



POSGRADOS

MAESTRÍA EN

ELECTRÓNICA Y AUTOMATIZACIÓN

RPC-SO-19-No.277-2018

OPCIÓN DE
TITULACIÓN:

PROYECTOS DE DESARROLLO

TEMA:

PLATAFORMA DE MONITOREO Y CONTROL REMOTO
OPEN SOURCE, BASADA EN COMUNICACIONES
INALÁMBRICAS. APLICACIÓN A CELDAS DE
MANUFACTURA BAJO ENTORNOS VIRTUALIZADOS
UTILIZANDO LA HERRAMIENTA PLANT SIMULATION

AUTORES:

JOHNNY MAURICIO SIGUA SALINAS
JOSÉ FRANCISCO GUILLÉN GUILLÉN

DIRECTOR:

JULIO CÉSAR ZAMBRANO ABAD

CUENCA - ECUADOR
2021

Autores:



Johnny Mauricio Sigua Salinas

Ingeniero Electrónico

Candidato a Magíster en Electrónica y Automatización,
Mención en Informática Industrial por la Universidad
Politécnica Salesiana - Sede Cuenca.

jsigua@est.ups.edu.ec



José Francisco Guillén Guillén

Ingeniero Electrónico

Candidato a Magíster en Electrónica y Automatización,
Mención en Informática Industrial por la Universidad
Politécnica Salesiana - Sede Cuenca.

jguilleng2@est.ups.edu.ec

Dirigido por:



Julio César Zambrano Abad

Ingeniero Electrónico.

Magíster en Automatización y Control Industrial.

Doctor en Automática, Robótica e Informática Industrial.

jzambranoa@ups.edu.ec

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS

©2021 Universidad Politécnica Salesiana
CUENCA – ECUADOR – SUDAMÉRICA
JOSÉ FRANCISCO GUILLÉN GUILLÉN
JOHNNY MAURICIO SIGUA SALINAS

***PLATAFORMA DE MONITOREO Y CONTROL REMOTO OPEN SOURCE, BASADA EN
COMUNICACIONES INALÁMBRICAS. APLICACIÓN A CELDAS DE MANUFACTURA
BAJO ENTORNOS VIRTUALIZADOS UTILIZANDO LA HERRAMIENTA PLANT
SIMULATION***

Índice general

Índice de Figuras	IV
Índice de Tablas	VII
Resumen	VIII
Abstract	VIII
1. Introducción	1
1.1. Descripción general del problema.....	2
1.2. Objetivos	3
1.2.1. Objetivo general	3
1.2.2. Objetivos específicos	3
1.3. Contribuciones	3
1.4. Organización del manuscrito	4
2. Marco referencial	5
2.1. Estado del Arte.....	6
2.1.1. Capa Física	6
2.1.2. Capa Comunicación	8
2.1.3. Capa de Aplicación.....	9
2.2. Formulación del Problema	10
3. Metodología y Diseño	11
3.1. Metodología.....	12
3.2. Requerimientos	13
3.3. Diseño de la Capa Física	13
3.3.1. Planta de Extracción de Aceite.....	14
3.3.2. Comunicación Industrial Profinet.....	30
3.3.3. Planta Embotelladora de Aceite.....	31
3.4. Diseño de la Capa de Comunicación.....	48
3.4.1. Gemelo digital del PLC con PLCSIM Advance.....	48
3.4.2. Diseño de la red	49

3.4.3. Direccionamiento	52
3.4.4. Configuración de los elementos de red	52
3.5. Diseño de la Capa de Aplicación	58
3.5.1. Docker	58
3.5.2. Comunicación S7	58
3.5.3. Protocolo MQTT	59
3.5.4. Plataforma Thingsboard	60
3.5.5. Diseño de la aplicación	60
3.5.6. Puesta en marcha del servidor IoT	61
3.5.7. Lectura y envío de datos hacia el servidor IoT	63
3.5.8. Diseño del panel de Thingsboard	67
4. Análisis de Resultados, Conclusiones y Trabajos Futuros	79
4.1. Pruebas Realizadas	80
4.1.1. Rendimiento de la Capa de Red	80
4.1.2. Rendimiento Capa de Aplicación	82
4.2. Rendimiento de la Plataforma ThingsBoard	84
4.3. Trabajos Futuros	87
4.4. Conclusiones	88

Índice de Figuras

2.1. Capas de la arquitectura implementada.....	6
3.1. Capas de la arquitectura implementada.	12
3.2. Planta Extractora de Aceite en 2D.....	15
3.4. Contenedor de Coco.....	15
3.3. Planta Extractora de Aceite en 3D.....	16
3.5. Banda transportadora de producto.	17
3.6. Estación de perforación de producto.	17
3.7. Estación de seccionamiento de producto.	18
3.8. Horno	18
3.9. Estación extractora de aceite.	19
3.10. Tanque Industrial.....	20
3.11. Almacenador de aceite.	20
3.12. Almacenador de aceite.	21
3.13. Panel de control de la planta extractora de aceite.	21
3.14. Interfaz de comunicación de Planta con PLC.	22
3.15. Interfaz de PLSIM Advance en software Tecnomatrix Plant Simulation.....	22
3.16. Variables de ingreso al PLC.....	24
3.17. Variables de salida al PLC.....	25
3.18. Variables tipo marca utilizadas en el PLC.	26
3.19. Variables tipo Data Block utilizados en el PLC.....	27
3.20. Variables tipo Data Block utilizados en el PLC.....	28
3.21. Función para conversión de variable tipo TIME a entero.....	29
3.22. Conversión de variables tipo TIME a entero.....	29
3.23. Bloque de conversión numérico a String en el PLC.	30
3.24. Configuración de Red Industrial entre PLCs.	30
3.25. Envío y recepción de datos por de Red Industrial PROFINET.	31
3.26. Planta Extractora de Aceite en 2D.....	33

3.27. Planta Embotelladora de Aceite en 3D.....	34
3.28. Almacenador de la Planta embotelladora de aceite.....	35
3.29. Vaciador de aceite de Planta embotelladora.....	35
3.30. Tanque de Planta embotelladora de aceite.	36
3.31. Estación de llenado de aceite en botellas.....	36
3.32. Banda transportadora de producto.....	37
3.33. Colocación de tapa a las botellas.....	37
3.34. Colocación de etiqueta a las botellas.	38
3.35. Botella final con tapa y etiqueta.....	38
3.36. Brazo robot de marca KUKA.....	39
3.37. Empaquetadora de botellas.....	39
3.38. Panel de control de la planta embotelladora de aceite.	40
3.39. Variables de ingreso al PLC.....	41
3.40. Variables de salida al PLC.....	43
3.41. Variables tipo marca utilizadas en el PLC.....	44
3.42. Variables tipo Data Block utilizados en el PLC.....	45
3.43. Función para conversión de variable tipo TIME a entero.....	46
3.44. Conversión de variables tipo TIME a entero.....	46
3.45. Bloque de conversión numérico a String en el PLC.....	47
3.46. Interfaz de comunicación con PLCSIM Advance.....	48
3.47. Topología de la Red Diseñada.....	49
3.48. RouterBoard hAP AC Lite.....	50
3.49. Unifi NanoLoco Station M5.....	51
3.50. Configuración del Puente o Bridge.....	53
3.51. Direcciónamiento del Router.....	53
3.52. Configuración NAT.....	54
3.53. Configuración del Firewall.....	54
3.54. Configuración de la antena base.....	55
3.55. Configuración de la antena estación.....	56
3.56. Configuración del Adaptador de red de PLCSIM.....	57
3.57. Configuración del Adaptador de red de PLCSIM.....	57
3.58. Diagrama de despliegue de la aplicación.....	61
3.59. Login Thingsboard.....	63
3.60. Etapas del algoritmo de Python.....	63
3.61. Diagrama de flujo para la inicialización de datos.....	65
3.62. Diagrama de flujo comunicación S7.....	66
3.63. Diagrama de flujo comunicación MQTT.....	67
3.64. Librería utilizada para diseño de paneles.....	68
3.65. Paquete de Widgets de Thingsboard.....	69
3.66. Paneles de visualización en plataforma Thingsboard.....	70

3.67. Panel:Estado de las Plantas.....	71
3.68. Boton para editar panel.	71
3.69. Creación de Alias de entidad en plataforma.....	72
3.70. Panel:Extractor de Aceite.....	73
3.71. Agregación de variables a observar en Plataforma.	73
3.72. Generación de código token para visualización de variables.....	74
3.73. Código en Python para cargar variables de PLC a Thingsboard.	74
3.74. Panel: Embotelladora de Aceite.....	75
3.75. Panel: Históricos de planta extractora de aceite.....	76
3.76. Gráficos de históricos de planta extractora de aceite.	76
3.77. Panel: Históricos de planta embotelladora de aceite.....	77
3.78. Gráficos de históricos de planta embotelladora de aceite.	78
4.1. Prueba de velocidad radio enlace estación.	81
4.2. Ancho de banda consumido por la comunicación S7.	82
4.3. Respuesta del comando docker image ls.....	83
4.4. Respuesta del comando docker stats.	84
4.5. Históricos de planta extractora de aceite.....	85
4.6. Gráficos en pastel de planta extractora de aceite.....	86
4.7. Históricos de planta embotelladora.	86
4.8. Gráficos en pastel de planta embotelladora.....	87

Índice de Tablas

3.1. Elementos de Planta Extractora de Aceite	14
3.2. Dirección IP utilizada en Planta extractora de aceite.....	28
3.3. Conversión de valor numérico a String.	30
3.4. Elementos de Planta embotelladora de Aceite.	32
3.5. Dirección IP utilizada en Planta embotelladora de aceite.	45
3.6. Conversión de valor numérico a String.	47
3.7. Direccionamiento de red.	52
3.8. Librerías de Python y su función.....	64
3.9. Paneles de visualización en plataforma Thingsboard.	69
4.1. Librerías de Python y su función.....	82
4.2. Peso de los contenedores instalados.....	83
4.3. Recursos consumidos por los contenedores.	84

Resumen

Actualmente, es posible encontrar diversas herramientas informáticas que pueden ser aplicadas a la industria 4.0 para simplificar procesos de implementación en sistemas SCADA. El presente proyecto muestra el desarrollo de una plataforma de monitoreo y control remoto basada en código abierto y comunicaciones inalámbricas para la gestión de variables en procesos de manufactura. En el mismo contexto de la industria 4.0 estos procesos son virtualizados a través de la herramienta "Plan Simulation".

Para el desarrollo del proyecto se utiliza una arquitectura estructurada por tres capas: 1) capa física que se encarga del diseño y simulación de la planta con los PLCs; 2) capa de comunicación que contempla todos los dispositivos que permiten el envío y recepción de datos y 3) capa de aplicación en donde se encuentran todos los servicios que permiten el monitoreo y control de las variables a monitorizar.

Los resultados evidencian el desempeño de cada capa. Aquí se detalla los parámetros a tomar en cuenta para la implementación de este sistema con el fin de aportar información técnica de las herramientas utilizadas para la virtualización de procesos industriales, comunicaciones inalámbricas y plataformas IoT, las cuales pretenden ser de utilidad para profesionales en el campo de la automatización, dado que hasta el momento son tecnologías que se encuentran en desarrollo.

Abstract

Currently, it is possible to find various computer tools that can be applied to industry 4.0 to simplify implementation processes in SCADA systems. This project shows the development of a remote control and monitoring platform based on open source and wireless communications for the management of variables in manufacturing processes. In the same context of Industry 4.0, these processes will be virtualized through the Plant Simulation tool.

For the development of the project, an architecture structured by three layers is used: 1) physical layer that is in charge of the design and simulation of the plant with the PLCs; 2) communication layer that includes all the devices that allow the sending and reception of data and 3) application layer where all the services that allow the monitoring and control of the variables to be monitored are located.

The results show the performance of each layer. Here the parameters to be taken into account for the implementation of this system are detailed in order to provide technical information on the tools used for the virtualization of industrial processes, wireless communications and IoT platforms, which are intended to be useful for professionals in the field. automation, given that so far these are technologies that are under development.

Capítulo 1

Introducción

Debido al crecimiento en los sistemas de automatización industrial, desde hace algunos años se vienen implementando nuevos métodos de adquisición de datos y señales, que permiten el control y monitoreo de variables de interés. Bajo esta nueva concepción, estas variables resultan útiles para generar indicadores de la planta con el fin de obtener un mejor rendimiento. Actualmente existen varias herramientas para realizar estas tareas como servicios Web que permiten el acceso remoto para la interacción del usuario final con el proceso. Con el desarrollo de las tecnologías y la creación de diferentes tipos de herramientas informáticas, hoy en día se puede realizar la simulación o puesta en marcha virtual (*Virtual Commissioning*) de plantas o procesos. En concreto, la puesta en marcha virtual consiste en recrear mediante modelos, generalmente en tres dimensiones, una serie de procesos industriales, para emular el comportamiento real de la misma, permitiendo tener un ambiente de pruebas para desarrollos de proyectos que serán implementados en un entorno de producción real. En este trabajo se integran las tecnologías antes mencionadas, pretendiendo obtener, las variables de interés de un sistema simulado, para su control y monitoreo en un servicio Web, al cual pueden acceder usuarios finales para la interacción con esta información.

1.1. Descripción general del problema

En la actualidad se han logrado adelantos importantes en la recolección, análisis y distribución de información relacionada con procesos industriales. Esta información tiene distintos fines, entre ellos destaca el almacenamiento o creación de bases de datos para su representación digital y obtención de indicadores de gestión.

En la actualidad la mayoría de industrias se ven limitadas en cuanto innovación debido a que sus sistemas de automatización son rígidos. Probar nuevas formas de automatizar muchas veces resulta imposible ya que existe el riesgo de que la producción se vea afectada por un mal funcionamiento del sistema. La nueva era industrial, conocida como Industria 4.0 abre un abanico de posibilidades para abatir este tipo de problemas. Bajo esta nueva forma de ver a la industria es necesario contar con herramientas que potencien la innovación. Hoy en día, contar con un gemelo digital de un proceso resulta altamente necesario si se quiere transitar por el camino de la innovación. Este tipo de herramientas permiten crear modelos de un proceso para que sobre este modelo puedan ser probadas nuevas formas de automatizar en aras de optimizar los procesos.

Actualmente, en nuestro país este tipo de herramientas son desconocidas o poco utilizadas en la industria. Bajo esta concepción, el presente proyecto de titulación pretende introducir el uso de una herramienta para virtualizar un sistema de producción, permitiendo así elaborar un programa de automatización que pueda ser probado y analizado previo a la implementación.

Por otra parte, el presente proyecto aborda otro problema relacionado con la infraestructura de comunicación para la recolección, distribución y monitoreo de la información de un proceso. Esta situación puede resultar crítica debido a que muchos procesos de producción se encuentran en zonas remotas o alejados de los centros de datos donde se almacena la información. La implementación con medios convencionales como el cobre o la fibra óptica, no siempre es factible por lo que es necesario utilizar tecnologías inalámbricas [Carvajal et al., 2012], que permitan la comunicación entre las distintas infraestructuras que se desea monitorizar y controlar.

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar una plataforma de monitoreo y control remoto basada en código abierto y comunicaciones inalámbricas para la gestión de variables en procesos de manufactura virtualizados a través de la herramienta Plan Simulation.

1.2.2. Objetivos específicos

- Analizar el programa informático Plant Simulation mediante una exploración práctica de sus herramientas para la creación de las celdas de manufactura virtualizadas.
- Definir los requerimientos de diseño mediante el análisis de las variables de los procesos para sentar las bases para el desarrollo de la plataforma de monitoreo y el sistema de comunicación inalámbrica.
- Implementar el sistema comunicación inalámbrico y la plataforma de monitoreo, para administrar las variables de los procesos virtualizados.
- Evaluar la funcionalidad y las prestaciones de la plataforma a través de pruebas experimentales donde se pueda poner en marcha los procesos virtualizados y adquirir las variables de interés.

1.3. Contribuciones

El proyecto pretende generar una infraestructura que utilice tecnología *open source* para el control de variables en entornos industriales, para esto se toma en cuenta un entorno del cual se obtiene datos de distintos lugares que se encuentren separados del centro de datos, por lo que se hará uso de radio enlaces que permitan el envío de información a distancias considerables. Con el presente proyecto se aportará información técnica importante como los rangos de distancia en que se puede usar un radio enlace, la velocidad de transmisión, ancho de banda, tiempos de respuesta y los problemas que presente al momento de implementar la comunicación inalámbrica, información que será de utilidad para profesionales en el campo de la automatización que requieran comunicar equipos que se encuentren distantes para el control y monitoreo de datos de diferentes procesos industriales, dado que hasta el momento es una alternativa poco conocida. También se

realizará un breve análisis de las plataformas *open source*, debido a que nos son muy conocidas, pero aportan un gran beneficio frente a las plataformas de suscripción o propietarias existentes en el mercado.

Otra de las contribuciones de este trabajo radica en el análisis de la herramienta *plant simulation* de siemens y el desarrollo de un entorno de producción virtualizado, lo cual puede resultar atractivo para las industrias, dado que un modelo virtual de un proceso permite desarrollar algunas acciones importantes, entre las que destaca la prueba de nuevas formas de automatizar un proceso sin la necesidad de generar una para o un posible daño en el sistema real.

1.4. Organización del manuscrito

De aquí en adelante este manuscrito se encuentra organizado de la siguiente manera: en el capítulo 2 se presenta un marco referencial del trabajo. A continuación, en el capítulo 3 se presenta la metodología y el diseño del proyecto. Finalmente, en el capítulo 4 se presentan los resultados y se reportan algunas conclusiones y trabajos futuros.

Capítulo 2

Marco referencial

En el presente capítulo se desarrolla un análisis del estado del arte tomando como ejes principales las herramientas informáticas necesarias para la implementación del modelo de producción para la puesta en marcha virtual, así como los equipos y herramientas necesarios para la implementación del sistema de monitoreo remoto. Adicionalmente se desarrollará de una manera más amplia la formulación del problema.

2.1. Estado del Arte

El presente trabajo de investigación se desarrolla sobre la base de una arquitectura integrada por tres capas (véase la Figura 2.1):

- Capa física: Virtualización de una planta de producción y su respectiva automatización.
- Capa de comunicación: Infraestructura inalámbrica para la adquisición de variables de la planta.
- Capa de aplicación: Sistema de monitoreo remoto basado en servicios web.

Con base en estas tres capas, a continuación se presenta una análisis del estado del arte:



Figura 2.1: Capas de la arquitectura implementada.

2.1.1. Capa Física

Dentro la capa física se contempla todos los elementos necesarios para la adquisición de los datos y variables a medir, como son sensores y el elemento que procesa esa información. Todos estos procesos se pueden simular con modelos virtuales que reflejen de manera precisa la operación de una planta.

Estos modelos también serán de utilidad para validar la programación de los PLCs. [Villacañas Sánchez, 2019]

En el contexto de la industria 4.0, hoy en día se cuenta con diversos programas informáticos que permiten simular plantas y procesos. Entre estas herramientas destacan:

- **Tecnomatrix Plant Simulation:** Es un software perteneciente a Siemens que permite realizar análisis de eficiencia de las plantas [Villacañas Sánchez, 2019], simular procesos virtuales e interactuar con equipos físicos como PLCs para determinar la eficiencia de la implementación de los mismos. Es una herramienta muy utilizada en procesos actuales [Sobrino et al., 2019].
- **Process Simulate:** El cual es un software que permite la simulación de procesos de ensamblaje de productos y permite la iteración con brazos robóticos, en los trabajos [Novak et al., 2017] [Guerra-Zubiaga et al., 2019] se presenta el uso de Process Simulate para la simulación de gemelos digitales con un enfoque en virtual commissioning para la simulación de brazos robóticos en procesos de ensamblaje.
- **Siemens NX:** En este software se puede diseñar muchos procesos, existen muchos trabajos desarrollados mediante este software como el modelamiento de un proceso de perforación geométrico realizado por [Osinde et al., 2019]. Es una herramienta de gran utilidad en la industria, muy empleado en procesos virtuales [Guerrero et al., 2014].
- **Factory IO:** Es un software utilizado en el campo educativo como entrenador de PLC puesto que permite su integración con diversos dispositivos físicos, en el trabajo presentado en [Satorres Mira, 2021] se implementa un SCADA simple para la automatización de un almacén de mercancías utilizando un PLC simulado en Codesys y FACTORY IO para la generación de datos.
- **Unity:** Este software es perteneciente a la compañía desarrolladora de videojuegos conocida como Unity Technologies, utilizada para la creación de animaciones y diseños en 3d [Canossa, 2013], actualmente se está utilizando en la industria para el desarrollo de procesos virtuales [Ospina Eslava and Santiago, 2021].
- **PSIMIT:** Es un programa de SIEMENS que permite la simulación de procesos se utiliza para desarrollar proyectos de virtual commissioning y también para entrenar a operadores en el uso de maquinaria dado

que posee módulos de educación, en el proyecto desarrollado por [Cortés-López, 2017] se puede observar la implementación de este programa en el campo de la educación.

En el trabajo presentado por [Villacañas Sánchez, 2019] se realiza la integración de PLCs con modelos virtuales de plantas industriales, para ello se utiliza un PLC virtual s7 1500 simulado en PLCSim Advance 3.0, aplicado en una línea de producción alimentaria de una fábrica de productos enlatados. El proceso que se ejecuta es el esterilizado y empaquetado del producto. El mismo que es simulado en Plant Simulation.

Otro programa utilizado quizá con un enfoque educativo es Factory IO, el cual es utilizado como un entrenador de PLCs, ya que permite su integración con diversos dispositivos físicos. Un trabajo previo que se puede mencionar es el presentado por [Abdulwahid and Wasel, 2020], donde se implementa un servicio de monitoreo con ThingsBoard para el control de nivel de un reservorio de agua utilizando un PLC s7 400 de Siemens. El proceso a controlar es simulado en Factory IO.

2.1.2. Capa Comunicación

Contempla todos los dispositivos que permiten el envío y recepción de datos entre la capa física y la capa de aplicación, a continuación se destacan algunas tecnologías que resultan de utilidad para realizar esta tarea:

El Simatic IoT 2040 es un equipo industrial diseñado para la adquisición, procesamiento e intercambio de datos para entornos de producción. Dado que se utiliza como una compuerta o gateway entre el servidor ubicado en la nube y el proceso a monitorizar o controlar, es muy útil porque admite numerosos protocolos de comunicación industrial. [Siemens A.G., 2021b]

En el trabajo desarrollado por [Sarmiento Paute, 2020], se utiliza el Simatic IoT 2040 como un gateway para enviar variables de un motor. Aquí se puede destacar la implementación de una red Modbus TCP para la adquisición de datos de un Sentron PAC además se utiliza un protocolo de comunicación s7 entre NODERED y el PLC para el envío de datos a la nube de Ubidots.

Otra alternativa interesante radica en el uso de antenas de radio enlace, que se indican en el trabajo desarrollado por [Dueñas Llanos and Chalacán Aguayo, 2017] en donde se comunica dos plantas didácticas utilizando un radio enlace conformado por dos antenas UBIQUITI NANOSTATIONS 5 a un SCADA simple desarrollado en LabVIEW, así también esta tecnología es aplicada en proyectos con mayor complejidad

como el trabajo presentado por [Smerpitak et al., 2018] en el que se realiza un control cooperativo de tres bandas y dos vehículos guiados el control lo realizan dos PLCs maestros y dos PLCs esclavos la interconexión de estos equipos lo realiza un puente inalámbrico conformado por dos antenas Ubiquiti Rocket M5 y Nano M5. Finalmente en el trabajo [Domínguez Cartolín, 2018] se presenta el control de nivel, de caudal en reservorios usando un controlador basado en lógica difusa, estos reservorios se encuentran dispersos por varios puntos de la ciudad y se interconectan mediante una red de radio enlace hacia una central que monitoree y controle en tiempo real el proceso.

2.1.3. Capa de Aplicación

Es la capa que facilita la interacción entre el humano y la máquina, aquí se encuentran todos los servicios que permiten el monitoreo y control de las variables de los procesos industriales. Existen plataformas en la nube que agilizan este proceso, sin embargo, tienden a ser de pago por ejemplo la plataforma de Amazon Web Server, el usuario tiene que pagar de acuerdo a la frecuencia de uso.

Una alternativa para optimizar los costos radica en el uso de tecnologías open source. En el trabajo desarrollado por [Aghenta and Iqbal, 2019] se presenta un SCADA de bajo costo utilizando la plataforma ThingsBoard y protocolo MQTT para el monitoreo de variables utilizadas en un proceso de energías renovables, a través del uso de paneles solares. De este trabajo se puede destacar el uso de la tarjeta ESP32 y una Raspberry Pi, donde se ejecuta el servidor que le convierte en una aplicación de bajo costo, óptima para pequeños y medianos procesos desarrollados en la industria.

Por otra parte, en el trabajo desarrollado por [Chalán Padilla, 2020], se presenta otra plataforma open source, EMQ, que funciona con el protocolo MQTT para el monitoreo de un sistema de tanques utilizando un PLC s7 1200.

Las plataformas previamente mencionadas como son ThingsBoard y EMQ ya se encuentran desarrolladas, siendo una alternativa a plataformas de suscripción como Ubidot, MindSphere, con la diferencia que son open source y las mismas se pueden instalar en un servidor para configurarse de acuerdo a las necesidades de la aplicación.

Como se puede apreciar en los trabajos previamente descritos se puede implementar servicios web open source que permitan el monitoreo y control de equipos industriales como son los PLC. Existe también la tecnología inalámbrica que permite la obtención de estos datos, por lo que en los puntos

posteriores se plantea un proyecto que combine las tecnologías antes descritas en las distintas capas y que sea de un costo accesible para la pequeña y mediana empresa.

2.2. Formulación del Problema

Existen tecnologías desarrolladas que permiten el control y monitoreo de dispositivos desde la nube, sin embargo, estas tecnologías son costosas, lo cual dificulta implementarlas en pequeños proyectos. Por esta razón se propone el presente proyecto, el cual tiene como finalidad entregar una alternativa al uso de estas tecnologías utilizando plataformas open source, las cuales son una opción a los servicios de suscripción que se encuentran en internet, cuya ventaja es que se pueden personalizar y no se está atado a términos y condiciones del proveedor.

Así mismo existen problemas para la obtención de datos en procesos industriales, sobre todo en los que se encuentran en zonas remotas o de difícil acceso, que no cuentan con el suficiente presupuesto para armar una infraestructura cableada, por lo que es necesario utilizar equipos con comunicación inalámbrica, ya que se ha visto que algunos de ellos alcanzan distancias teóricas de 10km, reduciendo los costos de implementación y aportando un desarrollo más simple para control y monitoreo de estos procesos.

El proyecto pretende dar solución a la problemática descrita en los párrafos anteriores, la cual se centra en la capa de aplicación y la capa de comunicación, por tal razón se hará uso de virtual commissioning siendo una tecnología que ayuda a simular plantas de procesos industriales, de la cual se obtendrá los indicadores de la planta que se desea monitorizar como pueden ser: el tiempo de producción, la cantidad de producción y variables de consumo eléctrico.

Capítulo 3

Metodología y Diseño

En este capítulo se presenta la metodología que se implementó para el desarrollo del trabajo. Se explica el desarrollo que se realizó en cada etapa del proyecto y el diseño y configuración de cada capa implementada.

3.1. Metodología

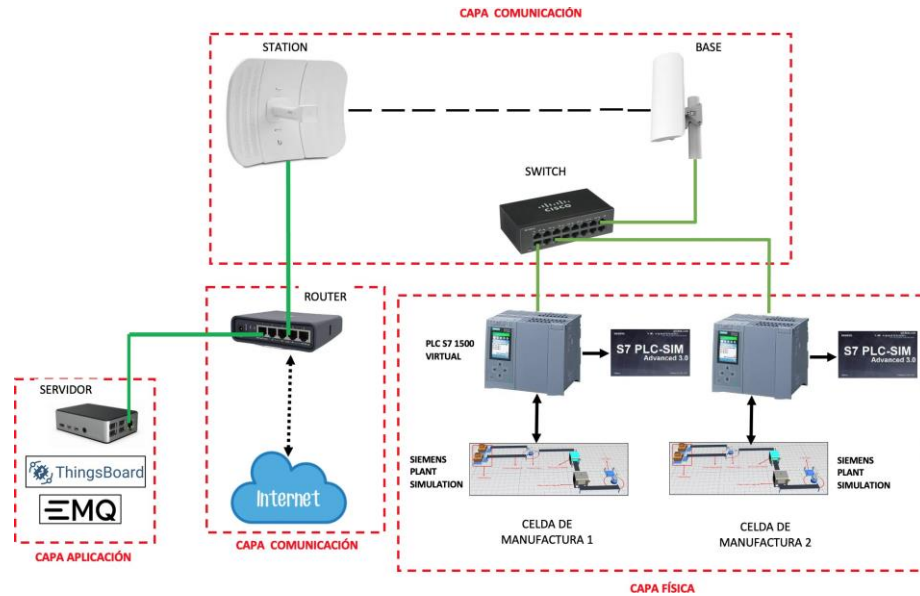


Figura 3.1: Capas de la arquitectura implementada.

La metodología propuesta para el presente trabajo se basa en la arquitectura presentada en el capítulo anterior. Esta arquitectura se ilustra de manera detallada en la figura 3.1. Para el desarrollo de este proyecto se inició con la implementación de la capa inferior o capa física. A continuación, se implementó la capa de comunicación y finalmente se diseñó e implementó la capa de aplicación.

Para la implementación de la capa física, la cual va a ser emulada por un proceso virtualizado se debe instalar el software del gemelo digital, familiarizarse con el mismo y diseñar la planta a monitorizar. Se debe tomar en cuenta los elementos electrónicos que pueden ser simulados por el software tales como: controladores, sensores y actuadores.

Posterior a simular la planta se implementa la capa de comunicación, lo que conlleva el diseño de la topología, el direccionamiento, configuración y conexión de los distintos dispositivos de red.

Al establecer una comunicación funcional entre los dispositivos y la planta, se implementa la capa de aplicación a través de algoritmos para el envío de los datos a monitorizar y la puesta en marcha del servidor IoT. Finalmente se realiza la interfaz visual HMI para la visualización de los datos pertenecientes a la planta. Terminando con pruebas de funcionamiento, donde se verifica el desempeño general de todo el sistema de monitoreo.

3.2. Requerimientos

Como punto de inicio se listan los requerimientos del sistema a desarrollar con el fin de buscar las herramientas que permitan la puesta en marcha del mismo:

1. El sistema debe contar con dos plantas simuladas en un software que permita el diseño y análisis de gemelos digitales. Para esta implementación se utilizó el programa informático Tecnomatrik Plant Simulation de Siemens.
2. Las plantas deben ejecutarse en dos equipos distintos y las mismas deben estar comunicadas a través de cualquier protocolo de comunicación industrial.
3. Los datos a analizar deben subirse a un servidor IoT, los cuales deben poder visualizarse desde cualquier cliente que se conecte al servidor.
4. Se debe realizar un radio enlace con la capacidad suficiente para enviar los datos de manera remota al servidor.

3.3. Diseño de la Capa Física

El desarrollo de esta capa abarca el diseño y simulación de la planta industrial, la comunicación entre los dos PLC que controlan cada una de las plantas y el envío de datos a las antenas mediante los PLCs.

En el presente proyecto se ha hecho uso del virtual commissioning [[Mortensen and Madsen, 2018](#)] el cual consiste en la emulación completa de una planta de manera virtual, para esto se ha empleado el software Tecnomatrix Plant Simulation [[Bangsow, 2020](#)] que es una herramienta de simulación que permite crear modelos digitales de sistemas productivos y de procesos, sistemas logísticos y flujo de materiales, que permite la exploración

de las características de los sistemas y la optimización de su rendimiento.

Se diseñó dos plantas industriales, la primera se encarga de realizar la extracción de aceite de coco. La segunda planta se encarga del embotellamiento y empaque del producto. Ambas plantas se simulan en el software Tecnomatrix Plant Simulation v16.0, el cual contiene varias herramientas que permiten integrar las funciones que posee una planta real con un PLC. Una de las consideraciones que se tomó en cuenta para ejecutar el presente trabajo fue el utilizar dos PLCs s7 1500 de la marca Siemens, los cuales a través del software PLSIM Advance 3.0 fueron simulados haciendo posible el funcionamiento de cada una de las plantas anteriormente mencionadas. A continuación se detalla cada una de ellas.

3.3.1. Planta de Extracción de Aceite

Esta planta consiste en una celda de manufactura para la extracción de aceite de coco, cuenta con una serie de procesos que son útiles para la obtención de datos de producción. La planta a simular está estructurada por los elementos que se pueden apreciar en la tabla 3.1.

Estación	Descripción
Contenedor de Producto	En esta estación se encuentra la materia prima con la que se va a trabajar que es el coco.
Perforación	Aquí utilizando un taladro se procede a perforar la materia prima con varios orificios.
Seccionamiento	En esta estación se realiza varios cortes al producto para ser procesado.
Horno	Se eleva a cierta temperatura al producto previamente seccionado.
Extractor	Se extrae el aceite el cual pasa posteriormente a los tanques de almacenamiento.
Taque 1	Almacena el aceite
Taque 2	Almacena el aceite
Taque 3	Almacena el aceite
Almacenador 1 y 2	Aquí se guarda todo el aceite que se extraiga de la planta que proviene de los 3 tanques.

Tabla 3.1: Elementos de Planta Extractora de Aceite

En la figura 3.2 se puede observar el diseño 2D de la planta extractora de

aceite. Para tener una mejor apreciación de la planta se puede observar un diseño en tres dimensiones en la figura 3.3.

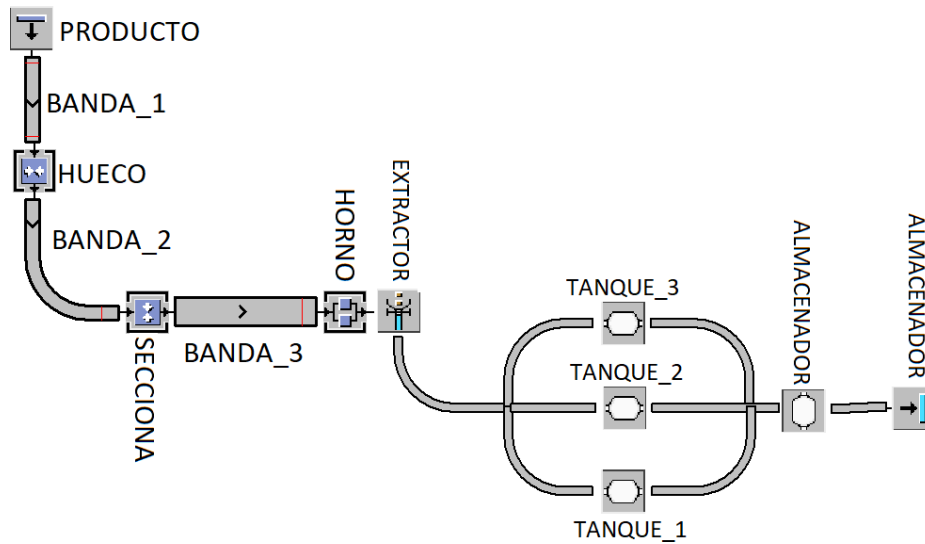


Figura 3.2: Planta Extractora de Aceite en 2D.

Funcionamiento de la Planta

Al ingreso de la planta se encuentra el contenedor de producto que se puede observar en la figura 3.4. En este contenedor se encuentra la materia prima, que en este caso sería el coco.

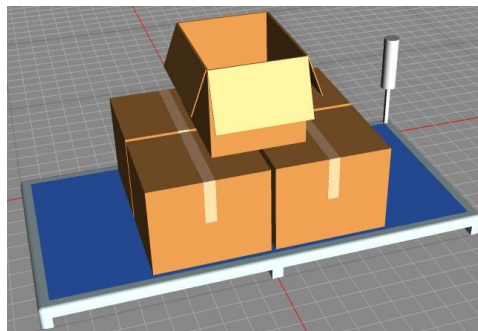


Figura 3.4: Contenedor de Coco.

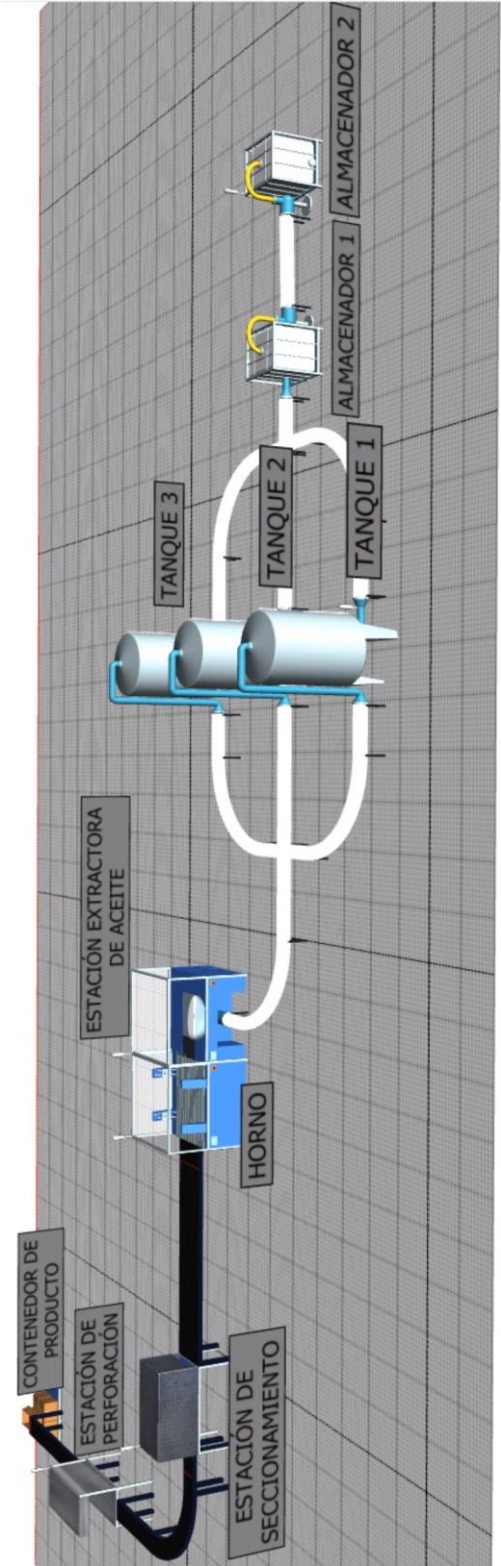


Figura 3.3: Planta Extractora de Aceite en 3D.

El producto pasa a través de una banda transportadora que se observa en la figura 3.32, la misma que contiene sensores de ingreso de producto y salida del mismo.

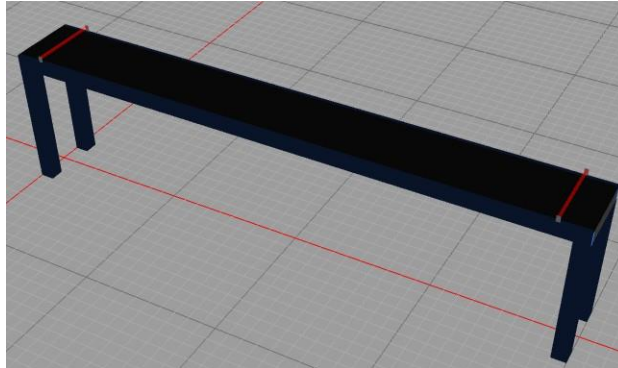


Figura 3.5: Banda transportadora de producto.

Luego el producto pasa a ser perforado en la estación que se observa en la figura 3.6.

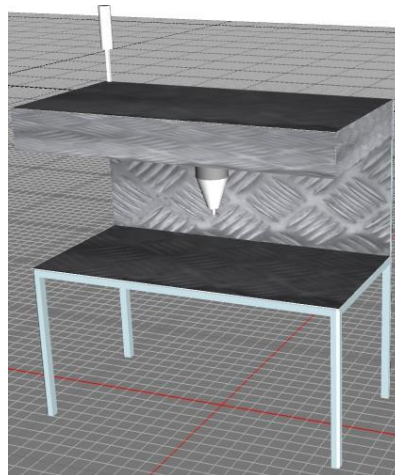


Figura 3.6: Estación de perforación de producto.

Por medio de una banda transportadora los cocos perforados pasan a ser seccionados en la estación que se observa en la figura 3.7.

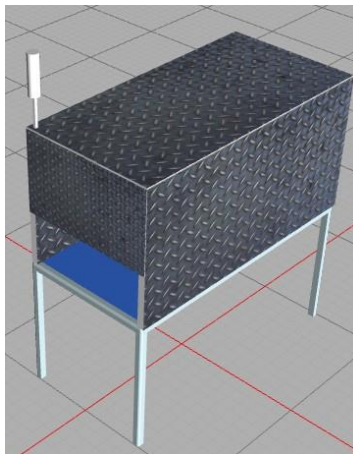


Figura 3.7: Estación de seccionamiento de producto.

Una vez ya seccionado el producto ingresa a la estación que se observa en la figura 3.8 que se encarga de elevar el producto a la temperatura necesaria para que se pueda obtener el aceite.

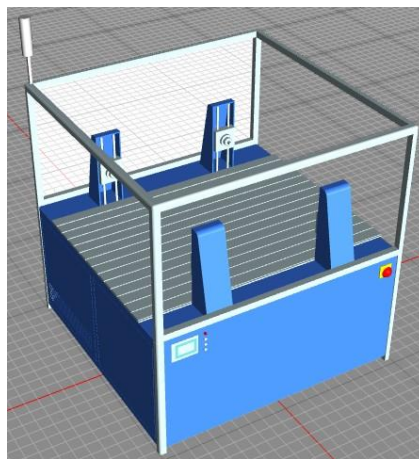


Figura 3.8: Horno

En la estación que se observa en la figura 3.9 se realiza el envío del aceite obtenido a las tuberías.

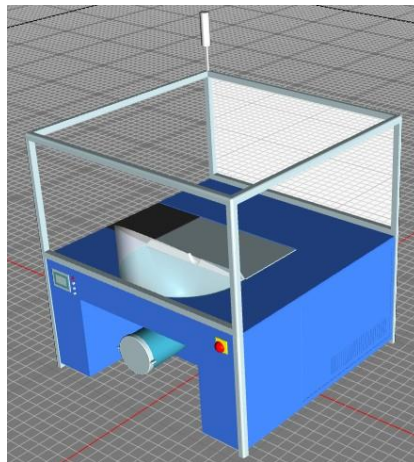


Figura 3.9: Estación extractora de aceite.

Las tuberías conducirán el líquido a los tres tanques que se observan en la figura 3.30. Estos poseen una capacidad de 30 litros cada uno. Mediante los PLCs se programa una secuencia de tal modo que al momento de llenarse el primer tanque, el ingreso del líquido al mismo se bloquea, permitiendo que el segundo tanque de la planta pase a llenarse, una vez llenado el segundo tanque se bloquea el ingreso de líquido al mismo, permitiendo que el tercer tanque pase a llenarse.

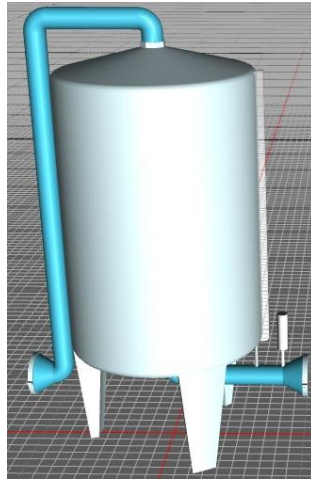


Figura 3.10: Tanque Industrial.

Cada vez que se llena cualquiera de los tanques, estos pasan a viciarse en el almacenador que se observa en la figura 3.11.

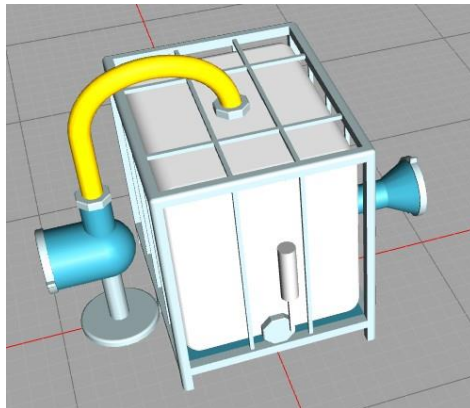


Figura 3.11: Almacenador de aceite.

Posterior a esto pasan a un segundo almacenador que se observa en la figura 3.12, del cual parte la segunda planta para poder empezar con el proceso de embotellamiento, etiquetado y empaquetado de las botellas.

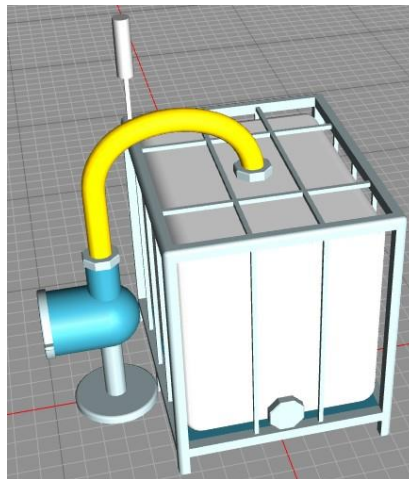


Figura 3.12: Almacenador de aceite.

Todo el proceso anteriormente descrito se lo puede controlar con el panel de control que se puede observar en la figura 3.13, que se encarga de encender y parar la planta, subir los datos de funcionamiento de cada estación y ver los estados de las válvulas de los tanques.

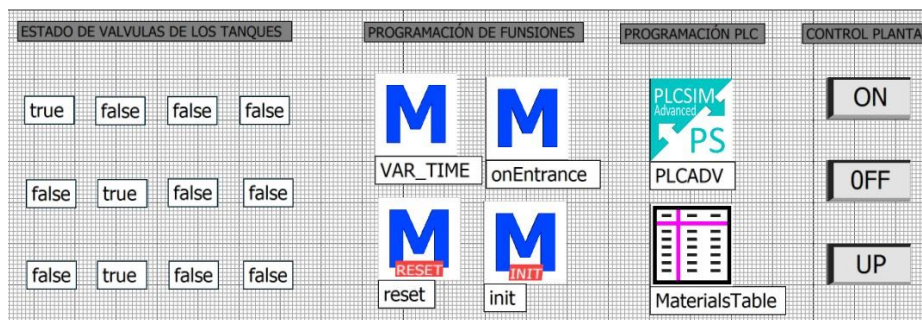


Figura 3.13: Panel de control de la planta extractora de aceite.

Comunicación de la Planta Extractora de aceite con el PLC

Para establecer la comunicación entre la Planta virtual y el PLC se utiliza la interfaz de comunicación que se observa en la figura 3.14. Este interfaz se encuentra en el software Plant Simulation.



Figura 3.14: Interfaz de comunicación de Planta con PLC.

Este módulo hay que cargarlo en el Layout que es la plantilla donde se va a colocar todos los elementos de la planta. La interfaz de configuración de este módulo se explica en la figura 3.15, mientras que cada campo es detallado a continuación:

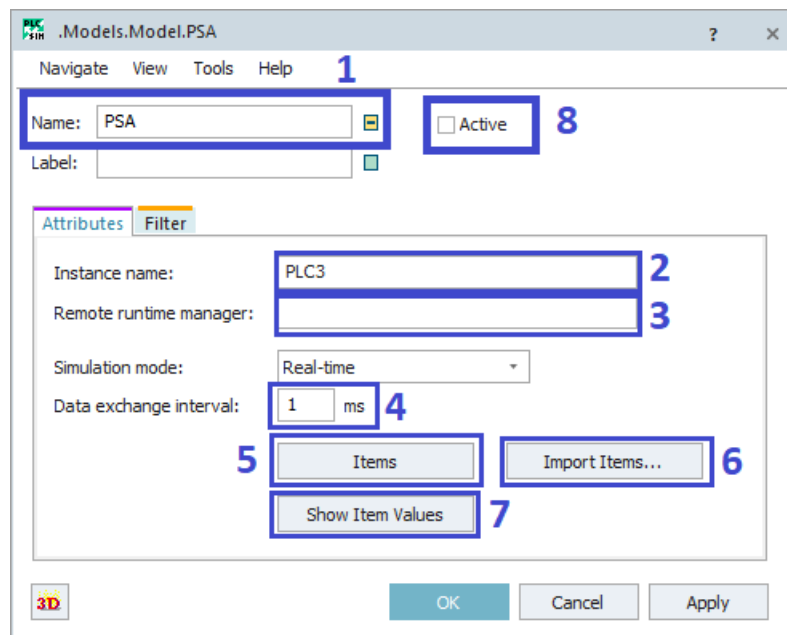


Figura 3.15: Interfaz de PLSIM Advance en software Tecnomatrix Plant Simulation.

1. **Name:** Es el nombre que se le asignará al módulo a trabajar dentro del Layout.
2. **Instance name:** En este apartado se coloca el mismo nombre que se puso al PLC simulado en el software PLCSIM Advance.

3. **Remote runtime manager:** Este campo se llena siempre y cuando el PLC virtual se encuentre en otra máquina distinta en donde se está corriendo la planta virtual. Basta con colocar la dirección IP del otro equipo.
4. **Data exchange interval:** En este apartado se coloca el tiempo que se recibe los datos enviados por el PLC a la planta virtual.
5. **Items:** En este campo se relaciona las señales utilizadas por el PLC con la variable que se esté usando en la Planta Virtual.
6. **Import items:** Cada vez que se grabe el programa desarrollado en TIA Portal al PLC virtual simulado en PLCSIM Advance se debe presionar este botón, al momento de hacerlo importará todas las variables usadas por el PLC para asociarlas a las variables de la planta.

En la figura 3.16 se puede observar las variables de ingreso al PLC que salen de la planta simulada en el software Tecnomatrix Plant Simulation. En la primera columna con el nombre de “Identifier” se tiene todas las variables que provienen del PLC, en la segunda columna con el nombre de “Data Type” se observa el tipo que tiene cada variable, siendo estas booleanas, enteras, reales, etc. La tercera columna cuyo nombre es “Alias” es una asignación que el usuario puede hacer a cada variable para poder identificarla rápidamente. Finalmente en la última columna con el nombre de “Simulation Model Attribute” se tienen las variables que van a ser controladas con el PLC en el software, es decir estas variables provienen de la planta simulada.

	Identifier	Data Type	Alias	Simulation Model Attribute
1	SenB1_IN	Boolean	SenB1_IN	
2	SenB1_OUT	Boolean	SenB1_out	
3	SenB2_OUT	Boolean	SenB2_OUT	
4	HUECO_OBJETO	Boolean	HUECO_OBJETO	
5	SenB1_IN	Boolean	SenB1_IN	BANDA_1.SensorID(1).isTriggered
6	SenB1_OUT	Boolean	SenB1_out	BANDA_1.SensorID(2).isTriggered
7	SenB2_OUT	Boolean	SenB2_OUT	BANDA_2.SensorID(2).isTriggered
8	HUECO_OBJETO	Boolean	HUECO_OBJETO	HUECO.Occupied
9	SECCIONA_OBJETO	Boolean	SECCIONA_OBJETO	SECCIONA.Occupied
10	HORNO_OBJETO	Boolean	HORNO_OBJETO	HORNO.Occupied
11	SenB3_OUT	Boolean	SenB3_OUT	BANDA_3.SensorID(1).isTriggered
12	PARTESHORNO	Int16	PARTESHORNO	HORNO.NumMU
13	NIVEL_TANQUE_1	Real32	NIVEL_TANQUE_1	TANQUE_1.CurrentAmount
14	NIVEL_TANQUE_2	Real32	NIVEL_TANQUE_2	TANQUE_2.CurrentAmount
15	NIVEL_TANQUE_3	Real32	NIVEL_TANQUE_3	TANQUE_3.CurrentAmount
16	VACIO_TANQUE_1	Boolean	VACIO_TANQUE_1	TANQUE_1.Empty
17	VACIO_TANQUE_2	Boolean	VACIO_TANQUE_2	TANQUE_2.Empty
18	VACIO_TANQUE_3	Boolean	VACIO_TANQUE_3	TANQUE_3.Empty
19	EXTRACTOR_OBJETO	Boolean	EXTRACTOR_OBJETO	EXTRACTOR.Occupied
20	FULL_TANQUE_1	Boolean	FULL_TANQUE_1	TANQUE_1.Full
21	FULL_TANQUE_2	Boolean	FULL_TANQUE_2	TANQUE_2.Full
22	FULL_TANQUE_3	Boolean	FULL_TANQUE_3	TANQUE_3.Full
23	PARO_GENERAL	Boolean	PARO_GENERAL	btn_2.APAGADO
24	DESBLOQUEA_CAJA	Boolean	DESBLOQUEA_CAJA	btn_1.START

Figura 3.16: Variables de ingreso al PLC.

A continuación se hace una lista las variables de ingreso al PLC, que se obtienen de cada estación.

1. Las variables con el nombre **BANDA_1**, **BANDA_2**, **BANDA_3**, corresponden a los sensores de ingreso y salida que han sido colocados en cada una de las bandas transportadoras.
2. Las variables con la extensión “variable.Occupied”, se han utilizado para los sensores que detectan el ingreso del producto a cada una de las diferentes estaciones: **HUECO.Occupied**, **SECCIONA.Occupied**, **HORNO.Occupied** y **EXTRACTOR.Occupied**.
3. Para obtener el nivel de cada uno de los tanques se utiliza la extensión “variable.CurrentAmount”, la cual es de tipo real, para este proyecto se obtuvo el valor del nivel de los tres tanques. Las variables usadas fueron:

TANQUE_1.CurrentAmount, TANQUE_2.CurrentAmount y TANQUE_3.CurrentAmount.

4. Para obtener el valor del estado del tanque se debe colocar la extensión “variable.Empty” que se activará si el tanque se encuentra vacío. La extensión “variable.Full” será activa mientras el tanque se encuentre lleno, estas extensiones se aplicaron para obtener el estado de los tres tanques utilizados en esta planta.
5. Para el control general de toda la planta en el encendido se utilizó la variable **btn_1.START**, mientras que para el apagado se utilizó la variable **btn_2.APAGADO**.

En la figura 3.17 se puede observar las variables de salida hacia el PLC. A continuación se explica cada una de estas:

	Identifier	Data Type	Alias	Simulation Model Attribute
1	NUMERO_COCOS	Int16	NUMERO_COCOS	
2	PAUSA_B1	Boolean	PAUSA_B1	
3	NUMERO_COCOS	Int16	NUMERO_COCOS	
4	PAUSA_B1	Boolean	PAUSA_B1	BANDA_1.Pause
5	PAUSA_B2	Boolean	PAUSA_B2	BANDA_2.Pause
6	PAUSA_B3	Boolean	PAUSA_B3	BANDA_3.Pause
7	BLOQUEO_TANQUE_1_OUT	Boolean	BLOQUEO_TANQUE_...	TANQUE_1.ExitLocked
8	BLOQUEO_TANQUE_2_OUT	Boolean	BLOQUEO_TANQUE_...	TANQUE_2.ExitLocked
9	BLOQUEO_TANQUE_3_OUT	Boolean	BLOQUEO_TANQUE_...	TANQUE_3.ExitLocked
10	BLOQUEO_TANQUE_1_IN	Boolean	BLOQUEO_TANQUE_...	TANQUE_1.EntranceLocked
11	BLOQUEO_TANQUE_2_IN	Boolean	BLOQUEO_TANQUE_...	TANQUE_2.EntranceLocked
12	BLOQUEO_TANQUE_3_IN	Boolean	BLOQUEO_TANQUE_...	TANQUE_3.EntranceLocked
13	LLENO_TANQUE_1	Boolean	LLENO_TANQUE_1	
14	LLENO_TANQUE_2	Boolean	LLENO_TANQUE_2	
15	LLENO_TANQUE_3	Boolean	LLENO_TANQUE_3	
16	INICIO_GENERAL	Boolean	INICIO_GENERAL	PRODUCTO.ExitLocked

Figura 3.17: Variables de salida al PLC.

1. Para frenar cada una de las bandas utilizadas en la planta virtual mediante el PLC, se utilizó la extensión “variable.Pause”, las variables utilizadas fueron: **BANDA_1.Pause, BANDA_2.Pause y BANDA_3.Pause.**
2. Para hacer un control de las válvulas de ingreso y salida del líquido a cada tanque se utilizó las extensiones: “variable.EntranceLocked” y

“variable.ExitLocked”, respectivamente. En esta planta se aplicó este control para cada uno de los tres tanques.

Las variables que son utilizadas como marcas en el PLC se pueden observar en la figura 3.18. A continuación se explica la utilidad de cada una de estas:

	Identifier	Data Type	Alias	Simulation Model Attribute
1	TIEMPO_SEGUNDOS	Int32	TIEMPO_SEGUNDOS	
2	TIEMPO_MILISEGUNDOS	Int32	TIEMPO_MILISEGUN...	
3	TIEMPO_FORMATO	Int32	TIEMPO_FORMATO	
4	HUECO_TIEMPO_TRABAJADO	Int16	HUECO_TIEMPO_TR...	HUECO.tiempoTrabajado
5	HUECO_TIEMPO_ESPERA	Int16	HUECO_TIEMPO_ES...	HUECO.tiempoEspera
6	HUECO_TIEMPO_BLOQUEA...	Int16	HUECO_TIEMPO_BL...	HUECO.tiempoBloqueado
7	HUECO_TIEMPO_FALLA	Int16	HUECO_TIEMPO_FA...	HUECO.tiempoFalla
8	SECCION_TIEMPO_TRABAJA...	Int16	SECCION_TIEMPO_T...	SECCIONA.tiempoTrabajado
9	SECCION_TIEMPO_ESPERA	Int16	SECCION_TIEMPO_E...	SECCIONA.tiempoEspera
10	SECCION_TIEMPO_BLOQUE...	Int16	SECCION_TIEMPO_B...	SECCIONA.tiempoBloqueado
11	SECCION_TIEMPO_FALLA	Int16	SECCION_TIEMPO_F...	SECCIONA.tiempoFalla
12	HORNO_TIEMPO_TRABAJADO	Int16	HORNO_TIEMPO_TR...	HORNO.tiempoTrabajado
13	HORNO_TIEMPO_ESPERA	Int16	HORNO_TIEMPO_ES...	HORNO.tiempoEspera
14	HORNO_TIEMPO_BLOQUEA...	Int16	HORNO_TIEMPO_BL...	HORNO.tiempoBloqueado
15	HORNO_TIEMPO_FALLA	Int16	HORNO_TIEMPO_FA...	HORNO.tiempoFalla
16	EXTRACTOR_TIEMPO_TRABA...	Int16	EXTRACTOR_TIEMP...	EXTRACTOR.tiempoTrabajado
17	EXTRACTOR_TIEMPO_ESPERA	Int16	EXTRACTOR_TIEMP...	EXTRACTOR.tiempoEspera
18	EXTRACTOR_TIEMPO_BLOQ...	Int16	EXTRACTOR_TIEMP...	EXTRACTOR.tiempoBloqueado
19	EXTRACTOR_TIEMPO_FALLA	Int16	EXTRACTOR_TIEMP...	EXTRACTOR.tiempoFalla
20	ESTADO_HUECO	Int16	ESTADO_HUECO	HUECO.ESTADO
21	ESTADO_SECCIONADOR	Int16	ESTADO_SECCIONA...	SECCIONA.ESTADO
22	ESTADO_HORNO	Int16	ESTADO_HORNO	HORNO.ESTADO
23	ESTADO_EXTRACTOR	Int16	ESTADO_EXTRACTOR	EXTRACTOR.ESTADO
24	ENERGIA_EXTRACTOR	Real32	ENERGIA_EXTRACTOR	EXTRACTOR.Energia
25	ENERGIA_HORNO	Real32	ENERGIA_HORNO	HORNO.StatEnergyTotalConsump...
26	ENERGIA_HUECO	Real32	ENERGIA_HUECO	HUECO.StatEnergyTotalConsumpt...
27	ENERGIA_SECCION	Real32	ENERGIA_SECCION	SECCIONA.StatEnergyTotalConsu...
28	NIVEL_ALMACENADOR	Real32	NIVEL_ALMACENAD...	ALMACENADOR.CurrentAmount
29	TIEMPO_TOTAL_SIMULADO	Int16	TIEMPO_TOTAL_SIM...	EventController.tiempoSimulacion

Figura 3.18: Variables tipo marca utilizadas en el PLC.

1. **tiempoSimulación:** Indica el tiempo total de la simulación.
2. **tiempoTrabajado:** Indica el tiempo que la estación estuvo en estado de Working.
3. **tiempoEspera:** Indica el tiempo que la estación estuvo en estado de

Waiting.

4. **tiempoFalla:** Indica el tiempo que la estación estuvo en estado de Failing.
5. **tiempoBloqueado:** Indica el tiempo que la estación estuvo en estado de Blocking.
6. **estado:** Indica el estado actual de la estación.
7. **energía:** Indica la energía en KW que consume la estación. Todos estos datos en conjuntos de datos fueron aplicados a cada una de las estaciones de la planta extractora de aceite que son: estación extractora, estación seccionadora, estación perforadora y horno.

Finalmente, los datos que son tipo **Data Block** que se utilizan en el PLC, se pueden observar en la figura 3.19.

	Identifíer	Data Type	Alias	Simulation Model Attribute
1	ALMACENADOR.ENTRADA_...	Boolean	ALMACENADOR_EN...	ALMACENADOR.EntranceLocked
2	ALMACENADOR.SALIDA_AL...	Boolean	ALMACENADOR_SAL...	ALMACENADOR.ExitLocked
3	HISTORICOS.uplinkFlag	Boolean	HISTORICOS_uplink...	btn_up.uplinkFlag

Figura 3.19: Variables tipo Data Block utilizados en el PLC.

A continuación se explica el uso de las variables dentro de los **Data Block** en la planta simulada.

1. Para hacer un control de las válvulas de ingreso y salida del líquido a los dos almacenadores existentes en la planta, se utilizó las extensiones: “variable.EntranceLocked” y “variable.ExitLocked” respectivamente.
2. Para cargar los datos de los históricos de cada planta, se programó un botón que permite subir el estado de trabajo de cada una de las estaciones. La variable utilizada fue **btn_up.uplinkFlag**

Programación en el PLC

El PLC que se utilizó para el control de la planta extractora de aceite es un s7 1500(CPU 1512C-1 PN) de la marca Siemens, el mismo que está configurado con la dirección IP y máscara de red que se puede observar en la tabla 3.2.

Tabla 3.2: Dirección IP utilizada en Planta extractora de aceite.

Tipo	Descripción
Dirección IP	10.0.2.100
Máscara de Subred	255.255.255.0

Las variables utilizadas en el PLC para la comunicación con laplanta virtual se las estructuró en forma de Data Block (véase la figura 3.20), el cual consisten en un bloque con un grupo de variables de diferente tipo como entero, booleano, real [Berger, 2012]. Estas envían y reciben información del estado de funcionamiento de cada estación.

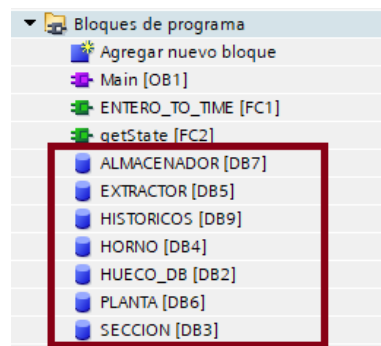


Figura 3.20: Variables tipo Data Block utilizados en el PLC.

Se programó dos funciones que se pueden observar en la figura 3.21. La primera **ENTERO_TO_TIME[FC1]** se encarga de convertir la variable de tipo tiempo a una de valor entero, estas variables muestran tanto el tiempo de trabajo, tiempo de espera, tiempo de fallo y bloqueo de cada una de las estaciones anteriormente descritas. La estructura de este bloque se puede observar en la figura 3.22.



Figura 3.21: Función para conversión de variable tipo TIME a entero.

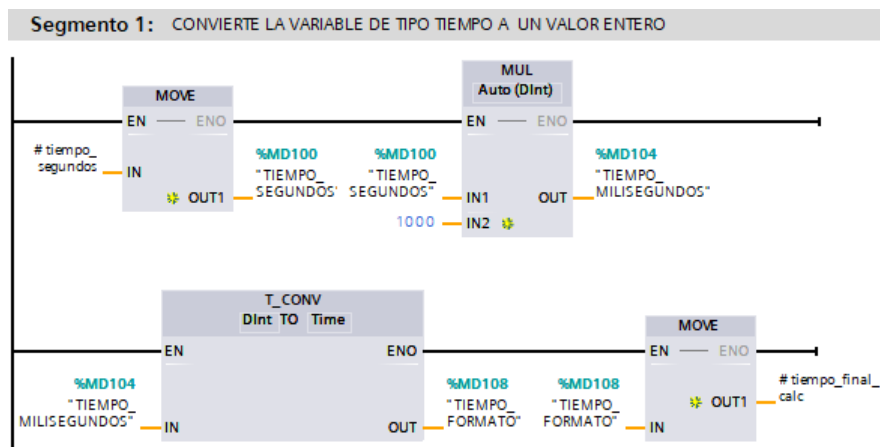


Figura 3.22: Conversión de variables tipo TIME a entero.

La segunda función **getState[FC2]** se encarga de convertir los valores numéricos que indican el funcionamiento de la planta en valores tipo string, como se puede apreciar en la tabla 3.3.

Estado	Nombre del String
1	Working
2	Waiting
3	Blocked
2	Falling

Tabla 3.3: Conversión de valor numérico a String.

En la figura 3.23 se puede observar el bloque de configuración realizado en el PLC.

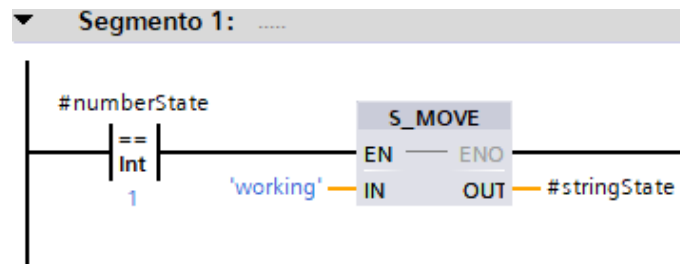


Figura 3.23: Bloque de conversión numérico a String en el PLC.

3.3.2. Comunicación Industrial Profinet

Para poder establecer un enlace entre las dos plantas descritas en el presente proyecto se realizó una red industrial PROFINET (véase la figura 3.24) entre los dos PLC que controlan cada planta.

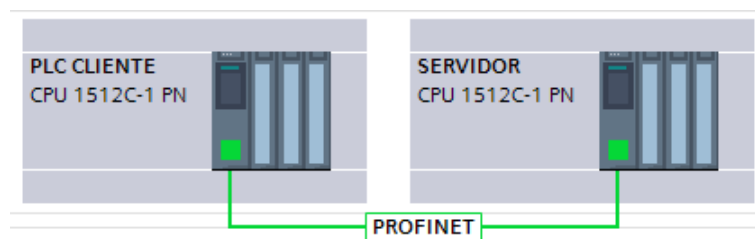


Figura 3.24: Configuración de Red Industrial entre PLCs.

El PLC servidor corresponde al que se utilizó para controlar la planta extractora de aceite, éste envía el dato del nivel del almacenador en la

memoria local con dirección **MD18** y reserva un espacio de memoria **MD60** en el PLC cliente. De igual manera se recibe un dato de tipo booleano del PLC cliente, el mismo que sirve para bloquear la salida de líquido en la estación de almacenamiento de la planta extractora de aceite, este bloque de programación se puede observar en la figura 3.25.

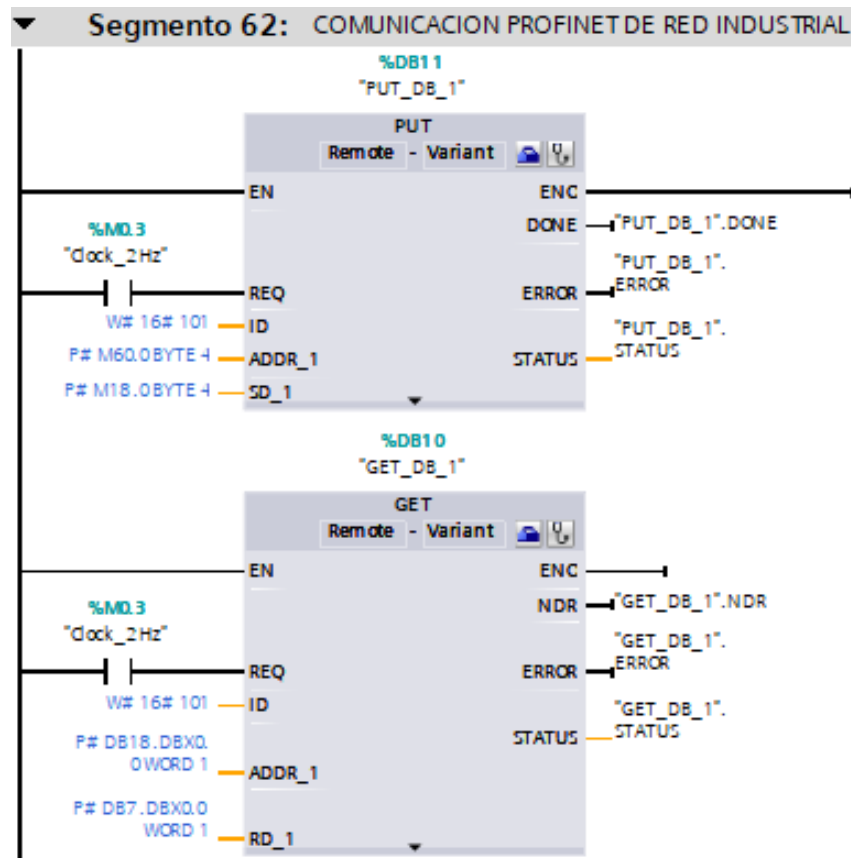


Figura 3.25: Envío y recepción de datos por de Red Industrial PROFINET.

3.3.3. Planta Embotelladora de Aceite

Esta planta consiste en una celda de manufactura que se encarga de embotellar con aceite de coco diferentes botellas, para luego proceder a ser empaquetadas y colocadas en cajas de distribución. La planta a simular está estructurada por los elementos que se pueden apreciar en la tabla 3.4

Estación	Descripción
Almacenador	Esta estación contiene la cantidad de aceite enviada de la planta extractora de aceite
Vaciador	Aquí se almacena el líquido que se vacía del tanque
Tanque	Carga el aceite para empezar con el llenado de las botellas
Contenedor de Producto	En esta estación se encuentran las botellas que van a ser llenadas de aceite.
Llenado de líquido	Ingresa las botellas para llenarlas de aceite
Tapadora de botella	Coloca una tapa en cada botella
Etiquetadora	Coloca una etiqueta en cada botella
Brazo Robot 1 y 2	Traslada cada botella a la caja
Empaquetadora	Traslada las cajas llenas de botellas

Tabla 3.4: Elementos de Planta embotelladora de Aceite.

Las estaciones descritas en la tabla anterior han sido controladas mediante un PLC S7-1500 de la marca Siemens.

Con el software TIA PORTAL V16.0 se programó cada bloque para hacer posible el funcionamiento de la planta, cada uno de los PLC fueron simulados mediante el software PLSIM Advance 3.0, el mismo que tiene compatibilidad con el programa Tecnomatrix Plant Simulation.

En la figura 3.26 se puede observar el diseño 2D de la planta extractora de aceite.

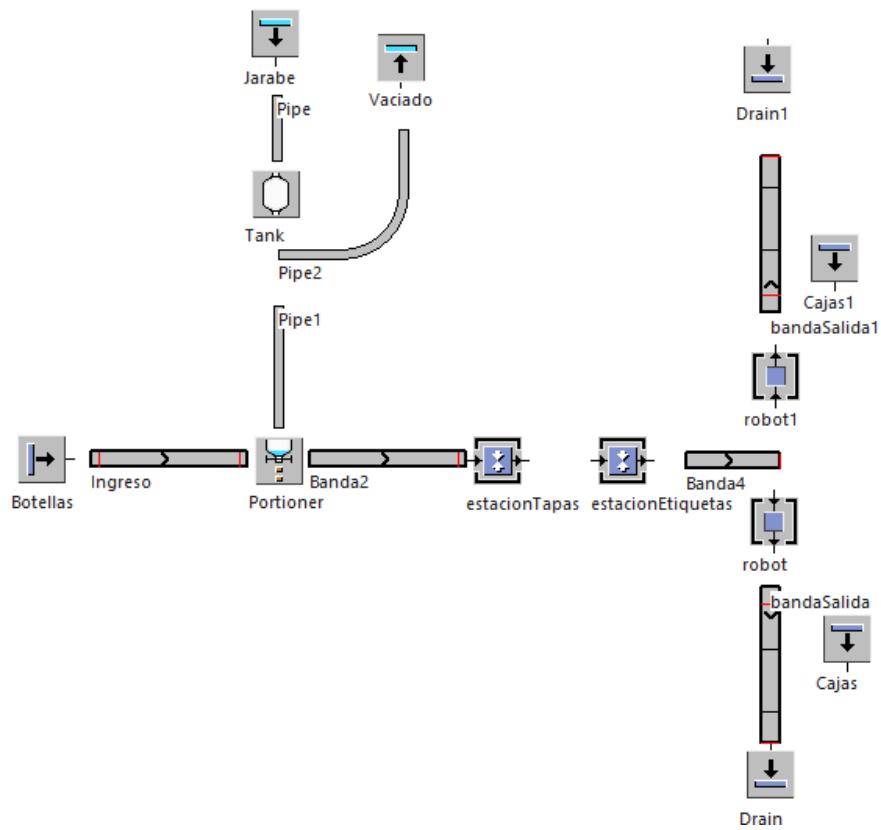


Figura 3.26: Planta Extractora de Aceite en 2D.

Para tener una mejor apreciación de la planta se puede observar en la figura 3.27 el diseño en 3D.

Funcionamiento de la Planta

Al ingreso de la planta se encuentra el almacenador que proviene de la primera planta extractora de aceite. Este almacenador se puede observar en la figura 3.28.

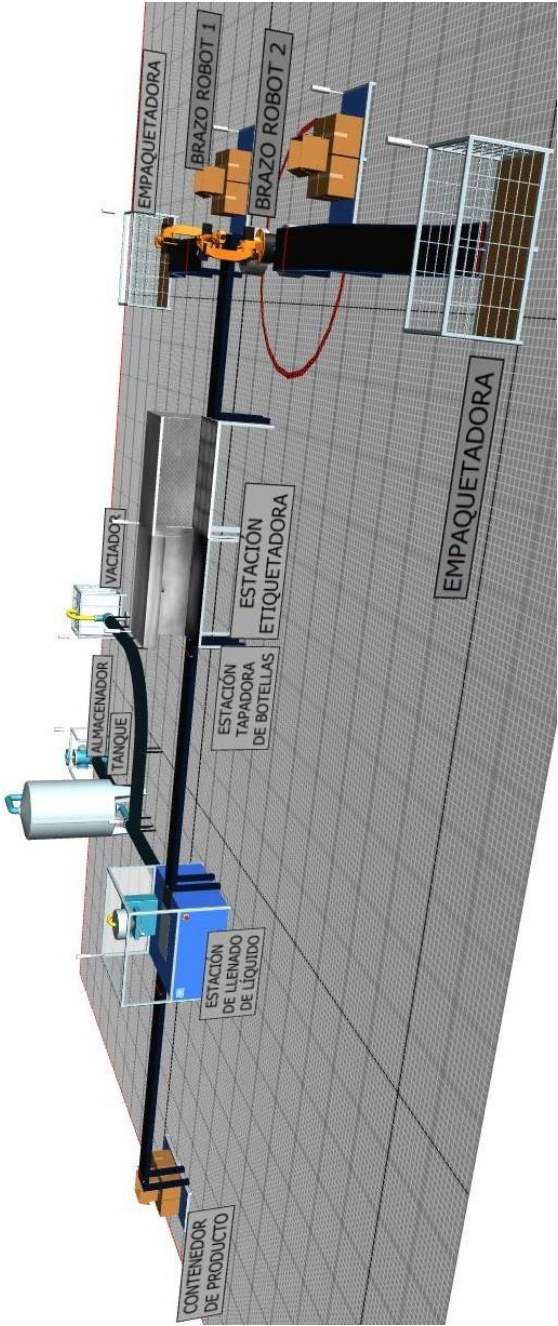


Figura 3.27: Planta Embotelladora de Aceite en 3D.

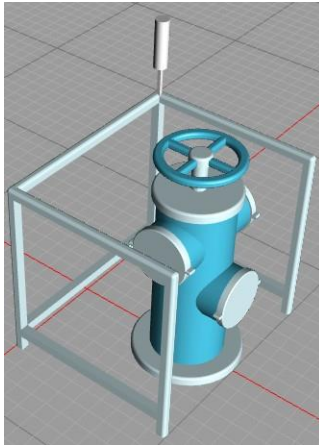


Figura 3.28: Almacenador de la Planta embotelladora de aceite.

En caso de quedar algún residuo de líquido en el tanque, se procede a vaciarlo en la estación de vaciado que se observa en la figura 3.29.

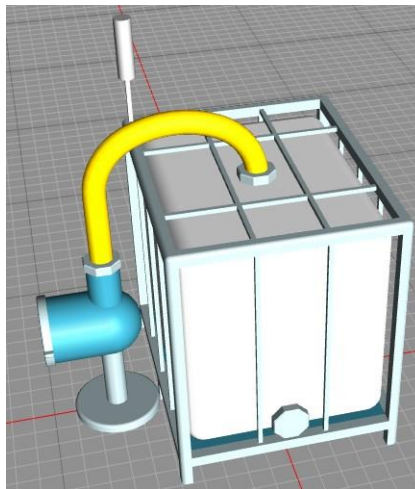


Figura 3.29: Vaciador de aceite de Planta embotelladora.

Una vez que se ha vaciado por completo el tanque, se empieza a llenarlo de aceite que proviene del almacenador. La cantidad de aceite que soporta el tanque es de 100 litros. Esto se puede observar en la figura 3.30.

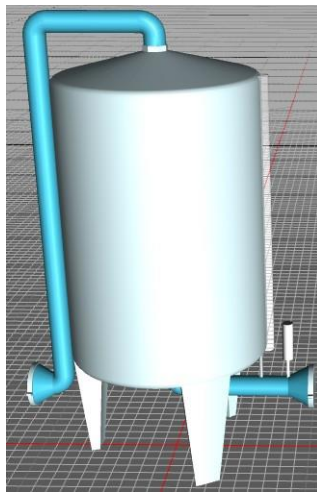


Figura 3.30: Tanque de Planta embotelladora de aceite.

Cuando el tanque se ha llenado por completo este empieza a vaciar su contenido a través de las tuberías a la estación de llenado de aceite que se observa en la figura 3.31. Aquí cada botella pasa a ser llenada con aceite.

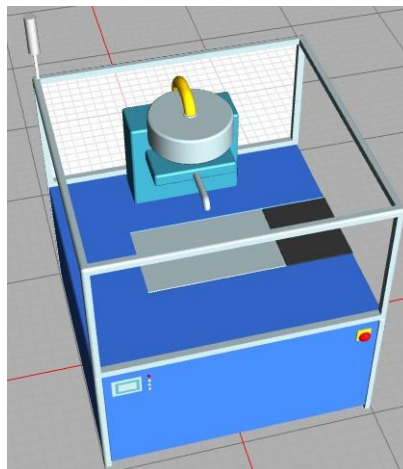


Figura 3.31: Estación de llenado de aceite en botellas.

Las botellas pasan a través de una banda transportadora que se observa en la figura 3.32, la misma que contiene sensores de ingreso de producto y

salida del mismo.

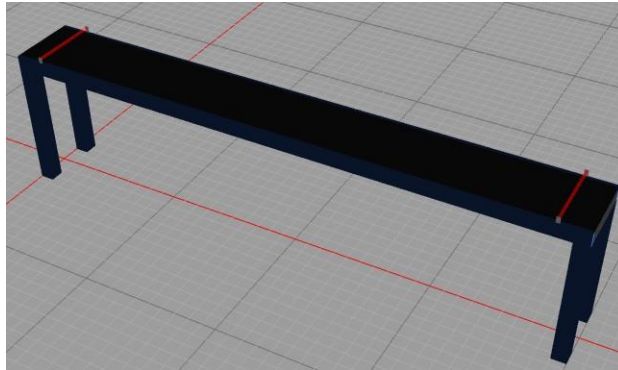


Figura 3.32: Banda transportadora de producto.

Las botellas pasan a ser colocadas su correspondiente tapa en la estación que se observa en la figura 3.33.

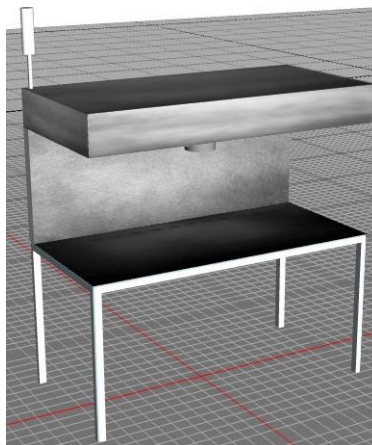


Figura 3.33: Colocación de tapa a las botellas.

Luego se coloca a cada botella su correspondiente etiqueta en la estación que se observa en la figura 3.34.

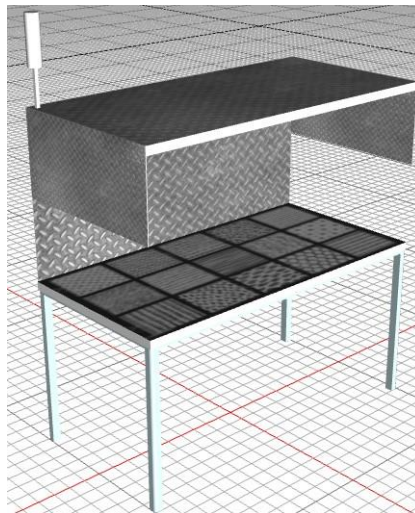


Figura 3.34: Colocación de etiqueta a las botellas.

Al terminar este proceso las botellas quedan listas para ser empaquetadas en sus correspondientes cajas. El producto final que se obtiene se puede observar en la figura 3.35.

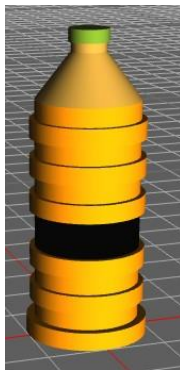


Figura 3.35: Botella final con tapa y etiqueta.

Se dispone de dos brazos robot de marca KUKA que se encargan de colocar las botellas en las diferentes cajas. La forma física de estos robots se puede apreciar en la figura 3.36.

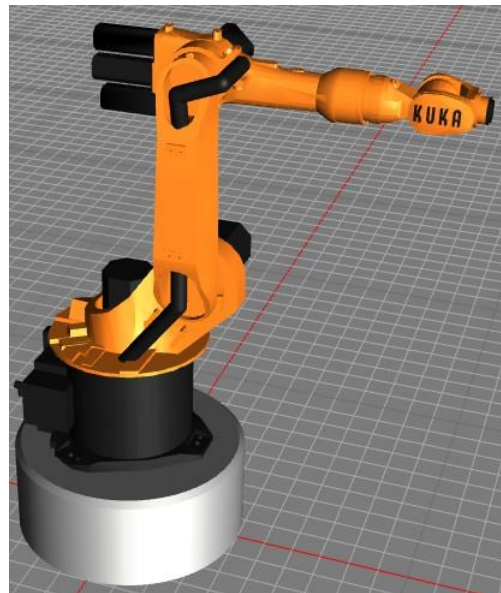


Figura 3.36: Brazo robot de marca KUKA.

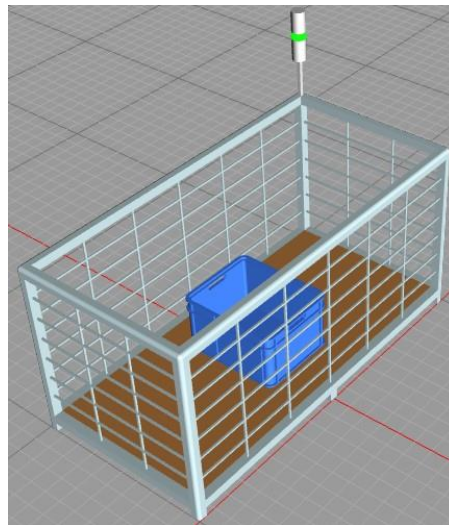


Figura 3.37: Empaquetadora de botellas.

Una vez colocadas las botellas en cada caja, estas se trasladan a la estación

que se observa en la figura 3.37.

Todo el proceso anteriormente descrito se lo puede controlar con el panel de control que se puede observar en la figura 3.38. A través de este panel se puede encender y parar la planta y subir los datos de funcionamiento de cada estación.

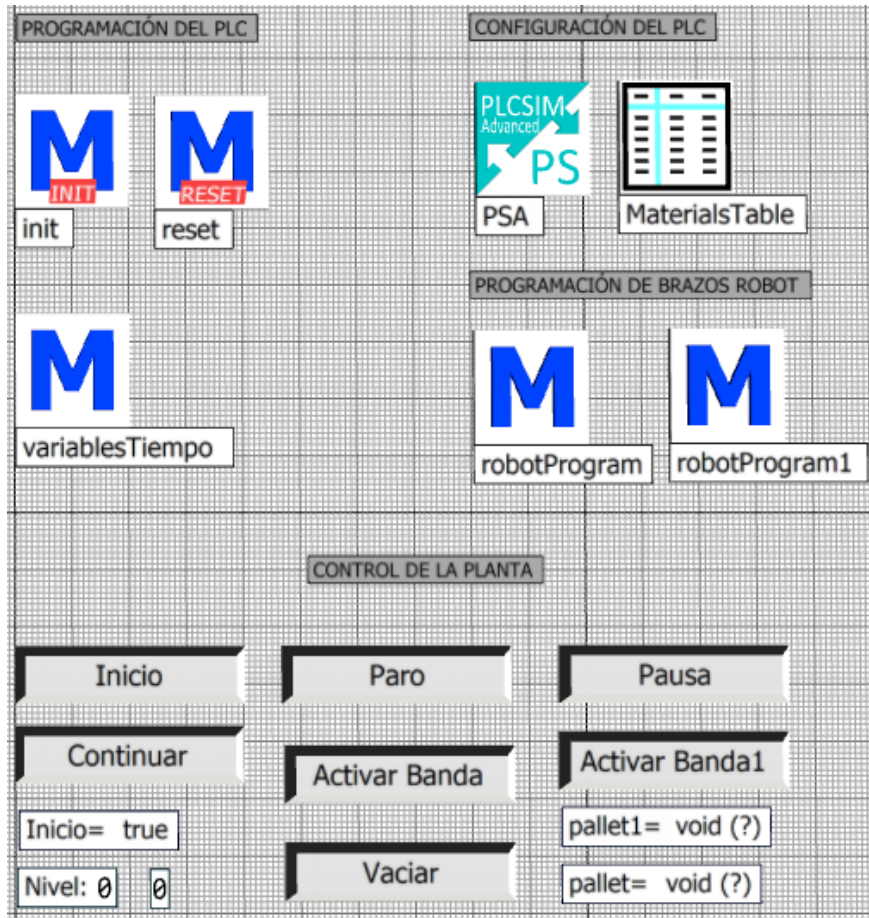


Figura 3.38: Panel de control de la planta embotelladora de aceite.

Comunicación de la Planta Embotelladora de aceite con el PLC

En la figura 3.39 se puede observar las variables de ingreso al PLC que salen de la planta simulada en el software Tecnomatrix Plant Simulation. Al ser variables de ingreso estas representan los sensores de cada estación,

información que es utilizada realizar la automatización de la planta. En el nombre de cada variable se detalla la estación y el dato que se obtiene separado con un guión bajo. Por otra parte, en la figura 3.38 se presentan las variables para cada botón del panel de control.

The screenshot shows a window titled ".Models.Model.PSA - Items[1, 1]" with a sub-header "portioner_estado". Below it is a table with 4 columns: Identifier, Data Type, Alias, and Simulation Model Attribute. The table lists 23 variables, each with a unique identifier, a data type (mostly Boolean, one Int16, one Real32), an alias, and a corresponding simulation model attribute.

Identifier	Data Type	Alias	Simulation Model Attribute
1 portioner_estado	Int16	portioner_estado	
2 tanque_Nivel	Real32	tanque_Nivel	Tank.CurrentAmount
3 tanque_sensorVacio	Boolean	tanque_sensorVacio	Tank.Empty
4 tanque_sensorLleno	Boolean	tanque_sensorLleno	Tank.Full
5 bandaIngreso_sensorSalida	Boolean	bandaIngreso_sens...	Ingreso.SensorID(1).isTriggered
6 portioner_Botella	Boolean	portioner_Botella	Portioner.Botella
7 estacionTapas_Botella	Boolean	estacionTapas_Botella	estacionTapas.Botella
8 banda2_sensorSalida	Boolean	banda2_sensorSalida	Banda2.SensorID(1).isTriggered
9 estacionEtiquetas_Botella	Boolean	estacionEtiquetas_B...	estacionEtiquetas.Botella
10 bandaIngreso_sensorIngreso	Boolean	bandaIngreso_sens...	Ingreso.SensorID(2).isTriggered
11 robot_Botella	Boolean	robot_Botella	robot.Botella
12 banda4_sensorSalida	Boolean	banda4_sensorSalida	Banda4.SensorID(1).isTriggered
13 bandaSalida_sensorIngreso	Boolean	bandaSalida_sensor...	bandaSalida.SensorID(1).isTriggered
14 bandaSalida_sensorSalida	Boolean	bandaSalida_sensor...	Drain.Occupied
15 banderaVaciado	Boolean	banderaVaciado	btnVaciado.flagActivacion
16 botonBanda	Boolean	botonBanda	btnSalida.Flag
17 bandaSalida	Boolean	bandaSalida	bandaSalida.Occupied
18 bandaSalida1_sensorIngreso	Boolean	bandaSalida1_sens...	bandaSalida1.SensorID(1).isTrigge...
19 bandaSalida1_sensorSalida	Boolean	bandaSalida1_sens...	Drain1.Occupied
20 robot1_Botella	Boolean	robot1_Botella	robot1.Botella
21 botonBanda1	Boolean	botonBanda1	btnSalida1.Flag
22 bandaSalida1	Boolean	bandaSalida1	bandaSalida1.Occupied
23 btnSubir	Boolean	btnSubir	btnParo.uplinkFlag

Figura 3.39: Variables de ingreso al PLC.

A continuación se listan las variables que se obtienen de cada estación.

1. **Nivel:** Utilizada en los tanques para indicar el Nivel de líquido.
2. **sensorVacio:** Utilizada en los tanques para indicar cuando el tanque esta completamente vacío.
3. **sensorLleno:** Utilizado en los tanques para indicar cuando el tanque esta lleno.
4. **sensorSalida:** Utilizado en las bandas transportadoras para indicar la presencia de un elemento al final de la banda.

5. **Botella**: Utilizado en las estaciones para indicar cuando ha ingresado una botella en una estación.
6. **sensorIngreso**: Utilizado en las bandas transportadoras para indicar la presencia de un elemento al inicio de la banda.
7. **banderaVaciado**: Se activa al presionar el botón de Vaciar y realiza el vaciado del tanque.
8. **botonBanda**: Se activa al presionar el botón de Activar Banda y activa la banda final del lado derecho, por lo general se lo activa al final de cada producción.
9. **bandaSalida**: Se activa al presionar el botón de Activar Banda 1 y activa la banda final del lado izquierdo, por lo general se lo activa al final de cada producción.
10. **btnSubir**: Se activa al momento de presionar el botón de paro y sube la información de la producción al sistema Thingsboard. Utilizado para generar los históricos.

En la figura 3.40 se puede observar las variables de salida hacia el PLC. Al ser variables de salida estas representan los actuadores de cada estación, información que es utilizada realizar la automatización de la planta, en el nombre de cada variable se detalla la estación y el dato que se controla separado con un guión bajo. A continuación se listan las variables que se obtienen de cada estación.

1. **bloqueoEntrada**: Utilizada en los tanques. Cuando esta variable está activa bloquea la entrada de líquido al tanque.
2. **bloqueoSalida**: Utilizada en los tanques. Cuando esta variable está activa bloquea la salida de líquido del tanque.
3. **Pausa**: Utilizada en las bandas transportadoras. Cuando esta variable está activa se detiene la banda.
4. **Start**: Utilizada en los brazos robot. Cuando esta variable está activa indica al brazo que funcione.

	Identifier	Data Type	Alias	Simulation Model Attribute
1	tanque_bloqueoEntrada	Boolean	tanque_bloqueoEnt...	Tank.EntranceLocked
2	tanque_bloqueoSalida	Boolean	tanque_bloqueoSali...	Tank.ExitLocked
3	bandaIngreso_Pausa	Boolean	bandaIngreso_Pausa	Ingreso.Pause
4	portioner_bloqueoEntrada	Boolean	portioner_bloqueoE...	Portioner.EntranceLocked
5	banda2_Pausa	Boolean	banda2_Pausa	Banda2.Pause
6	banda3_Pausa	Boolean	banda3_Pausa	
7	botella_Pausa	Boolean	botella_Pausa	Botellas.Pause
8	banda4_Pausa	Boolean	banda4_Pausa	Banda4.Pause
9	bandaSalida_Pausa	Boolean	bandaSalida_Pausa	bandaSalida.Pause
10	cajas_Pausa	Boolean	cajas_Pausa	cajas.Pause
11	robot_Start	Boolean	robot_Start	robot.Start
12	vaciador_bloqueoEntrada	Boolean	vaciador_bloqueoE...	Vaciado.EntranceLocked
13	robot_Pausa	Boolean	robot_Pausa	robot.Stopped
14	robot1_Start	Boolean	robot1_Start	robot1.Start
15	bandaSalida1_Pausa	Boolean	bandaSalida1_Pausa	bandaSalida1.Pause
16	cajas1_Pausa	Boolean	cajas1_Pausa	Cajas1.Pause

Figura 3.40: Variables de salida al PLC.

Las variables que son utilizadas como marcas en el PLC se pueden observar en la figura 3.41. Estas marcas son utilizadas para el envío de información de los parámetros de las estaciones hacia el PLC, dado que estos se leen con métodos generados en Plant Simulation. Así también se realiza el control de botones del panel 3.38. A continuación se listan las variables que se obtienen de cada estación.

1. **banderaInicio:** Se activa con el botón de inicio, e inicia el funcionamiento de la planta.
2. **banderaContinuar:** Se activa con el botón de inicio y se desactiva con el botón de Pausa. Permite pausar el funcionamiento de la planta.
3. **tiempoTotal:** Indica el tiempo total de la simulación.
4. **tiempoTrabajado:** Utilizado en estaciones para indicar el tiempo que la estación estuvo trabajando (Working).
5. **tiempoEspera:** Utilizado en estaciones para indicar el tiempo que la estación estuvo en estado de para (Waiting).
6. **tiempoFalla:** Utilizado en estaciones para indicar el tiempo que la estación estuvo en estado de Fallo (Failing).

7. **tiempoBloqueado:** Utilizado en estaciones para indicar el tiempo que la estación estuvo en estado de bloqueo (Blocking).
8. **estado:** Utilizado en estaciones para indicar el estado actual de la estación.
9. **energia:** Utilizado en estaciones para indicar la energía en KW que consume la estación.

The screenshot shows a window titled ".Models.Model.PSA - Items[1,3]" with a close button. Below the title bar is a table with the following columns: "Identifer", "Data Type", "Alias", and "Simulation Model Attribute". The table lists 33 variables, each with a unique identifier, a data type (Boolean, Int16, or Real32), an alias, and a corresponding simulation model attribute.

	Identifer	Data Type	Alias	Simulation Model Attribute
1	banderalnicio	Boolean	banderalnicio	Inicio
2	banderaContinuar	Boolean	banderaContinuar	btnContinuar.Flag
3	tiempoTotal	Int16	tiempoTotal	EventController.tiempoSimulacion
4	portioner_tiempoTrabajado	Int16	portioner_tiempoTr...	Portioner.tiempoTrabajado
5	portioner_tiempoEspera	Int16	portioner_tiempoEs...	Portioner.tiempoEspera
6	portioner_tiempoFalla	Int16	portioner_tiempoFa...	Portioner.tiempoFalla
7	portioner_tiempoBloqueado	Int16	portioner_tiempoBl...	Portioner.tiempoBloqueado
8	estacionTapas_tiempoTrabaj...	Int16	estacionTapas_tiem...	estacionTapas.tiempoTrabajado
9	estacionTapas_tiempoEspera	Int16	estacionTapas_tiem...	estacionTapas.tiempoEspera
10	estacionTapas_tiempoFalla	Int16	estacionTapas_tiem...	estacionTapas.tiempoFalla
11	estacionTapas_tiempoBloqu...	Int16	estacionTapas_tiem...	estacionTapas.tiempoBloqueado
12	estacionEtiquetas_tiempoTr...	Int16	estacionEtiquetas_ti...	estacionEtiquetas.tiempoTrabajado
13	estacionEtiquetas_tiempoEs...	Int16	estacionEtiquetas_ti...	estacionEtiquetas.tiempoEspera
14	estacionEtiquetas_tiempoFa...	Int16	estacionEtiquetas_ti...	estacionEtiquetas.tiempoFalla
15	estacionEtiquetas_tiempoBl...	Int16	estacionEtiquetas_ti...	estacionEtiquetas.tiempoBloquea...
16	robot_tiempoTrabajado	Int16	robot_tiempoTrabaj...	robot.tiempoTrabajado
17	robot_tiempoEspera	Int16	robot_tiempoEspera	robot.tiempoEspera
18	robot_tiempoFalla	Int16	robot_tiempoFalla	robot.tiempoFalla
19	robot_tiempoBloqueado	Int16	robot_tiempoBloqu...	robot.tiempoBloqueado
20	robot1_tiempoTrabajado	Int16	robot1_tiempoTraba...	robot1.tiempoTrabajado
21	robot1_tiempoEspera	Int16	robot1_tiempoEspera	robot1.tiempoEspera
22	robot1_tiempoFalla	Int16	robot1_tiempoFalla	robot1.tiempoFalla
23	robot1_tiempoBloqueado	Int16	robot1_tiempoBloq...	robot1.tiempoBloqueado
24	portioner_Estado	Int16	portioner_Estado	portioner.Estado
25	estacionTapas_Estado	Int16	estacionTapas_Estado	estacionTapas.Estado
26	estacionEtiquetas_Estado	Int16	estacionEtiquetas_E...	estacionEtiquetas.Estado
27	robot_Estado	Int16	robot_Estado	robot.Estado
28	robot1_Estado	Int16	robot1_Estado	robot1.Estado
29	portioner_energia	Real32	portioner_energia	portioner.Energy
30	estacionTapas_energia	Real32	estacionTapas_ener...	estacionTapas.StatEnergyTotalCon...
31	estacionEtiquetas_energia	Real32	estacionEtiquetas_e...	estacionEtiquetas.StatEnergyTotal...
32	robot_energia	Real32	robot_energia	robot.StatEnergyTotalConsumption
33	robot1_energia	Real32	robot1_energia	robot1.StatEnergyTotalConsumpti...

Figura 3.41: Variables tipo marca utilizadas en el PLC.

Programación en el PLC

El PLC que se utilizó para el control de la planta embotelladora de aceite es un s7 1500(CPU 1512C-1 PN) de la marca Siemens. La configuración de la dirección IP y la máscara de red de este autómata se puede observar en la tabla 3.5.

Tipo	Descripción
Dirección IP	10.0.2.101
Máscara de Subred	255.255.255.0

Tabla 3.5: Dirección IP utilizada en Planta embotelladora de aceite.

Las variables utilizadas en el PLC para la comunicación con la planta virtual se las estructuró en forma de Bloque de Datos (Data Block), el cual consiste de un grupo de variables de diferente tipo como entero, booleano, real [Berger, 2012], estas envían y reciben información del estado de funcionamiento de cada estación. Los bloques de datos creados se pueden apreciar en la figura 3.42.

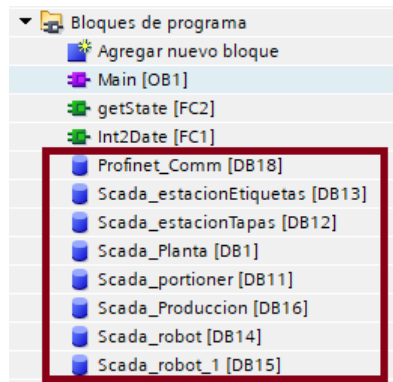


Figura 3.42: Variables tipo Data Block utilizados en el PLC.

Se programó dos funciones que se pueden observar en la figura 3.43, la primera **Int2Date[FC1]** se encarga de convertir la variable de tipo tiempo a una de valor entero. Estas variables muestran el tiempo de trabajo, el tiempo de espera, el tiempo de fallo y el tiempo de bloqueo de cada una de las estaciones anteriormente descritas. La estructura de este bloque se puede observar en la figura 3.44.

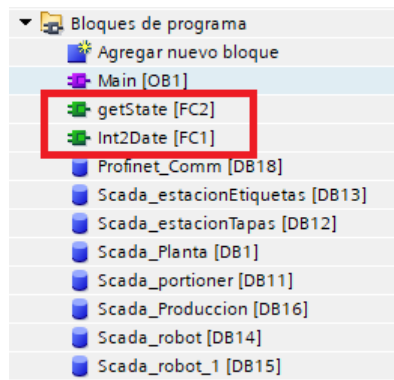


Figura 3.43: Función para conversión de variable tipo TIME a entero.

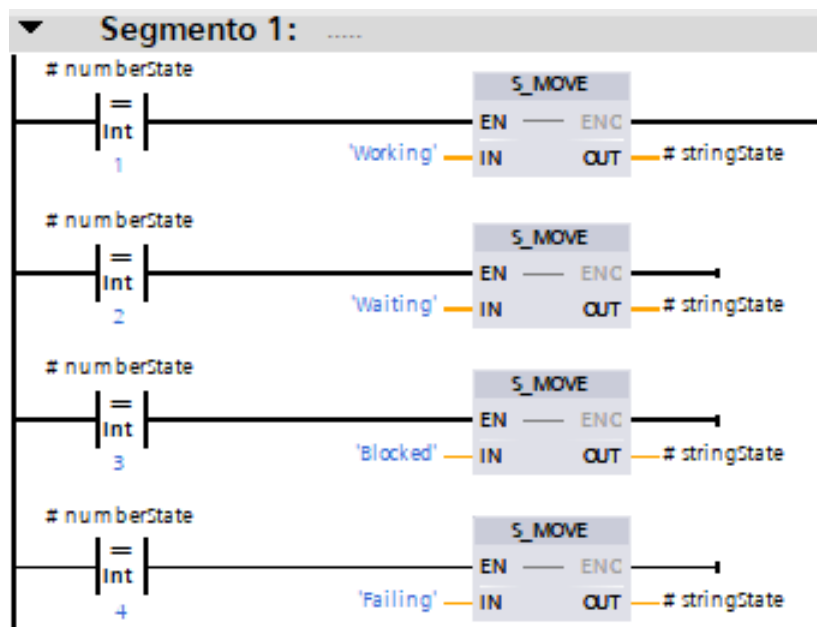


Figura 3.44: Conversión de variables tipo TIME a entero.

La segunda función **getState[FC2]** se encarga de convertir los valores numéricos que indican el funcionamiento de la planta en valores tipo string, como se puede apreciar en la tabla 3.6

Estado	Nombre del String
1	Working
2	Waiting
3	Blocked
2	Falling

Tabla 3.6: Conversión de valor numérico a String.

En la figura 3.45 se puede observar el bloque de configuración realizado en el PLC.

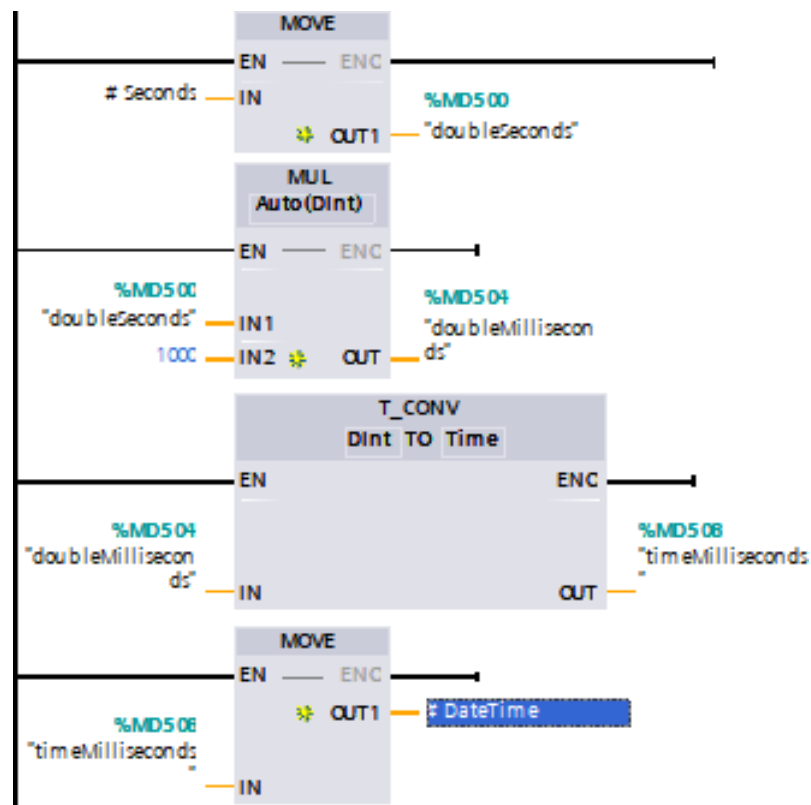


Figura 3.45: Bloque de conversión numérico a String en el PLC.

3.4. Diseño de la Capa de Comunicación

El diseño de red del sistema abarcará la topología y configuraciones necesarias que se deben realizar para que los diferentes dispositivos listados en la capa física puedan comunicarse entre ellos y de esta manera poder compartir información de interés que favorecerá al monitoreo de la planta.

3.4.1. Gemelo digital del PLC con PLCSIM Advance

De acuerdo con el manual de PLCSIM Advanced [Siemens A.G., 2016], al momento de realizar la instalación de este programa por defecto se crea un adaptador de red virtual con nombre Siemens PLCSIM Virtual Ethernet Adapter, el mismo que funciona como un switch virtual el cual favorece la interconexión de los distintos PLC virtuales que se creen y el software que se utiliza para realizar la carga del PLC. Esta arquitectura de conectividad se ilustra en la figura 3.46

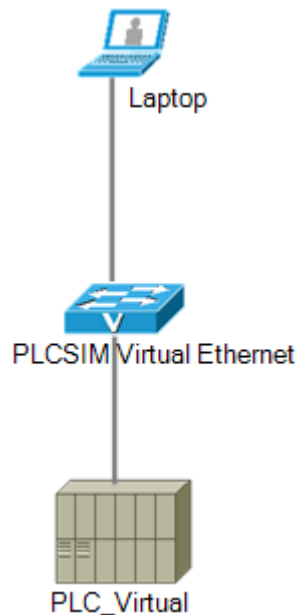


Figura 3.46: Interfaz de comunicación con PLCSIM Advance.

Para poder realizar una correcta implementación del sistema de comunicación se debe tener esta consideración, puesto que todos los dispositivos a configurar deben tener una dirección IP dentro de la misma red.

3.4.2. Diseño de la red

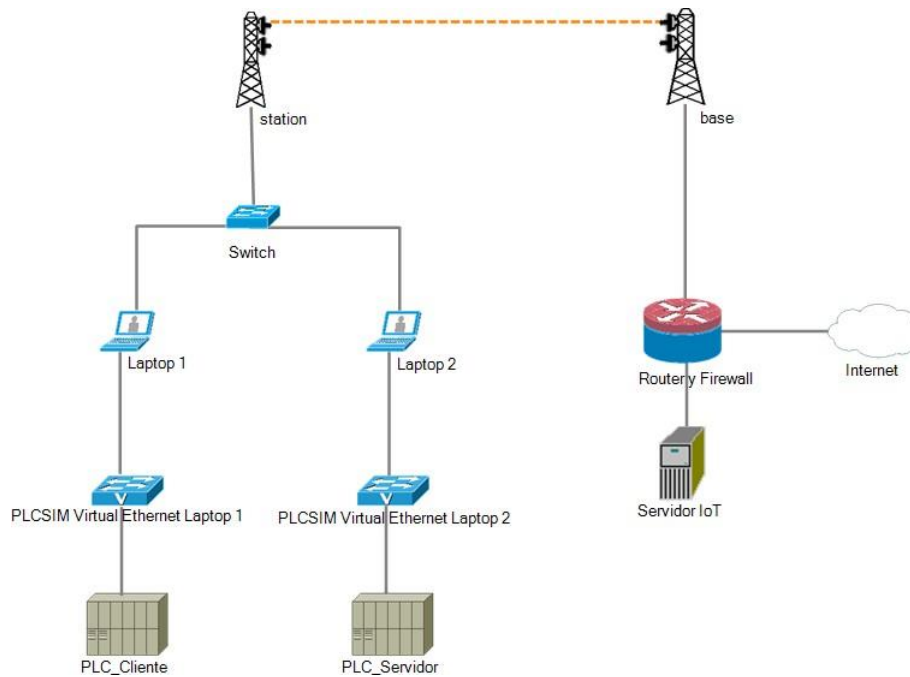


Figura 3.47: Topología de la Red Diseñada.

En la figura 3.47 se muestra la topología diseñada para el sistema en base a los requerimientos mencionados en la sección 3.2. Esta topología está estructurada con los siguientes elementos:

Router

El router es el elemento más importante de la red dado que realiza el enrutamiento y direccionamiento de los distintos elementos que conforman el sistema. De igual manera, este equipo proveerá salida a internet, lo que

hará posible el acceso externo al servidor. Se buscó un equipo en el que se puedan configurar los siguientes servicios:

- Servidor DHCP, para el direccionamiento de los dispositivos.
- Firewall, para establecer reglas de ciberseguridad.
- Port forwarding, para el direccionamiento de los diferentes puertos.
- NAT, para la salida a internet.

Se decidió utilizar un router integrado de la marca Mikrotik modelo hAP AC Lite dado que opera con el sistema operativo RouterOS, el cual viene con un conjunto de herramientas muy útiles para el monitoreo de red, además de cumplir con las características antes mencionadas y contar con una excelente relación costo/beneficio comparados con otros equipos en el mercado [Escalante, 2019]. El equipo se puede observar en la figura 3.48



Figura 3.48: RouterBoard hAP AC Lite.

Radio enlace

Un radio enlace es la interconexión de dos terminales de telecomunicación, un transmisor y un receptor, que utilizan ondas de radios para transportar información a través del espacio libre [Escalante, 2021]. Se utilizará en este

sistema para el envío de datos desde la estación donde se encuentran los PLC hacia la base donde está ubicado el servidor. El equipo seleccionado debe contar con las siguientes características:

- Frecuencia de operación: 5GHz.
- Rango \geq 1Km.
- Throughput \geq 1Mbps.

Se decidió utilizar un par de antenas de Nano Station Loco M5 de la marca Unifi dado que cumplen con las características antes mencionadas además de que constan con una buena relación calidad/precio y son de fácil instalación. Un gráfico del equipo escogido se puede observar en la figura 3.49



Figura 3.49: Unifi NanoLoco Station M5.

Switch

Dado que la antena de radio enlace cuenta solo con un puerto LAN, es necesario un elemento de red que permita interconectar los diferentes dispositivos. Para este caso se utilizó un switch Linksys de 5 puertos no administrable.

3.4.3. Direccionamiento

El direccionamiento consiste en asignar una dirección IP a cada dispositivo que pertenece a la red [Escalante, 2019], en este proyecto se utilizó la subred 10.0.2.0/24 la que posee 252 direcciones IP utilizables, las que son más que suficientes para la asignación de las direcciones IP, la tabla 3.7 muestra el direccionamiento realizado.

Dispositivo	Dirección IP
Router	10.0.2.1
Servidor IoT	10.0.2.10
Antena Base	10.0.2.11
Antena Estación	10.0.2.12
Laptop 1	10.0.2.23
Laptop 2	10.0.2.24
PLCSIM Virtual Ethernet L1	10.0.2.98
PLCSIM Virtual Ethernet L2	10.0.2.99
PLC Cliente	10.0.2.100
PLC Servidor	10.0.2.101

Tabla 3.7: Direccionamiento de red.

3.4.4. Configuración de los elementos de red

En este apartado se presenta detalles de la configuración de los dispositivos, de acuerdo al diseño y al direccionamiento previamente especificado.

Configuración del Router

Como se indica en la figura 3.50, se colocaron las interfases eth2, eth3, eth4 y et5 en puente de manera que cualquier equipo que se conecte a dichas interfases se encuentre dentro de la red LAN, por lo que se generó una interfaz virtual a la que se llamo bridge_vlan que abarca las interfaces antes mencionadas.

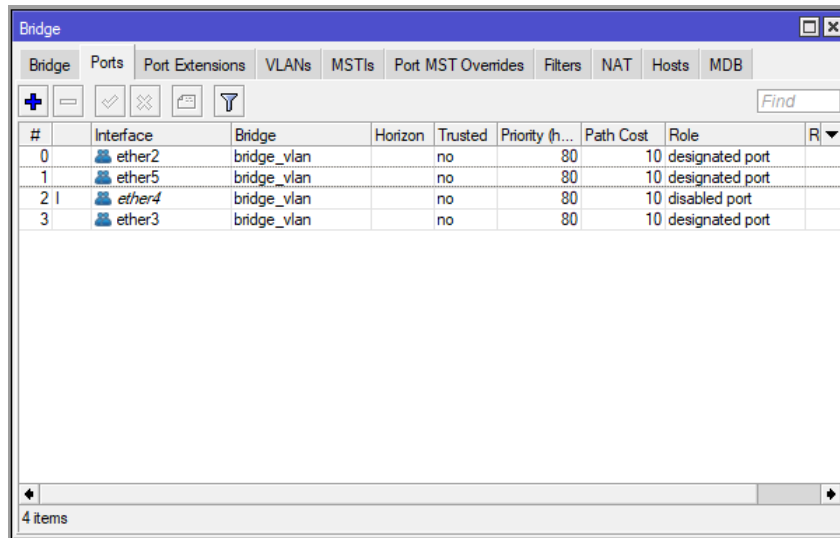


Figura 3.50: Configuración del Puente o Bridge.

A la interfaz virtual bridge_vlan se le asignó la dirección 10.0.2.1/24 según lo que se indica en la tabla 3.7. Además se configuró un servidor DHCP para la asignación de direcciones de los dispositivos que se conecten a esta red.

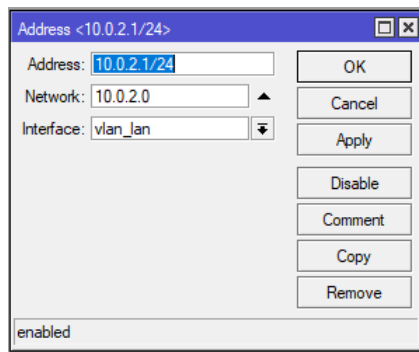


Figura 3.51: Direcccionamiento del Router.

Se configuró en la interfaz eth1 un cliente DHCP para una asignación automática de una dirección IP pública. De esta manera se podrá tener acceso remoto al sistema. También se configuró una regla de NAT en esta interfaz para la salida a internet. Los detalles de esta configuración se muestran en

la figura 3.52.

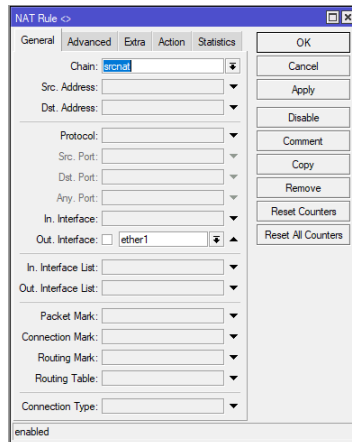


Figura 3.52: Configuración NAT.

Finalmente, se levantó un firewall que controla el acceso desde fuera de la red. En este caso se realiza una redirección del puerto 8080 del servidor al puerto 9090 del router y se coloca una lista de acceso. Esto con el fin de precautelar la seguridad y denegar el acceso no deseado. Los detalles de esta configuración se muestran en la figura 3.53.

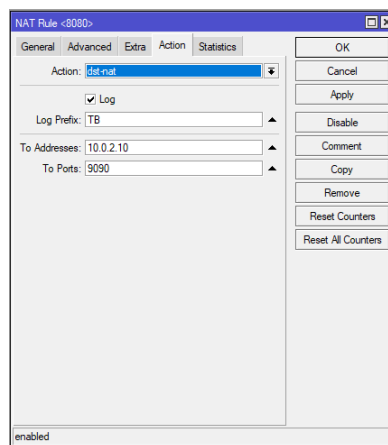


Figura 3.53: Configuración del Firewall.

Configuración de la antena Base

La antena base es un equipo inalámbrico el cual opera como punto de acceso, encargándose de anunciar la red inalámbrica a través del espectro radio eléctrico, por lo general son equipos de alto rendimiento que son capaces de soportar múltiples conexiones [Escalante, 2021]. La figura 3.54 muestra la configuración de la antena base utilizada para este proyecto.

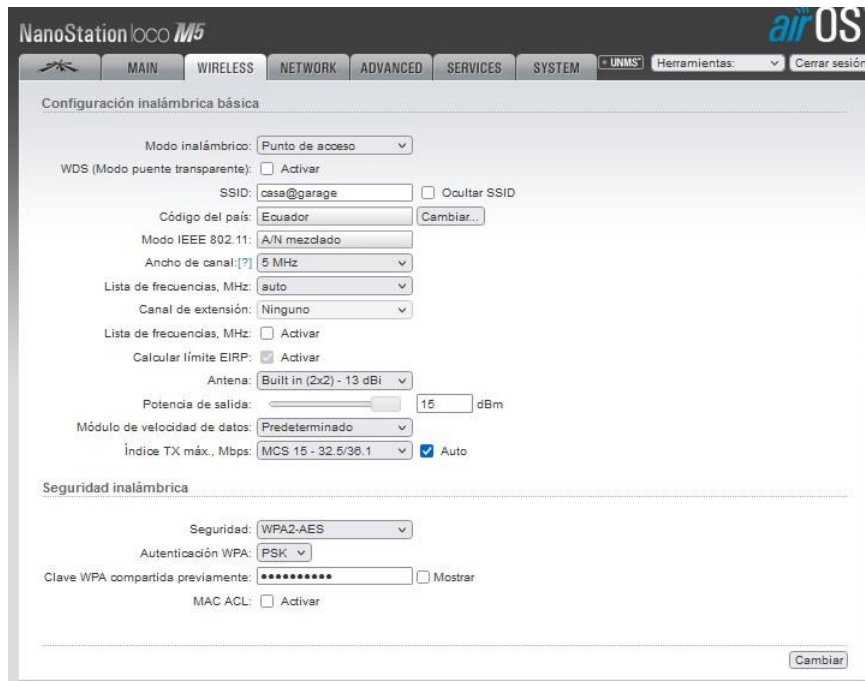


Figura 3.54: Configuración de la antena base.

Como se puede observar la antena opera en modo punto de acceso y anuncia la SSID datacenter@fabricacion. Se configuró la máxima potencia de salida con el fin de realizar pruebas de alcance.

Configuración de la antena estación

La antena estación es un equipo inalámbrico que busca y se conecta a la red inalámbrica que anuncia la antena base. Por lo general son equipos que utilizan antenas con un ángulo de cobertura estrecho con el fin de evitar la interferencia y el ruido de estaciones vecinas [Siemens A.G., 2021a]. En la

figura 3.55 se puede observar la configuración de la antena estación.

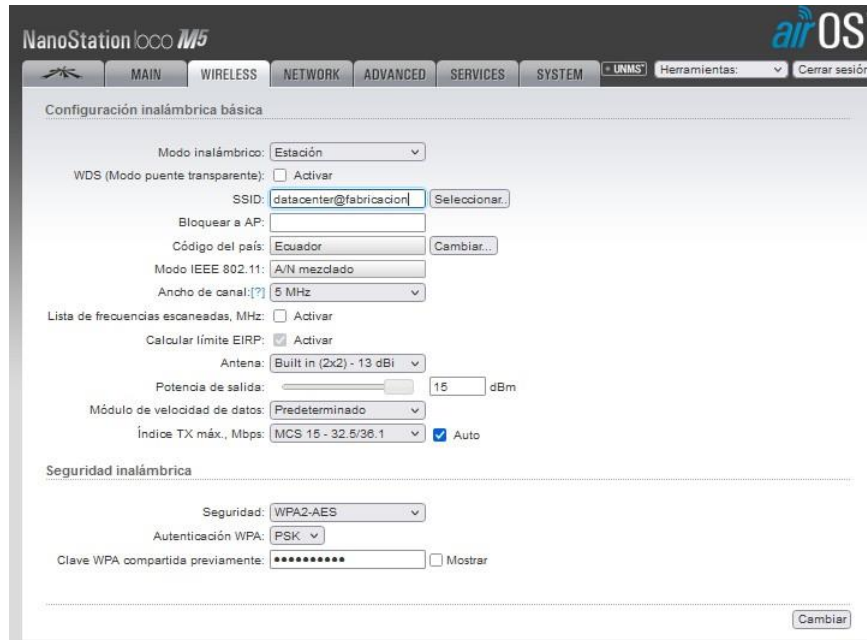


Figura 3.55: Configuración de la antena estación.

Parámetros a destacar dentro de esta configuración es que el modo de operación de la antena debe ser de estación, el SSID debe coincidir con el que difunde la antena base. Asimismo, el resto de parámetros deben ser idénticos a los configurados en la antena base para garantizar el correcto funcionamiento del equipo.

Configuración de PLCSIM Advance

Como se explicó en la sección 3.4.1, PLCSIM crea un adaptador de red virtual el cual se considera dentro de la tabla 3.7 de direccionamiento. Para ello, en Windows, hay que dirigirse a la ventana de Conexiones de Red, en ella aparecen todos los adaptadores de red que se encuentran configurados en el sistema operativo, se debe ingresar en las propiedades de IPv4 del adaptador cuyo nombre de dispositivo coincida con: Siemens PLCSIM Virtual Ethernet Adapter y colocar los datos según el diseño realizado. La configuración del adaptador virtual se puede observar en la figura 3.56.

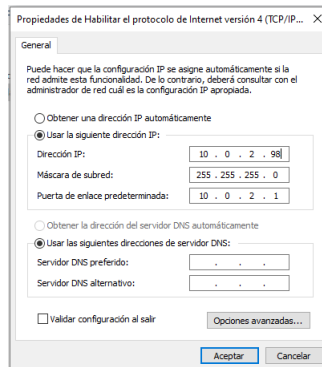


Figura 3.56: Configuración del Adaptador de red de PLCSIM.

Finalmente, se asigna la dirección al PLC virtual en la interfaz de PLCSIM Advance, tal como se indica en la figura 3.57. La consideración que se debe tener es la de escoger la tarjeta de red que se encuentre conectada al switch, con la finalidad de que todos los elementos se encuentren en la misma red de área local.

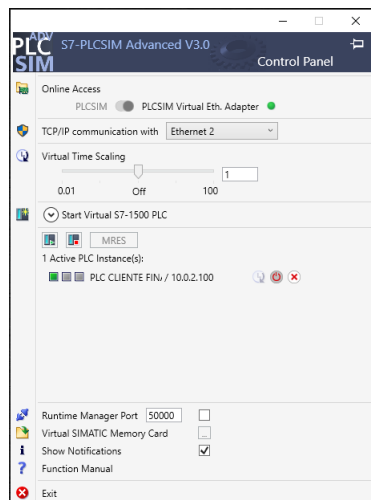


Figura 3.57: Configuración del Adaptador de red de PLCSIM.

3.5. Diseño de la Capa de Aplicación

3.5.1. Docker

Docker es una plataforma de software, desarrollada a partir de los contenedores de Linux, que permite la creación e implementación de aplicaciones de manera rápida y eficiente [Emaz and Una, 2017]. Esta plataforma se encargan de virtualizar un sistema operativo empaquetando el código y servicios a ejecutar en pequeñas unidades estandarizadas llamadas contenedores. Su implementación se basa en el uso de imágenes, las que incluyen bibliotecas, herramientas de sistema, código y tiempo de ejecución, es decir todo lo necesario para que el software se ejecute. Esto permite compartir una aplicación con todas las dependencias necesarias para su funcionamiento, otorgando la capacidad de distribuirla de manera rápida en los entornos que se necesite, así como tener total control sobre las versiones que se desarrollen [Red Hat, 2017].

El propósito de los contenedores es esta independencia, ya que poseen la capacidad de ejecutar varios procesos y aplicaciones por separado haciendo un mejor uso de su infraestructura instalada y permitiendo conservar la seguridad informática que tendrían varios sistemas separados [Emaz and Una, 2017].

Estas plataformas tienen ventaja sobre las máquinas virtuales dado que los contenedores generan archivos más ligeros puesto que no tienen la necesidad de virtualizar el hardware, lo que permite la ejecución de estos servicios en equipos de bajo recursos como: ordenadores de placas reducidas, routers entre otros.

3.5.2. Comunicación S7

La comunicación S7 es un protocolo propietario de SIMATIC desarrollado para la comunicación entre CPUs. Se utiliza principalmente para la migración y la conexión de PLC. Los datos se transmiten a través del bus PROFIBUS, lo que permite que la comunicación opere en altas velocidades. Esta comunicación se caracteriza porque sus enlaces de comunicación son dinámicos, es decir, la aplicación los crea y los libera de nuevo. Esta comunicación se encuentra asignada en la capa de aplicación del modelo OSI y utiliza el puerto 102 TCP para el envío y recepción de datos [Siemens A.G., 2021a].

En el presente proyecto se utilizará el paquete de Snap7 que permite establecer una comunicación S7 entre un CPU y un PLC para realizar la lectura de datos, se optó por esta manera de comunicación ya su implementación es sencilla y rápida.

3.5.3. Protocolo MQTT

MQTT son las siglas de MQ Telemetry Transport, se define como un protocolo de comunicación Machine to machine, de tipo message queue. Una de sus principales características es que este protocolo mantiene abierta la conexión permitiendo de esta manera reutilizarla para cada comunicación que se realice [Hillar., 2017].

El funcionamiento de este protocolo se basa en una arquitectura de publicador y suscriptor, en la cual los clientes se conectan a un servidor central denominado broker y este se encarga del envío y recepción de datos. Dicha comunicación se establece por el puerto TCP 1883 y es el broker quien mantiene un registro de todos los clientes que establezcan comunicación con él [Hunkeler et al., 2008].

Así mismo en el broker se disponen de tópicos que se encuentran organizados jerárquicamente, de tal manera que un cliente publique los mensajes en cada uno de los tópicos y otros dispositivos clientes puedan suscribirse a dichos tópicos y el broker será el encargado de hacerles llegar la información requerida [Soni and Makwana, 2017].

Para establecer comunicación con el broker el cliente debe enviar una petición CONNECT con la información necesaria, nombre de usuario y contraseña. A su vez el broker responde a esta petición con un mensaje de CONNACK, que indica si la petición fue aceptada o rechazada [Hillar., 2017].

Para enviar mensajes el cliente los envía a través de mensaje PUBLISH, los cuales contienen el tópico a publicar y el contenido del mismo, que en términos de esta comunicación se lo denomina PAYLOAD [Soni and Makwana, 2017].

Los mensajes de SUBSCRIBE y UNSUBSCRIBE son utilizados para la suscripción y de suscripción al broker, el cual responde a estas peticiones con mensajes de SUBACK y UNSUBACK [Soni and Makwana, 2017].

Cada cierto tiempo establecido se comprueba si la conexión se encuentra activa, esto se lo realiza a través de peticiones de tipo PINREQ y la misma se responde por parte del servidor con peticiones de tipo PINGRESP [[Soni and Makwana, 2017](#)].

Finalmente, el cliente se desconecta del broker a través de un mensaje de DISCONNECT. [[Hillar, 2017](#)]. La comunicación MQTT es utilizada en el presente proyecto para el envío de datos desde el dispositivo encargado de realizar la lectura del PLC hacia el broker MQTT a utilizar en este caso es Thingsboard.

3.5.4. Plataforma Thingsboard

Thingsboard es una plataforma open-source que sirve para el almacenamiento, procesamiento y visualización de datos. Funciona como un broker MQTT y es utilizada ampliamente para proyectos de IoT.

Tal y como se indicó en la sección anterior para establecer comunicación con el broker es necesario enviarle una petición de tipo CONNECT, en la que tiene que ir incluido el access token asignado al dispositivo, el cual es un código alfanumérico que genera el propio Thingsboard al momento de crear el dispositivo [[Thingsboard](#)].

Para realizar la suscripción, el contenido o payload debe ser enviado en una estructura de clave valor en formato JSON, en la cual la clave siempre será de tipo string mientras que el valor puede ser de cualquier tipo, mismo mensaje que debe ser enviado al tópico:

```
v1/devices/me/telemetry
```

3.5.5. Diseño de la aplicación

En la figura [3.58](#) se indica el diseño que se realizó para la capa de aplicación del proyecto. Como se puede observar se utiliza una arquitectura de contenedores en donde se tiene un contenedor por cada PLC que se encarga de la lectura de los datos de interés, dicha lectura la realiza por comunicación S7 y los datos recibidos se envían a través de MQTT a otro contenedor el cual ejecuta el servicio de Thingsboard. Posteriormente esta información es almacenada en una base de datos en Postgres SQL para su posterior visualización por el usuario final a través de una interfaz web.

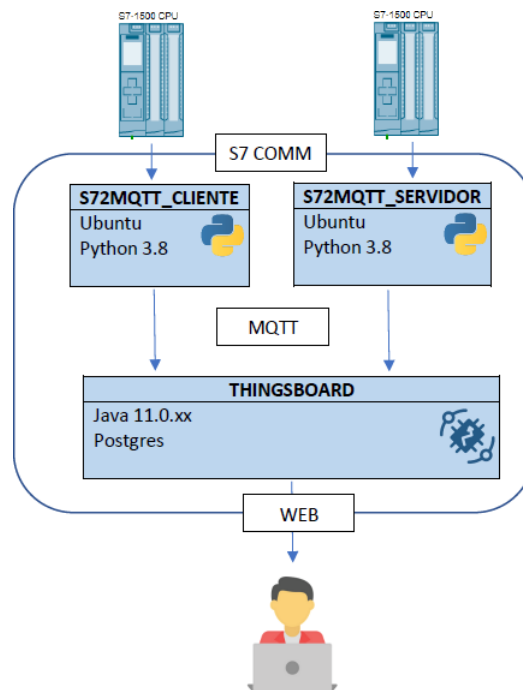


Figura 3.58: Diagrama de despliegue de la aplicación.

3.5.6. Puesta en marcha del servidor IoT

Una vez definido el diseño de la aplicación el primer paso es realizar la puesta en marcha del servicio de Thingsboard, para ello se va a hacer uso de la herramienta de compose de docker.

Según la documentación oficial de docker, compose es una herramienta que sirve para definir y ejecutar aplicaciones multi-contenedor. Su modo de funcionamiento se basa en definir en un archivo de formato YALM, las características con las que se desea que funcione los servicios de la aplicación, para que con un único comando se pueda crear y ejecutar todos los servicios que necesita la aplicación [Docker Inc., 2021].

Para realizar la puesta en marcha del servicio de Thingsboard la configuración del mismo debe ser colocada en un archivo de formato YAML. Afortunadamente esta configuración puede encontrarse en la web oficial por lo que para implementar el servicio se realizan los siguientes pasos:

1. Crear el archivo de composición con formato YAML para ello utilizamos el comando:

```
nano thingsboard-compose.yml
```

2. Colocar la configuración del archivo, la misma se consulta en la pagina oficial de Thingsboard, sección de instalación en la opción de Docker (Linux o MAC OS). La configuración que se extrajo se coloca a continuación:

```
version: ' 2.2'
services:
  mytb:
    restart: always
    image: "thingsboard/tb-postgres"
    ports:
      - "8080:9090"
      - "1883:1883"
    environment:
      TB_QUEUE_TYPE: in-memory
    volumes:
      - ~/.mytb-data:/data
      - ~/.mytb-logs:/var/log/thingsboard
```

3. Se corre construye y ejecuta el contenedor con la siguiente sentencia:

```
docker-compose -f thingsboard-compose.yml up
```

4. Finalmente, se comprueba el funcionamiento de la pagina web colocando en la barra de dirección de un navegador la dirección IP del servidor seguido de :8080 que indica el puerto al cual conectarse. En este caso sería: 10.0.2.101:8080. Si el servidor se ejecuta correctamente se debe visualizar la siguiente pagina web:

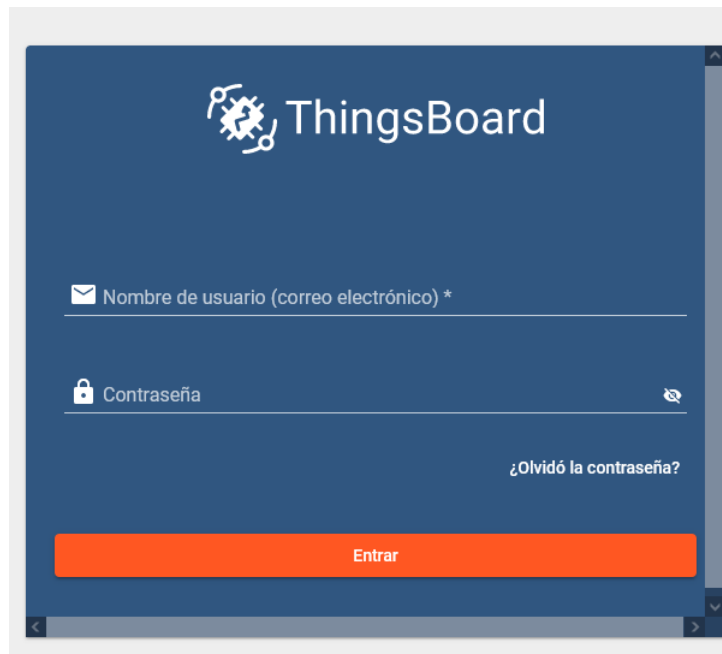


Figura 3.59: Login Thingsboard

3.5.7. Lectura y envío de datos hacía el servidor IoT

Para el envío de datos de los PLC al servidor MQTT se desarrolló un algoritmo en Python que funciona como intermediario entre los dispositivos y se encarga de realizar la lectura de los datos a través de una conexión S7 y subirla por MQTT, por lo que se dividió la estructura de este algoritmo en tres etapas las mismas pueden observarse en la figura 3.60.

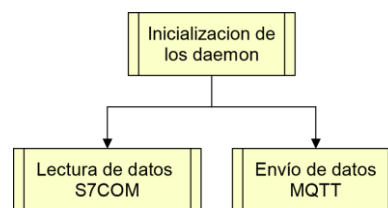


Figura 3.60: Etapas del algoritmo de Python

Dichas etapas se modelaron como procesos en paralelos que dependen de

un proceso de inicialización y comparten datos entre ellos, por lo que para el desarrollo del algoritmo se utilizó una arquitectura basada en hilos la que permite la programación de procesos en paralelo. Para el desarrollo de estas etapas se utilizaron las librerías que se indican en la tabla 3.8:

Librería	Función
Logging	Realiza la impresión del estado del algoritmo en consola
Snap7	Realiza la lectura de los datos del PLC
Threading	Creación de hilos que funcionan como daemons
Queue	Comunicación entre los daemons
Paho MQTT Client	Para la publicación de datos mediante MQTT
Json	Para el formato de los mensajes a publicar
Configparser	Lectura de la configuración cargada desde un archivo de texto plano

Tabla 3.8: Librerías de Python y su función.

La figura 3.61 presenta un diagrama de flujo de la primera etapa del algoritmo la que tiene como principal tarea la lectura de los datos de configuración necesarios para el correcto funcionamiento de la comunicación S7 y la comunicación MQTT, así como la creación e inicialización del daemon para estas comunicaciones.

En la figura 3.62 se presenta el diagrama de flujo para realizar la comunicación S7, en el cual primero se pregunta si la conexión esta establecida en caso de que no lo este trata de conectarse al PLC, al momento de que la conexión este establecida se procede a realizar la lectura de cada DB del PLC que se encuentre configurada en el archivo config.plc. Tras esto estas variables se almacenan en una variable de tipo diccionario cuya clave es el nombre de la variable y el valor es el dato leído del PLC, esta estructura de datos es enviada mediante una queue al daemon de MQTT para subirla al servidor.

En la figura 3.63 se presenta el diagrama de flujo para el envío de datos hacia el servidor MQTT, en el cual primero se establece la comunicación MQTT y cuando esta esta establecida se leen los datos que envió el daemon de lectura del PLC y los mismos se publican en el servidor MQTT.

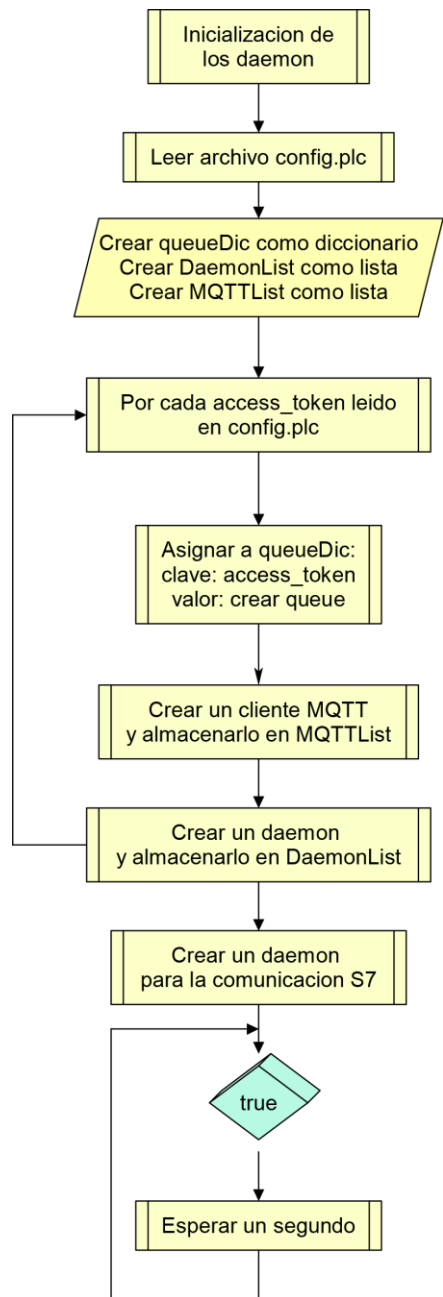


Figura 3.61: Diagrama de flujo para la inicialización de datos.

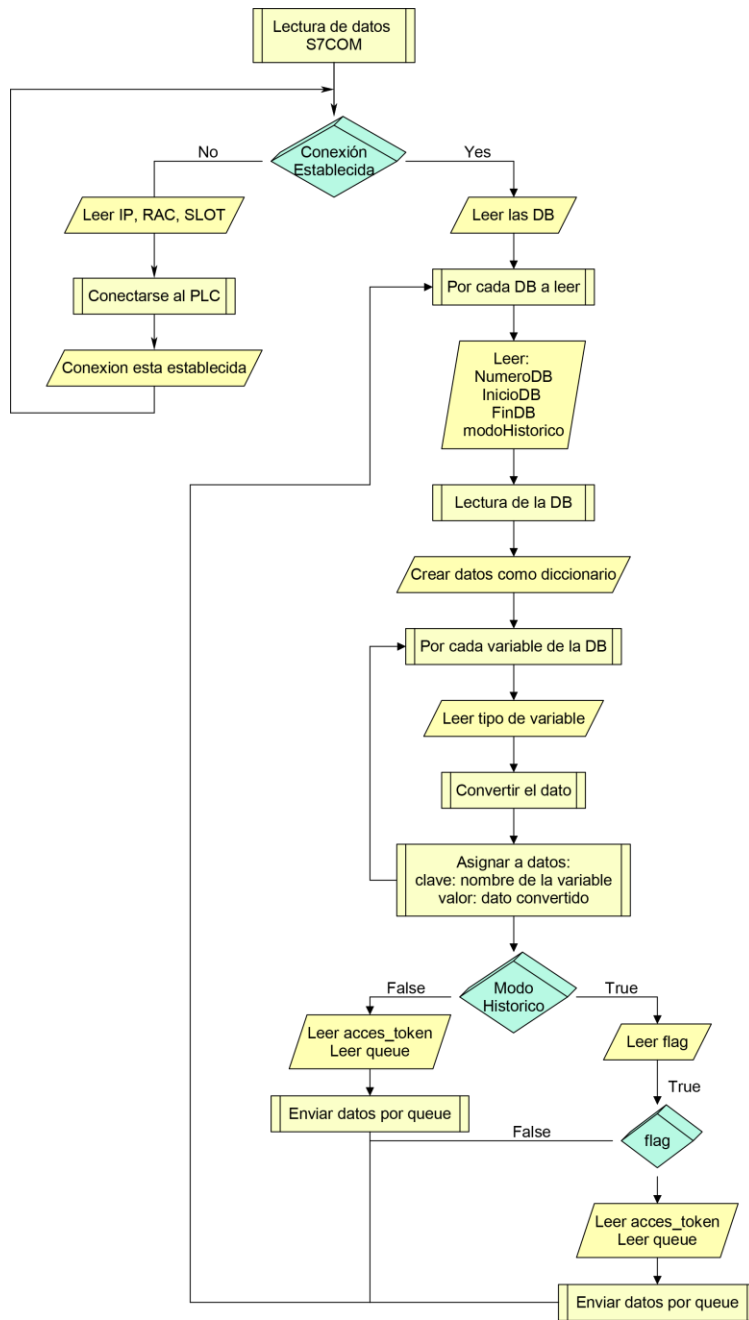


Figura 3.62: Diagrama de flujo comunicación S7.

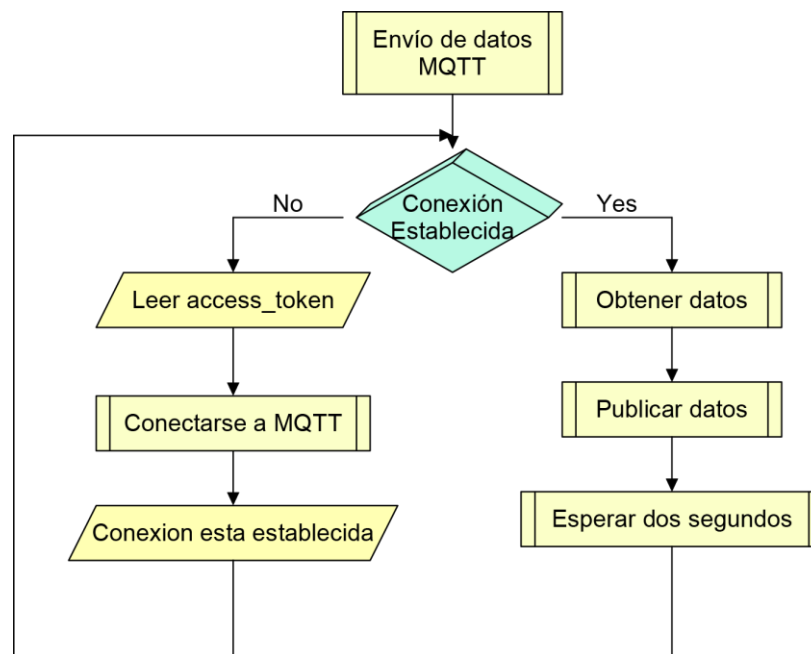


Figura 3.63: Diagrama de flujo comunicación MQTT.

3.5.8. Diseño del panel de Thingsboard

Para la visualización de los datos adquiridos de las diferentes plantas, se utilizó la plataforma Thingsboard, la misma que permite crear gadgets personalizados [Van Tam and Thien, 2020], o tener una variedad que ya están preestablecidos por la misma plataforma. Para la creación de los mismos se puede trabajar con librerías ya creadas, a estas se les conoce como **widgets**, como se puede observar en la figura 3.65, que dependiendo del tipo de utilidad que se le vaya a dar se elige con la que se va a trabajar. Para la creación de paneles se ha utilizado del paquete de widgets el que tiene por nombre **Navigation widgets**, como se aprecia en la figura 3.64



Figura 3.64: Librería utilizada para diseño de paneles.

En el presente trabajo se diseñó un panel que cuenta con varias opciones de visualización, las mismas que se aprecian en la tabla 3.9. La presentación para el usuario de estos paneles se puede observar en la figura 3.66.



Figura 3.65: Paquete de Widgets de Thingsboard.

Nombre Panel	Descripción
Estado de Plantas	Visualización del funcionamiento de las 2 plantas en tiempo real
Embotelladora	Visualiza datos más específicos de la planta embotelladora
Extractor de Aceite	Visualiza datos más específicos de la planta extractora de aceite
Históricos Embotelladora	Visualizar datos de funcionamiento de la planta cargados a la plataforma de diferentes fechas
Históricos Extractor de aceite	Visualizar datos de funcionamiento de la planta cargados a la plataforma de diferentes fechas

Tabla 3.9: Paneles de visualización en plataforma Thingsboard.

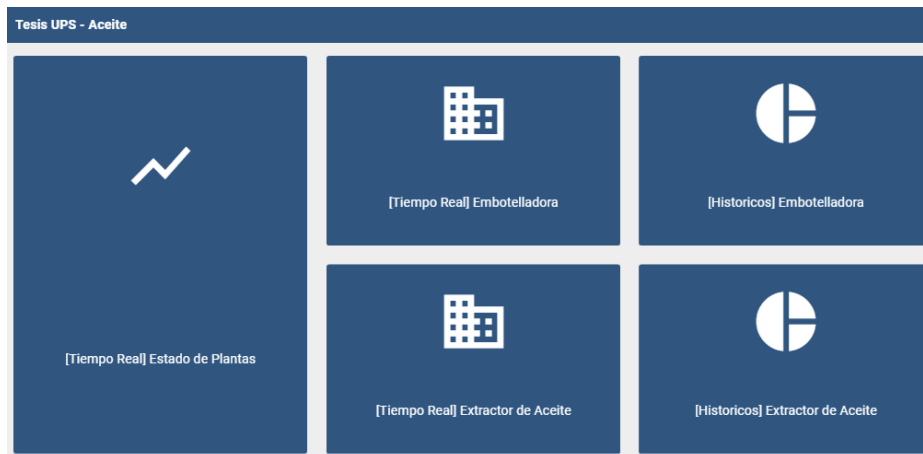


Figura 3.66: Paneles de visualización en plataforma Thingsboard.

Panel: Estado de plantas

Al dar click en este panel que se observa en la figura anterior, se obtiene el estado de la planta extractora de aceite y de la planta embotelladora. La presentación que se da en este panel es en forma booleana. En el caso de la planta embotelladora de aceite se observará que se encienden los leds dependiendo si la misma esta trabajando, si se ha llenado el tanque, si se está vaciando el tanque, si la planta está en pausa, si la planta está detenida. Por otro lado también se puede apreciar el estado de la planta extractora de aceite, donde se encenderán los leds si el tanque 1,2 y 3 están llenos o vacíos, también permite visualizar el nivel de cada tanque y la cantidad de cocos que se está procesando en esa celda de manufactura. Esto se puede apreciar de mejor manera en la figura [3.67](#).

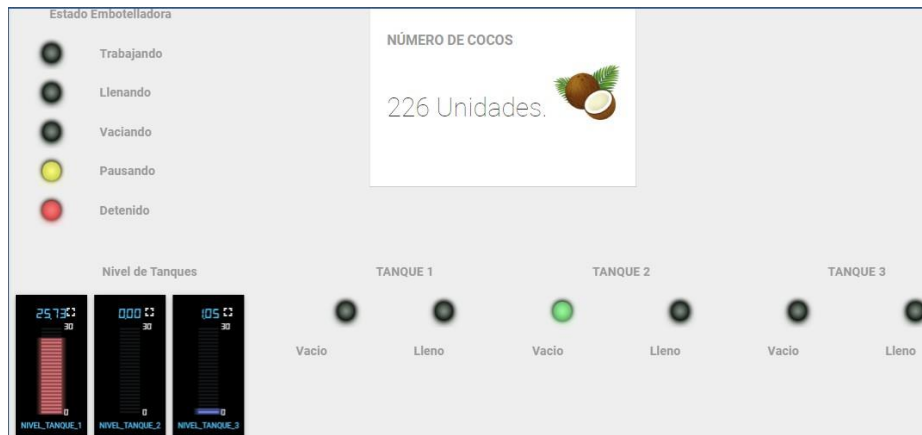


Figura 3.67: Panel:Estado de las Plantas.

Para poder relacionar los datos obtenidos de la planta simulada con el panel anteriormente presentado se siguen los siguientes pasos: Primero se ingresa al panel de navegación con el nombre **Estado de Plantas**, el cual contiene los botones de señalización y demás variables que se deseen visualizar, en la parte inferior derecha se selecciona el botón que dice **Modo Edición** como se observa en la figura 3.68.

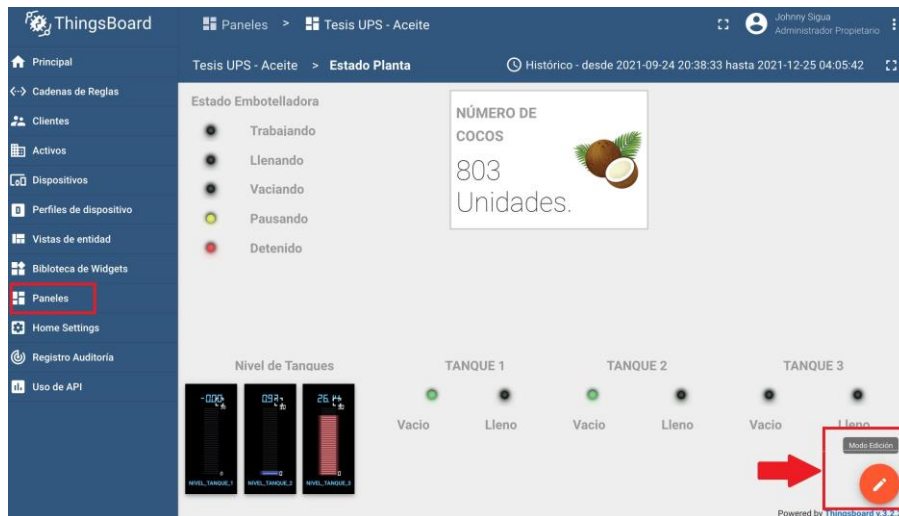


Figura 3.68: Boton para editar panel.

Una vez dentro de la opción anteriormente descrita, se da click en la opción de nombre **Alias de entidad**, esta opción permite ligar un set de datos a un alias, permitiendo leer los datos que han sido cargados de la planta a la plataforma ThingsBoard, esto se puede apreciar en la figura 3.69.

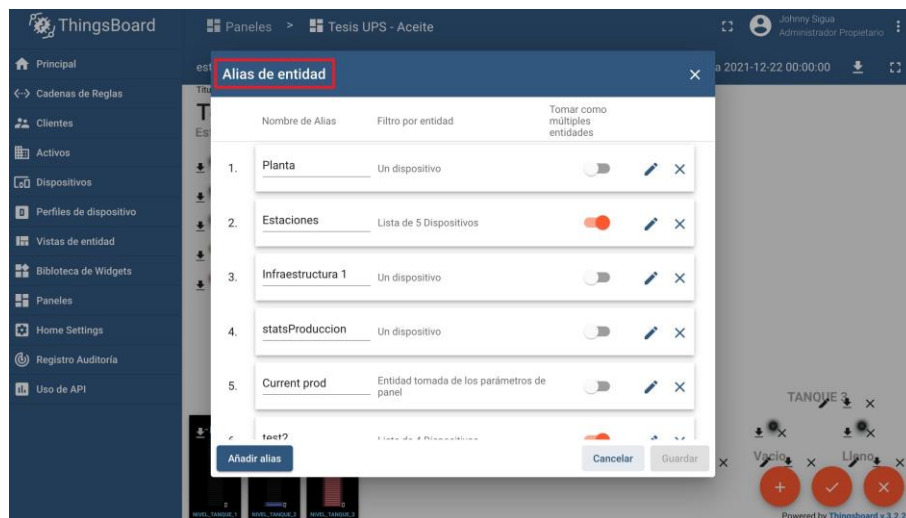


Figura 3.69: Creación de Alias de entidad en plataforma.

Panel: Extractor de Aceite

En este panel se puede observar a detalle el estado de la planta extractora de aceite, los datos a visualizar son: nivel del tanque 1, nivel del tanque 2, nivel del tanque 3, número de cocos, nivel del almacenador, energía consumida por la planta y tiempo de simulación transcurrido. En una tabla de datos también se puede observar el tiempo de trabajo de cada una de las estaciones: extractora, horno, perforadora y seccionadora, esto se puede visualizar en la figura 3.70.

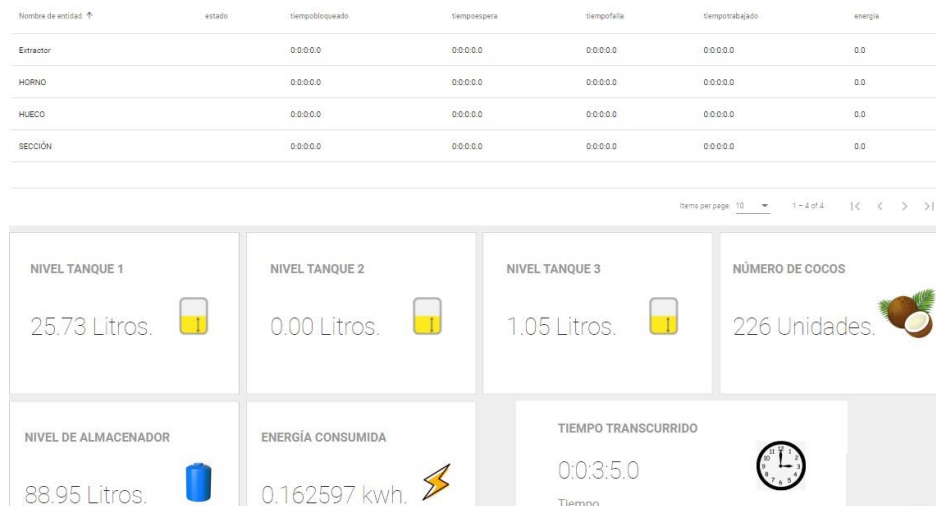


Figura 3.70: Panel:Extractor de Aceite.

Para cargar todas las variables que se deseen observar en la plataforma de ThingsBoard, se debe agregar una por una en la opción de **Dispositivos** ubicada en la parte lateral izquierda y se debe elegir la opción **Agregar dispositivo**, tal como se aprecia en la figura 3.71.

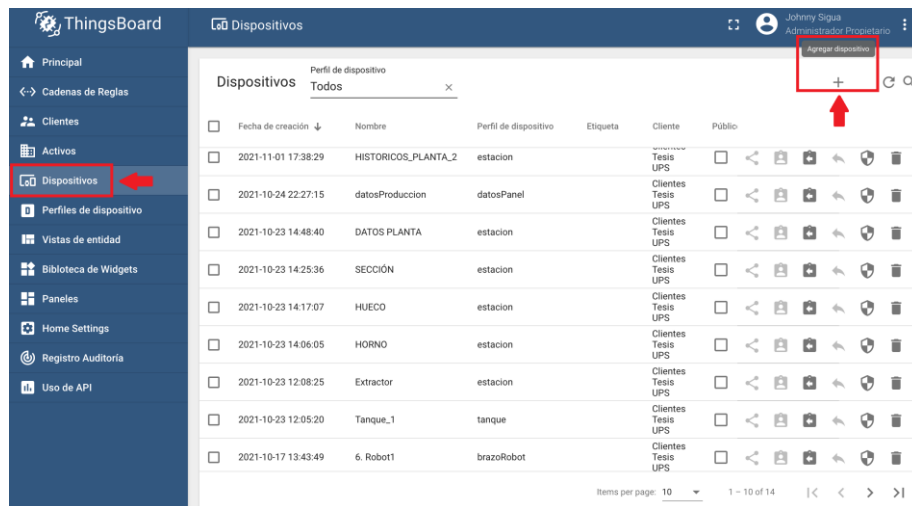


Figura 3.71: Agregación de variables a observar en Plataforma.

En la figura 3.72 se puede observar como se agrega todos los datos de la estación de nombre **HUECO** correspondientes a la planta extractora de aceite. Para poder relacionar los datos de la planta generados con el PLC que se encuentran en forma de **DataBlocks** a la plataforma Thingsboard, se debe elegir la opción **Copiar access token**, este código generado será pegado en el código desarrollado en Python, cuyo código contiene el tipo variable que se está utilizando, el ancho del **DataBlock**, el número del DataBlock utilizado en el PLC y el código Token generado por Thingsboard.

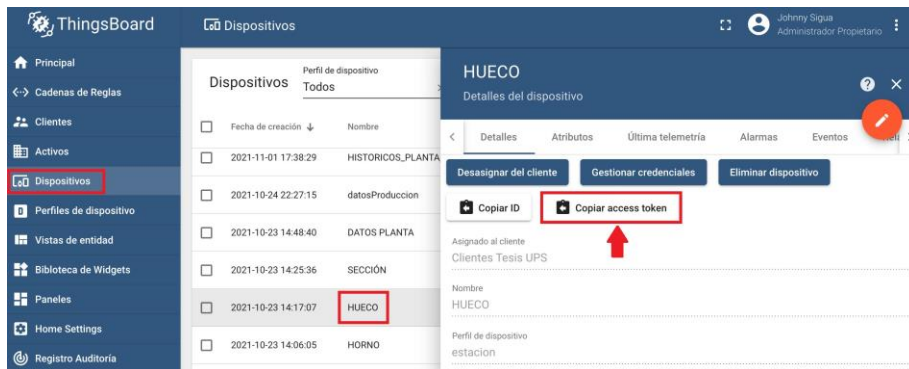


Figura 3.72: Generación de código token para visualización de variables.

En la siguiente figura 3.73 se puede apreciar el código generado en Python donde se pega el código token generado y se configura los datos del **DataBlock** proveniente del PLC.

```
[HUECO]
DBNumber = 2
DBInit = 0
DBEnd = 38
uplinkFlag = 0
ACCESS_TOKEN = access_token z5c0ZBc1bhRBDKvldsXB
Estado = String 0
tiempoTrabajado = Time 18
tiempoEspera = Time 22
tiempoBloqueado = Time 26
tiempoFalla = Time 30
ENERGIA = real 34
```

Figura 3.73: Código en Python para cargar variables de PLC a Thingsboard.

Panel: Embotelladora de Aceite

En este panel se puede observar a detalle el estado de la planta embotelladora, los datos a visualizar son: número de botellas a producir, número de botellas producidas, nivel de tanque, número de empaques utilizados, tiempo de simulación transcurrido. También se puede observar el tiempo de trabajo de cada una de las estaciones: Llenado de aceite, estación de colocación de tapas, estación de colocación de etiquetas, brazo robot 1, brazo robot 2. Esto se puede apreciar de mejor manera en la figura 3.74.



Figura 3.74: Panel: Embotelladora de Aceite.

Panel: Históricos Extractor de Aceite

En este panel se pueden visualizar datos del funcionamiento de la planta que han sido cargados en diferentes fechas, el usuario podrá ajustar la fecha que desee para observar los datos, esto se puede apreciar en la figura 3.75.

Selector de fecha 20 Oct - 28 Nov 2021		Intervalo Intervalo personaliza...		Número de pasos < Día >	
Timestamp	tiempotrabajo	energiasconsumida	litrosaceite_total	numero_de_cocos	
2021-11-07 12:55:05	0:0:0.0	0.1629732842445374	126.7119979858984	64	
2021-11-07 12:55:01	0:0:0.0	0.16164756449718475	125.60399427665547	64	
2021-11-07 12:54:59	0:0:0.0	0.16113072633743286	124.94999994824219	64	
2021-11-07 12:54:56	0:0:0.0	0.16070909798145204	124.52402007519531	64	
2021-11-07 12:54:54	0:0:0.0	0.16049908101558685	124.0199966430664	64	
2021-11-07 12:54:46	0:0:0.0	0.1592008024541168	122.07599493892378	64	
2021-11-07 12:54:44	0:0:0.0	0.15868043389652292	121.47000122070012	64	
2021-11-07 12:54:34	0:0:0.0	0.15649212896823883	118.83399963378906	62	
2021-11-07 12:54:30	0:0:0.0	0.15592758357524872	117.72799682617188	62	
2021-11-07 12:54:28	0:0:0.0	0.15511655807495117	117.24400329589844	61	

Figura 3.75: Panel: Históricos de planta extractora de aceite.

Al momento de dar click sobre cualquiera de los datos presentados en los históricos, al usuario se le presenta una gráfica tipo pastel donde se puede observar el tiempo de trabajo, tiempo de bloqueo, tiempo de espera y tiempo de falla de cada una de las estaciones de la planta. Esto se puede observar en la figura 3.76.

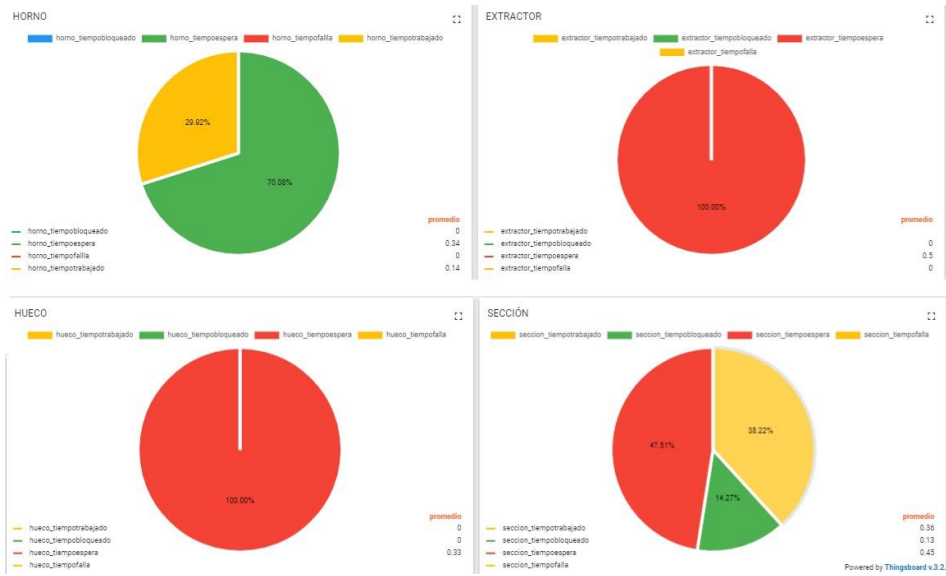


Figura 3.76: Gráficos de históricos de planta extractora de aceite.

Panel: Históricos de la embotelladora de aceite

En este panel se pueden visualizar datos del funcionamiento de la planta que han sido cargados en diferentes fechas, el usuario podrá ajustar la fecha que desee para observar los datos, esto se puede apreciar en la figura 3.77.

Timestamp	tiempotrabajo	energiaconsumida	litrosaceite	botellasutilizadas	empaquesutilizados
2021-11-03 08:25:41	0:0:5:40.0	0.9262465238571167	0	100	26
2021-11-02 18:40:42	0:0:8:59.0	0.9068132042884827	0	100	0
2021-11-02 18:36:04	0:0:8:2.0	0.9068132042884827	9449	0	100

Figura 3.77: Panel: Históricos de planta embotelladora de aceite.

Al momento de dar click sobre cualquiera de los datos presentados en los históricos, al usuario se le presenta una gráfica tipo pastel donde se puede observar el tiempo de trabajo, tiempo de bloqueo, tiempo de espera y tiempo de falla de cada una de las estaciones de la planta. Esto se puede observar en la figura 3.78.



Figura 3.78: Gráficos de históricos de planta embotelladora de aceite.

Capítulo 4

Análisis de Resultados, Conclusiones y Trabajos Futuros

En este capítulo se presentan los resultados tras la implementación del presente proyecto. En primera instancia se analiza el rendimiento de la capa de red. A continuación se analiza la capa de aplicación. Finalmente, en este mismo capítulo se reportan algunas conclusiones y trabajos futuros a implementarse.

4.1. Pruebas Realizadas

Las pruebas de funcionamiento se realizaron en el sector rural de la ciudad de Cuenca, parroquia El Valle, dado que este ambiente favorece para que las antenas de radio enlace cuenten con línea de vista, al ser un ambiente con pocos obstáculos. El sistema se implementó en dos casas separadas aproximadamente 200 metros, cubriendo esta distancia con las antenas correspondientes. El montaje de los equipos se realizó según se indica en la topología 3.47.

Inicialmente se realizaron pruebas de ping con el fin de comprobar la conectividad en todos los equipos de red antes de correr la simulación en Plant Simulation. Posteriormente se ejecutaron los contenedores instalados en el servidor de Linux con el fin de levantar los servicios necesarios para el monitoreo. Finalmente se encendieron las instancias de PLC creadas en el programa PLCSIM Advance de cada ordenador.

El sistema se corrió aproximadamente por una hora, que es el tiempo en que se tardan en producir mil botellas de aceite. En este tiempo se aprovechó para comprobar el rendimiento de cada capa implementada con el fin de determinar la factibilidad del sistema. El análisis realizado en las capas correspondientes se presenta a continuación:

4.1.1. Rendimiento de la Capa de Red.

Una vez que se pusieron en marcha los equipos se realizaron las pruebas de funcionamiento con el fin de evaluar el rendimiento de toda la plataforma, para ello se implementaron dos estaciones, una base y una estación, de acuerdo a lo presentado en el capítulo anterior, figura 3.47.

Las estaciones se distanciaron para propósito de evaluación aproximadamente 200 metros. La comunicación entre las dos estaciones se cubrió en su totalidad por el radio enlace, al cual se le aplicaron las siguientes configuraciones:

- Ancho de banda: 5GHz
- Potencia: 15 dbm

Se realizaron mediciones de velocidad en la antena colocada en la estación y se obtuvo los siguientes resultados.

- Velocidad de Subida (TX): 11.74 Mbps.
- Velocidad de Bajada (RX): 10.74 Mbps.

Estos resultados también se pueden evidenciar a través de la figura 4.1.

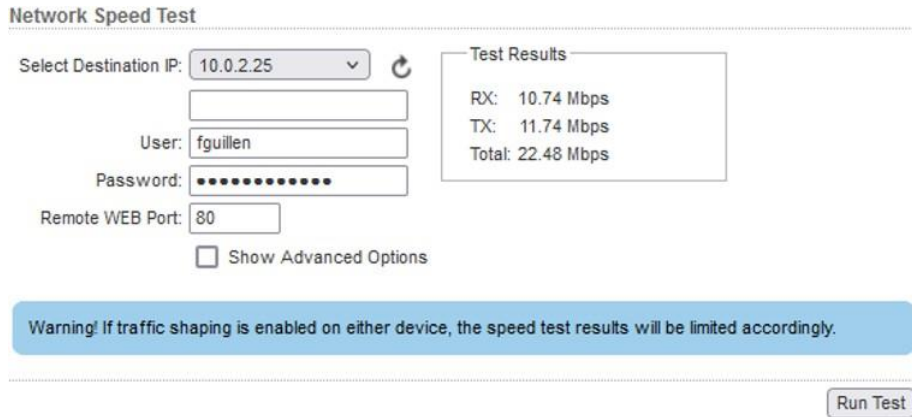


Figura 4.1: Prueba de velocidad radio enlace estación.

De acuerdo a estos resultados, se estima que las antenas entregaron un ancho de banda promedio de aproximadamente 10 Mbps. Asimismo, se procedió con la medición del ancho de banda que necesita cada PLC para el correcto envío de datos (véase la figura 4.2). Para ello, utilizando la herramienta torch presente en los routers Mikrotik se realizó un escaneo del ancho de banda que consume todas los paquetes enviados hacia el puerto TCP 102 que es el que utiliza la comunicación S7. En la tabla 4.1 se presenta un resumen con contiene el numero de variables enviadas por cada PLC y el ancho de banda que estas consumen.

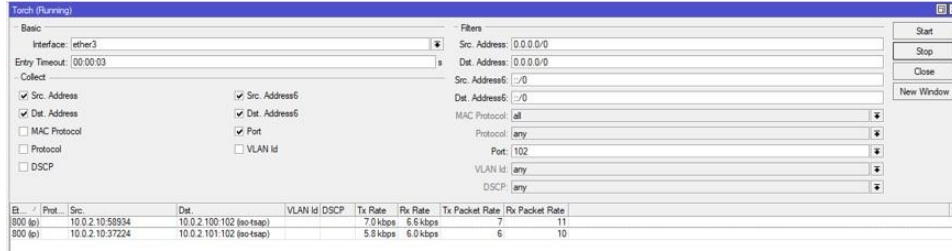


Figura 4.2: Ancho de banda consumido por la comunicación S7.

PLC	Nro. de datos	Ancho de banda
PLC Cliente	70 datos	6.6 kbps
PLC Servidor	57 datos	6 kbps

Tabla 4.1: Librerías de Python y su función.

Como se puede observar en la tabla 4.1 se envían un total de 127 datos de los PLCs teniendo un consumo total de 12.6 Kbps. Como se mencionó anteriormente, el radio enlace entrega un ancho de banda de 10 Mbps, es decir que el ancho de banda que consume la comunicación S7 entre los PLC y el servidor consume aproximadamente el 0.1 % del ancho de banda disponible; esto indica que el radio enlace implementado entrega la capacidad suficiente para el correcto funcionamiento del proyecto.

4.1.2. Rendimiento Capa de Aplicación

Para el análisis de la capa de aplicación se utilizaron las herramientas de monitoreo presentes en la paquetería de docker engine, esto con el fin de monitorizar el coste de recursos que representa la implementación de la aplicación realizada y de esta manera obtener los requerimientos mínimos.

Utilizando el comando:

```
docker image ls
```

es posible listar las imágenes de docker instaladas y el dato del peso. La figura 4.3 muestra la ejecución de dicho comando y su respuesta, en el mismo se puede notar el peso de cada imagen utilizada en el proyecto.

```
xissco@server1:~/Docker$ docker image ls | grep 'thingsboard\|s72mqtt'
s72mqtt          latest          e72c33926706   13 days ago    542MB
ivanmorenoj/thingsboard v3.2.2         acaee97e846a   7 months ago   834MB
thingsboard/tb-postgres latest         51aeaa041fde   8 months ago   1.69GB
xissco@server1:~/Docker$
```

Figura 4.3: Respuesta del comando docker image ls.

La tabla 4.2 presenta una tabla resumen de los datos necesarios para el análisis.

Imagen	Función	Peso
s72mqtt	Lectura de datos de los PLC	542 MB
Thingsboard	Aplicación Thingsboard	843 MB
Postgres	Servicio de base de datos	1.69 GB

Tabla 4.2: Peso de los contenedores instalados.

En la implementación de la aplicación se decidió por utilizar una imagen del postgres independiente a la imagen de thingsboard con el fin de poder acceder al servicio de base de datos el cual es el encargado de almacenar los datos leídos del PLC y toda la configuración del servicio de Thingsboard, por lo que es importante respaldar esta información y a ello se debe el peso de la misma.

Se recomienda utilizar un volumen de alta capacidad de almacenamiento como un Disco Duro para alojar los datos que entrega esta imagen así como realizar respaldos de la misma.

Las imágenes del servicio de S72MQTT y Thingsboard consumen un total de 1.385 GB siendo imágenes ligeras las cuales se podrían implementar en un servicio en la nube que soportaría esta capacidad, incluso podrían ser instaladas en un cliente ligero como una Raspberry o similares siempre y cuando se tenga en cuenta la recomendación dada en el punto anterior de colocar la información de la base de datos en una unidad aparte.

Adicionalmente se utilizó el comando:

```
docker stats
```

El cual indica los recursos de CPU y memoria que consume los servicios obteniendo los datos que se presentan en la figura 4.4

```

CONTAINER ID   NAME                                CPU %     MEM USAGE / LIMIT     MEM %     NET I/O       BLOCK I/O     PIDS
e165dc515fed   docker_s72mqtt-servidor_1         1.66%    18.38MiB / 7.431GiB   0.24%    0B / 0B       0B / 0B       14
cab9a718f8ed   docker_s72mqtt-cliente_1         2.05%    18.34MiB / 7.431GiB   0.24%    0B / 0B       0B / 0B       16
8b55ac87d9e2   docker_thingsboard_1             3.08%    729.3MiB / 7.431GiB  9.58%    9.69MB / 23.8MB  26.9kB / 94.2kB 263
7c9cf727689   docker_db-postgres_1             0.85%    58.1MiB / 7.431GiB   0.76%    16.7MB / 6.46MB  426kB / 108MB   23
8aa4b173f347   8aa4b173f347_traefik            0.09%    22.29MiB / 7.431GiB  0.29%    14.2MB / 14.6MB  66.3MB / 0B    18
    
```

Figura 4.4: Respuesta del comando docker stats.

La tabla 4.3 presenta una tabla resumen de los datos necesarios para el análisis.

Imagen	CPU	Memoria
s72mqtt_cliente	2.05 %	18.34 MiB
s72mqtt_servidor	1.66 %	18.38 MiB
Thingsboard	3.08 %	729.3 MiB
Postgres	0.85 %	58.1 MiB

Tabla 4.3: Recursos consumidos por los contenedores.

Como se puede observar en la tabla de 4.3 la aplicación consume pocos recursos de CPU y memoria lo que permitiría que la misma pueda correr en un ordenador de una capacidad mínima de 2GB de memoria RAM. Se debe tener en cuenta que mientras se realizaban las pruebas el consumo de CPU del servicio de Thingsboard se elevó al 40 % mientras se cargaba la interfaz web, sin embargo este pico de consumo fue transitorio y tras tener cargados todos los widgets el consumo se mantuvo en lo indicado en la tabla 4.3.

4.2. Rendimiento de la Plataforma ThingsBoard

Al arrancar las dos plantas activando el botón de **ON** en los diferentes paneles de control ubicados en cada una de las plantas como se ilustra en las figuras 3.13 y 3.38, empieza todo el proceso a ejecutarse, dentro de la plataforma de ThingsBoard se observó que los datos visualizados son en tiempo real, para esto se puso en línea el PLC mediante el programa **Tia Portal V16**, donde los cambios de las variables eran inmediatos en ambas plataformas. Para poder observar los datos de los históricos que generan las plantas, hay que presionar los botones de **UP** que se encuentran ubicados en los paneles de control anteriormente mencionados. En la figura 4.5 se puede observar una prueba realizada a la planta extractora de aceite cuyo tiempo de prueba fue de 55 minutos aproximadamente. En este tiempo se

puede observar que la planta produce una cantidad de 1170.3 litros de aceite, utilizando una cantidad total de 803 cocos, con un total de 1.740 KWh.

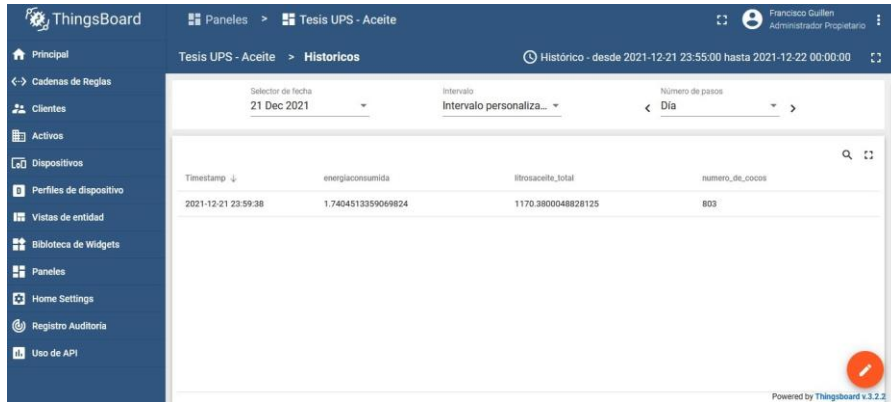


Figura 4.5: Históricos de planta extractora de aceite.

De manera más detallada se puede observar en la figura 4.6 que se presentan los datos de trabajo de cada una de las estaciones pertenecientes a la planta extractora de aceite, aquí se aprecia el tiempo de trabajo, el tiempo que estas permanecen en espera, el tiempo de falla que ha tenido cada una y la energía que consume cada estación.



Figura 4.6: Gráficos en pastel de planta extractora de aceite.

En la figura 4.7 se puede observar una prueba realizada a la planta embotelladora cuyo tiempo de prueba es de 52 minutos aproximadamente. En este tiempo se puede observar que la planta produce una cantidad de 1000 botellas, también se produce una cantidad de 251 empaquetados, la energía consumida por la misma durante este tiempo es de 8,601 KWh.

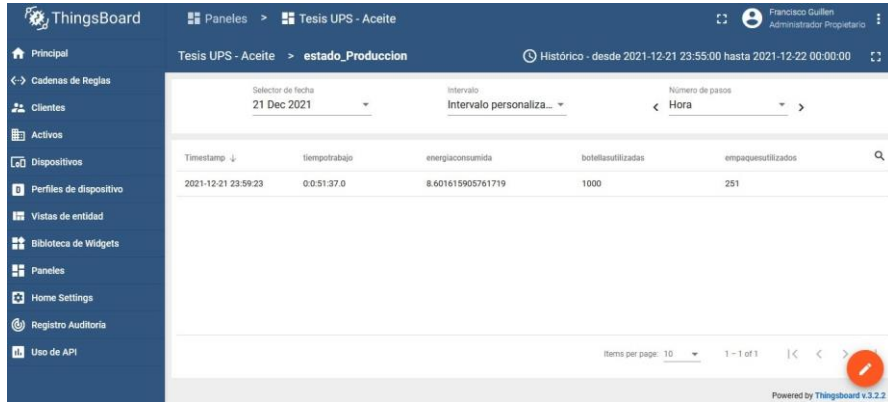


Figura 4.7: Históricos de planta embotelladora.

De manera más detallada se puede observar en la figura 4.8 los datos de trabajo de cada una de las estaciones pertenecientes a la planta embotelladora. Aquí se aprecia el tiempo de trabajo, el tiempo que estas permanecen en espera, el tiempo de falla que ha tenido cada una y la energía que consume cada estación.

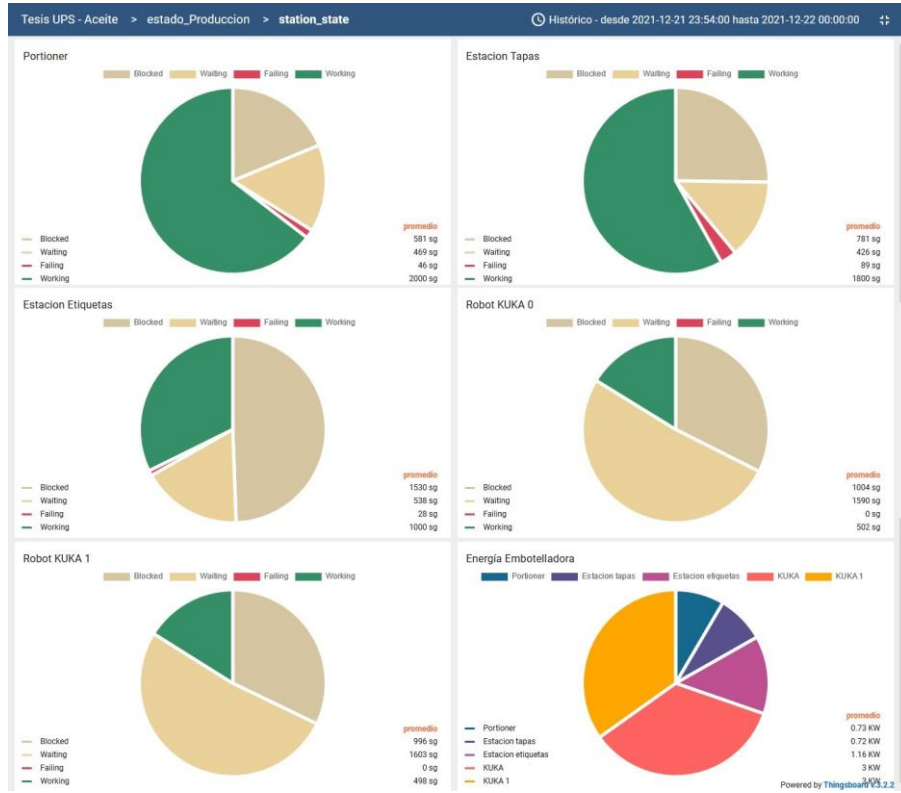


Figura 4.8: Gráficos en pastel de planta embotelladora.

4.3. Trabajos Futuros

Si se desea continuar con el presente trabajo, el siguiente paso sería ejecutarlo en una planta real, en vista de que la aplicación se encuentra desarrollada en su totalidad y su arquitectura es simple de aplicar, el tiempo de implementación sería corto, dado que solo se tendría que configurar el envío de datos de los Bloques de Datos que provienen de cada PLC.

Una mejora que se le podría realizar es el microservicio desarrollado en el lenguaje de programación Python, ya que se ejecuta sobre una imagen con el sistema operativo Ubuntu 20.0.4, el mismo podría mudarse al sistema operativo Alpine, que es mucho más ligero permitiendo ejecutarlo en dispositivos con recursos limitados.

La infraestructura desarrollada puede llegar a simplificarse al ejecutar el microservicio de Python en un router Mikrotik, dado que en base a la última actualización de su sistema operativo, permite la ejecución de contenedores, sin embargo tamaño del microservicio debe ser más ligero para poder ser ejecutado.

4.4. Conclusiones

Dentro de los resultados obtenidos en el presente proyecto, al establecer una distancia aproximada de 200m entre la antena base y la estación, la velocidad de subida de datos tiene un ancho de banda de 11,74 Mbps, mientras que la velocidad de los datos de bajada es de 10,74 Mbps, esto significa que el ancho de banda es más que suficiente si se desea enviar información entre PLCs, dado que al momento de realizar las pruebas de consumo de datos en las comunicaciones de los mismos, el PLC cliente utiliza un ancho de banda de 6,6 kbps al enviar 70 datos, mientras que el PLC servidor envía 57 datos con un ancho de banda de 6 kbps. Al analizar estos resultados se puede confirmar que las antenas utilizadas son muy eficientes para lograr una comunicación a larga distancia.

En el desarrollo del proyecto se ha hecho uso de diferentes protocolos de comunicación, llegando a ser cada uno de ellos muy eficiente al momento de probar su funcionamiento. Para la lectura de datos de los PLCs con el servidor, se utilizó el protocolo de comunicación s7, el mismo que permite un bajo consumo de ancho de banda para el envío y recepción de datos. En los resultados obtenidos se puede estimar que el ancho de banda que consumen los dos PLCs con el servidor es de un 0.1%, esto significa que si se desea a futuro conectar más PLCs para controlar diferentes plantas y los mismos necesitan comunicarse entre sí, se puede disponer de la misma infraestructura de comunicación utilizada, sin realizar ningún cambio en la antena base y la estación, esto ayuda de gran manera a los técnicos que trabajan en el campo, dado que no se tiene que invertir más presupuesto en

hardware para la comunicación inalámbrica entre diferentes dispositivos.

Otro de los protocolos de comunicación que se probó en este trabajo fue el MQTT, el mismo que se utilizó para subir los datos obtenidos de los PLCs por el servidor a la plataforma Thingsboard, dando como resultado un buen rendimiento. En un análisis obtenido se pudo notar que los mismos se cargan con un tiempo máximo de demora de 2 segundos, esto se puede observar de mejor manera en el diagrama de flujo expuesto en la figura 3.63.

Para la conexión a la página desarrollada en la plataforma Thingsboard, se utilizó el protocolo de comunicación http, el cual demostró dar un buen rendimiento al momento de observar los datos obtenidos por las dos plantas virtuales en tiempo real. Independientemente del computador que se utilizara, la visualización y funcionamiento fue el adecuado al momento de registrar los datos de cada estación.

Al hacer un análisis de la cantidad de consumo de memoria RAM de los servidores empleados en el presente trabajo, se pudo observar que es muy bajo, estos valores se los aprecia en la figura 4.3, donde se puede concluir que este servidor puede ejecutarse en un equipo que no requiera muchas características técnicas como procesador y memoria RAM, dando como resultado un gran beneficio al momento de ejecutarlo en la industria, puesto que no se necesita invertir mucho presupuesto para levantar un servidor utilizado para comunicación entre diferentes dispositivos.

En el desarrollo de este trabajo no se dispuso de plantas físicas, pero las mismas fueron ejecutadas mediante el software Tecnomatrix Plant Simulation, el cual resultó ser un software amigable para el usuario, dado que se puede obtener muchos datos del desempeño de la planta como tiempo de trabajo, tiempo de fallo, energía consumida por cada estación empleada, etc. Estos datos son de suma importancia para los usuarios, ya que se toman como referencia para tener un estimado de cómo sería la implementación de la misma. Además estos datos ayudan al usuario a tomar una decisión acertada de los elementos que se vayan a comprar, tipo de mantenimiento a emplear, costo-beneficio , etc.

El desarrollo del Virtual Commissioning abre un campo muy amplio para investigadores, estudiantes y empresas, que deseen realizar un estudio previo del comportamiento de la planta industrial antes de implementarla, se puede también tener un conocimiento más elaborado del desempeño de la

misma. Permite también obtener un modelado de control del sistema, una programación de control, un modelado de dispositivos físicos (geométricos y cinemáticos), un modelado de dispositivos lógicos y construcción de plantas y dispositivos virtuales.

Con la comunicación realizada en este proyecto, al haber empleado un enlace de corta distancia, se pudo constatar que las antenas utilizadas no necesariamente deben ser de tipo industrial, dado que el ancho de banda que estas poseen cubren en su totalidad la comunicación con los diferentes PLCs que se emplearon.

Para la visualización de información que se pudo extraer de las diferentes plantas, se utilizó la plataforma thingsboard, en donde para que estos datos sean mas amigables con el usuario, se colocó diferentes tipos de dashboards, cuya configuración no requiere mucha complejidad y al ser un software libre se puede llegar a personalizar el diseño en caso de así desearlo. Actualmente esta plataforma está en desarrollo por lo que no se encuentra mucha información en la red, pero abre las puertas a que pueda ser utilizada ampliamente en el campo de la automatización y control.

La propuesta de trabajo realizada fue desarrollar la comunicación de dos plantas mediante PLCs utilizando como medio de comunicación antenas inalámbricas, para esto, en su mayor parte se empleó software libre tanto para la comunicación de los servidores, como para la subida de los datos de los PLCs. Al emplear este tipo de recurso, permite a desarrolladores relacionarse mucho más con la industria 4.0 sin tener que hacer uso de un gran presupuesto para la implementación de la misma.

Bibliografía

- M. M. Abdulwahid and N. B. M. Wasel. Design and implementation of water level tank model by using scada system. *Informatica: Journal of Applied Machines Electrical Electronics Computer Science and Communication Systems*, 1(1):63–69, 2020.
- L. O. Aghenta and T. Iqbal. Design and implementation of a low-cost, open source iot-based scada system using esp32 with oled, thingsboard and mqtt protocol. *AIMS Electronics and Electrical Engineering*, 4(1):57–86, 2019.
- S. Bangsow. *Tecnomatix plant simulation*. Springer, 2020.
- H. Berger. *Automating with SIMATIC: controllers, software, programming, data*. John Wiley & Sons, 2012.
- A. Canossa. Interview with nicholas francis and thomas hagen from unity technologies. In *Game Analytics*, pages 137–142. Springer, 2013.
- R. G. Carvajal, E. E. T. S. de Ingenieros, M. D. D. REBOLLO, and P. Superior. Tecnologías inalámbricas. *Segur. y Medio Ambient*, pages 10–21, 2012.
- V. A. Chalán Padilla. Desarrollo de un controlador óptimo lqr utilizando herramientas iot para un sistema de presión constante controlado remotamente. Master’s thesis, 2020.
- Ó. Cortés-López. Estudio teórico experimental con simit. 2017.
- Docker Inc. Overview of docker compose, Nov 2021. URL <https://docs.docker.com/compose/>.
- R. Domínguez Cartolín. Diseño de un sistema de control distribuido con control pid difuso para control de nivel de aguas para una planta de distribución de servicios de agua potable. 2018.

- F. A. Dueñas Llanos and N. V. Chalcán Aguayo. Diseño, implementación y monitoreo de una red inalámbrica, entre dos plantas de didáctica industrial, usando antenas ubiquiti networks nanostation y un scada bajo labview. B.S. thesis, Universidad Politécnica Salesiana, 2017.
- A. Emaz and P.-R. Una. Contenedores de docker | ¿que es docker? | aws, 2017. URL <https://aws.amazon.com/es/docker/>.
- M. Escalante. *Conceptos fundamentales de Mikrotik RouterOS*, volume 1. Academy Xperts, 6 edition, 2019.
- M. Escalante. *UAS-AIR-CPM Configuración de enlace Punto a Punto y Multipunto*, volume 1. Academy Xperts, 1 edition, 2021.
- D. A. Guerra-Zubiaga, A. Bondar, G. Escobedo, and A. Schumacher. Digital twin in a manufacturing integrated system: Siemens tia and plm case study. In *ASME International Mechanical Engineering Congress and Exposition*, volume 59384, page V02BT02A008. American Society of Mechanical Engineers, 2019.
- L. V. Guerrero, V. V. López, and J. E. Mejía. Virtual commissioning with process simulation (tecnomatix). *Computer-Aided Design and Applications*, 11(sup1):S11–S19, 2014.
- G. C. Hillar. *MQTT essentials: a lightweight IoT protocol: the preferred IoT publish-subscribe lightweight messaging protocol*. Packt Publishing, 2017.
- U. Hunkeler, H. L. Truong, and A. Stanford-Clark. Mqtt-s — a publish/subscribe protocol for wireless sensor networks. In *2008 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pages 791–798, 2008. doi: 10.1109/COMSWA.2008.4554519.
- S. T. Mortensen and O. Madsen. A virtual commissioning learning platform. *Procedia Manufacturing*, 23:93–98, 2018.
- P. Novak, P. Kadera, and M. Wimmer. Model-based engineering and virtual commissioning of cyber-physical manufacturing systems — transportation system case study. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, 2017. doi: 10.1109/ETFA.2017.8247743.

- N. O. Osinde, J. B. Byiringiro, M. M. Gichane, and H. Smajic. Process modelling of geothermal drilling system using digital twin for real-time monitoring and control. *Designs*, 3(3):45, 2019.
- D. M. Ospina Eslava and A. F. Santiago. Virtual commissioning of robotic cell using cloud-based technologies and advanced visualization system, 2021.
- Red Hat, 2017. URL <https://www.redhat.com/es/topics/containers/what-is-docker>.
- A. F. Sarmiento Paute. Adquisición y visualización de parámetros eléctricos de un motor trifásico mediante el uso de un módulo de internet industrial de las cosas. B.S. thesis, Universidad del Azuay, 2020.
- S. Satorres Mira. *Automatización de un paletizador industrial. Programación con TIA Portal y virtualización con Factory IO*. PhD thesis, Universitat Politècnica de València, 2021.
- Siemens A.G. *S7-PLCSIM Advanced, Function Manual*. Siemens A.G., 1 edition, 2016. URL https://cache.industry.siemens.com/dl/files/153/109739153/att_895955/v1/s7-plcsim_advanced_function_manual_en-US_en-US.pdf.
- Siemens A.G. *SIMATIC S1-1500, ET 200MP, ET 200SP, ET200AL, ET200pro*, volume 1. Siemens, 1 edition, 2021a.
- Siemens A.G. Lista de precios productos industriales electricos. B.S. catalog, 2021b.
- K. Smerpitak, W. Jearnpanitpong, A. Julsereewong, and T. Thepmanee. Multi-plc control system based on wireless bridge/base stations for work-in-process movements in corrugated box manufacturer. In *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, pages 1175–1180, 2018.
- D. R. D. Sobrino, R. Ružarovský, R. Holubek, and K. Velíšek. Into the early steps of virtual commissioning in tecnomatix plant simulation using s7-plcsim advanced and step 7 tia portal. In *MATEC Web of Conferences*, volume 299, page 02005. EDP Sciences, 2019.
- D. Soni and A. Makwana. A survey on mqtt: a protocol of internet of things (iot). In *International Conference On Telecommunication, Power Analysis And Computing Techniques (ICTPACT-2017)*, volume 20, 2017.

- Thingsboard. Mqtt device api reference. URL <https://thingsboard.io/docs/reference/mqtt-api/>.
- N. Van Tam and N. D. Thien. using open-source platform thingsboard. *Journal of Science-Quy Nhon University*, 14(3):89–97, 2020.
- D. Villacañas Sánchez. Integración de plc con modelos virtuales de plantas industriales. B.S. thesis, 2019.