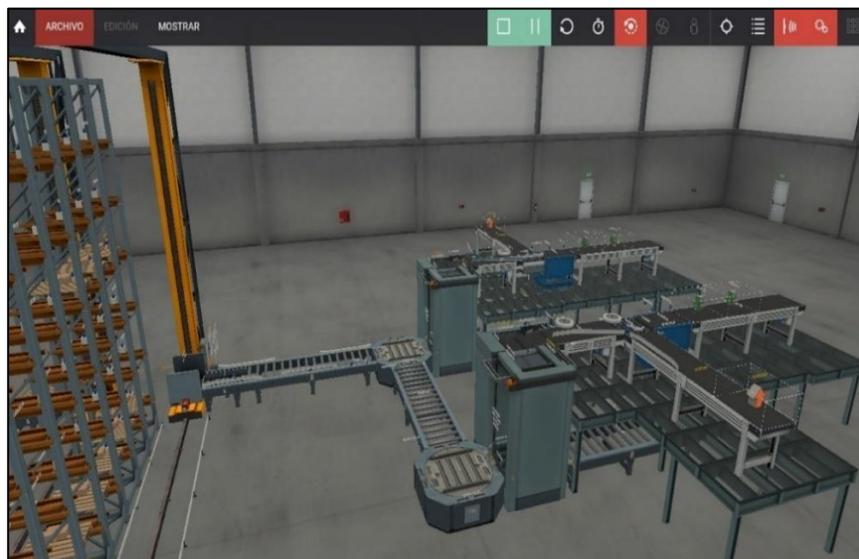
- Bandas transportadoras de cajas con actuadores analógicos o digitales
- Bandas transportadoras de *pallets* con actuadores analógicos o digitales.
- Balanzas electrónicas de pesaje para clasificación de cajas.
- Máquinas de paletizado.
- Robots cartesianos.
- Estantes de almacenamiento.
- Sensores inductivos, reflectivos.
- Sensores RFID.

Se ha armado la planta estructurando y combinando cada uno de estos componentes hasta

completar el diseño planteado. También se asignan etiquetas a cada uno de los sensores y actuadores, de tal forma que el PLC leerá y/o escribirá datos en estos dispositivos, y ejecutará la lógica programada.

Figura 15

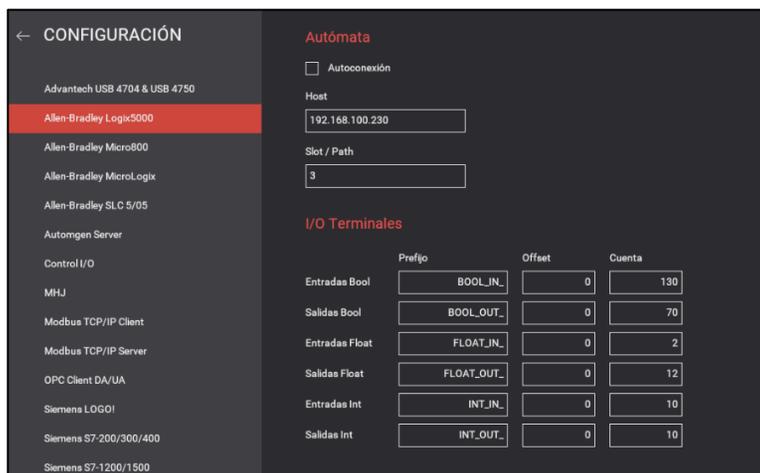
Componentes de la planta virtual de paletizado



Se configura en el *software* la dirección IP del PLC, tomando en cuenta el *slot* en el cual el controlador se encuentra ubicado. Se han utilizado un total de 234 etiquetas correspondientes a entradas y salidas, las cuales son procesadas por el PLC, como lo indica la figura 16:

Figura 16

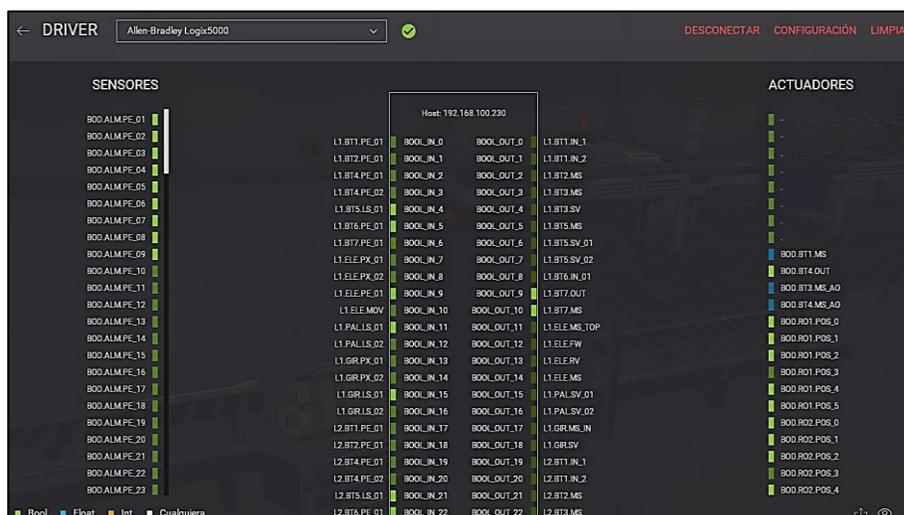
Enlace del software Factory IO con el PLC virtual mediante Ethernet



A continuación, se enlazan los *tags* de proceso del *software Factory IO* con el PLC.

Figura 17

Asignación de etiquetas de variables de Factory IO hacia el PLC



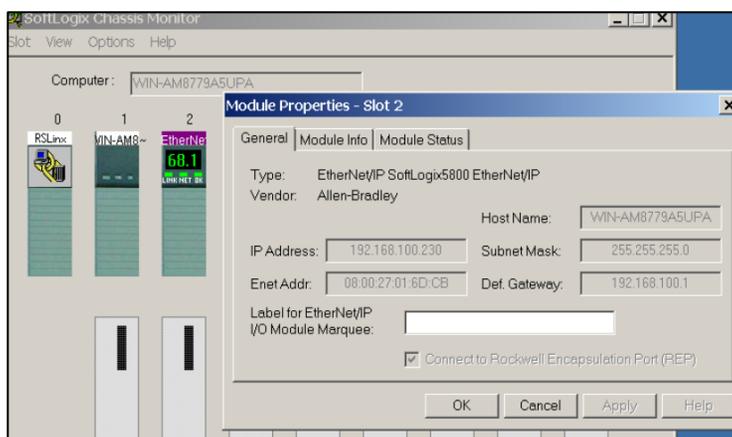
Configuración SoftLogix

El programa *SoftLogix5800* de simulación de PLC's, permite la programación de un PLC virtual de alta gama de la familia *Logix* de *Allen Bradley*.

Dispone de una interfaz de un chasis virtual para añadir y configurar los módulos requeridos. Se añade y configura la tarjeta de red *Ethernet* según el direccionamiento IP planteado anteriormente.

Figura 18

Configuración de tarjeta de red Ethernet de Softlogix

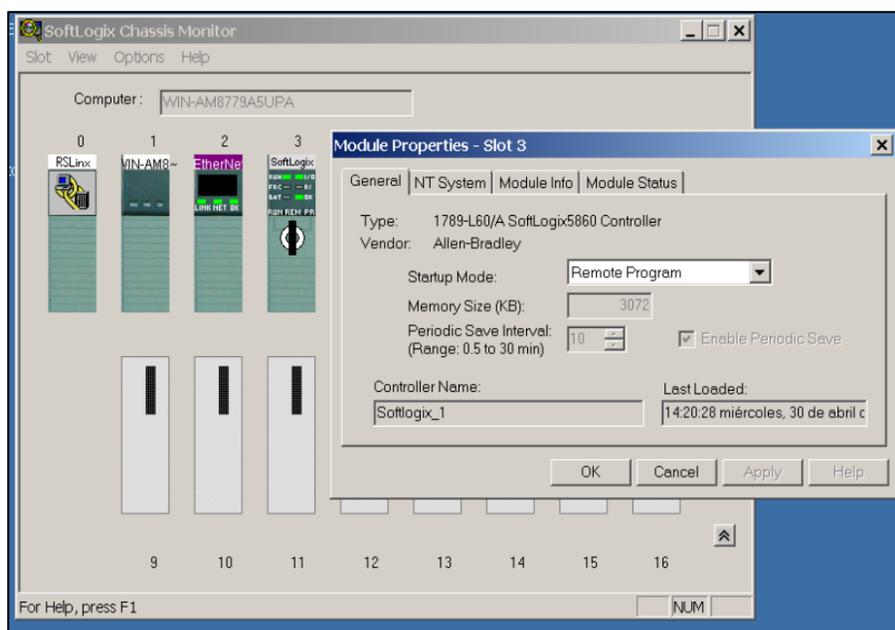


El siguiente paso es añadir al chasis virtual el controlador *SoftLogix*; para esto es necesario insertar el controlador virtual en uno de los *slots* disponibles, y se debe asignar dicho *slot* en el programa de edición.

Se configura el controlador virtual con una memoria de 3072 Kb.

Figura 19

Configuración del PLC virtual



Programación del Software RSLogix 5000 para el Proceso de Paletizado.

Una vez configurada la tarjeta de comunicación, se utiliza el *software RSLinx* para la comunicación del PLC virtual con el *software RSLogix 5000*, donde se encuentra programada la secuencia de control automático del proceso de paletizado.

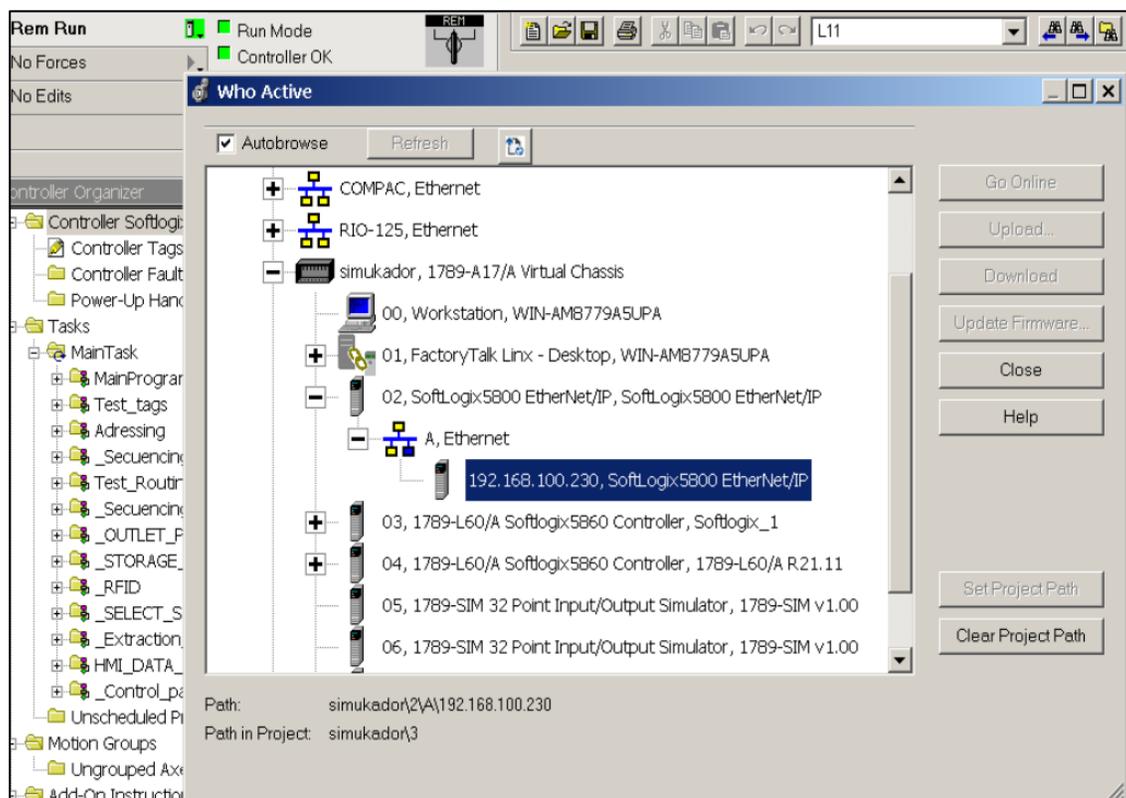
Es necesario indicarle al programa *RSLogix 5000* la ruta mediante la cual se va a comunicar con el PLC virtual, ya que, en el mismo chasis se pueden configurar uno o más controladores virtuales.

Esta ruta servirá para realizar la descarga de la lógica de control hacia el PLC virtual y para controlar sus modos de operación: *arranque*, *arranque remoto* y *programación*.

Se comprueba la comunicación entre PLC y *software* de programación como ilustra la figura:

Figura 20

Comunicación del PLC virtual con el software RSLogix 5000



Programas y Rutinas.

Se han creado varios programas para organizar la secuencia del proceso de paletizado; cada uno con rutinas específicas, las cuales contienen los códigos de programa para:

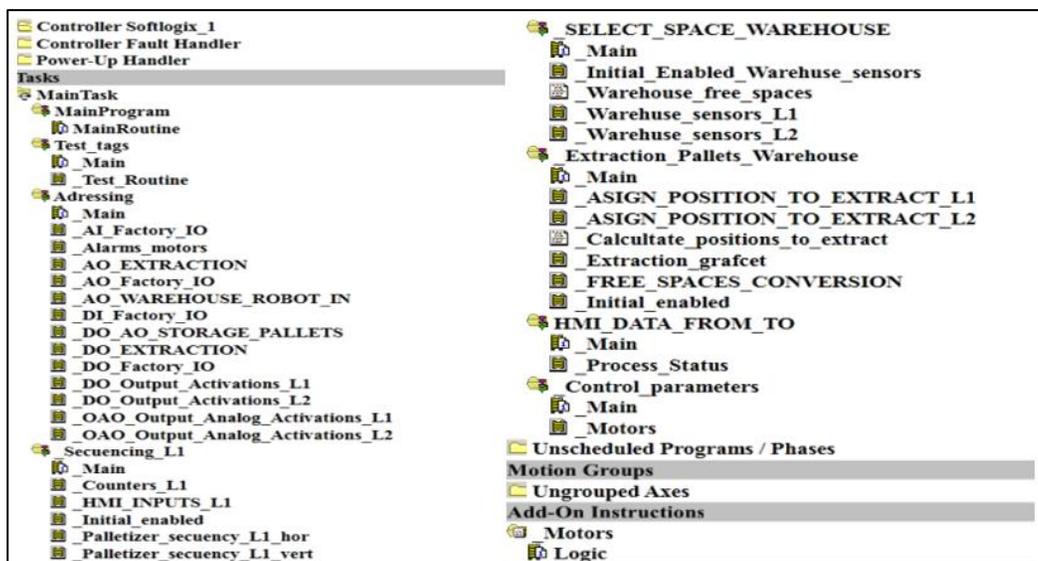
- Direccionamiento de sensores y actuadores de dispositivos de planta.
- Programación en lenguaje escalera para la secuencia de pesaje, paletizado, transporte, almacenamiento y extracción.
- Programación en lenguaje de texto estructurado para el cálculo de estantes disponibles de almacenamiento y despacho.
- Programación de etiquetas de lectura y escritura de variables provenientes del SCADA-MES.
- Programación de lectura de sensores RFID.

- Programación de “Add-On” para lectura de *status* de motores y sus respectivos horómetros.

Se muestran varias de las rutinas creadas en el PLC en la siguiente figura:

Figura 21

Detalle de rutinas creadas en el PLC



En total se han creado 732 variables en el PLC para realizar el control del proceso, entre entradas, salidas, temporizadores, contadores, cálculos, variables de SCADA-MES, entre otros.

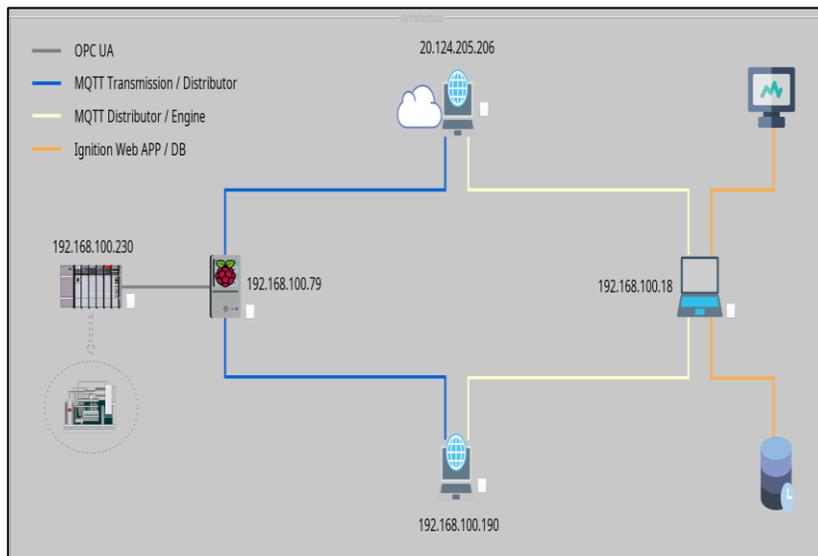
Arquitectura IIoT Redundante Planteada

- La arquitectura IIoT redundante consta en detalle de los siguientes elementos:
- PLC *SoftLogix* para el control del proceso de paletizado.
- Módulo de *Ignition* para adquisición de datos del PLC mediante OCP UA.
- Módulo de *Ignition* para conversión de datos de OPC UA a paquetes MQTT.
- *Broker Cloud* primario de paquetes MQTT.
- *Broker LAN* redundante de paquetes MQTT.
- Módulo de *Ignition* para recepción de paquetes MQTT.
- Módulo de *Ignition* para configuración de SCADA-MES y base de datos.

- Cliente donde se ejecutará la aplicación.

Figura 22

Detalle de componentes de la arquitectura IIoT redundante

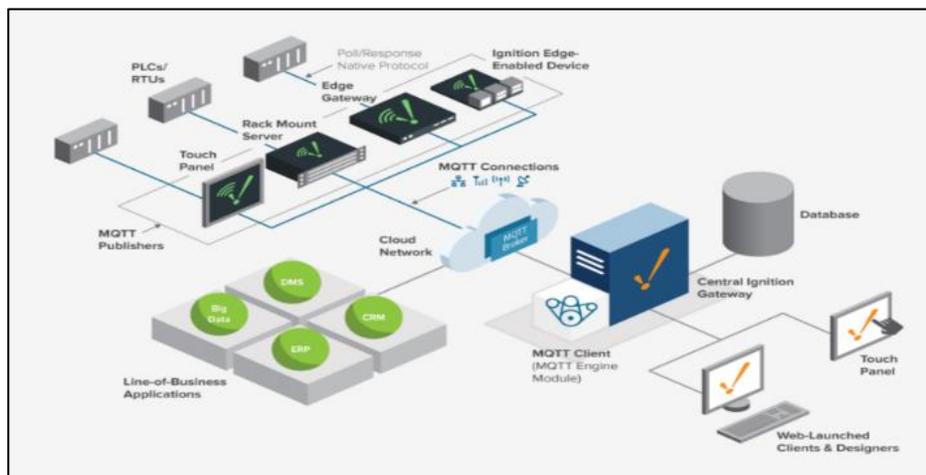


Configuración de Ignition Edge en Raspberry PI.

El *software Ignition Edge* es un aplicativo para dispositivos de control que adquieren datos directamente desde un controlador y transforman los paquetes del protocolo nativo al protocolo MQTT.

Figura 23

Esquema de funcionamiento de software Ignition Edge



Nota. Tomado de Cirrus Link (2023)

En el dispositivo *Raspberry Pi* se ha instalado el aplicativo *Ignition Edge*, y se ha configurado el *driver* OPC UA correspondiente a la familia *Logix* de *Allen Bradley* para adquirir los datos del PLC virtual *SoftLogix*, asignando la respectiva dirección IP y su respectivo *slot* en el chasis virtual.

Figura 24

Configuración de driver OPC UA para controladores Logix 5000

General	
Name	PLC TESIS
Description	PLC for palletizer project
Enabled	<input checked="" type="checkbox"/> (default: true)
Connectivity	
Hostname	192.168.100.230 Hostname or IP address of the controller or ethernet module.
Port	44818 Port to connect to on the remote device. (default: 44818)
Local Address	Address of network adapter to connect from. (default:)
Timeout	2000 Max amount of time to wait, in milliseconds, for responses from the processor. (default: 2.000)
Max Concurrent Requests	2 Controls both the max number of concurrent unconnected requests and the max number of concurrent connected requests (and therefore CIP connections) allowed. (default: 2)
Slot Number	3 The slot number in which the processor resides. If not in a rack, usually 0. (default: 0)
Connection Path	Optionally, the connection path to the processor. If specified, the slot number setting is ignored, so be sure to terminate the path with backplane and slot number. (default:)
<input type="checkbox"/> Show advanced properties	
Save Changes	

Se verifica la comunicación con el controlador desde la pestaña “*Quick Client*”, lo que permite asegurar que los datos del PLC son leídos correctamente, según se aprecia en la figura 25:

Figura 25

Verificación de lectura de datos del PLC mediante el protocolo OPC-UA

TYPE	ACTION	TITLE
Server	refresh	Ignition OPC UA Server
Object		Devices
Object		[PLC TESIS]
Object		Controller:Global
Object		Program:Addressing
Object		Program:HMI_DATA_FROM_TO
Object		Program:MainProgram
Object		Program:Test_Routine
Object		Program:Test_tags
Object		Program:_Control_parameters
Object		Program:_Extraction_Pallets_Ware...
Object		Program:_OUTLET_PALLETS_L1_L2
Object		Program:_RFID
Object		Program:_SELECT_SPACE_WAREHOUSE

Una vez establecida la comunicación OPC UA, se configura el módulo de Transmisión MQTT, según el cual, el dispositivo *Raspberry PI* convertirá los paquetes OPC UA en paquetes MQTT y los publicará a los *brokers* redundantes MQTT según el protocolo de transmisión *Sparkplug B*, diseñado para transmisión de datos en tiempo real. Para este fin, en la pestaña “*Servers*” es necesario indicarle al transmisor MQTT las direcciones IP de los *brokers*, con su respectivo usuario de autenticación y accesos, según se indica continuación:

Figura 26

Configuración de módulo de Transmisión MQTT

Name	URL	Server Set	Username	Connected	
CLOUD-BROKER	tcp://20.124.205.206:1883	Default	MrDowsebf	1 of 1	delete edit
LAN-BROKER	tcp://192.168.100.190:1883	Default	MrDowsebf	0 of 1	delete edit

Como se puede apreciar, el *broker Cloud* es el primario y se encuentra conectado al transmisor MQTT, mientras que el redundante está en espera. En caso de una desconexión, el transmisor cambia de *broker* y permite mantener la comunicación de toda la red.

En la pestaña “*Transmitters*” se define el detalle del direccionamiento de los datos del protocolo *Sparkplug B* MQTT que se desea publicar, ya que, para leer exitosamente los datos en el receptor, es necesario seguir la misma estructura de identificación de datos.

Se ha configurado la ruta MQTT de las variables a publicar en la siguiente estructura:

- *TESIS*
- *EDGE_ARM*
- *RASPBERRY*

Figura 27

Configuración de protocolo Sparkplug B en el módulo de Transmisión MQTT

Sparkplug Settings	
Group ID	<input type="text" value="TESIS"/> An ID representing a logical grouping of Edge Nodes and Devices (optional)
Edge Node ID	<input type="text" value="EDGE_ARM"/> An ID representing an Edge or Network (EoN) Node (optional)
Device ID	<input type="text" value="RASPBERRY"/> An ID representing a Device (optional)

Figura 28

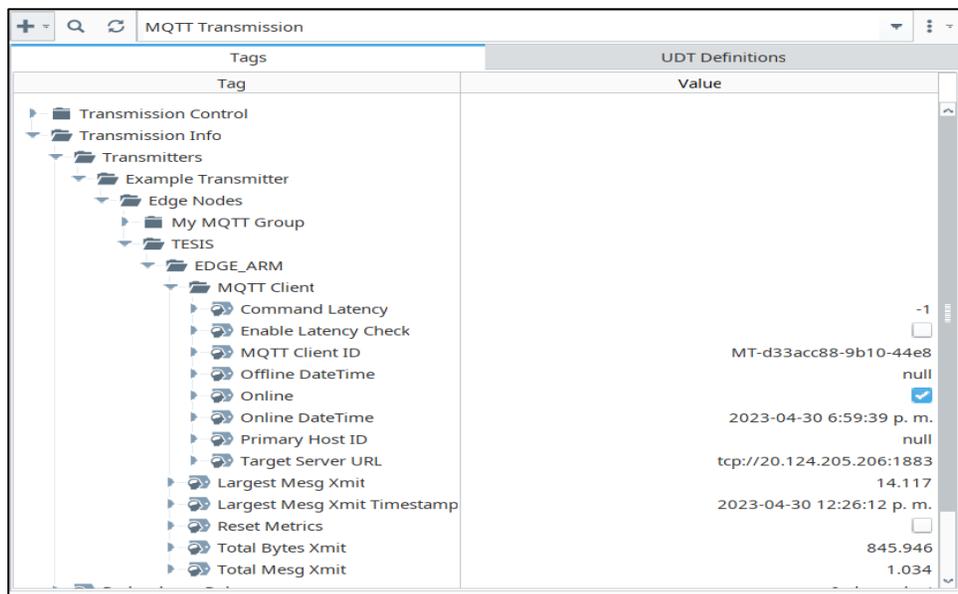
Selección de rutas de tags a publicar por el módulo de Transmisión MQTT

General Servers Sets Transmitters Records Files							
Name	Enabled	Tag Provider	Tag Path	Set	History Store	Sparkplug IDs	
Example Transmitter	true	edge	MQTT Tags	Default		TESIS/EDGE_ARM/RASPBERRY	<input type="button" value="delete"/> <input type="button" value="edit"/>
→ Create new Settings...							

Una vez configurado el módulo OPC UA, el módulo de Transmisión MQTT y el ID del protocolo, se pueden transferir las etiquetas del PLC hacia el árbol de “Tags” para que el *broker* pueda adquirirlos y distribuirlos al receptor MQTT. Esta configuración se la realiza en el *software* de desarrollo de *Ignition*, llamado “*Ignition Designer-Launcher*”. Para ello, se verifica que la ruta configurada en *Sparkplug B* tenga comunicación con el *broker*.

Figura 29

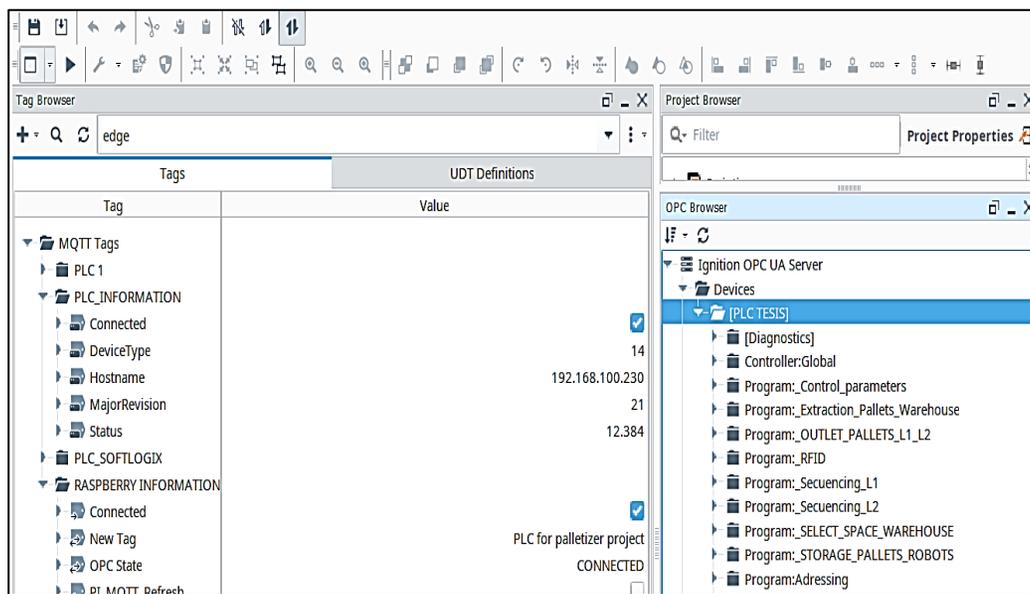
Verificación de comunicación de tags MQTT según Sparkplug B



A continuación, se deben “arrastrar” los tags del PLC desde el servidor OPC UA hacia el árbol de tags MQTT, en la pestaña de tags asignada en la configuración del transmisor MQTT.

Figura 30

Ubicación de tags del PLC hacia el árbol de tags MQTT



Como se puede apreciar, el *software* detecta automáticamente ciertas variables referentes al PLC, y el resto de variables de proceso se deben “arrastrar” desde el controlador. No es necesario crear nuevas variables de tipo derivado o referente, sino que la red IIoT lo hace automáticamente. De esta manera, las variables requeridas para control y monitoreo ya se encuentran publicadas y listas para su tratamiento en el sistema SCADA-MES.

Configuración del Broker Primario MQTT

El *broker* primario MQTT es el principal componente de la red, ya que es el servidor “*máster*” de la arquitectura IIoT. Consta de una máquina virtual alojado en la nube de *Microsoft Azure*, a través del servicio de *Cloud Computing*, con una dirección IP pública 20.124.205.206, a la cual se puede acceder mediante aplicativo RDP a través del puerto 3389. Las características completas se aprecian en la figura 31:

Figura 31

Características de la máquina virtual del broker primario MQTT

Máquina virtual		Redes	
Nombre del equipo	MQTT-1	Dirección IP pública	20.124.205.206 (Interfaz de red mqtt-1389)
Sistema operativo	Windows	Dirección IP pública (IPv6)	-
Publicador	MicrosoftWindowsDesktop	Dirección IP privada	10.2.0.4
Oferta	Windows-10	Dirección IP privada (IPv6)	-
Plan	win10-21h2-pro-g2	Red virtual/subred	MQTT-1-vnet/default
Generación de VM	V2	Nombre DNS	Configurar
Arquitectura de VM	x64		
Grupo host	Ninguno	Tamaño	
Host	-	Tamaño	Standard B2s
Grupo con ubicación por proximidad	-	vCPU	2
Estado de ubicación	N/D	RAM	4 GiB
Grupo de reserva de capacidad	-	Disco	
Tipo de controladora de disco	SCSI	Disco del SO	MQTT-1_disk1_ee915c647f2a447b8871a9f20fcd6cd
		Cifrado en el host	Deshabilitado
		Azure Disk Encryption	No habilitado

Con fines de seguridad informática, las conexiones TCP entrantes y salientes son bloqueadas por defecto. Para poder acceder a los puertos necesarios de *Ignition*, se deben abrir los puertos creando reglas de entrada y salida de los puertos TCP 1883, 8080-8090, así como el protocolo ICMP, tal como se ilustra a continuación:

Figura 32

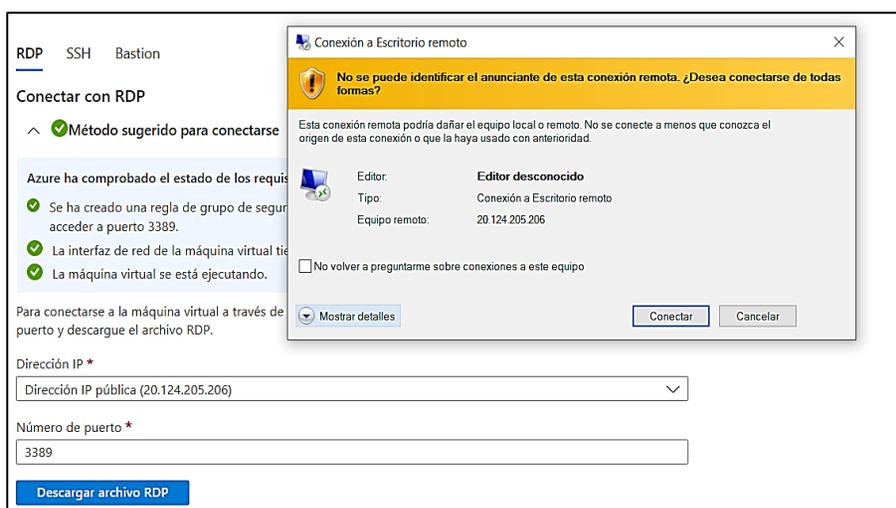
Configuración de apertura de puertos TCP en la máquina virtual

Prioridad	Nombre	Puerto	Protocolo	Origen	Destino	Acción	
1000	default-allow-rdp	3389	TCP	Cualquiera	Cualquiera	Permitir	***
1010	AllowAnyCustom8080Inbound	8080	TCP	Cualquiera	Cualquiera	Permitir	***
1020	AllowAnyCustomAnyInbound	Cualquiera	ICMP	Cualquiera	Cualquiera	Permitir	***
4000	AllowAnyCustom8080-8090Inbound	8080-8090	TCP	Cualquiera	Cualquiera	Permitir	***
4096	AllowAnyCustom1883Inbound	1883	TCP	Cualquiera	Cualquiera	Permitir	***
65000	AllowVnetInBound	Cualquiera	Cualquiera	VirtualNetwork	VirtualNetwork	Permitir	***
65001	AllowAzureLoadBalancerInBound	Cualquiera	Cualquiera	AzureLoadBalancer	Cualquiera	Permitir	***
65500	DenyAllInBound	Cualquiera	Cualquiera	Cualquiera	Cualquiera	Denegar	***

Una vez realizada la apertura de los puertos, se accede a la máquina virtual mediante el aplicativo RDP a través de escritorio remoto según se detalla en la figura 33:

Figura 33

Acceso a máquina virtual mediante aplicativo RDP



Una vez que se acceda remotamente a la máquina virtual, se instala el aplicativo de *Ignition Gateway* con el respectivo módulo de *broker* MQTT, llamado *MQTT Distributor Module*.

La configuración del *broker* se lo realiza dentro del módulo MQTT, y requiere:

- Habilitar el servidor MQTT.
- Habilitar las conexiones TCP para el *broker* MQTT.
- Habilitar el puerto de acceso.

Por defecto, el puerto correspondiente es el 1883, y es a través de este puerto por el cual el *broker* recibe los paquetes MQTT del transmisor y los publica hacia los receptores, según lo indica la figura 34:

Figura 34

Configuración del broker dentro del módulo MQTT

General		Users
Main		
Enabled	<input checked="" type="checkbox"/> Enable the MQTT Server	
Non-TLS Settings		
Enable TCP	<input checked="" type="checkbox"/> Enable plain TCP connections for the MQTT Server	
Port	1883 Non-TLS MQTT Server port	
Enable Websocket	<input checked="" type="checkbox"/> Enable Websocket connections for the MQTT Server	
Websocket Port	8090 Non-TLS MQTT Server Websocket port	

Lo siguiente a configurar es el nivel de acceso de los transmisores y receptores. El *software* de *Ignition* dispone de varias opciones de seguridad de acceso, sin embargo, en el presente proyecto se ha utilizado el usuario autenticado como control de acceso, según se aprecia a continuación:

Figura 35

Configuración de usuario autenticado para acceso a los paquetes MQTT

Config > Mqtt distributor > MQTT Distributor Settings

Trial Mode 1:53:44 We're glad you're test driving our software. Have fun. [Activate Ignition](#)

Main	
Username	MrDowsebf MQTT Username to use during connection establishment
Change Password?	<input type="checkbox"/> Check this box to change the existing password.
Password	[Redacted] MQTT Password to use during connection establishment
Password	[Redacted] Re-type password for verification.
ACLs	RW # Comma separated list of permissions associated with this user of the form [RW topic],[RW topic]...

[Save Changes](#)

La última configuración requerida es el modo de redundancia. Para el servidor *Cloud*, como ya se ha manifestado, se configurará como *broker* primario. Esto se lo realiza en la pestaña de “Redundancia” mediante la selección de la opción “Master”, de acuerdo a la siguiente figura:

Figura 36

Configuración de modo de redundancia en el broker primario

Config > System > Redundancy and Network Configuration

Trial Mode 1:51:01 We're glad you're test driving our software. Have fun. [Activate Ignition](#)

Redundancy Settings	
Mode	Master Enable or disable redundancy, and specify this node's role. There should be one master and one backup node per redundant pair. <code>Independent</code> turns off redundancy.
Standby Activity Level	Warm How the node should run when it is not currently the active node. If <code>cold</code> , the node will perform minimal operations until it becomes active. If <code>warm</code> , the node will run at a high level, reducing failover times.
Failover Timeout	10000 The time of inactivity, in milliseconds, before the backup assumes responsibility. (default: 10000)
Startup Connection Allowance	30000 The time in milliseconds that the system will wait at startup for a connection before making a decision on the node's responsibility level. (default: 30000)

El *broker* primario está listo para recibir paquetes MQTT del distribuidor *Raspberry Pi* y transmitirlos al receptor MQTT.

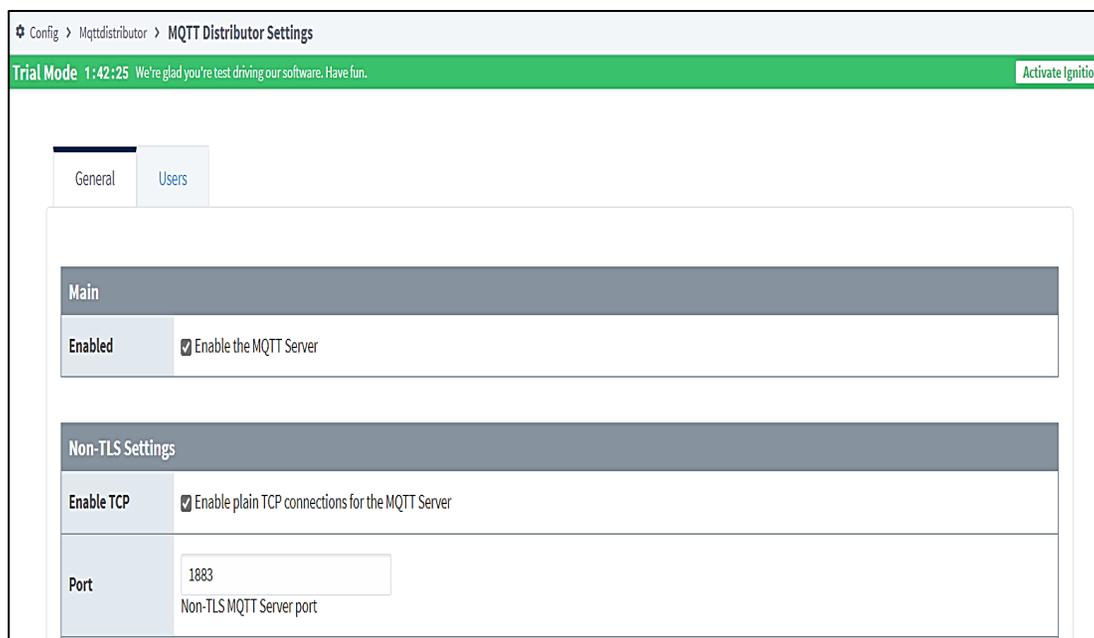
Configuración del Broker MQTT de Respaldo

Según el diseño de la arquitectura planteada, el *broker* de respaldo constituye un ordenador que se encuentra dentro de la red de área local y permanece en “espera” hasta detectar una desconexión del *broker* primario.

La configuración del *broker* de respaldo es similar a la del *broker* primario. Se instala el módulo *MQTT Distributor Module*, se habilita el servidor y el puerto 1883, a través del cual se puede acceder a los datos MQTT tanto del transmisor como del receptor, según ilustra la figura 37:

Figura 37

Detalle de configuración de broker MQTT de respaldo



La configuración de la redundancia también es similar a la del *broker* primario: Se accede a la pestaña de “Redundancia” y se configura con el modo “*Back Up*”. Adicionalmente, se indica al *broker* de respaldo la dirección IP del *broker* primario, para que lo monitoree continuamente y entre en servicio en caso de falla, según se indica en la figura 38.

Figura 38

Detalle de configuración de redundancia en broker de respaldo

Redundancy Settings	
Mode	Backup <small>Enable or disable redundancy, and specify this node's role. There should be one master and one backup node per redundant pair. <code>Independent</code> turns off redundancy.</small>
Standby Activity Level	Warm <small>How the node should run when it is not currently the active node. If <code>cold</code>, the node will perform minimal operations until it becomes active. If <code>warm</code>, the node will run at a high level, reducing failover times.</small>
Backup Node Settings	
Master Node Address	20.124.205.206 <small>The address of the master Ignition system.</small>
Port	8088 <small>The Gateway Network port used by the master. (default: 8060)</small>

De esta manera queda configurada la redundancia de *brokers* MQTT, en caso de falla del primario, el *broker* de respaldo entra en funcionamiento y mantiene la comunicación de la red IIoT.

Configuración del Gateway SCADA-MES

Una vez configurado el transmisor y los *brokers* MQTT, según la arquitectura planteada, es necesario configurar el receptor MQTT, para lo cual se realiza la instalación del módulo *MQTT Engine Module*.

Dentro de este módulo, se configuran las direcciones IP los dos *brokers*, tanto del primario como del redundante, y se verifica la conectividad. Se configura el usuario de acceso, mediante el cual el sistema permitirá la comunicación de los nodos conectados, como se ilustra a continuación:

Figura 39

Configuración de brokers MQTT en el módulo de Recepción

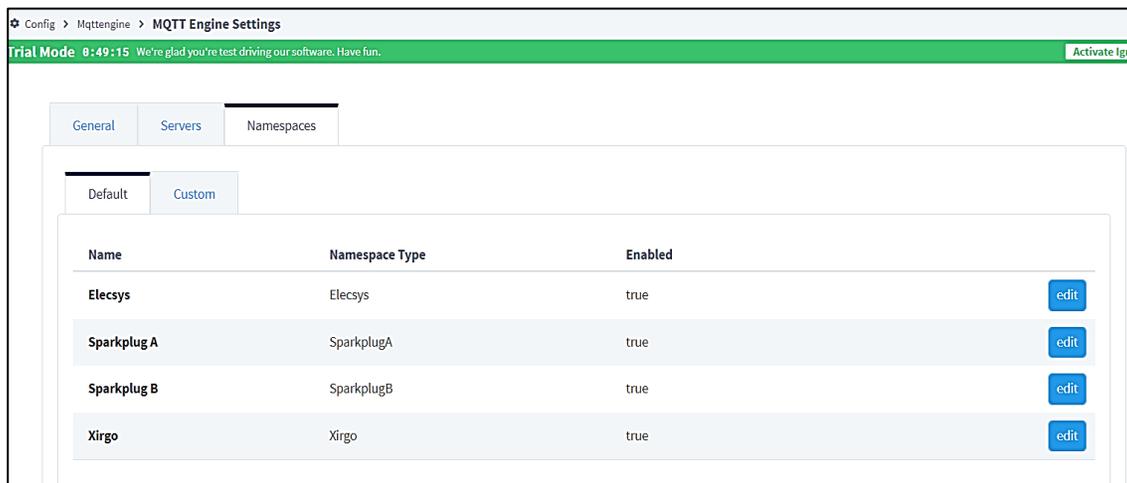
MQTT Engine Settings				
Trial Mode: 9:51:46 We're glad you're test driving our software. Have fun. Activate Ignition				
General Servers Namespaces				
Settings Certificates				
Name	URL	Username	Status	
CLOUD-BROKER	tcp://20.124.205.206:1883	MrDowsebf	Connected	delete edit
LAN-BROKER	tcp://192.168.100.190:1883	MrDowsebf	Connected	delete edit

→ Create new MQTT Server Setting...

Es necesario verificar la configuración activa del protocolo *Sparkplug B*, para la adquisición de datos en tiempo real desde el transmisor MQTT, como lo ilustra la figura 40.

Figura 40

Verificación de configuración activa de protocolo Sparkplug B

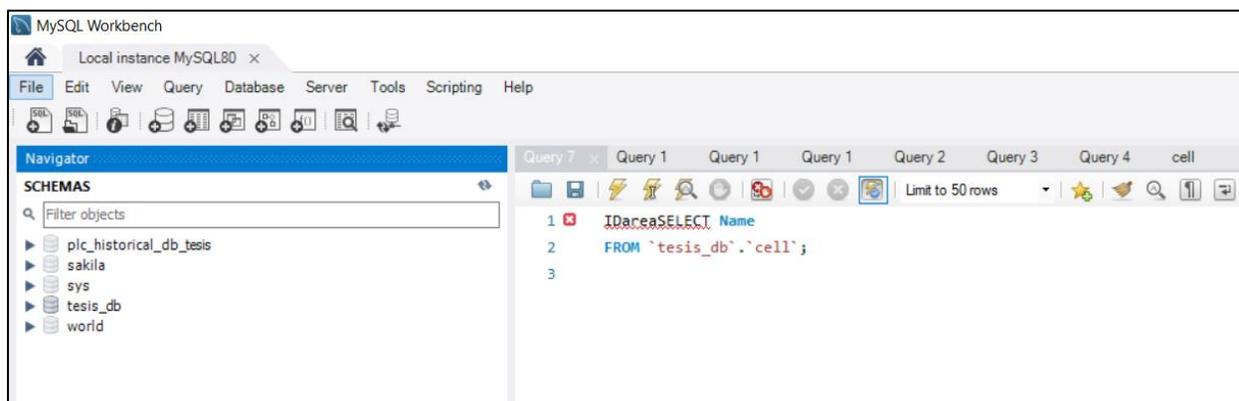


Configuración de la Base de Datos

Para realizar la historización de variables y enlazar la aplicación MES en el sistema de monitoreo, es necesario enlazar una base de datos a la plataforma *Ignition*. Se crea una base de datos en el ordenador *Gateway*, a través del aplicativo *MySQL Workbench*, con el nombre de “*tesis_db*”, como se ilustra a continuación:

Figura 41

Creación de base de datos a través de MySQL Workbench



Una vez creada la base de datos, se configura la conexión de ésta hacia la plataforma *Ignition*. Dentro del aplicativo *Ignition*, en el apartado de “*Bases de datos*”, se crea la conexión, y se direcciona a la base de datos creada (*tesis_db*), según indica la figura 42:

Figura 42

Detalle de configuración de conexión a plataforma Ignition

The screenshot shows the 'Database Connections' configuration window. The 'Main Properties' section includes the following fields:

- Name:** DB_MY_SQL_CONNECTION
- Description:** (empty text area)
- JDBC Driver:** MySQL
- Connect URL:** jdbc:mysql://localhost:3306/tesis_db
- Username:** root
- Change Password?:** Check this box to change the existing password.
- Password:** (masked)
- Password:** (masked)

Una vez creada la conexión, se debe verificar que sea “*válida*”. Esto indica que la base de datos está configurada correctamente, que *Ignition* está comunicándose exitosamente y que la base de datos está habilitada para almacenar los datos requeridos por los módulos MES, según se aprecia en la figura:

Figura 43

Verificación de validez de conexión de la base de datos

Name	Description	JDBC Driver	Translator	Status		
DB_MY_SQL_CONNECTION		MySQL	MYSQL	Valid	delete	edit
PLC_HISTORICAL_DB_CONNECTION	Database used for storage PLC variables independent from MES	MySQL	MYSQL	Valid	delete	edit

Configuración de Módulos MES.

Los módulos del sistema de Monitoreo de Ejecución de Manufactura permiten realizar gestión de procesos de producción en tiempo real. En el presente proyecto se utiliza el módulo de *Producción de Ignition* como gestor principal y el *Módulo de Recetas*, para cargar los datos de “batch” requeridos por el usuario al proceso de paletizado. Se requiere instalar los módulos en el aplicativo *Ignition*, como muestra la figura 44:

Figura 44

Detalle de instalación de módulos MES en el aplicativo Ignition

Sepasoft Inc.					
View Certificate					
Name	Version	Description	License	State	
Instrument Interface	2.81.0 (b2112272059)	(SP-3) A module that provides interfacing to instruments.	Trial	Running	More restart
OEE Downtime V2	2.81.0 (b2022011818)	(SP-17) A module that provides OEE and downtime version 2 functionality with task routing.	Trial	Running	More restart
Production	2.81.0 (b2022011818)	(SP-17) A module that provides the core MES production model functionality with task routing.	Trial	Running	More restart
Production Simulator	2.81.0 (b2112272059)	(SP-3) A module that provides a production simulator driver.	Trial	Running	More restart
Recipe	2.81.0 (b2022011818)	(SP-17) A module that provides recipe management.	Trial	Running	More restart

Una vez instalados los módulos, es necesario enlazar la conexión de la base de datos creada anteriormente al módulo de Producción, para lo cual se añade dicha conexión en las propiedades del módulo MES, como se ilustra a continuación:

Figura 45

Detalle de enlace de base de datos al módulo de Producción

Config > Mes > MES Module Settings
 Trial Mode 11:41:22 We're glad you're testing our software. Have fun.

Runtime Datasource

Runtime Database: DB_MY_SQL_CONNECTION
 The database connection to store runtime production data.
 (Stop all production runs before changing this setting.)

Analysis Datasource

Analysis Database: DB_MY_SQL_CONNECTION
 The database connection to store historical analysis production data to. Multiple sites can be set to the same analysis database to allow enterprise reporting.
 (Stop all production runs before changing this setting.)

Analysis Database (Auxiliary): DB_MY_SQL_CONNECTION
 The auxiliary or mirror database connection to store historical analysis production data to.
 (Stop all production runs before changing this setting.)

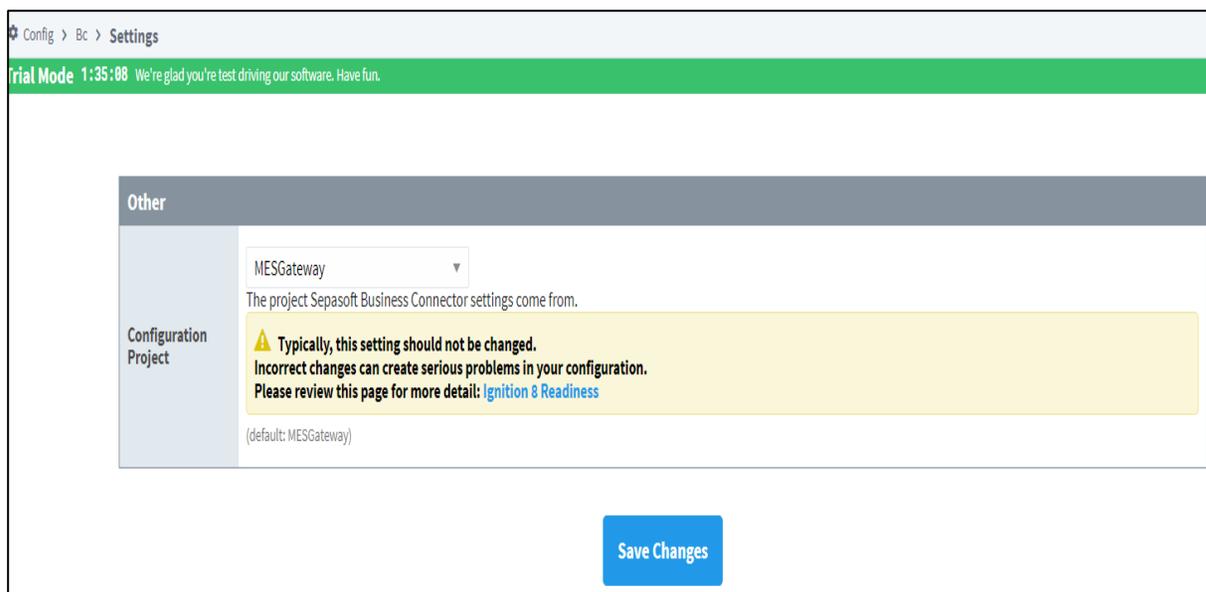
Analysis Query Cache Duration: 300
 Number of seconds to cache analysis results. Increasing this setting will reduce the load on the database but, will delay the propagation of current production information to the analysis results.

Analysis Read Lock Timeout: 30
 When executing analysis, the number of seconds to wait when the analysis engine configuration is being updated. The analysis engine configuration is updated when equipment, equipment modes, equipment states, counters or additional factors are changed.

A continuación, se debe asignar el proyecto de edición en el cual se van a utilizar los módulos MES, es decir, el programa de ingeniería donde se desarrollará el aplicativo SCADA-MES. El nombre del proyecto ha sido designado como “MESGateway”, como lo muestra la figura 46:

Figura 46

Asignación del proyecto de edición para enlace del módulo de Producción



Una vez configurados los dispositivos de la red IIoT redundante, los módulos MQTT, MES, base de datos, PLC y proceso de paletizado, se diseña el aplicativo de control y monitoreo en tiempo real SCADA-MES.

Diseño de la Aplicación SCADA-MES

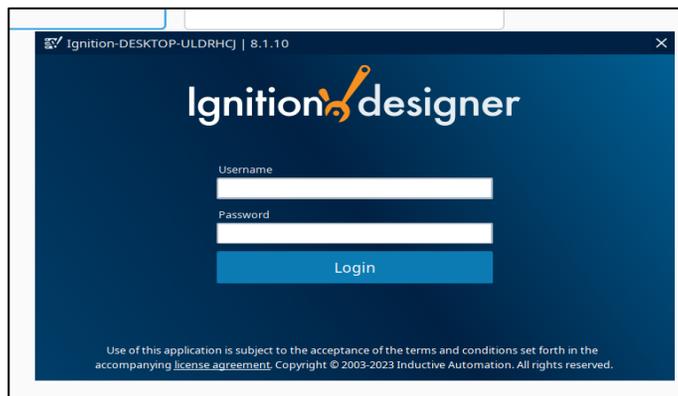
El *software* en el cual se diseña el aplicativo es “*Ignition Designer Launcher*”, herramienta que permite el diseño de pantallas, acceso a *tags* MQTT, historización de variables y lectura/escritura de datos del sistema SCADA-MES.

Una ventaja significativa de esta herramienta, es que permite acceder a los módulos de *Ignition* no solamente del dispositivo local, sino también en cualquier dispositivo conectado a la red, de forma que se puede obtener información de todos los dispositivos IIoT de forma remota.

El aplicativo exige la autenticación del usuario, como se muestra a continuación:

Figura 47

Detalle de autenticación de usuario en el software de edición



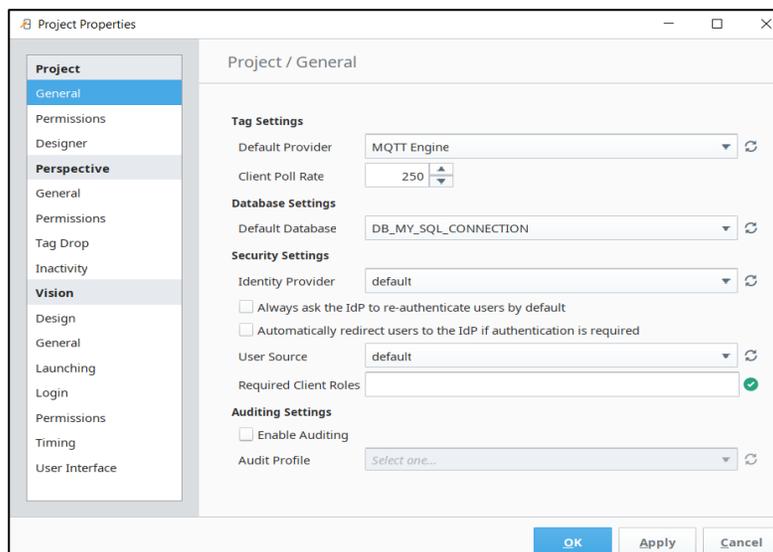
Evaluación Inicial.

Es necesario asignar al proyecto la conexión de la base de datos configurada en el “Gateway”, de lo contrario no se obtendrá acceso a la base de datos de MySQL ni a los paquetes de datos proveídos por Ignition y el módulo MES.

La asignación se la realiza en las propiedades del proyecto de edición.

Figura 48

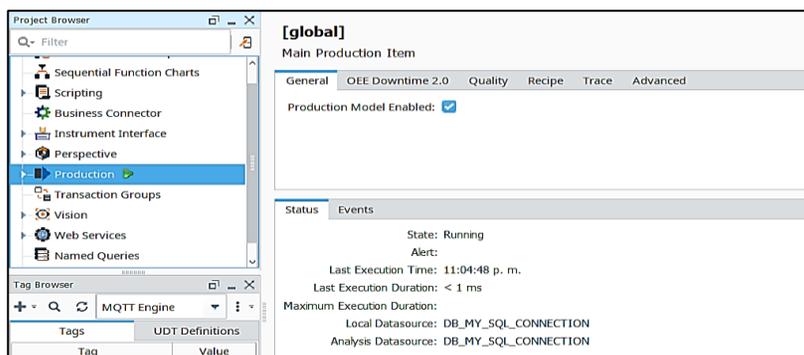
Detalle de asignación de la base de datos en el software de edición



Es necesario verificar que el módulo de “Producción” se encuentre ejecutándose correctamente, con status de “Activo” y tenga comunicación exitosa con el motor de la base de datos, como se indica en la siguiente figura:

Figura 49

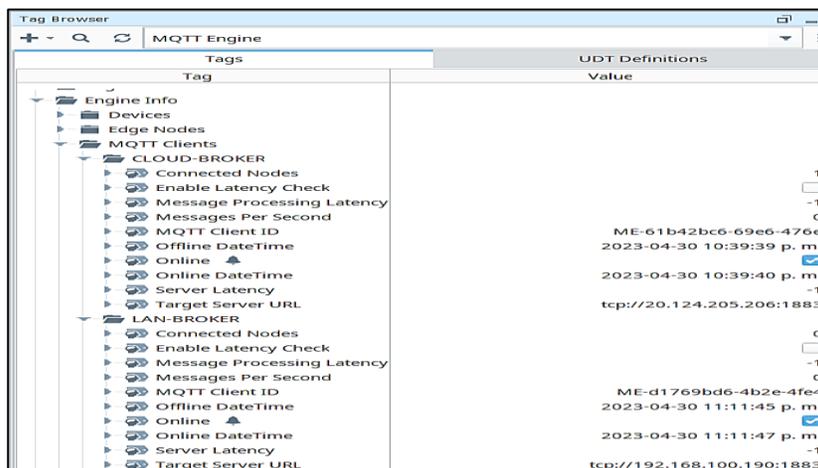
Verificación del módulo de Producción



Es necesario verificar en los *tags* MQTT, que exista conexión con los *broker* primario y redundante, con el fin de garantizar que los paquetes MQTT están siendo transmitidos y recibidos correctamente. Si todas las conexiones son exitosas, se debe visualizar que ambos *broker* están en línea (*online*), pero el primario tiene el nodo conectado a la red IIoT, lo que verifica la redundancia configurada en estos dispositivos, como se ilustra en la figura 50:

Figura 50

Verificación de status de brokers MQTT



Se debe verificar en los *tags* MQTT que el módulo de Recepción tenga comunicación con el módulo de Transmisión a través del *broker* MQTT, verificando el ID del protocolo *Sparkplug B* configurado en el dispositivo *Raspberry Pi*. Cabe destacar que no es necesario crear ningún *tag* de lectura de los paquetes MQTT, ya que *Ignition* los crea automáticamente a través de la red IIoT si todos los dispositivos están configurados correctamente y la autenticación de seguridad es exitosa. Se debe visualizar la información de la red IIoT como se muestra en la figura 51:

Figura 51

Comunicación entre publicador/suscriptor MQTT a través del broker primario

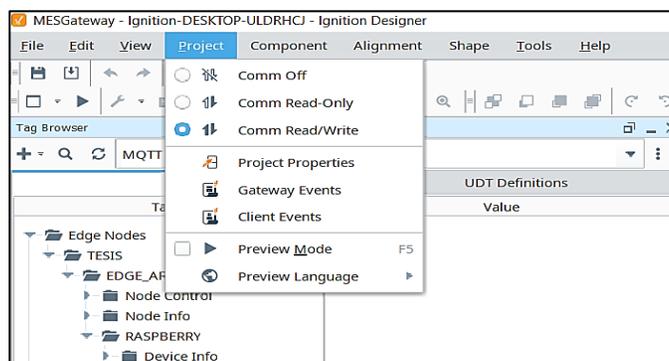
Tag	Value
bdSeq	9
Birth Count	10
Current Server	CLOUD-BROKER
Data Latency	-441
Death Count	4
Message Schema	Sparkplug B v1.0
Offline DateTime	2023-04-30 11:11:45 p. m.
Online	<input checked="" type="checkbox"/>
Online DateTime	2023-04-30 11:11:50 p. m.
Reset Info	<input type="checkbox"/>
seq	16
Total Bytes Recvd	882.017
Total Bytes Xmit	4.547
Transmission Version	4.0.8 (b2021071520)

Lectura de los Tags MQTT

Una vez verificada la comunicación con los dispositivos IIoT, se deben leer los *tags* provenientes del dispositivo *Raspberry Pi*, los cuales son los datos del PLC transmitidos a través del protocolo MQTT, para ello es necesario configurar que los *tags* tengan las propiedades de lectura/escritura, como se detalla a continuación:

Figura 52

Detalle de revisión de propiedades de Lectura/Escritura (Read/Write)



Configuradas las variables con acceso de lectura/escritura, se debe verificar que los *tags* que han sido publicados por el PLC se puedan leer correctamente en el *Gateway*. Para esto, no es necesario copiar las variables desde el transmisor MQTT, ya que son levantadas automáticamente por la red, con lo cual se pueden utilizar los *tags* necesarios para diseñar las pantallas del aplicativo SCADA-MES. Los *tags* leídos deben coincidir con el ID *Sparkplug B* configurado en el módulo de Transmisión “(TESIS/EDGE_ARM/RASPERRY)”; de esta manera se obtienen lecturas en tiempo real de las variables del PLC, según se ilustra a continuación:

Figura 53

Verificación de tags publicados por el PLC a través de la red IIoT

The screenshot shows the MQTT Engine interface with a tree view of tags and a table of values. The tree view shows 'Edge Nodes' expanded to 'TESIS', which contains 'EDGE_ARM', 'RASPERRY', 'PLC 1', 'PLC_INFORMATION', and 'PLC_SOFTLOGIX'. The 'PLC_SOFTLOGIX' folder is expanded to show 'COMANDOS HMI' with various tags and their values.

Tag	Value
BOOL_HMI_L1	1
BOOL_HMI_L2	1
EXT_CANT_TYP_1_HMI	3
EXT_CANT_TYP_2_HMI	3
L1_CT_ELE_01_HMI	1
L1_QTY_REQ	3
L1_RFID_READ_SERIAL_NUM_HMI	0
L1_Start_command_HMI	1
L2_CT_ELE_01_HMI	1
L2_QTY_REQ	2
L2_RFID_READ_SERIAL_NUM_HMI	0
L2_Start_command_HMI	2
SEL_EXT_L1_L2	2
START_EXT_L1_HMI	1
START_EXT_L2_HMI	1

Desarrollo de la Aplicación SCADA-MES

La aplicación SCADA-MES ha sido desarrollada tomando en cuenta las variables principales del proceso de paletizado, de tal forma que el usuario obtenga información útil del proceso y pueda interactuar con el mismo.

Se han desarrollado 8 pantallas con las siguientes cabeceras:

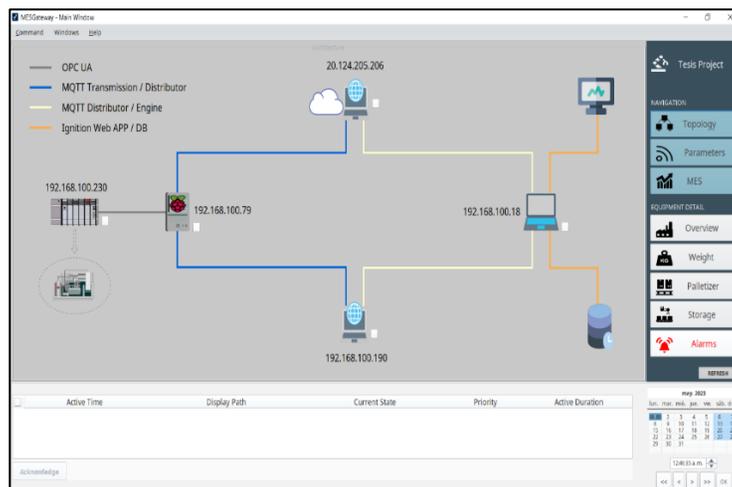
- Topología
- Parámetros
- MES
- Vista general
- Pesaje
- Paletizado
- Almacenamiento
- Alarmas

Pantallas Ancladas

Constan del menú de navegación, un “*banner*” de alarmas y un calendario.

Figura 54

Menú de navegación para acceso a pantallas de control, monitoreo y alarmas

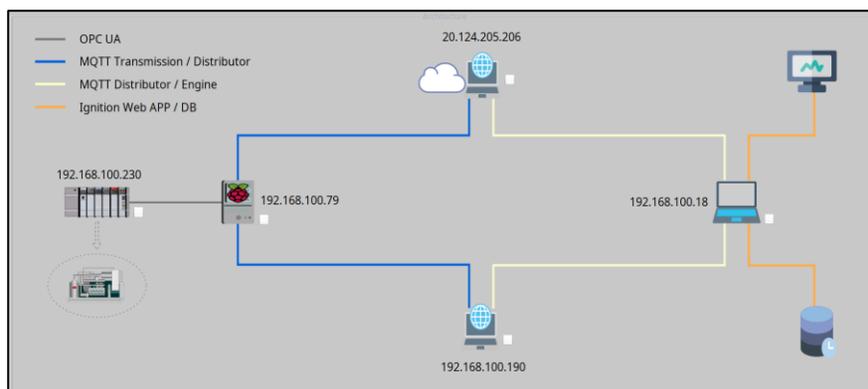


Topología

Muestra la arquitectura de la red, con sus respectivos dispositivos, *status* y direcciones IP, y una indicación visual de los distintos módulos y protocolos utilizados en la arquitectura IIoT:

Figura 55

Detalle de arquitectura de la red con elementos y protocolos de arquitectura IIoT



Parámetros

Permite visualizar los parámetros de red de los dispositivos IIoT, con sus respectivas variables relevantes como direcciones IP, *status*, nodos conectados, nombres de dispositivos conectados, y cantidad de paquetes transmitidos.

Figura 56

Detalle de visualización de parámetros de red de los dispositivos IIoT

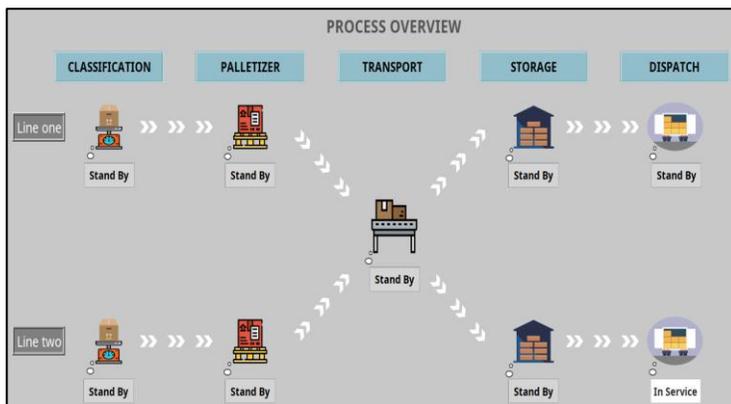
CLOUD BROKER		ENGINE NODE	
Target Server URL	tcp://20.124.205.206:1883	Current Server	LAN-BROKER
Online DateTime	1 may. 2023 0:05:29	Online DateTime	1 may. 2023 0:11:20
Offline DateTime		Offline DateTime	1 may. 2023 0:08:48
Connected nodes	0	Last node to Connect	TESIS/EDGE_ARM
Messages per second	0	Total Data Mesg Count	1.355
Online	ON	Devices Online	1
LAN BROKER		EDGE NODE	
Target Server URL	tcp://192.168.100.190:1883	Message Schema	Sparkplug B v1.0
Online DateTime	1 may. 2023 0:05:42	PLC Description	PLC for palletizer project
Offline DateTime		OPC State	CONNECTED
Connected nodes	1	System Name	Ignition-raspberrypi
Messages per second	0	Hostname	192.168.100.230
Online	ON	Online	ON

Vista General

Permite visualizar de forma general del proceso de paletizado, con sus respectivos subprocesos y el estado de cada uno.

Figura 57

Detalle de interfaz general de proceso de paletizado



Pesaje

Muestra el proceso de pesaje de las cajas de alimentación a las paletizadoras, el *status* de cada uno de los motores, el *setpoint* de pesaje, los comandos de arranque y los botones de simulación de falla de los motores para gestión de las alarmas.

Figura 58

Detalle de interfaz de proceso de pesaje

The interface is split into two panels for 'Line One' and 'Line Two'. Each panel contains the following information:

- Parameters:** Setpoint (2.5 Kg), Box size (Regular/Oversize checkboxes).
- Motors Status:**

Motors Status	Data
Box feeder	Passed 2.243
Electronic scale	Rejected 27
Out conveyor	Total 2.243
- Controls:** START button, Motor Fail indicator, and Reset button.

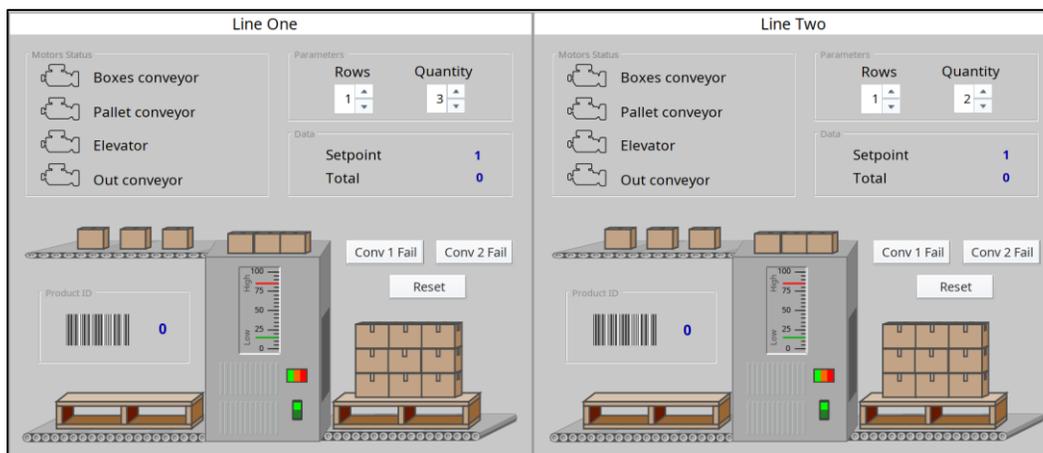
At the bottom of each panel is a visual representation of a weighing scale with a box on it, showing '0 Kg'.

Paletizado

Permite visualizar las líneas de paletizado, el *status* de los motores, los parámetros de producción, la lectura de los sensores RFID de cada uno de los *pallets*, los botones de simulación de falla de motores, botones de *reset* y el total de *pallets*.

Figura 59

Detalle de interfaz de proceso de paletizado

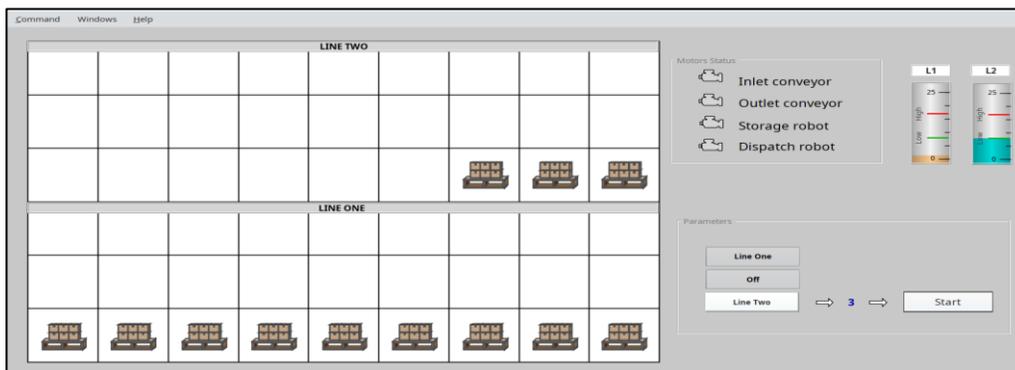


Almacenamiento

Permite visualizar los estantes de almacenamiento de cada una de las líneas de producción con sus respectivos lugares disponibles, *status* de los motores de ingreso y despacho, cantidad de *pallets* a despachar y el comando de inicialización de despacho.

Figura 60

Detalle de interfaz de almacenamiento



Alarmas

Permite visualizar las alarmas existentes y pasadas con su respectivo estado. Las alarmas configuradas son: Falla de motores y pérdida de comunicación de los dispositivos IIoT.

Figura 61

Detalle de visualización de alarmas

Active Time	Clear Time	Display Path	Current State	Active Duration	Priority
1/5/23 0:39	1/5/23 0:41	CLOUD BROKER COMM LOST	Cleared, Unacknowledged	1 minutes, 58 seconds	Critical
30/4/23 22:03	1/5/23 0:11	OPC UA Communication lost	Cleared, Unacknowledged	2 hours, 8 minutes, 1 seconds	Critical
1/5/23 0:08	1/5/23 0:11	MQTT Engine Communication lost	Cleared, Unacknowledged	2 minutes, 30 seconds	Critical
1/5/23 0:08	1/5/23 0:11	Raspberry PI Communication lost	Cleared, Unacknowledged	2 minutes, 30 seconds	Critical
1/5/23 0:03	1/5/23 0:06	MQTT Engine Communication lost	Cleared, Unacknowledged	3 minutes, 30 seconds	Critical
1/5/23 0:03	1/5/23 0:06	Raspberry PI Communication lost	Cleared, Unacknowledged	3 minutes, 29 seconds	Critical
1/5/23 0:03	1/5/23 0:05	LAN BROKER COMM LOST	Cleared, Unacknowledged	2 minutes, 24 seconds	Critical
1/5/23 0:03	1/5/23 0:05	CLOUD BROKER COMM LOST	Cleared, Unacknowledged	2 minutes, 12 seconds	Critical
30/4/23 23:11	30/4/23 23:11	MQTT Engine Communication lost	Cleared, Unacknowledged	4 seconds	Critical
30/4/23 23:11	30/4/23 23:11	Raspberry PI Communication lost	Cleared, Unacknowledged	4 seconds	Critical
30/4/23 23:11	30/4/23 23:11	LAN BROKER COMM LOST	Cleared, Unacknowledged	2 seconds	Critical
30/4/23 22:04	30/4/23 22:39	CLOUD BROKER COMM LOST	Cleared, Unacknowledged	35 minutes, 9 seconds	Critical
30/4/23 22:03	30/4/23 22:04	MQTT Engine Communication lost	Cleared, Unacknowledged	42 seconds	Critical
30/4/23 22:03	30/4/23 22:04	Raspberry PI Communication lost	Cleared, Unacknowledged	41 seconds	Critical
30/4/23 22:02	30/4/23 22:03	LAN BROKER COMM LOST	Cleared, Unacknowledged	50 seconds	Critical
30/4/23 22:02	30/4/23 22:03	MQTT Engine Communication lost	Cleared, Unacknowledged	40 seconds	Critical
30/4/23 22:02	30/4/23 22:03	Raspberry PI Communication lost	Cleared, Unacknowledged	39 seconds	Critical

MES

Permite visualizar los parámetros de paletizado seleccionados en el módulo de Producción y en el módulo de Recetas. Es decir, en esta pantalla se configuran cuántos *pallets* se desean procesar, el *setpoint* de las líneas de pesaje y la cantidad de *pallets* que se requieren para ser despachados.

Figura 62

Detalle de interfaz de módulo de Recetas del sistema MES

Name	Recipe Setting	Live Value	Manual Value	Description
L1 EXTRACTOR SP	5	3	5	

Line One Weight	L1 WEIGHT SP1 L1 WEIGHT SP2	Line Two Weight	L2 WEIGHT SP1 L2 WEIGHT SP2
Line One Palletizer	L1 PALLETIZER SP1 L1 PALLETIZER SP2	Line Two Palletizer	L2 PALLETIZER SP1 L2 PALLETIZER SP2
Line One Extraction	L1 EXTRACTOR SP1 L1 EXTRACTOR SP2	Line Two Extraction	L2 EXTRACTOR SP1 L2 EXTRACTOR SP2
Search Line	Search Cell	Search SP	
LINE ONE	CTOR ROBOT	L1 EXTRACTO...	

Todas estas variables están configuradas en la base de datos del módulo de Producción MES, de tal forma que se pueden configurar diferentes recetas según se requiera, y los datos son almacenados y jerarquizados de acuerdo a la norma ISA-95, como se aprecia a continuación:

Figura 63

Jerarquización del proceso productivo MES según ISA-95

The screenshot displays the configuration of a recipe in a SCADA-MES system. The interface is divided into several sections:

- Tree View (Left):** Shows a hierarchical structure of the production process. The selected path is: Production > SF INDUSTRIES > ECUADOR > PALLETIZER > LINE ONE > PALLETIZER.
- Main Configuration Panel (Right):** Contains the following fields:
 - General: OEE Downtime 2.0, Quality, Recipe (selected), Trace, Advanced.
 - Licensed: Yes
 - Sub Recipe Mask: (empty)
 - Inherit Recipe Values Mode: Parent Recipe
 - Recipe Values: A table with columns Name, Description, Tag, Request V..., Enable Sc..., and Enable Va...

Name	Description	Tag	Request V...	Enable Sc...	Enable Va...
L2 PALLETS SP		[default]SF ...		true	true
L2 ROW PALLETIZER SP		[default]SF ...		true	true
- Tag Browser (Bottom Left):** Shows a list of tags under the 'default' group, including 'SF INDUSTRIES' and 'Test_tags'.
- Status Panel (Bottom Right):** Displays the current state of the recipe:
 - State: Running
 - Alert:
 - Last Execution Time: 12:58:54 a. m.
 - Last Execution Duration: < 1 ms
 - Maximum Execution Duration:
 - Local Datasource: DB_MY_SQL_CONNECTION
 - Analysis Datasource: DB_MY_SQL_CONNECTION

Una vez configuradas y programadas las pantallas SCADA-MES, se puede realizar el control y monitoreo del proceso de paletizado en tiempo real.

Capítulo IV

Implementación

Prueba de Comunicación entre Publicadores y Suscriptores

Como primer paso para verificar la conectividad entre dispositivos, se realiza un comando “ping” hacia todos los nodos, verificando sus tiempos de respuesta, obteniéndose los siguientes resultados:

Tabla 4

Pruebas de comando “ping” para verificar comunicación entre dispositivos IIoT

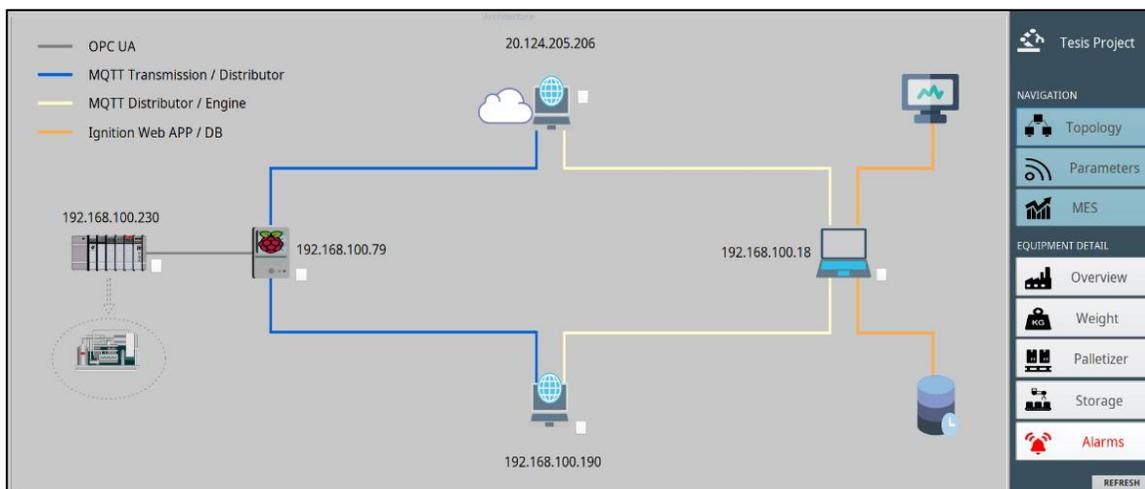
PRUEBA DE CONECTIVIDAD ICMP		
Equipo origen	Equipo destino	Velocidad (ms)
	192.168.100.230	4
Ignition Gateway	192.168.100.190	4
192.168.100.18	192.168.100.79	9
	20.124.205.206	105

Prueba de Redundancia y Comunicación de Dispositivos MQTT.

El paso más importante de validación de la arquitectura es comprobar la redundancia de la red IIoT. Para ello, se verifica que todos los dispositivos se encuentren intercomunicados en la pantalla de “Topología”:

Figura 64

Interfaz de comprobación de redundancia de la red IIoT



También se valida que el *broker* primario esté conectado al nodo de recepción.

Figura 65

Interfaz de comprobación de conexión de broker primario

The screenshot displays the 'Tesis Project' interface with a navigation sidebar on the right containing 'Topology', 'Parameters', 'MES', 'Overview', 'Weight', 'Palletizer', 'Storage', and 'Alarms'. The main content area is divided into four panels:

- CLOUD BROKER:** Target Server URL: tcp://20.124.205.206:1883; Online DateTime: 1 may. 2023 11:10:15; Offline DateTime: 1 may. 2023 11:10:15; Connected nodes: 1; Messages per second: 0,5; Online: ON.
- ENGINE NODE:** Current Server: CLOUD-BROKER; Online DateTime: 1 may. 2023 11:11:24; Offline DateTime: 1 may. 2023 11:11:21; Last node to Connect: TESIS/EDGE_ARM; Total Data Mesg Count: 1.887; Devices Online: 1.
- LAN BROKER:** Target Server URL: tcp://192.168.100.190:1883; Online DateTime: 1 may. 2023 11:11:23; Offline DateTime: 1 may. 2023 11:11:21; Connected nodes: 0; Messages per second: 0; Online: ON.
- EDGE NODE:** Message Schema: Sparkplug B v1.0; PLC Description: PLC for palletizer project; OPC State: CONNECTED; System Name: Ignition-raspberrypi; Hostname: 192.168.100.230; Online: ON.

Para verificar la redundancia, se desconecta el *broker* Cloud, y se analiza la respuesta de la red.

Figura 66

Interfaz de respuesta del sistema por desconexión de broker primario

The screenshot shows the same interface as Figure 65, but with updated status and an alarm log:

- CLOUD BROKER:** Online: OFF.
- ENGINE NODE:** Current Server: LAN-BROKER; Online DateTime: 1 may. 2023 11:14:20; Offline DateTime: 1 may. 2023 11:14:19; Total Data Mesg Count: 1.998.
- LAN BROKER:** Connected nodes: 1; Messages per second: 0,5; Online: ON.
- EDGE NODE:** Same as Figure 65.

An alarm log is visible at the bottom, showing a critical event:

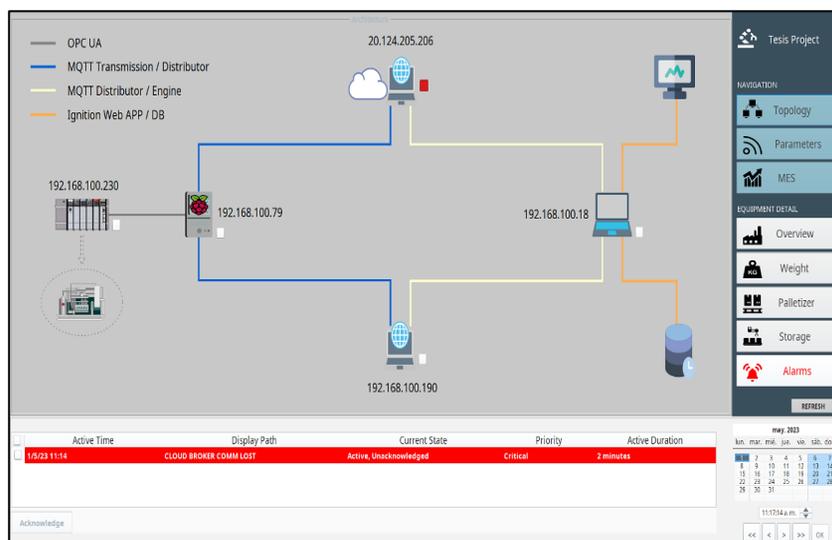
Active Time	Display Path	Current State	Priority	Active Duration
1/5/23 11:14	CLOUD BROKER COMM LOST	Active, Unacknowledged	Critical	17 seconds

The interface also includes a calendar for May 2023 and a clock showing 11:15:08 a.m.

Como se puede apreciar, la red IIoT detecta inmediatamente la pérdida de comunicación del *broker* primario, sin embargo, el *broker* de respaldo entra en servicio, lo que permite mantener la comunicación. También se evidencia esta operación en la pantalla de topología:

Figura 67

Interfaz de detección de pérdida de conexión de broker primario



Una vez restablecida la comunicación con el *broker* primario, éste entra en servicio y se restaura la redundancia de la red. Se realiza la misma prueba de conectividad con los equipos *Raspberry Pi*, PLC y *broker* LAN; al realizar la desconexión se activan las alarmas de comunicación en el SCADA, según indican las figuras 68 y 69:

Figura 68

Pruebas de conexión y desconexión de nodos de la red IloT

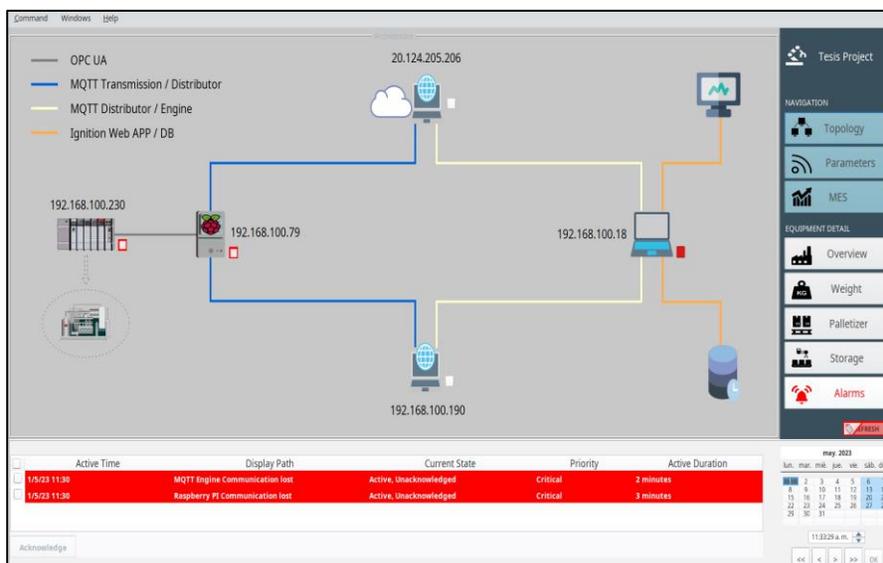


Figura 69

Verificación de pérdida de comunicaciones con dispositivos IIoT

The screenshot displays a SCADA-MES interface for 'Tesis Project'. It is divided into several sections:

- CLOUD BROKER:** Target Server URL: tcp://20.124.205.206:1883; Online DateTime: 1 may. 2023 11:20:38; Offline DateTime: 1 may. 2023 11:20:38; Connected nodes: 0; Messages per second: 0; Online: ON.
- ENGINE NODE:** Current Server: LAN-BROKER; Online DateTime: 1 may. 2023 11:14:20; Offline DateTime: 1 may. 2023 11:30:28; Last node to Connect; Total Data Mesg Count: 2.619; Devices Online: 0.
- LAN BROKER:** Target Server URL: tcp://192.168.100.190:1883; Online DateTime: 1 may. 2023 11:11:23; Offline DateTime: 1 may. 2023 11:11:21; Connected nodes: 0; Messages per second: 0; Online: ON.
- EDGE NODE:** Message Schema: Sparkplug B v1.0; PLC Description: C for palletizer project; OPC State: UNCONNECTED; System Name: nition-raspberrypi; Hostname: 2.168.100.230; Online: N.

On the right, there is a sidebar with 'NAVIGATION' (Topology, Parameters, MES) and 'EQUIPMENT DETAIL' (Overview, Weight, Palletizer, Storage, Alarms). At the bottom right, there is a calendar for May 2023 and a clock showing 11:34:07 a.m.

At the bottom, an alarm log table shows the following entries:

Active Time	Display Path	Current State	Priority	Active Duration
1/5/23 11:30	MQTT Engine Communication lost	Active, Unacknowledged	Critical	3 minutes
1/5/23 11:30	Raspberry PI Communication lost	Active, Unacknowledged	Critical	3 minutes

Se procede a restaurar la comunicación entre dispositivos y el sistema recupera conectividad con todos los dispositivos de la red IIoT.

Pruebas de Desempeño de la Aplicación SCADA-MES

En el apartado anterior se ha evaluado la aplicación respecto a la conectividad de la red IIoT, sin embargo, en este apartado se evaluará el desempeño en tiempo real del control y monitoreo del proceso de paletizado.

El punto de partida es la pantalla MES, ya que, como se había indicado, en esta pantalla están programadas las recetas que le indican a las líneas de producción cuántos *pallets* procesar y cuántos se desean despachar. Se verifica que los *setpoints* se carguen correctamente al PLC, para que ejecute la lógica de programación de las secuencias.

Para este requerimiento puntual, se han seleccionado los *setpoints* detallados en la tabla 5:

Tabla 5

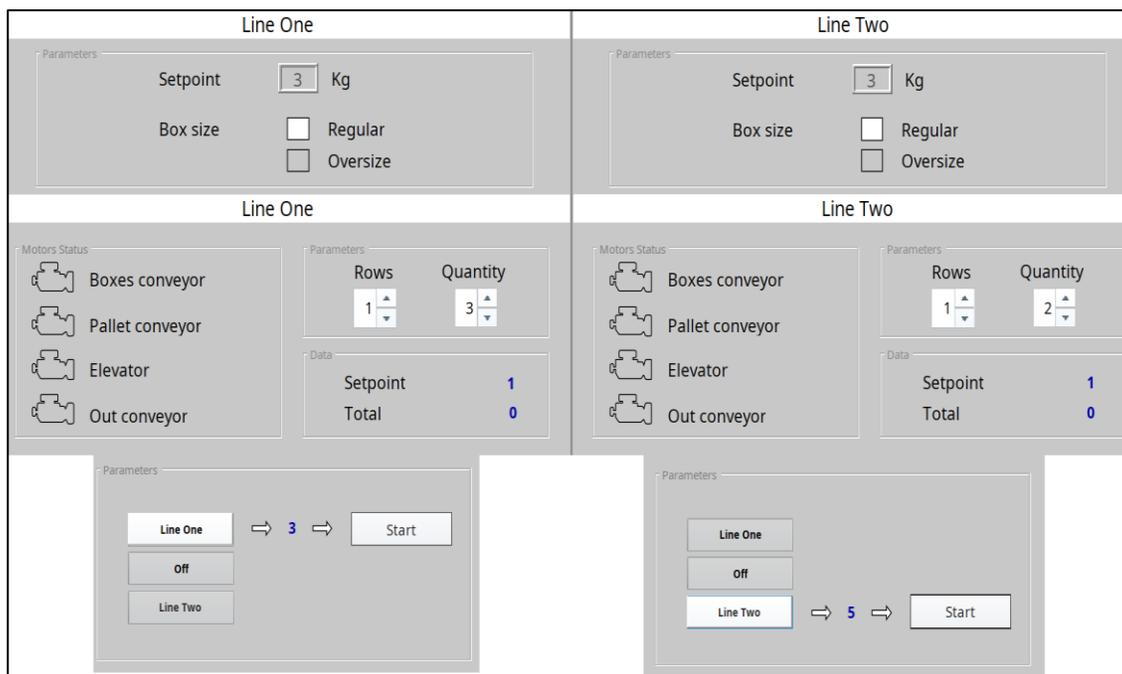
Detalle de selección de setpoints

RECETAS PARA LÍNEAS DE PALETIZADO				
	Pesaje (kg)	Pallets (u)	Filas de cajas (u)	Despacho (u)
Línea Uno	3	3	1	3
Línea Dos	3	2	1	5

Se comprueban los *setpoints* en las pantallas del sistema SCADA-MES, según la figura 70.

Figura 70

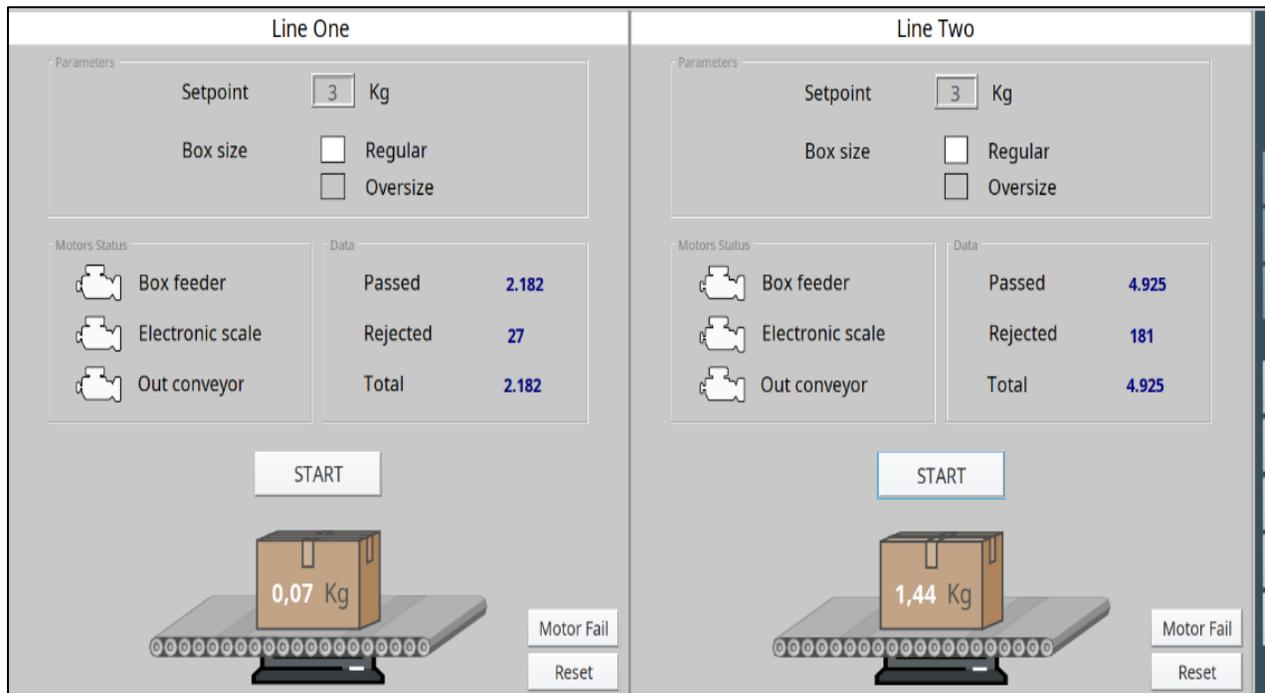
Detalle de comprobación de setpoints en las pantallas del sistema SCADA-MES



Una vez configurados los *setpoint*, se puede arrancar el proceso de paletizado mediante los botones de las pantallas correspondientes. Se evalúa el desempeño del programa verificando que las líneas de paletizado cumplan con lo requerido en las recetas y las secuencias se ejecuten correctamente en el *software Factory IO*.

Figura 71

Pantalla SCADA – comando de arranque de secuencias

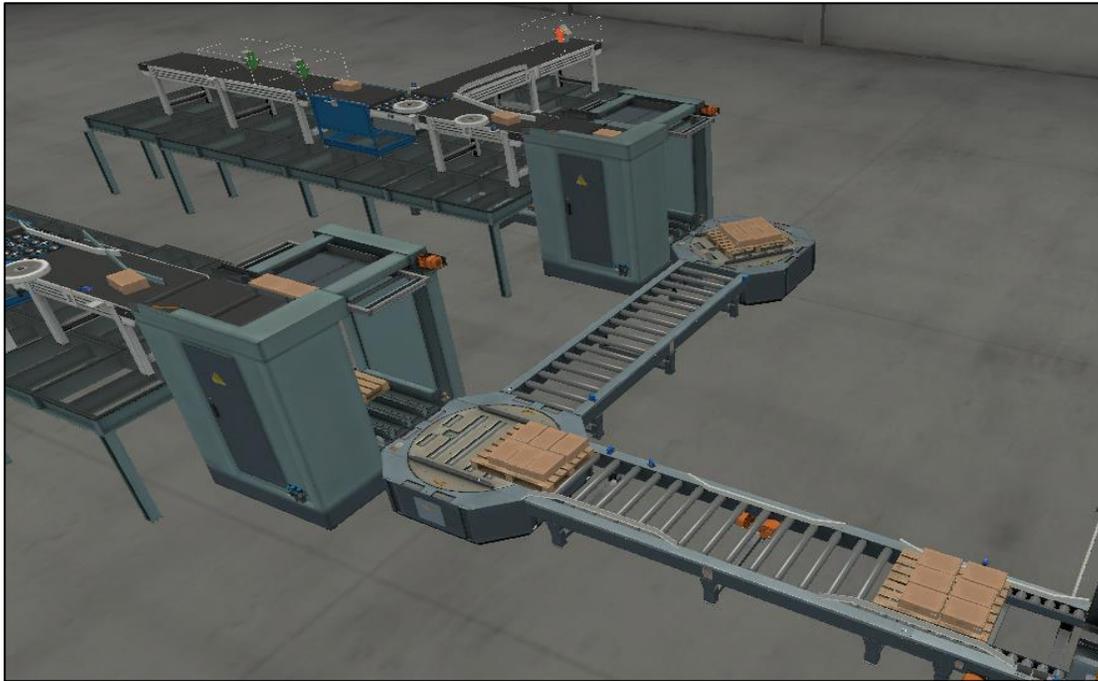
**Figura 72**

Factory IO – Líneas de paletizado en ejecución



Figura 73

Factory IO – Bandas de transporte en ejecución

**Figura 74**

Factory IO – Proceso de almacenamiento automático en marcha



Figura 75

Factory IO – Proceso de almacenamiento automático finalizado



Figura 76

Pantalla SCADA – Comando de arranque para despacho según receta

The SCADA interface displays the status of two production lines. The 'LINE TWO' section shows a 3x8 grid with six pallets in the bottom row. The 'LINE ONE' section shows a 3x8 grid with nine pallets in the bottom row and three pallets in the middle row. The 'Motors Status' panel includes indicators for Inlet conveyor, Outlet conveyor, Storage robot, and Dispatch robot, along with two level gauges (L1 and L2) showing low and high levels. The 'Parameters' panel features buttons for 'Line One', 'Off', and 'Line Two', and a 'Start' button. A blue arrow points from 'Line One' to a '3' in a box, which then points to the 'Start' button.

Figura 77

Factory IO – Despacho finalizado



Evaluación de Historización de Variables.

Como se había mencionado previamente, la base de datos debe estar configurada, instalada y comunicada correctamente al *Gateway*, ya que, de otra forma, no es posible levantar la comunicación con los *tags* de Producción provistos por *Ignition* para la gestión de datos.

Se evalúa la historización de variables en el aplicativo *MySQL Workbench*, verificando que las variables de Producción se encuentren historizadas correctamente, según indica la figura 78:

Figura 78

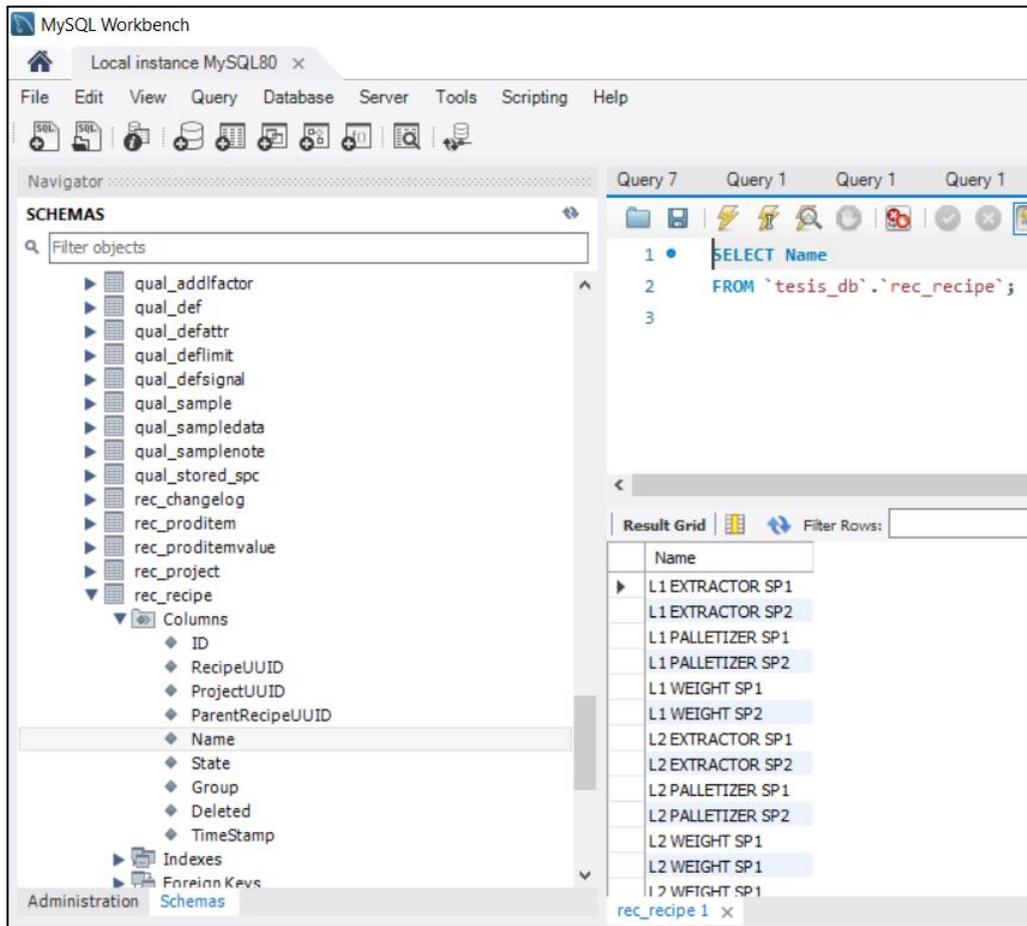
Verificación de almacenamiento de datos de líneas de producción

ID	ParentID	Name	TimeStamp
1	1	FEEDING CONVEYOR	2023-04-27 22:25:54
2	2	FEEDING CONVEYOR	2023-04-28 16:27:14
3	3	FEEDING CONVEYOR	2023-04-28 16:32:25
4	3	ELECTRONIC BALANCE	2023-04-28 16:32:25
5	4	Feeder_conveyor_1	2023-04-28 16:37:06
6	4	Storage_robot	2023-04-28 16:37:06
7	4	Feeder_conveyor_2	2023-04-28 16:37:06
8	5	Feeder conveyor	2023-04-28 16:37:06
9	5	Feeder pallet	2023-04-28 16:37:06
10	5	Elevator	2023-04-28 16:37:06
11	5	Out_pallet_conveyor	2023-04-28 16:37:06
12	5	Out_pallet_conveyor_2	2023-04-28 16:37:06
13	6	Extract_robot	2023-04-28 16:37:06

Se verifica que los datos de las recetas se almacenen correctamente en la base de datos, como se muestra a continuación:

Figura 79

Verificación de almacenamiento de recetas en base de datos

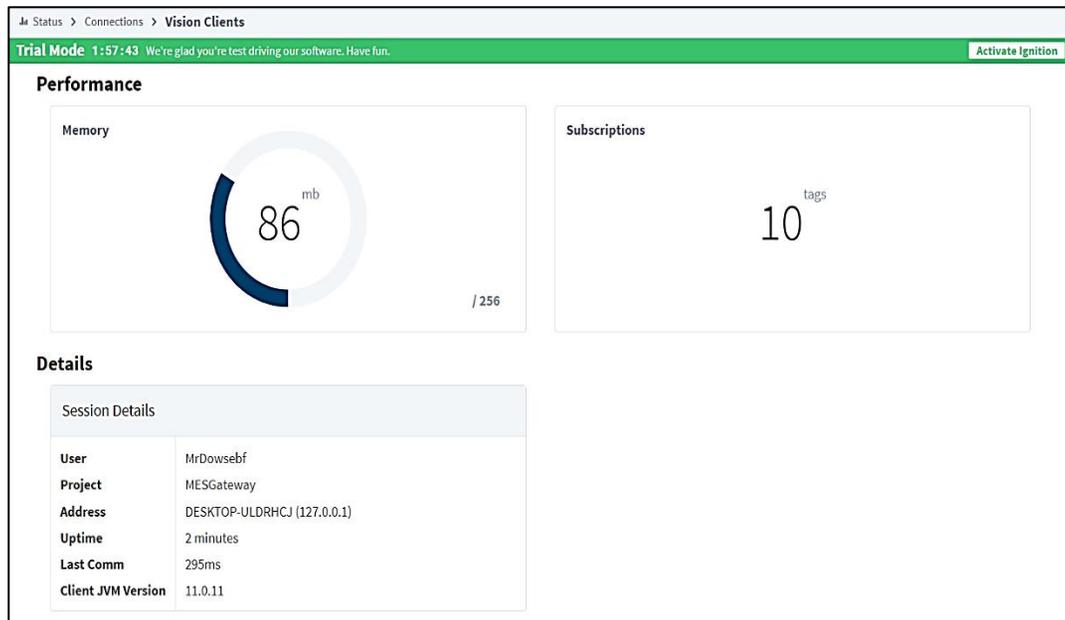


Pruebas de Desempeño de la Red en Tiempo Real

Una vez que se ha comprobado el correcto funcionamiento de las secuencias de programación del proceso de paletizado a través de una arquitectura redundante IIoT, se verifica el desempeño de la arquitectura implementada en tiempo real. Se verifica conectividad entre todos los componentes y se ejecuta el programa de aplicación cliente *Ignition*. Se navega por las pantallas del sistema SCADA-MES y se analiza la velocidad de comunicación en la arquitectura IIoT desde el *Gateway* de la aplicación, según lo indica la siguiente figura:

Figura 80

Detalle de análisis de velocidad de comunicación en la arquitectura IIoT



Según indica la figura, hay varios parámetros que el sistema calcula para determinar la velocidad de transmisión de los paquetes MQTT, y están regidos principalmente por la pantalla en uso y el número de tags MQTT suscritos en cada una.

Analizando los datos se obtienen los siguientes resultados:

Tabla 6

Detalle de parámetros de velocidad de transmisión de paquetes MQTT

Pantalla	Parámetros de red IIoT (tiempo real)		
	Tags	Uso de memoria (Mb)	Velocidad (ms)
Topología	10	78	325
Parámetros	66	79	299
MES	66	81	277
Vista general	84	87	165
Pesaje	123	93	297
Paletizado	160	90	229
Almacenamiento	240	94	361
Alarmas	240	97	126

Validez de la Hipótesis

Se pudo verificar que la máxima latencia de comunicaciones de la red IIoT corresponde a 361 milisegundos, lo que corresponde a un valor menor a dos segundos; indicador mediante el cual se valida la hipótesis.

De la misma manera, se evidencia que las variables de proceso, control y monitoreo han sido transmitidas al 100%, y que la redundancia de la red IIoT funciona correctamente, lo que corrobora la validez de la hipótesis planteada.

Conclusiones

- Se ha implementado exitosamente una arquitectura IIoT redundante, logrando el monitoreo en tiempo real de un proceso de paletizado simulado en un entorno virtual 3D, mediante la configuración adecuada de los dispositivos IIoT, y el uso del protocolo MQTT.
- La implementación de una arquitectura IIoT permite no solamente el monitoreo remoto de un proceso industrial, también ofrece el acceso a los datos y el análisis de forma más rápida y automática, agilizando la aplicación de cambios y mejoras en los procesos de manera remota. También permite dar un seguimiento a los planes de producción, detectar desviaciones, realizar control de calidad y, en definitiva, generar un entorno en el que las decisiones basadas en información sean prioritarias.
- La programación de una secuencia automática de un proceso de paletizado simulado en un entorno 3D, ha sido de gran importancia para el desarrollo y fortalecimiento de las competencias en Automatización Industrial, ya que ha permitido manipular variables de proceso y probar diversas secuencias y lógicas de control, sin la necesidad de contar con un proceso físico de producción para realizar dichas pruebas.
- La plataforma *SoftLogix* ha sido de gran utilidad para la integración del PLC con la red IIoT, debido a la tarjeta de red *Ethernet* que tiene embebida; lo cual permite la comunicación del PLC virtual con cualquier red TCP/IP, como si se tratara de un PLC físico. Además, al ser de la familia *Logix* de *Allen Bradley*, permite el uso de herramientas de programación de PLC's de alta gama.
- La arquitectura IIoT redundante implementada permite la comunicación en tiempo real de manera segura y confiable, permitiendo que los paquetes de datos MQTT sean transmitidos, distribuidos y recibidos correctamente y sin pérdidas. El envío de datos según el protocolo *Sparkplug B* es de fácil entendimiento, configuración y amigable al usuario, lo que permite minimizar confusiones en los datos que se desea enviar y recibir.

- Se ha configurado satisfactoriamente dos *broker* MQTT gracias a las recomendaciones técnicas emitidas por *Ignition* y *Cirrus Linx*. La redundancia ha sido probada tanto en el *broker* primario como en el de respaldo; el acceso a la configuración está protegida contra accesos no autorizados y la configuración de *Ignition* en el *broker* alojado en la nube se lo ha realizado sin problemas de compatibilidad y/o desempeño.
- Se ha desarrollado una aplicación SCADA-MES para el monitoreo del proceso, haciendo uso de las herramientas de programación SCADA de la plataforma *Ignition*. Se han utilizado diferentes módulos del aplicativo de *Ignition* para la creación y edición de pantallas gráficas.
- Se ha logrado la integración de las variables del módulo MES con las variables MQTT y con la base de datos, de tal forma que el resultado es una aplicación integral que permite alcanzar los objetivos planteados.
- Se ha logrado enlazar una base de datos al *Gateway* SCADA-MES, para lo cual se ha seguido las recomendaciones de *Ignition* de acuerdo con los manuales técnicos. La base de datos MySQL ha demostrado robustez y ser de fácil configuración para el usuario.

Recomendaciones

- Se recomienda realizar un análisis técnico exhaustivo de las herramientas y *software* a utilizar, ya que es necesario verificar la interoperabilidad de los dispositivos que se conectan a la red; su compatibilidad, protocolos de comunicación, puertos disponibles y robustez a nivel físico y lógico.
- Es recomendable realizar un análisis detallado de las arquitecturas que se deseen implementar, ya que dentro de las amplias topologías IIoT disponibles, cada una cuenta con sus propias especificaciones que deben ser analizadas rigurosamente para ser utilizadas en cualquier proyecto.
- Se recomienda investigar a profundidad el modo de funcionamiento del protocolo MQTT en una arquitectura IIoT, específicamente en los módulos de *Ignition*, ya que cada módulo tiene funciones y configuraciones específicas que permiten la comunicación de paquetes MQTT, y cada una de estas configuraciones deben ser analizadas antes de su configuración y puesta en servicio.
- Se recomienda implementar funciones de ciberseguridad cuando se trabaja con servidores en la nube, ya que es deseable evitar ataques externos. En el presente proyecto se ha optado por el acceso a través de un usuario registrado y autenticado, sin embargo, *Ignition* dispone otro tipo de estrategias para fortalecer la seguridad informática a sus sistemas IIoT.
- Se recomienda el entrenamiento previo sobre programación de PLC, sistemas SCADA, Redes Industriales y bases de datos, ya que el presente proyecto abarca prácticamente cuatro de los cinco peldaños de la pirámide de Automatización, partiendo en el nivel de Planta y terminando en el nivel de control de Producción.
- Se recomienda seguir las instrucciones técnicas publicadas por *Sepasoft* y *Cirrus Linx*, ya que se debe verificar la compatibilidad de versiones tanto del *Gateway* de *Ignition* como de los

módulos MQTT y MES. Si no existe una verificación de compatibilidad, los módulos no podrán ser instalados o presentarán fallas al momento de su ejecución.

- Se recomienda el uso de un dispositivo IIoT robusto para aplicaciones reales. En el presente proyecto se ha utilizado el equipo *Raspberry Pi*, el cual cumple satisfactoriamente como conversor de protocolo OPC UA – MQTT; sin embargo, en una aplicación industrial real, se deberá considerar la robustez y confiabilidad de los equipos usados para este propósito.
- Antes de la implementación de un proyecto con enfoque en Industria 4.0, se recomienda planificar el alcance, los objetivos y las metas a lograr, así como también realizar un análisis de la infraestructura tecnológica disponible y la selección de *software* con tecnología probada y confiable para garantizar el flujo confiable de datos.
- Se recomienda seguir las buenas pautas de diseño de sistemas SCADA y MES, según las normas ISA-101 e ISA-95, las cuales proporcionan información detallada y normada, que simplifican el diseño y permiten al programador estructurar de mejor manera su proyecto.

Bibliografía

- Acharya, V., & Kumar, S. (2018). Analyzing the factors in industrial automation using analytic hierarchy process. *Computers & Electrical Engineering*, Volume 71, October 2018, 9.
- Aguilar, L. (2021). *Internet de las cosas. Un futuro hiperconectado: 5G, inteligencia artificial, Big Data, Cloud, Blockchain y ciberseguridad*. Marcombo S.L.
- Aguiló, J. (2019). *Industria 4.0, la transformación digital en la industria*. Editorial UOC, S.L.
- Barbara. (2021). *IoT industrial: la respuesta a la digitalización industrial*. Obtenido de <https://barbaraiot.com/es/blog/el-iot-industrial-la-respuesta-a-la-digitalizacion-del-sector-industrial>
- Castillo, J. (2021). *Sistemas Secuenciales Programables*. Editex.
- Cirrus Link. (20 de abril de 2023). *MQTT Software for IIoT*. Obtenido de <https://cirrus-link.com/mqtt-software-for-iiot-scada/#>
- Conway, J. (2015). *The Industrial Internet of Things: an evolution to a smart manufacturing enterprise*. Obtenido de <https://es.rs-online.com/euro/img/general/pdf/iiot-evolution.pdf>
- Díaz, A., Cruz, H., & Botto-Tobar, M. (2022). Recent Advances in Electrical Engineering, Electronics and Energy. *Proceedings of the CIT 2021 Volume 1, Tomo 1*.
- Escaño, J., García, A., & García, J. (2019). *Integración de Sistemas de Automatización Industrial*. Ediciones Paraninfo.
- Factoryio. (20 de abril de 2023). *Diseñar y simular*. Obtenido de <https://factoryio.com/features>
- Gallardo, S. (2013). *Técnicas y procesos en instalaciones domóticas y automáticas*. Ediciones Paraninfo.
- IBM. (20 de abril de 2023). *Cómo las tecnologías de la Industria 4.0 están cambiando la fabricación*. Obtenido de IBM: <https://www.ibm.com/es-es/topics/industry-4-0>
- iControls. (20 de abril de 2023). *Ignitio Add on Modules. Even more functionality to Ignition's already powerful core features*. Obtenido de iControls:

<https://www.icontrols.com.au/products/ignition/add-on-modules/>

Jaidka, H., Sharma, N., & Singh, R. (2020). Evolution of IoT to IIoT: Applications & Challenges.

Proceedings of the International Conference on Innovative Computing & Communications (ICICC) 2020, 6.

Karmakar, A., Dey, N., Baral, T., Chowdhury, M., & Rehan, M. (2019). Industrial Internet of Things: A Review. *International Conference on Opto-Electronics and Applied Optics (Optronix)*, 6.

Luque, J. (20 de abril de 2023). PLC-HMI-SCADAS. Obtenido de <https://plc-hmi-scadas.com/111.php>

Moro, M. (2013). *Infraestructuras de redes de datos y sistemas de telefonía*. Ediciones Paraninfo.

Peláez, J. (2020). *Internet de las cosas (IOT) con ESP. Manual práctico*. Ediciones Paraninfo.

Rockwell Automation. (20 de abril de 2023). *Sistemas de control SoftLogix*. Obtenido de

<https://www.rockwellautomation.com/es-mx/products/hardware/allen-bradley/programmable-controllers/large-controllers/SoftLogix-controllers.html>

Sánchez, J. (2018). *Diseño de un control distribuido usando Factory IO y Codesys V3*. Universidad de Sevilla. Departamento de Ingeniería de Sistemas y Automática.

SAP. (20 de Abril de 2023). *¿Qué es la industria 4.0?* Obtenido de

<https://www.sap.com/latinamerica/insights/what-is-industry-4-0.html>

SAP Insights. (20 de abril de 2023). *¿Qué es un MES (sistema de ejecución de fabricación)?* Obtenido de

<https://www.sap.com/latinamerica/insights/what-is-mes-manufacturing-execution-system.html>

Sepasoft Inc. (20 de abril de 2023). *Settings & Changeover Module MES BUILT ON IGNITION*. Obtenido de

https://s3.amazonaws.com/files.sepasoft.com/website/datasheets/Settings+%26+Changeover_032323.pdf

Sunitha, G., Avanija, J., & Goundar, S. (2021). *Innovations in the Industrial Internet of Things (IIoT) and Smart Factory*. IGI Global.

Zurawski, R. (2018). *Integration Technologies for Industrial Automated Systems*. CRC Press.

