



**Diseño de un prototipo de arquitectura IoT para el monitoreo y control de una estación de nivel mediante un autómatas simulado.**

Semanate Esquivel, Luis Clinton

Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Maestría en Electrónica y Automatización Mención Redes Industriales

Trabajo de Titulación, previo a la obtención del título de Magister en Electrónica y  
Automatización Mención Redes Industriales

Ing. Pilatásig Panchi, Marco, Mgtr.

19 de abril de 2023

Latacunga

## Reporte de verificación de contenido

TESIS\_CORREGIDA\_FINAL.pdf

Scanned on: 15:55 December 12, 2022 UTC



Overall Similarity Score



Results Found



Total Words in Text

Identical Words	61
Words with Minor Changes	71
Paraphrased Words	142
Omitted Words	0

MARCO  
ANTONIO  
PILATASIG  
PANCHI

Firmado digitalmente por MARCO ANTONIO PILATASIG PANCHI



Website | Education | Businesses

**Ing. Pilatasig Panchi, Marco Antonio, Mgtr.**

**Director**

**C.C.: 0502375876**



## Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

### Centro de Posgrados

#### Certificación

Certifico que el trabajo de titulación: “**Diseño de un prototipo de arquitectura IoT para el monitoreo y control de una estación de nivel mediante un autómatas simulado**” fue realizado por el señor **Semanate Esquivel, Luis Clinton**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 19 de abril de 2023

MARCO  
ANTONIO  
PILATASIG  
PANCHI

Firmado digitalmente por  
MARCO ANTONIO  
PILATASIG PANCHI

.....

**Ing. Pilatagsig Panchi, Marco Antonio, Mgtr.**

**Director**

**C.C.: 0502375876**



**Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología**

**Centro de Posgrados**

**Responsabilidad de Autoría**

Yo **Semanate Esquivel, Luis Clinton**, con cédula de ciudadanía n° 1716439763, declaro que el contenido, ideas y criterios del trabajo de titulación: “ **Diseño de un prototipo de arquitectura IoT para el monitoreo y control de una estación de nivel mediante un autómatas simulado**” es de mí autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 19 de abril de 2023

.....  
**Semanate Esquivel, Luis Clinton**

**C.C.: 1716439763**



## Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

### Centro de Posgrados

### Autorización de Publicación

Yo **Semanate Esquivel, Luis Clinton** con cédula de ciudadanía n° 1716439763 autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: “**Diseño de un prototipo de arquitectura IoT para el monitoreo y control de una estación de nivel mediante un autómata simulado**” en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 19 de abril de 2023

.....  
**Semanate Esquivel, Luis Clinton**

**C.C.: 1716439763**

## **DEDICATORIA**

El presente proyecto de tesis está dedicado a Dios, a mis padres, hermanas y familia en general, ya que gracias a sus consejos y apoyo constante me ayudaron a seguir adelante y cumplir mis objetivos. A mis amigos y todas aquellas personas que confiaron en mi contribuyendo a la obtención de este logro.

Luis Clinton Semanate Esquivel

## **AGRADECIMIENTO**

Agradezco a Dios por bendecirme con este regalo llamada vida, a mis padres, Susana y Marcelo, quienes siempre me han dado su apoyo incondicional. A mis hermanas, Mayuri y Marcela, que siempre me dieron su cariño, animó y motivación para la culminación del presente proyecto.

Luis Clinton Semanate Esquivel

**ÍNDICE DE CONTENIDO**

<b>Cáratula .....</b>	<b>1</b>
<b>Reporte de verificación de contenido .....</b>	<b>2</b>
<b>Certificación .....</b>	<b>3</b>
<b>Responsabilidad de Autoría.....</b>	<b>4</b>
<b>Autorización de Publicación .....</b>	<b>5</b>
<b>Dedicatoria .....</b>	<b>6</b>
<b>Agradecimiento.....</b>	<b>7</b>
<b>Índice de contenido .....</b>	<b>8</b>
<b>Índice de figuras .....</b>	<b>12</b>
<b>Índice de tablas.....</b>	<b>16</b>
<b>Tabla de acrónimos .....</b>	<b>17</b>
<b>Resumen .....</b>	<b>18</b>
<b>Abstract .....</b>	<b>19</b>
<b>Capítulo I: Planteamiento del problema de investigación .....</b>	<b>20</b>
<b>Introducción.....</b>	<b>20</b>
<b>Problemas Principales .....</b>	<b>21</b>
<b>Justificación .....</b>	<b>22</b>
<b>Hipótesis de investigación .....</b>	<b>23</b>

<b>Categorización de las variables de investigación.....</b>	<b>23</b>
<i>Variable Independiente .....</i>	<i>23</i>
<i>Variable Dependiente.....</i>	<i>23</i>
<b>Operacionalización de variables.....</b>	<b>23</b>
<i>Operacionalización de variables.....</i>	<i>24</i>
<b>Antecedentes .....</b>	<b>26</b>
<b>Objetivos .....</b>	<b>27</b>
<i>Objetivo general del Proyecto .....</i>	<i>27</i>
<b>Objetivos específicos del proyecto.....</b>	<b>28</b>
<b>Capitulo II: Marco teorico .....</b>	<b>29</b>
<b>Introducción .....</b>	<b>29</b>
<b>El Internet de las cosas IoT .....</b>	<b>31</b>
<i>Principales Características.....</i>	<i>32</i>
<b>Protocolos IoT.....</b>	<b>33</b>
<i>OPC UA .....</i>	<i>33</i>
<i>HTTP (REST/JSON) .....</i>	<i>35</i>
<i>MQTT.....</i>	<i>36</i>
<i>CoAP .....</i>	<i>36</i>
<i>DDS .....</i>	<i>36</i>
<i>AMQP .....</i>	<i>37</i>
<b>Protocolo MQTT .....</b>	<b>37</b>

<b>Modelo publicación/suscripción .....</b>	<b>38</b>
<b>Cliente MQTT .....</b>	<b>39</b>
<b>El broker MQTT.....</b>	<b>39</b>
<b>Topic.....</b>	<b>40</b>
<b>QoS (Quality of Service) MQTT.....</b>	<b>40</b>
<b>Intercambio de mensajes MQTT MQTT.....</b>	<b>41</b>
<b>Hosting.....</b>	<b>43</b>
<b>Hosting gratuito.....</b>	<b>43</b>
<b>Hosting compartido.....</b>	<b>43</b>
<b>Hosting virtual .....</b>	<b>44</b>
<b>Hosting dedicado .....</b>	<b>44</b>
<b>Hosting en la nube .....</b>	<b>44</b>
<b>El internet de las cosas en el ambito Industrial IIoT .....</b>	<b>45</b>
<b>Sistemas de control en lazo Cerrado .....</b>	<b>47</b>
<b>Características de un sistema de control de lazo cerrado .....</b>	<b>48</b>
<b>Elementos básicos de un sistema de control de lazo cerrado.....</b>	<b>49</b>
<b>Elementos de comparación .....</b>	<b>49</b>
<b>Elemento de control .....</b>	<b>49</b>
<b>Elemento de corrección .....</b>	<b>49</b>
<b>Elemento de proceso .....</b>	<b>50</b>
<b>Elemento de medición .....</b>	<b>50</b>

Control PID .....	50
Método de sintonía Lambda.....	51
Node Red .....	52
<i>Datos y variables en Node-RED</i> .....	55
<i>Contexto de las variables</i> .....	55
<i>Librería de nodos</i> .....	55
Factory IO .....	56
<i>Navegación</i> .....	57
<i>Orbit Cámara</i> .....	58
<i>Fly Cámara</i> .....	59
<i>First person</i> .....	60
<i>Apertura de una escena</i> .....	62
Plataforma Ubidots .....	63
Siemens y el internet de las cosas .....	66
Librería LMQTT .....	67
PLC Sim advanced.....	68
<i>Ventajas del programa S7 PLCSIM Advanced</i> .....	68
Capitulo III: Desarrollo y Simulación .....	70
Descripción general del proyecto.....	70
Descripción del proceso y componentes Factory IO .....	72
<i>Tanque de agua</i> .....	72

<b><i>Válvulas de Carga y Descarga</i></b> .....	<b>73</b>
<b><i>Baliza</i></b> .....	<b>74</b>
<b><i>Sirena</i></b> .....	<b>75</b>
<b><i>Tablero de Control</i></b> .....	<b>76</b>
<b>Descripción de los bloques de control y el algoritmo implementado.</b> .....	<b>79</b>
<b>Configuración y creación de variables en Node-Red</b> .....	<b>86</b>
<b>Creación de dispositivos y dashboard en Ubidots.</b> .....	<b>91</b>
<b>Capítulo IV: Experimentación y Resultados</b> .....	<b>94</b>
<b>Prueba de la Instancia PLC-Sim y Tia Portal</b> .....	<b>94</b>
<b>Prueba de comunicación Factory IO</b> .....	<b>98</b>
<b>Pruebas de comunicación Node Red</b> .....	<b>100</b>
<b>Sintonía del lazo de control</b> .....	<b>103</b>
<b>Capítulo V: Validación de la hipótesis</b> .....	<b>107</b>
<b>Conclusiones</b> .....	<b>108</b>
<b>Recomendaciones</b> .....	<b>108</b>
<b>Bibliografía</b> .....	<b>110</b>
<b>Anexos</b> .....	<b>113</b>

## ÍNDICE DE FIGURAS

<b>Figura 1</b> <i>Intenet de las cosas IOT</i> .....	<b>32</b>
<b>Figura 2</b> <i>OPC UA</i> .....	<b>34</b>
<b>Figura 3</b> <i>HTTP(REST/JSON)</i> .....	<b>35</b>
<b>Figura 4</b> <i>Modelo publicación y suscripción</i> .....	<b>38</b>

<b>Figura 5</b> <i>Ejemplo de cómo controlar un topic</i> .....	40
<b>Figura 6</b> <i>Intercambio de mensajes MQTT/MQTT</i> .....	41
<b>Figura 7</b> <i>Sistema de control de lazo cerrado formulas y variables</i> .....	48
<b>Figura 8</b> <i>Ejemplo del sistema de control de lazo cerrado</i> .....	49
<b>Figura 9</b> <i>Curva del método Lambda</i> .....	51
<b>Figura 10</b> <i>Método de sintonía Lambda</i> .....	52
<b>Figura 11</b> <i>Nodo Red</i> .....	53
<b>Figura 12</b> <i>Factory IO programa</i> .....	57
<b>Figura 13</b> <i>Tipos de cámaras</i> .....	58
<b>Figura 14</b> <i>Cámara First person</i> .....	62
<b>Figura 15</b> <i>Escenas del sistema</i> .....	63
<b>Figura 16</b> <i>Plataforma Ubidots</i> .....	64
<b>Figura 17</b> <i>Plataforma Ubidots funcionando</i> .....	66
<b>Figura 18</b> <i>Aquitectura y la nube</i> .....	71
<b>Figura 19</b> <i>Imagen general del proceso</i> .....	72
<b>Figura 20</b> <i>Tanque de agua</i> .....	73
<b>Figura 21</b> <i>Válvulas de carga y descarga</i> .....	74
<b>Figura 22</b> <i>Baliza del sistema</i> .....	75
<b>Figura 23</b> <i>Sirena de alerta</i> .....	75
<b>Figura 24</b> <i>Tablero de direccionamiento de variables</i> .....	77
<b>Figura 25</b> <i>Bloques de funciones creados para el control</i> .....	79
<b>Figura 26</b> <i>Segmento de programación 1</i> .....	80
<b>Figura 27</b> <i>Tipo de datos, atributos y direcciones creadas</i> .....	80
<b>Figura 28</b> <i>Segmento de programación 2</i> .....	81
<b>Figura 29</b> <i>Segmento de programación 3</i> .....	82
<b>Figura 30</b> <i>Segmento de programación 4</i> .....	83

<b>Figura 31</b> <i>Segmento de programación 5</i> .....	84
<b>Figura 32</b> <i>Segmento de programación 6</i> .....	84
<b>Figura 33</b> <i>Segmento de programación modificado</i> .....	85
<b>Figura 34</b> <i>Ejecución de node Red</i> .....	86
<b>Figura 35</b> <i>Interfaz de Node Red</i> .....	87
<b>Figura 36</b> <i>Flujo del proceso Node Red</i> .....	87
<b>Figura 37</b> <i>Nodo de función</i> .....	88
<b>Figura 38</b> <i>Bloque de función Node Red</i> .....	89
<b>Figura 39</b> <i>Configuración del servidor Ubidots</i> .....	90
<b>Figura 40</b> <i>Configuración del Nodo MQTT</i> .....	90
<b>Figura 41</b> <i>Creación del servidor Ubidots</i> .....	92
<b>Figura 42</b> <i>Conectividad con el dispositivo Ubidots</i> .....	92
<b>Figura 43</b> <i>Creación de variables</i> .....	93
<b>Figura 44</b> <i>Gráfico de tendencias e ingreso de valores</i> .....	93
<b>Figura 45</b> <i>Dirección asignada</i> .....	94
<b>Figura 46</b> <i>Direccionamiento IPv4</i> .....	95
<b>Figura 47</b> <i>Programación del PLC</i> .....	95
<b>Figura 48</b> <i>IP adicional</i> .....	96
<b>Figura 49</b> <i>Red virtual en el PLC</i> .....	96
<b>Figura 50</b> <i>Conexión con la dirección añadida</i> .....	97
<b>Figura 51</b> <i>Conexión online entre Tia Portal y el dispositivo virtual</i> .....	97
<b>Figura 52</b> <i>Parámetros de control</i> .....	98
<b>Figura 53</b> <i>Conexión Exitosa</i> .....	99
<b>Figura 54</b> <i>Sistema en modo Run</i> .....	99
<b>Figura 55</b> <i>Configuración del nodo PLC</i> .....	100
<b>Figura 56</b> <i>Configuración del nodo del servidor Ubidots</i> .....	101

<b>Figura 57</b> <i>Ventana de ejecución del Node Red</i> .....	101
<b>Figura 58</b> <i>Comunicación del servidor</i> .....	102
<b>Figura 59</b> <i>Estado de Conexión</i> .....	103
<b>Figura 60</b> <i>Respuesta del proceso</i> .....	104
<b>Figura 61</b> <i>Sincronización del lazo PID</i> .....	104
<b>Figura 62</b> <i>Procedimiento del método lambda</i> .....	105
<b>Figura 63</b> <i>Respuesta del sistema de CV</i> .....	105
<b>Figura 64</b> <i>Respuesta a escalones</i> .....	106
<b>Figura 65</b> <i>Gráfica de tendencias en el servidor Ubidots</i> .....	106

**ÍNDICE DE TABLAS**

<b>Tabla 1</b> <i>Operacionalización de variables</i> .....	24
<b>Tabla 2</b> <i>Ejemplos de aplicación de la recopilación y análisis de datos</i> .....	30
<b>Tabla 3</b> <i>Tipo de mensaje con sus códigos</i> .....	42
<b>Tabla 4</b> <i>Las diferencias entre el IOT y la industria 4.0</i> .....	46
<b>Tabla 5</b> <i>Orbit cámara control Factory IO</i> .....	59
<b>Tabla 6</b> <i>Cámara de control Factory IO</i> .....	60
<b>Tabla 7</b> <i>Control de Frist person</i> .....	61
<b>Tabla 8</b> <i>Tabla de las versiones de los softwares utilizados</i> .....	71
<b>Tabla 9</b> <i>Tabla de direccionamiento de variables</i> .....	77

**Tabla de acrónimos**

<b>IoT:</b>	Internet of things, internet de las cosas.
<b>WSN:</b>	Wireless network sensor.
<b>6LoWPAN:</b>	IPv6 over Low-Power Wireless Personal Area Networks.
<b>M2M:</b>	Machine to machine.
<b>AMQP:</b>	Advanced Message Queuing Protocol.
<b>MQTT:</b>	Message queue telemetry transport.
<b>CoAP:</b>	Constrained Application Protocol.
<b>TLS:</b>	Transport layer security.
<b>SSL:</b>	Secure socket layers.
<b>SASL:</b>	Simple Authentication and Security Layer.
<b>STOMP:</b>	Simple/Streaming text oriented messaging protocol.
<b>PII:</b>	Personally identifiable information.
<b>SPI:</b>	Sensitive personal information.
<b>TCP:</b>	Transmission control protocol.
<b>UDP:</b>	User datagram protocol.
<b>AQI:</b>	Air quality index.
<b>LPM:</b>	Low power mode.

## Resumen

En el proyecto se implementa una arquitectura IoT para monitorear y controlar una estación de nivel virtual, se cuenta con un proceso de nivel desarrollado en el software Factory IO, el algoritmo de control del proceso se lo implementa en TIA Portal V16 y se lo ejecuta en PLC Sim Avance, la comunicación para IoT inicia con Node-red el cual adquiere los datos del PLC y finalmente los transmite al broker MQTT, para este proyecto se ha utilizado el servidor de Ubidots, en el cual se ha implementado el dashboard el mismo que permite el acceso a los diferentes host a la arquitectura implementada, además recopila y guarda la información de tendencias de los principales parámetros con lo que se tiene la posibilidad de analizar el comportamiento de las variables y eventos del proceso. Con la incorporación del monitoreo y control en la nube se mejora la eficiencia operativa, mediante el acceso remoto se tiene acceso a los modos de operación del sistema tanto local como remoto, así como el monitoreo de alarmas HH y LL además de situaciones de emergencia. En cualquier instante de tiempo el control en modo local tiene una priorización mayor que el modo remoto, con esto se emula un proceso físico que podría operar normalmente a pesar de que exista una falla de comunicación con el servidor. Finalmente, en el escenario desarrollado se consideran vallas de seguridad y sensores de ingreso al proceso con lo que se contempla aspectos de seguridad e integridad, así como de blindaje de máquina.

*Palabras Clave:* IoT, bloque de datos, Ubidots, virtualización, MQTT

### **Abstract**

In the project, an IoT architecture is implemented to monitor and control a virtual level station, there is a level process developed in the Factory IO software, the process control algorithm is implemented in TIA Portal V16 and it is executed in PLC Sim Avance, the communication for IoT begins with Node-red which acquires the data from the PLC and finally transmits it to the MQTT broker, for this project the Ubidots server has been used, in which the same dashboard has been implemented. it allows access to the different hosts to the implemented architecture, it also collects and saves the trend information of the main parameters with which it is possible to analyze the behavior of the variables and events of the process. With the incorporation of monitoring and control in the cloud, operational efficiency is improved, through remote access you have access to both local and remote system operation modes, as well as monitoring of HH and LL alarms in addition to emergency situations. At any time, control in local mode has a higher priority than remote mode, with this a physical process that could operate normally is emulated despite the fact that there is a communication failure with the server. Finally, in the scenario developed, security fences and process entry sensors are considered, which contemplates aspects of security and integrity, as well as machine shielding.

*Keywords:* IoT, data block, Ubidots, virtualization, MQTT

## Capítulo I

### Planteamiento del problema de investigación

#### Introducción

La Universidad de las Fuerzas Armadas ESPE Sede Latacunga, provee a sus estudiantes entrenamiento de alto nivel en el laboratorio de Mecatrónica, sus componentes funcionales acorde a lo esperado, pero su actualización a la nueva ola de innovaciones es deseable y necesaria, por lo que es de vital importancia proponer nuevas soluciones en base a la nueva corriente de la Industria 4.0.

Actualmente por el apareamiento de nueva tecnología, la implementación de nuevos sistemas de entrenamiento para la Industria 4.0 es todavía costoso, las Universidades, que han hecho una fuerte inversión en tecnología de la tercera generación de automatización no tienen los recursos para justificar una nueva adquisición y muchos de los equipos disponibles no cumplen el tiempo de vida útil.

Lamentablemente, la falta de material didáctico disponible con la cuarta generación de equipos de automatización hace que el entrenamiento de los nuevos profesionales sea en parte incompleta y hace más evidente el retraso tecnológico en comparación con países e instituciones que si puedan costear un nuevo cambio de tecnología.

En la actualidad, múltiples iniciativas de software libre se han puesto a la cabeza para dar una nueva vida útil a los autómatas programables de la tercera generación, su implementación es correcta, pero en el país todavía no existe una decisión firme de

implementarlo en un sistema funcional, por lo que su aplicación no solo es útil, también es necesaria.

El reacondicionamiento de equipo actual no sólo trae las ventajas de la conectividad, también la posibilidad de crear historiales de uso, horas de servicio, etc. Dichos datos son valiosos para generar un correcto mantenimiento preventivo y asegurar la disponibilidad adecuada de repuestos de automatización para cuando éstos lleguen a su vida útil.

Aparte, la monitorización de entradas y salidas es importante, pues su correcta transmisión y visualización permite que el entorno académico pueda conocer de primera mano el funcionamiento del autómatas, depurar programas de PLC sin acceder a las conexiones físicas y poder generar señales de fallo e implementar paros de emergencia remotos.

### **Problemas Principales**

- Actualmente no existen tecnologías de vanguardia, con precios inaccesibles para masificar o actualizar los laboratorios de la institución.
- Al carecer de esta tecnología en algún equipo, limita el aprendizaje o entrenamiento a los nuevos profesionales retrasándolos tecnológicamente.
- No existen datos reales del equipo en tiempo real que permita saber cuándo es necesario efectuar un mantenimiento.
- Actualmente no se ha implementado esta migración de acoplar esta tecnología moderna a equipos antiguos, con lo planteado sería un paso importante a la tecnología 4.0.

- En el mundo industrial, el robo de datos se ha vuelto más serio y las industrias se han vuelto más vulnerables a las filtraciones de datos, estas nuevas tecnologías permitirán evitar todo este tipo de situaciones.

## **Justificación**

La investigación enfatizará el diseño de un prototipo de arquitectura IoT para el monitoreo y control de una estación de nivel mediante un autómata simulado, ya que, debido a la incursión del internet de las cosas en la industria, los procesos y procedimientos buscan incrementar su eficiencia mediante la aplicación de la tecnología de vanguardia, dando lugar al IoT. El presente trabajo se convertirá en la base de desarrollo de otro tipo de escenarios y arquitecturas simuladas lo que brinda la posibilidad de realizar pruebas a costos relativamente bajos.

El análisis de la solución desarrollada, permitirá distinguir con claridad las compatibilidades del software integrado y presenta la posibilidad de desarrollar alternativas para la resolución de problemas. A través de esto se busca incrementar la eficiencia en el diseño y dimensionamiento de proyectos, además de ofrecer mejoras en los proyectos que ya se encuentran trabajando, para perfeccionarlos de acuerdo a diferentes condiciones y necesidades presentes.

El desarrollo de proyectos de IoT enfocados en entornos de proceso virtualizados, permite desarrollar y evaluar procesos con una considerable reducción de recursos si se lo compara con la implementación física del mismo.

El documento presenta el resultado de una investigación experimental, así como la evaluación en base a simulaciones de un sistema de control de nivel, comunicaciones entre dispositivos, intercambio de información cliente/servidor, número de dispositivos

interconectados, seguridad en la comunicación. El método se fundamenta en la evaluación experimental, con el objetivo de desarrollar una arquitectura útil para promover proyectos de IoT y evaluar su funcionalidad.

### **Hipótesis de investigación**

¿El diseño de un prototipo de arquitectura IoT, permitirá monitorear y controlar la estación de nivel mediante el autómata simulado?

### **Categorización de las variables de investigación (variables independientes y dependiente)**

#### ***Variable Independiente***

Diseño e implementación de un prototipo de arquitectura IoT.

#### ***Variable Dependiente***

Monitoreo y Control de la estación de nivel mediante el autómata simulado.

### **Operacionalización de variables**

**Tabla 1***Operacionalización de variables*

<b>VARIABLES</b>	<b>TIPO</b>	<b>DEFINICIÓN CONCEPTUAL</b>	<b>DEFINICIÓN OPERACIONAL</b>	<b>DIMENSIONES</b>	<b>INDICADORES</b>
Diseño e implementación de un prototipo de arquitectura IoT	Independiente	Un programa IoT permite realizar operaciones para el monitoreo, supervisión y control de los dispositivos que se conectan en cualquier tipo de estación, permitiendo realizar la gestión de los mismos para implementar programas de mantenimiento preventivo y predictivo	Un programa IoT mediante software libre permite resolver situaciones de mantenimiento y optimizar recursos en cualquier tipo de estación de trabajo	<ul style="list-style-type: none"> <li>• Costo de operación.</li> <li>• Tiempo de respuesta</li> </ul>	<ul style="list-style-type: none"> <li>• Costo mínimo de operación del sistema.</li> <li>• Porcentaje de disminución del tiempo de respuesta</li> </ul>

VARIABLES	TIPO	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
Monitoreo y Control de la estación de nivel mediante el automática simulado	Dependiente	El monitoreo por medio de IoT con software libre se basa en reducir al máximo la intervención física del operador y a la vez obteniendo información para establecer predicciones optimizando recursos.	Las IoT generan grandes cantidades de datos que permiten encontrar soluciones a las estaciones de trabajo a las que estén aplicadas.	<ul style="list-style-type: none"> <li>• Generación de datos y alarmas</li> <li>• Mantenimiento predictivo y correctivo</li> </ul>	<ul style="list-style-type: none"> <li>• Alarmas producidas en las Estaciones.</li> <li>• Volumen máximo de datos generados.</li> </ul>

## Antecedentes

El uso de nuevas soluciones con el IoT, entre estos el desarrollo industrial, ha permitido que con estas herramientas se pueda llegar a planificar y controlar efectivamente los diferentes tipos de estaciones de control. Actualmente en las ciudades se puede apreciar el uso de las tecnologías relacionadas con el IoT. El rápido crecimiento de algunas urbes hace necesario el desarrollo de una infraestructura de control que ayude a crear caracterizaciones que lleven al mejoramiento de la calidad de vida de sus habitantes.

Como ejemplo de esto están los trabajos encargados de implementar plataformas de control ambiental a través del uso de las IoT. Aquí se tiene el control de variables medioambientales en áreas urbanas (Gómez et al, 2017), en las que por medio de estaciones de monitoreo se recolecta información y luego es puesta a disposición de la población y las autoridades competentes. En otras partes del mundo se desarrollan iniciativas que aprovechan este tipo de trabajos y que tienen como propósito el mejoramiento de su capacidad alimentaria.

La aplicación de diversos sistemas como los relacionados con la monitorización para cultivos en invernaderos que buscan el control de los diferentes parámetros inmersos en el crecimiento de los cultivos a través del uso e implementación de redes de sensores y control de actuadores que ayuden a hacer un efectivo control de estos cultivos, se hace de manera real, mediante el monitoreo de la humedad relativa y temperatura, contenido volumétrico de agua en el suelo y demás factores que inciden en el desarrollo de los cultivos protegidos y que podrían ser monitorizados y controlados con sistemas de red de sensores e IoT. (Cama-Pinto, A., Gil-Montoya, F., Manzano-Agugliaro, F., García-Cruz, A., & Gómez-López, J. , 2014).

Este trabajo muestra la implementación y aplicación de una red de sensores, el uso en general de hardware libre y el uso de un modelo de aplicación sensor – cloud – server. El usuario obtendría la información, basada en tablas y gráficos relacionados con los parámetros a estudiar. Las conclusiones de este estudio están relacionadas con la facilidad de aplicación y control en los cultivos y presenta los retos obvios de un proceso de producción de bajo costo, frente al alto costo y baja configurabilidad de los sistemas ofrecidos en el mercado. Los resultados se pueden ver en cultivos protegidos en Europa, en especial en el mediterráneo en el que se ha visto un incremento de este tipo de cultivos, después de China. (Cama-Pinto, A., Gil-Montoya, F., Manzano-Agugliaro, F., García-Cruz, A., & Gómez-López, J. , 2014).

Actualmente, han surgido una diversidad de plataformas en respuesta al rápido incremento de dispositivos inalámbricos aplicados a la medición y automatización de procesos industriales y residenciales. Las plataformas IoT permiten el registro de datos adquiridos por sensores, el enrutamiento de datos en tiempo real y el control remoto de dispositivos. Además, proveen una interfaz como punto de acceso a sus recursos a través de protocolos de Internet; que permiten interactuar con lenguajes de alto nivel como LabView, Matlab y Python, para realizar procesamiento de datos y accionar remotamente un dispositivo. (Kerkez, 2016).

## **Objetivos**

### ***Objetivo general del Proyecto***

Diseñar un prototipo de arquitectura IoT para el monitoreo y control de una estación de nivel mediante un autómata simulado.

### ***Objetivos específicos del proyecto***

- Obtener y manipular los datos de red del autómata programable simulado para controlar el mismo a través de software libre.
- Utilizar un broker MQTT para el control y monitoreo remoto de los sensores y actuadores de la Estación de nivel virtual.
- Utilizar un protocolo de comunicación de red ligero para el intercambio de datos entre el PLC, bróker, dispositivos suscriptor y publicador.
- Registrar datos de uso de los sensores y actuadores a través del servidor, para recomendar acciones de mantenimiento preventivas del sistema.
- Realizar pruebas de la operación del sistema

## Capítulo II

### Marco teórico

#### Introducción

La tecnología actual ha permitido que las empresas y sectores productivos innoven su enfoque para interactuar con la sociedad. Los usuarios se sienten más cómodos con la tecnología digital y utilizan dispositivos inteligentes para su beneficio. Las empresas proporcionan todos sus servicios o parte de ellos en línea. Desde la comodidad del hogar se puede comprar alimentos en línea, reservar viajes, realizar pedidos de ropa y mantenerse conectado con sus amigos y familiares.

Los hogares inteligentes pueden contar con sensores de movimiento, sensores de agua, luz, temperatura, etc. Puede existir sensores en los semáforos, transporte público, garajes, cámaras de seguridad, trenes, aviones. Todos estos sensores y dispositivos de medición recopilan y transmiten sus propios datos. Los datos pueden almacenarse y analizarse en una fecha posterior o pueden ser analizados de forma inmediata y ser utilizados para modificar ciertos parámetros o ejecutar acciones de control en procesos de cualquier tipo. La combinación de métodos, infraestructura, equipos, elementos e instrumentos utilizados en la tendencia expuesta converge para dar paso al Internet de las cosas o IoT. En la tabla 2 se revisa algunos ejemplos de la aplicación de la recopilación y análisis de datos a través de herramientas de IoT.

**Tabla 2**

*Ejemplos de aplicación de la recopilación y análisis de datos.*

<b>INSTITUCIÓN</b>	<b>APLICACIÓN</b>
<b>Empresas</b>	Determinar patrones de compra Pronóstico de tendencias Optimización de la producción
<b>Gobiernos Nacionales</b>	Pronosticar tendencias de la población. Planificación de servicios de seguro social. Tasa de delitos.
<b>Gobiernos Locales</b>	Control de tráfico. Monitoreo de estacionamiento y garajes públicos. Asistencia de instituciones de emergencia. Gestión de residuos.

## El Internet de las cosas IoT

El internet de las cosas (IOT) está en auge, por lo que no hay una definición estándar sugerida por algunas organizaciones o autores el concepto de sujeto según sus diferentes criterios y puntos de vista, luego cubriremos los más relevantes. El IOT es una red de dispositivos físicos conectados de forma inteligente con la capacidad de recopilar datos del entorno en el que interactúan las cosas, las comunicaciones y aplicaciones para realizar un análisis de los datos recopilados. Un concepto más amplio define el internet de las cosas como una infraestructura de red global, dinámica y autónoma, en la cual, interactúan hardware y software (cosas físicas y virtuales), con un alto grado de autonomía de captura de datos, transferencia de eventos, conectividad de red e interoperabilidad.

El RFID group define la IoT como: “La red mundial de objetos interconectados direccionables basado exclusivamente en estándares de protocolos de comunicación”, mientras que para el Cluster of European Research Projects: “En la IoT se espera que “las cosas” conviertan a los participantes activos en procesos sociales, de información y negocios en donde “estas cosas” sean capaces de interactuar y comunicarse entre ellos mismos y con el ambiente a través del intercambio de datos e información detectada a su alrededor, mientras reaccionan automáticamente a los eventos del “mundo físico/real” y son influenciados a través de procesos en ejecución que disparan acciones y crean servicios con o sin la intervención directa del ser humano”.

Los diversos estándares del concepto del IoT Internet de las Cosas coinciden en que la red comunicación es uno de los principales pilares del Internet de las Cosas, ayuda a conectar un dispositivo, máquinas, sensores o "cosas" que genera datos desde cualquier punto geográfico del planeta. (Chuchico-Arcos, C., & Rivas-Lalaleo, D., Diciembre 2020)



el control de la calefacción o la luz en las viviendas, incluso sistemas industriales automatizados para el transporte de productos a escala mundial.

Un dispositivo IoT se caracteriza por ser un sistema electrónico de pequeño tamaño equipado con un procesador, sensores que le permiten medir el entorno, actuadores que le permiten realizar determinadas acciones en respuesta a los datos recibidos y módulos de comunicación vía cable, wifi o bluetooth.

Todas estas características le permiten comunicarse y cooperar con otros dispositivos, ser fácilmente configurables de forma remota y localizables, tanto físicamente, como en la red de la que constituye. (Cepymenews, 2022)

## **Protocolos IoT**

Las plataformas IoT proporcionan múltiples beneficios a las empresas del entorno industrial: desde potenciar la automatización y la productividad a generar nuevos modelos de gestión. No obstante, sin protocolos de red estos sistemas no se podrían ejecutar. (Barbaraiot, 2021)

Los protocolos IoT son un conjunto de normas y reglas que consienten de dos entidades a entenderse e intercambiar información, proporcionando la comunicación Machine2Machine (M2M). Los protocolos tienen permitido conectarse a dispositivos industriales con plataformas IoT, tratando de conectar dos aplicaciones y las dos soportan HTTP cada protocolo debe saber cuándo y cómo usarse:

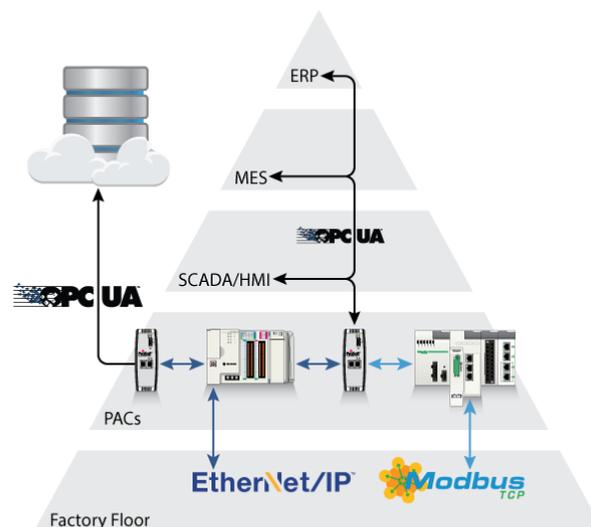
## **OPC UA**

El OPC UA (Unified Architecture, 'arquitectura unificada') es el estándar de nueva generación que le sigue a OPC Foundation. OPC clásico es bien conocido en la

industria y proporciona una interfaz estándar para comunicarse con los PLC (Programmable Logic Controller, “controlador lógico programable”). OPC UA procura expandir la compatibilidad de OPC al nivel de los dispositivos y de las empresas. OPC UA es un protocolo cliente/servidor. Los clientes se conectan, navegan, leen y escriben al equipamiento industrial.

La extensa base instalada en el mundo industrial del OPC UA. Es un buen recurso para enlazar información de sensores y PLC en aplicaciones industriales ya existentes como sistemas MES (Manufacturing Execution System, 'sistema de ejecución de manufactura') y SCADA (Supervisory Control and Data Acquisition, 'supervisión, control y adquisición de datos'), en donde la conectividad OPC y OPC UA ya estén disponibles. (Semle, Aron., 2016)

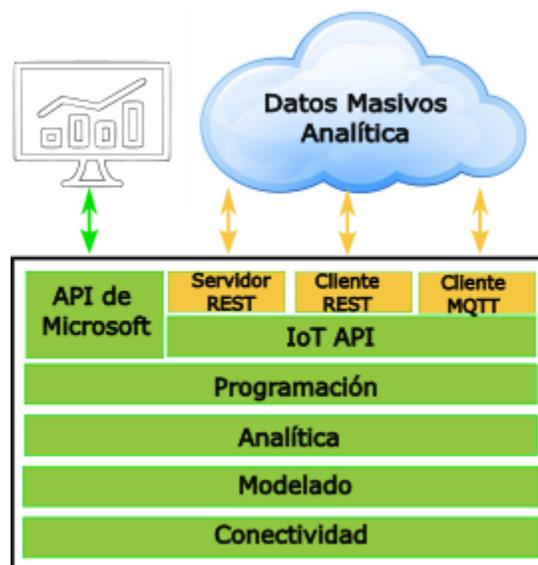
**Figura 2**  
*OPC UA*



### **HTTP (REST/JSON)**

HTTP (Hypertext Transfer Protocol, 'protocolo de transferencia de hipertexto') es un protocolo cliente/servidor sin conexión ubicuo en TIC y en la web. Dado que existen inmensurables herramientas de código abierto que usan HTTP, y que todo lenguaje de codificación tiene bibliotecas HTTP, es muy accesible. El foco de HTTP en IoT gira en torno a REST (Representational State Transfer, transferencia de estado representacional'), que es un modelo sin estados previos donde los clientes logran acceder a recursos en el servidor a través de pedidos. En la mayoría de los casos, un recurso es un dispositivo y la información que tal dispositivo contiene. HTTP provee transporte, pero no define la presentación de la información. (Semle, Aron., 2016)

**Figura 3**  
*HTTP (REST/JSON)*



## **MQTT**

MQTT (Message Queuing Telemetry Transport, 'Cola de mensajes telemetría y transporte') es un protocolo publicar/suscribir diseñado para SCADA y redes remotas. Se centra en un mínimo encabezado (dos bytes de cabeza) y comunicaciones confiables. También es muy simple. Tal como HTTP, la carga MQTT es concreta para la aplicación, y la colectividad de las implementaciones utilizan un formato JSON personalizado o binario. (Semle, Aron., 2016)

## **CoAP**

CoAP (Constrained Application Protocol, 'protocolo de aplicación restringida') fue fundado por IETF (Internet Engineering Task Force, 'Grupo de Trabajo de Ingeniería de Internet') para suministrar la compatibilidad de HTTP con una mínima carga. CoAP es similar a HTTP, pero usa UDP/multicast en lugar de TCP.

Además, facilita el encabezado HTTP y comprime el tamaño de cada requerimiento. CoAP se utiliza en dispositivo de borde en donde HTTP sería demandante de recursos, y a menudo, las plataformas de IoT lo utilizan como tercer protocolo, después de HTTP y MQTT. Similar a HTTPS, CoAP usa DTLS (Datagram Transport Layer Security, 'seguridad en la capa de transporte datagrama') para proteger las comunicaciones. (Semle, Aron., 2016)

## **DDS**

DDS (Data Distribution Service, 'servicio de distribución de datos') es un protocolo publicar/ suscribir que se focaliza en el borde de la comunicación en la red. DDS es un estándar abierto operado por OMG (Object Management Group, 'Grupo de Gestión de Objetos'). A diferencia de MQTT, que requiere de un agente centralizado,

DDS está descentralizado. Los nodos de DDS se comunican directamente punto a punto a través de UDP/multidifusión (multicast). Esto forma que no sea necesario una gestión centralizada de la red y que DDS sea un protocolo más veloz, con una resolución por debajo del milisegundo. DDS es una buena salida para la entrega de información de forma confiable y en tiempo real. Úselo para comunicaciones rápidas M2M (machine to machine, 'máquina a máquina'). DDS soporta a los gestores para integrar redes DDS con la empresa, pero en la destreza no está bien posicionada como punto de composición entre la industria y TIC; como gestores, son a menudo secundarios para la red DDS (Semle, Aron., 2016).

### **AMQP**

AMQP (Advanced Message Queuing Protocol) es otro protocolo tipo publicar/suscribir que procede del fragmento de servicios financieros. Tiene su representación en TIC, pero suficientemente limitada en la industria. El mayor beneficio de AMQP es su modelo firme en comunicaciones que soporta transacciones. A diferencia de MQTT, AMQP puede garantizar transacciones completas lo cual, no obstante, es útil, no siempre es algo que soliciten las aplicaciones IoT AMQP se agrupa a menudo con protocolos IoT y es uno, pero su mayor contra es que se trata de un protocolo pesado. Fue propuesto para sistemas TIC, y no para el límite de la red. (Semle, Aron., 2016).

### **Protocolo MQTT**

El protocolo MQTT (Message Queue Telemetry Transport) se trata de un protocolo de mensajería asíncrona, usado en el ámbito de las comunicaciones machine-to-machine (M2M), en el campo de IoT (Internet de las cosas). Su funcionamiento se basa

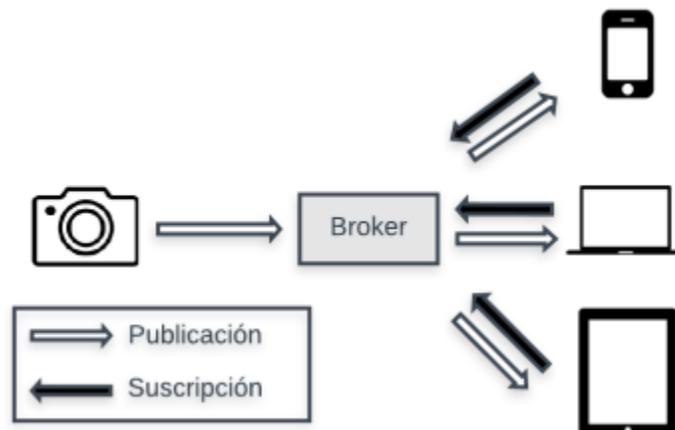
en el intercambio de mensajes mediante el modelo de publicación y suscripción. Este protocolo sutil permite su implementación en redes con un ancho de banda definido y con alta latencia. Por otra parte, su flexibilidad, accede su aplicación en diferentes dispositivos y servicios de IoT. (Mahedero Biot, F., 2020)

### **Modelo publicación/suscripción**

El modelo publicación y suscripción funciona a través de la conexión a un nodo central, llamado broker, por parte de los clientes. Tanto para el envío de los mensajes, como para su posterior recibimiento, ambos se suscriben a un topic donde se colocan los mensajes, convirtiendo así este modelo en un modelo no-bloqueante. (Mahedero Biot, F., 2020)

#### **Figura 4**

*Modelo publicación y suscripción*



Este desacople viene dado en tres dimensiones:

- Tiempo: ambos clientes no tienen por qué estar funcionando al mismo tiempo.
- Espacio: ambos clientes no tienen por qué conocerse entre sí.

- Sincronización: no se produce ningún resultado de interrupción en los clientes al publicar o recibir un mensaje de un topic. (Mahedero Biot, F., 2020)

### ***Cliente MQTT***

Los suscriptores y los publicadores, son cada uno de ellos un cliente MQTT. Un cliente MQTT puede ser tanto suscriptor como publicador, o ambos a la vez. La ventaja que tienen los clientes MQTT es la facilidad de implementación, y al ser un protocolo ligero, el hardware no es un problema, ya que un cliente puede ser desde un ordenador hasta cualquier otro dispositivo como un microcontrolador. En este trabajo se ha utilizado como Cliente MQTT una placa ESP32-CAM y un ordenador. (Mahedero Biot, F., 2020)

### ***El broker MQTT***

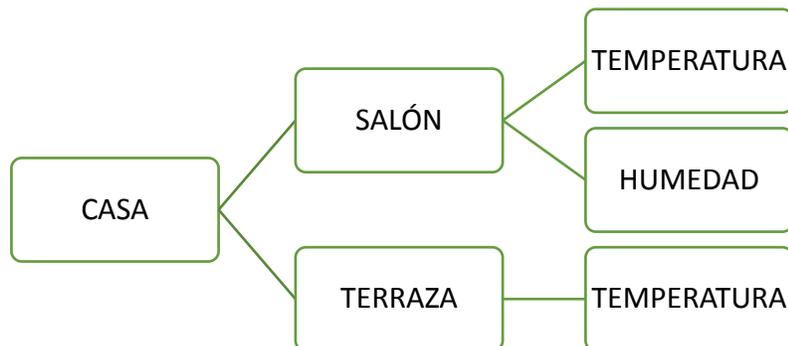
Se conoce como bróker MQTT al servidor encargado de la distribución de los mensajes a los receptores. El broker recoge los mensajes, hace un chequeo del topic al que están suscritos y los orienta hacia los clientes suscritos a este topic. El broker ofrece la posibilidad de que los mensajes sean persistentes, y a la vez que se guarden los mensajes hasta que el cliente al que va encaminado se conecte. Además, en las funciones del broker es verificar la identidad de los clientes como prevención de seguridad. En este trabajo se utiliza como hardware del broker un ordenador de placa simple llamado Raspberry Pi con el software Raspbian, versión adaptada de Debian. Por otra parte, el software instalado para llevar a cabo la función de broker será Mosquitto. (Mahedero Biot, F., 2020)

## **Topic**

El topic es el argumento o tema del mensaje, es indicar, a quien va dirigido ese mensaje o de quien se quiere recibir tal mensaje. El topic sirve como identificador al broker para poder encaminar cada uno de los mensajes recibidos hacia los clientes apropiados que también se posean la suscripción al topic. Los topics son case sensitive, es decir, diferencian entre mayúsculas y minúsculas. Estos brindan la posibilidad de poder jerarquizarlos. Por ejemplo: casa/salon/temperatura casa/salon/humedad casa/terraza/temperatura. (Mahedero Biot, F., 2020)

### **Figura 5**

*Ejemplo de cómo controlar un topic*



### **QoS (Quality of Service) MQTT**

Poseen un mecanismo para gestionar el envío de mensajes al cliente ante fallos.

Los diferentes niveles que se aprecian son:

- QoS 0: El mensaje se envía una sola vez y no se recibe confirmación de entrega.
- QoS 1: El mensaje se envía hasta que el receptor recibe la entrega. Se pueden producir mensajes duplicados.

- QoS 2: Se garantiza la entrega del mensaje y solo una vez. Un punto en contra es que esto aumenta la sobrecarga y disminuye el rendimiento del sistema.

(Mahedero Biot, F., 2020)

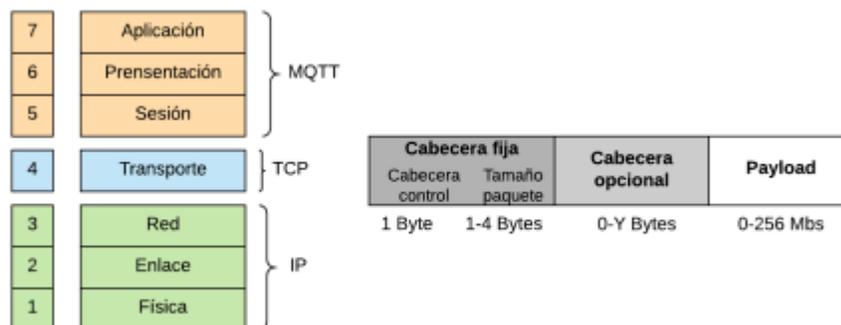
### ***Intercambio de mensajes MQTT / MQTT***

Está basado en TCP/IP y los mensajes del protocolo MQTT se dividen en tres partes:

- Cabecera fija: Consta de 2 a 5 bytes. Estos forman la cabecera de control y la longitud del mensaje. Esta longitud está codificada de 1 a 4 bytes, en los que se usan los 7 primeros bits y el restante se trata de un bit de continuidad.
- Cabecera variable: Esta cabecera es opcional.
- Payload: Está formado por el contenido real del mensaje. Su máximo es de 256 MB, aunque en las implementaciones reales el máximo se encuentra entre 2 y 4 kB.

### **Figura 6**

*Intercambio de mensajes MQTT / MQTT*



La conexión entre los clientes y el broker se hace por el puerto 1883 y con el 8883 si la conexión funciona sobre TLS. TLS proporciona autenticación y privacidad de la información entre extremos. Los tipos de mensajes junto con sus códigos son los siguientes:

**Tabla 3**  
*Tipo de mensajes con sus códigos*

<b>TIPO DE MENSAJE</b>	<b>CÓDIGO</b>	<b>DESCRIPCIÓN</b>
CONNECT	0X10	Petición del cliente para conectarse al servidor
CONNACK	0X20	Reconocimiento de la conexión
PUBLISH	0X30	Edición del mensaje
PUBACK	0X40	Reconocimiento de la edición
PUBREC	0X50	Edición recibida
PUBREL	0X60	Edición liberada
PUBCOMP	0X70	Edición completa
SUSCRIBE	0X80	Petición de suscripción del cliente
SUBACK	0X90	Reconocimiento de suscripción
UNSUBSCRIBE	0XA0	Petición desuscripción del cliente
UNSUBACK	0XB0	Reconocimiento de desuscripción
PINGREQ	0XC0	Petición de ping
PINGRESP	0XD0	Respuesta de ping
DISCONNECT	0XE0	Cliente desconectado

Las peticiones PING se envían de forma regular al broker desde el cliente. El broker responde con un PING response, método que admite a ambas partes determinar si una aún sigue disponible. (Mahedero Biot, F., 2020)

## **Hosting**

El significado de hosting en castellano es alojamiento, como un acercamiento al concepto del contrato de hosting, se puede deducir como un alojamiento web, el servicio que provee a los usuarios de internet, un sistema de dominio para poder almacenar información, la cual incluye videos, música, imágenes, correos cualquier contenido accesible vía web. (Baeza Martin, F. J., 2019)

### ***Hosting gratuito***

Este tipo de convenio está pensado para clientes que tienen un blog o que están principiando en un negocio y para promocionarlo crean una página web, disponiendo de poco presupuesto, existen muchas empresas que ofrecen hosting gratuito, pero no te engañes, no hay nada gratis, ya que empezar y crear una página web requiere una serie de gasto, como contratar el dominio, ya que sin un nombre no podrás crear la web. Para crear tu propia página, necesitas un programa para ello, este nada te sale gratis, en resumen, no te cobran por el hosting, pero si por el resto de servicios que necesitas. Es paralelo a un tipo de alojamiento, se puede equiparar a los albergues, donde se puede pasar la noche sin coste alguno, pero madrugando al día siguiente para llegar temprano al próximo y repetir la operación. (Baeza Martin, F. J., 2019)

### ***Hosting compartido***

Este es el servicio más elemental y más barato, este contrato aloja varias webs en el mismo servidor compartiendo entre todos los alojados los mismos recursos, el

espacio del disco, CPU, memoria RAM, ancho de banda, IP, etc. (Baeza Martin, F. J., 2019)

### ***Hosting virtual***

Conocido como VPS que significa, virtual private server, en castellano servidor virtual privado. La idea es fraccionar el servidor en varias entidades virtuales separadas, se obtiene un espacio independiente, con su propio sistema operativo. Así cada cliente tendrá unos recursos asignados y podrá configurar el servidor a sus necesidades. Se puede comparar al alojamiento en una casa rural, que te hospedas en habitaciones que están separadas, pero compartes zonas comunes, cocina, baños.

El tipo de hosting virtual tiene una serie de ventajas como la privacidad ya que no compartes la I.P, lo puedes personalizar con tu propio sistema operativo, autocontrol por el cliente, recursos dedicados, puedes aumentar o reducir tu capacidad según las necesidades del momento, y la relación calidad precio muy buena. (Baeza Martin, F. J., 2019)

### ***Hosting dedicado***

Es un servicio exclusivo, en el que se dedica un servidor para el proyecto del cliente, no se comparten los recursos y tienes acceso completo al software y al hardware. Es como alojarte en una villa de uso exclusivo, que va subiendo el precio según contrates los extras; piscina, jardín, casa de invitados. (Baeza Martin, F. J., 2019)

### ***Hosting en la nube***

Más conocido como cloud hosting, consiste en que la página web del cliente no está alojada en un solo servidor, sino que está en varios servidores interconectados en la nube, se debe de explicar que la nube no es una entidad física, sino una red enorme de servidores remotos de todo el mundo que están conectados para funcionar como un

único sistema, si un servidor falla se compensa con los otros, y se adapta a las necesidades en tiempo real. La analogía sería que te compras unas acciones de una cadena hotelera, y puedes alojarte en cualquier parte del mundo donde tenga instalaciones para el hospedaje dicha cadena. (Baeza Martin, F. J., 2019).

### **El internet de las cosas en el ambito Industrial IIoT**

El Internet de las cosas (IoT) está renovando el mundo, y las aplicaciones desarrolladas en la industria son consideradas como la nueva revolución industrial formando nuevas oportunidades en la economía y la sociedad. La adopción del Internet de las Cosas al desarrollo de la industria 4.0 renueva los procesos y sistemas de fabricación proyectándose a nuevos desafíos.

La información de estas aplicaciones está relacionada con el estado de la producción, el comportamiento de consumo de energía de máquinas, el movimiento de materiales, órdenes de clientes, datos de proveedores y en general, cualquier tipo de datos asociados a elementos inteligentes implementados en los procesos, los principios en los que se basa, los elementos y tecnologías disponibles para obtener la comunicación entre personas, objetos y aplicaciones que se han desarrollado en esta área e indica la importancia de su Implementación (Rodríguez Molano, José Ignacio; Cueva Lovelle, Juan Manuel , 27).Las diferencias entre el iot y la industria 4.0 son las siguiente:

**Tabla 4***Las diferencias entre el IOT y la industria 4.0*

<b>INDUSTRIA 4.0</b>	<b>IOT</b>
<b>Digitalización</b>	Big data y analítica
<b>Automatización</b>	Dispositivos interconectados
<b>Nuevos materiales</b>	Mantenimiento predictivo
<b>Técnica de mejora en los procesos</b>	Seguridad industrial

La tecnología IoT es una de las múltiples soluciones a los desafíos que presenta la industria 4.0: engloba el intercambio de datos e información entre varios de los dispositivos que incorpora el almacén. Lo opuesto a la industria 4.0 define el nuevo paradigma y los nuevos retos a los que se enfrenta las compañías la mejora de procesos, la automatización y la digitalización para mantener la competitividad.

Los beneficios del IoT en la industria son los siguientes:

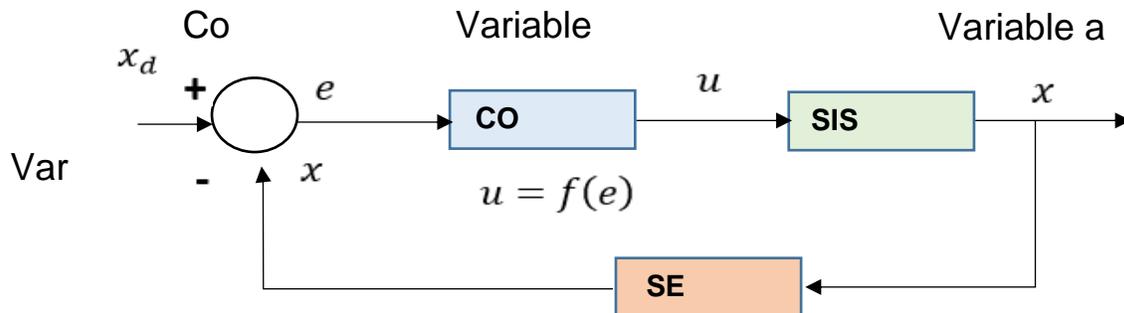
- Aumento de la eficiencia y el rendimiento en almacén: la interconexión entre personas y maquinaria facilitará la optimización de cada proceso logístico.
- Mantenimiento predictivo de la maquinaria: las máquinas contarán con sensores que, sin intervención humana, avisarán a los operarios de cualquier riesgo de avería antes de que se produzca de acuerdo con la función, se acrecentará la productividad operativa.

- Nuevas oportunidades de negocio: monitorizar de los datos que produce la compañía permitirá pronosticar las nuevas tendencias del mercado y prever qué fragmento de negocio puede ser más interesante para la empresa.
- Mejora de la eficiencia energética: la incorporación del internet de las cosas cambiará el paradigma de eficiencia energética, debido a que se dotará al almacén de sensores que ayudarán a optimizar los consumos energéticos.
- Seguridad industrial: la conectividad es la aliada de la seguridad en el almacén. Los procesos en los que intervengan elementos IoT contarán con sensores y monitores para prevenir cualquier accidente que ocurra en la instalación. (Mecalux, 2020).

### **Sistemas de control en lazo Cerrado**

El control en lazo cerrado aparece ante la necesidad de corregir las desviaciones, de la salida frente a la referencia de forma automática. Esto introduce un nuevo elemento en el sistema, un automatismo o procesador que se encargará de evaluar los valores del sensor y actuar en consecuencia. (Sinadrives, 2021)

**Figura 7**  
 Sistema de control de lazo cerrado formulas y variables

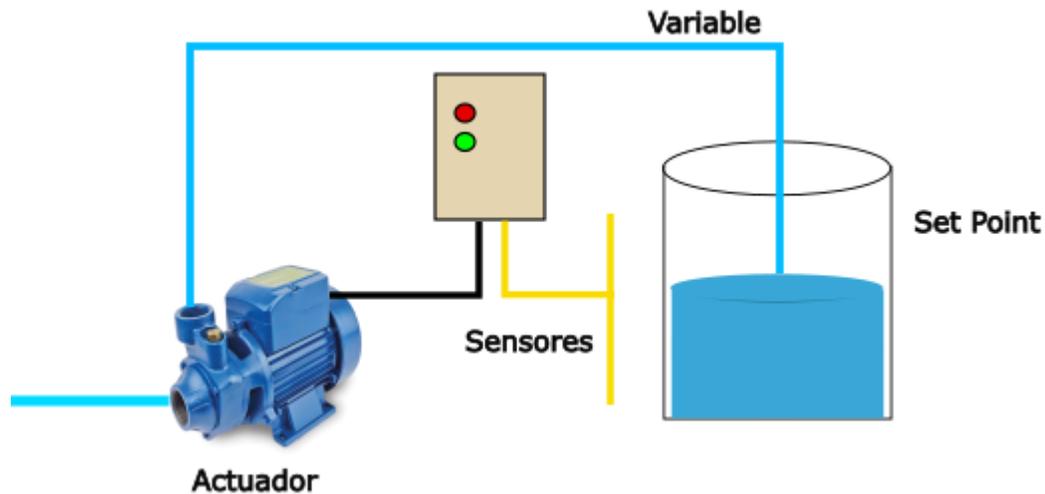


*Nota:* Ejemplo de un sistema de control de lazo cerrado. Tomado de (Trejo, Christian, 2021).

#### **Características de un sistema de control de lazo cerrado**

- La entrada consigue información de la salida esto quiere indicar que es un lazo realimentado.
- Su configuración ayuda a corregir el efecto de las perturbaciones que se presentan.
- Los sistemas de lazo cerrado son más complejos que los sistemas de lazo abierto por tal motivo el costo de implementación aumenta. Es más preciso y exacto que el lazo abierto.
- Un sistema de lazo cerrado tiene una precisión y exactitud que un sistema de lazo abierto no posee.

**Figura 8**  
Ejemplo del sistema de control de Lazo Cerrado



### Elementos básicos de un sistema de control de lazo cerrado

#### ***Elementos de comparación***

Este tipo de elemento compara el valor requerido o de referencia de la variable por controlar con el valor medido de lo que se consigue a la salida y produce una señal de error la cual indica la diferencia del valor obtenido a la salida y el valor requerido.

(Torres Capuz & Chanoluisa Martinez, 2015)

#### ***Elemento de control***

Este elemento decide que acción debe tomar cuando recibe una alerta de error.

#### ***Elemento de corrección***

Este elemento se utiliza para producir un cambio en el proceso al eliminar el error.

### ***Elemento de proceso***

El elemento de proceso o planta es el encargado del control de la variable.

### ***Elemento de medición***

Este elemento produce una señal relacionada con la condición variable controlada y proporciona la señal de retroalimentación al elemento de comparación para determinar si hay o no un error. (Torres Capuz & Chanoluisa Martinez, 2015)

## **Control PID**

El control PID es un mecanismo necesario para el control que a través de un lazo de retroalimentación permite regular la velocidad, temperatura, presión y flujo entre otras variables de un proceso en general. El controlador PID calcula la diferencia entre una variable real y una variable deseada, el claro ejemplo es un sistema de bombeo regularmente interesa mantener la presión o el flujo constante por lo tanto el control PID mide la diferencia entre la presión de la tubería y la presión requerida y actúa variando la velocidad del motor para que se pueda tener la presión o flujo continuo.

El parámetro proporcional (P) mide la diferencia entre el valor actual y el set-point (en porcentaje) y aplica el cambio en aplicaciones sumergibles, el valor recomendado es 50% y para aplicaciones centrifugas el valor que se recomienda es el 10%, mientras que el parámetro integral (I) se refiere al tiempo en el que se lleva la acción correctiva, el valor mientras sea más pequeño el ajuste es más rápido pero puede causar inestabilidad en el sistema, oscilaciones, vibración de motor o bomba el valor recomendando sumergible es de 0.5 segundos y para centrifugas es 1 segundo.

El parámetro derivativo (D) emite una acción predictiva es decir prevé el error he inicia una acción oportuna. (Torres Capuz & Chanoluisa Martinez, 2015)

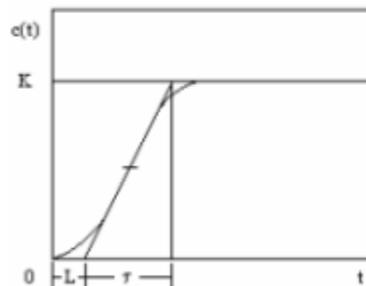
## Método de sintonía Lambda

Las insuficiencias de la industria provocan que la generación y aplicación de tecnologías para solucionar problemas enfocados en obtener excelentes resultados a menor costo, aplicando mayor confiabilidad en el control de proceso. (Guagchinga, F., Calvopiña, P., Alpúsig, S., & Cajas Buenaño, M., 2021).

En este método se obtiene experimentalmente la respuesta de la planta al aplicar un escalón unitario, como se muestra en la figura 9, si la planta no incluye integradores o polos dominantes complejos conjugados, la curva de respuesta al escalón unitario puede tener el aspecto de una curva en forma de S, como se observa en dicha figura, en el caso en que la curva no presente esta forma, no se puede aplicar el método.

### Figura 9

*Curva del método Lambda*



Los requisitos que debe tener un método lambda son:

- Requisitos: modelo de primer orden, estable o integral con tiempo muerto
- Parámetros de diseño: constante de tiempo de lazo cerrado. ( $\lambda$ )
- Factor lambda

T representa el tiempo del proceso,  $k_p$  es la ganancia en porcentaje con respecto a la amplitud del escalón y L representa el tiempo muerto del proceso.

(Almachi Chiguano, J. W., & Naranjo Barrera, E. A. , 2021)

**Figura 10**  
*Método de sintonía Lambda*

PID	
Agresivo	Robusto
$K_c = \frac{1}{k_p} * \frac{\frac{L}{2} + T}{\frac{L}{2} + T}$	$K_c = \frac{1}{k_p} * \frac{\frac{L}{2} + T}{\frac{L}{2} + 3 * T}$
$T_i = T + L/2$	$T_i = T + L/2$
$T_d = \frac{TL}{L + 2T}$	$T_d = \frac{TL}{L + 2T}$

## Node Red

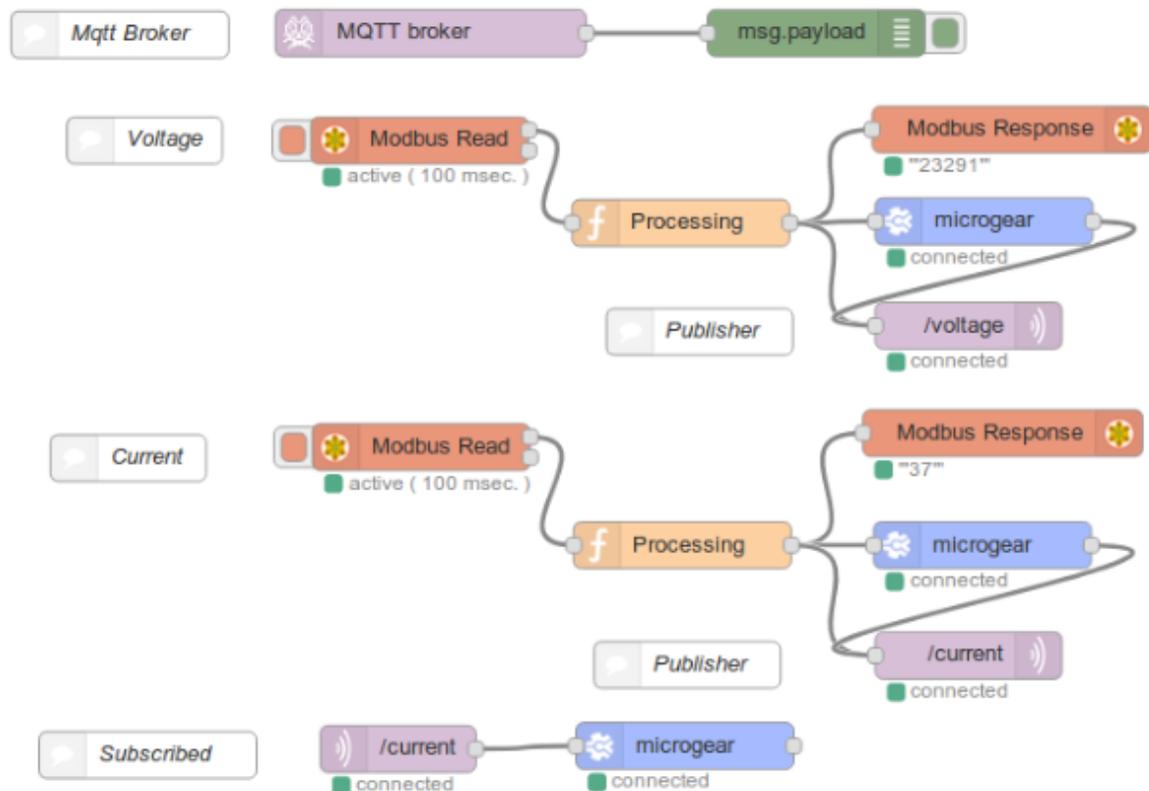
Node-RED es herramienta de desarrollo open-source basada en programación visual que fue creada por IBM para conectar dispositivos de hardware, APIs y servicios en línea.

Una herramienta fuerte, de fácil aprendizaje, y que no requiere altos conocimientos de programación que se ha consolidado como una de las principales aplicaciones de gestión y transformación de datos en tiempo real para soluciones IoT e Industria 4.0, permite conectar gráficamente bloques predefinidos, llamados nodos, para desarrollar una tarea concreta. La conexión de los nodos, generalmente una

combinación de nodos de entrada, nodos de procesamiento y nodos de salida, forman lo que se conoce como flow.

Entre los nodos disponibles se puede hallar desde protocolos estándar como MQTT, REST, Modbus, OPC-UA, Bacnet, Websocket, hasta integraciones a APIs de terceros como Microsoft Azure, Amazon Web Services, Twitter, Facebook y muchas más.

**Figura 11**  
*Nodo RED*



El nodo red, el microcontrolador actuará principalmente como publicador en el servidor Mosquitto. La herramienta que hará de subscriptor y que recogerá los datos publicados es Node-RED. Node-RED es una herramienta de programación visual

mediante la cual se le permite al usuario programar sin escribir código, ya que muestra visualmente las relaciones y las funciones. Está basado en navegador y permite añadir, eliminar y conectar entre sí nodos con el fin de que se comuniquen entre ellos. (Moral-Pérez, M. A., 2021)

En Node-Red se clasifican en tres tipos de nodos:

- **Iniciadores:** Son nodos que inician un flujo de datos, únicamente poseen un puerto de salida. Pueden ser de dos tipos:  
  
Activos: Se activan de una forma periódica bajo la supervisión de un evento determinado. Por ejemplo, que se envíe la fecha actual cada cinco segundos. Pasivos: Están a la escucha de algún evento. Es el caso de MQTT. Este nodo está a la espera de que se actualice un topic en el broker, y cuando lo hace recibe este dato actualizado.
- **Intermedios:** Son nodos que reciben un flujo de datos, realizan las funciones pertinentes dependiendo del nodo que se trate y generan una salida con el resultado de esa operación. Por lo tanto, tienen un puerto de entrada y otro de salida. En este tipo de nodos, cabe hacer referencia al “function node”, el cual nos permite incorporar directamente código propio de JavaScript para implementar nuestra propia función.
- **Terminales:** Se sitúan al final de las ramas del flujo de datos comentado anteriormente, por lo tanto, posee únicamente una salida. Se utiliza, sobre todo, para ejecutar otros flujos de la aplicación y para generar eventos, por ejemplo, el envío de datos al broker. (Moral-Pérez, M. A., 2021)

### ***Datos y variables en Node-RED***

Cuando se forma un dato en Node-RED, sólo podrán acceder a él, los nodos que se encuentren conectados, es decir, que pertenezcan al mismo flujo de datos. Por defecto, la variable se generará en el contexto del nodo, aunque existen tres tipos de contextos. (Moral-Pérez, M. A., 2021)

### ***Contexto de las variables***

Node-RED, proporciona un perfil de almacenamiento de la información, para que esta sea compartida, sin necesidad de que los nodos se hallen en el mismo flujo de datos. Esto es el “contexto” de las variables.

- El alcance que tenga un valor va a determinar con quién se comparte y por lo tanto su contexto. Hay tres niveles de alcance según su contexto:
- **Nodo:** Solo será visible para el nodo que estableció el valor y los nodos que se encuentren en su mismo flujo de datos.
- **Flujo (Flow):** Será visible para todos los nodos que se encuentren en la misma pestaña del flujo.
- **Global:** Será visible para todos los nodos que se encuentren en la herramienta Node-RED, independientemente si se encuentran conectados o en el mismo flujo de datos. (Moral-Pérez, M. A., 2021)

### ***Librería de nodos***

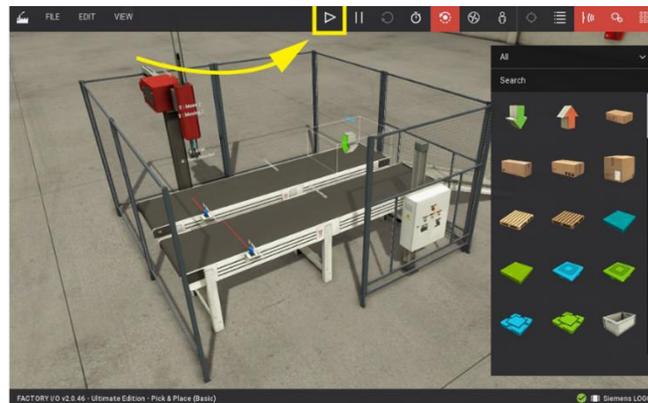
La librería de nodos es la medio de herramienta, siendo de software libre, y la facilidad por desarrollar nuevos nodos, confluyen en que la librería de nodos de Node-RED crezca cada día con nuevas aportaciones de la comunidad.

Actualmente existen más de 2500 nodos disponibles en la librería oficial de Node-RED, incluyendo integraciones de Smart Home, conversión entre protocolos IoT, funciones de geolocalización, autenticación vía OAuth2 y mucho más. (Pickdata, 2020).

## **Factory IO**

El Factory IO es un software de virtualización manejado para el aprendizaje de los sistemas de automatización que se encuentra en cualquier empresa. El programa admite construir una fábrica virtual utilizando una opción de piezas industriales, también incluye muchas escenas inspiradas en aplicaciones industriales típicas, que presentan varios niveles de dificultad. Este programa ofrece muchas posibilidades a la hora de elegir el controlador, los más utilizados en la industria son los PLC de los que se pueden encontrar de diferentes marcas y prestaciones. Sin embargo, también se pueden utilizar otros controladores como microcontroladores, SoftPLC y Modbus entre otros tipos de tecnología. El Factory IO es el programa indicado para realizar virtualizaciones, ya que es muy sencillo conectarlo con otros programas y ofrece una gran cantidad de opciones a la hora de crear la fábrica virtual. (Satorres Mira, S., 2021).

**Figura 12**  
*Factory IO Programa*

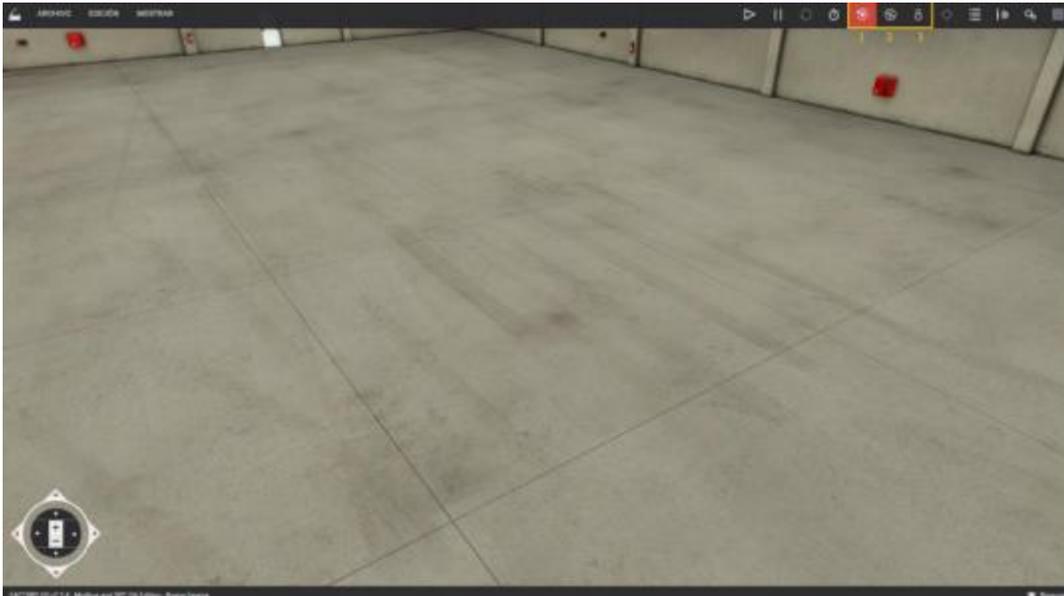


### ***Navegación***

La destreza para utilizar las cámaras en FACTORY IO constituye un papel fundamental en esta categoría. Como se sabe, el software que se está tratando genera una escena en tres dimensiones, por lo cual, debemos ser capaces de navegar en dicho espacio tridimensional sin dificultad alguna. Las cámaras importantes para interactuar con las distintas partes presentes en el programa y serán totalmente necesarias a la hora de crear o editar una escena. (Marín Sánchez, J., 2018)

## Figura 13

### *Tipos de cámaras*



### ***Orbit Cámara***

La Orbit camera es una cámara predeterminada e imprescindible en la construcción de escenas ya que es la única que admite al usuario el movimiento a través de las partes presentes en la escena sin producir colisión alguna con las mismas. Su funcionamiento es bastante intuitivo, girando alrededor de un punto de interés, que se encuentra indicado por un punto blanco. El punto de interés se establece haciendo doble clic izquierdo en cualquiera de las partes presentes en la escena. Además, cada vez que se mueve una parte (o selección de partes), el punto de interés se establece automáticamente en el centro de la pieza. Una vez definido el punto de interés, se podrá girar la cámara alrededor del mismo manteniendo presionado el botón derecho y arrastrando el ratón. Como inciso, es importante saber que los ítems que se arrastren desde la paleta se crearán a la altura definida por este punto, a excepción de las partes

que normalmente se colocan en el suelo, como cintas transportadoras, estaciones, etc.  
(Marín Sánchez, J., 2018)

**Tabla 5**

*Orbit cámara control Factory IO*

<b>CONTROL</b>	<b>ACCIÓN</b>
	Establece el punto de interés de la cámara.
<b>Doble clic izquierdo</b>	La cámara girará alrededor de este punto y se crean nuevas piezas a distinta altura.
<b>Clic derecho + arrastre</b>	Gira la cámara alrededor del punto de interés.
<b>Clic rueda + arrastre</b>	Desplaza la cámara horizontalmente.
<b>Rueda del ratón</b>	Acerca la cámara hacia dentro y hacia fuera
<b>Backspace</b>	Restablece la cámara a la posición y rotación predeterminadas.

### ***Fly Cámara***

Es utilizada para moverse libremente en el espacio tridimensional. A diferencia de la anterior, esta cámara colisiona con las distintas partes en se encuentran en la escena, aunque es totalmente imperceptible para los sensores que en ella se hallan. De modo que, con esta cámara se podrá inspeccionar el espacio 3D de forma segura y rápida, logrando alcanzar perspectivas imposibles por parte de sus alternativas. (Marín Sánchez, J., 2018)

**Tabla 6***Cámara de control Factory IO*

<b>CONTROL</b>	<b>ACCIÓN</b>
<b>Doble clic izquierdo</b>	La cámara mira hacia donde apunta el cursor del mouse.
<b>Clic derecho + arrastre</b>	Gira la cámara
<b>Rueda del ratón</b>	Desplaza la cámara verticalmente
<b>Clic izquierdo + derecho</b>	Mueve la cámara hacia delante
<b>W</b>	Mueve la cámara hacia delante
<b>S</b>	Mueve la cámara hacia atrás
<b>A</b>	Mueve la cámara hacia la izquierda
<b>D</b>	Mueve la cámara hacia la derecha

***First person***

Esta cámara muestra la visión de una persona de 1,80 metros de altura y debe ser utilizada a la hora de simular a una persona en una fábrica virtual. Choca con las partes de la escena y, de forma predeterminada, no es detectada por los sensores.

NOTA: Es posible configurarla para que sea detectada por los sensores. Para ello se debe abrir la consola tal y como indican las Figuras en la misma camera.fp\_detected = 1. (Marín Sánchez, J., 2018)

**Tabla 7***Control de First person*

<b>CONTROL</b>	<b>ACCIÓN</b>
<b>Doble clic izquierdo</b>	La cámara mira hacia donde apunta el cursor del mouse.
<b>CONTROL</b>	<b>ACCIÓN</b>
<b>Clic derecho + arrastre</b>	Gira la cámara
<b>Clic izquierdo + derecho</b>	Mueve la cámara hacia delante
<b>W</b>	Mueve la cámara hacia delante
<b>S</b>	Mueve la cámara hacia atrás
<b>A</b>	Mueve la cámara hacia la izquierda
<b>D</b>	Mueve la cámara hacia la derecha
<b>Space</b>	Salta

**Figura 14**  
*Cámara First person*

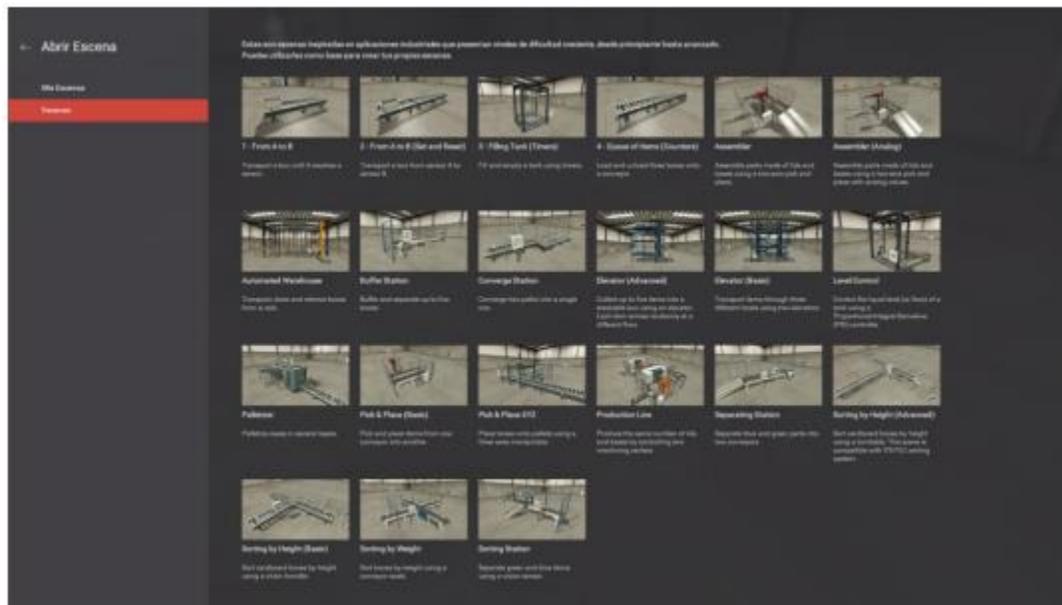


### ***Apertura de una escena***

El proceso para abrir una escena es sencillo, solo será necesario hacer clic en Archivo y seleccionar la opción abrir. FACTORY IO ofrece la posibilidad de abrir escenas de forma inmediata pulsando Ctrl + O en cualquier momento. Una vez hecho esto, se seleccionará la escena que se desee haciendo doble clic izquierdo sobre la misma.

Tal y como se ha comentado anteriormente, FACTORY IO ofrece escenas ya creadas, escenas típicas que pueden encontrarse en la industria. La dificultad en cuanto a la elaboración del código necesario para controlar las escenas va agrandando de una forma progresiva para facilitar el aprendizaje del usuario. Algo frecuente es aprovechar las escenas, usarlas como punto de partida para una nueva, así se tendrá que abrir la escena, modificarla según las necesidades particulares que requiera para guardarla con un nuevo nombre. De este modo, se mantendrá la escena original intacta en todo momento. (Marín Sánchez, J., 2018)

**Figura 15**  
*Escenas del sistema*



Una vez creada la escena que requiere, se recomienda controlar manualmente una escena antes de controlar la misma con un controlador externo. (Marín Sánchez, J., 2018)

## Plataforma Ubidots

Ubidots es una plataforma para construir nuevos productos y aplicaciones a partir de objetos conectados a internet donde se puede realizar desde una aplicación que registre el agua que se requiere durante el día y alerte con un mensaje de texto si se excede cierto nivel, hasta un bote que registre su temperatura en tiempo y eficacia. (Espinosa Apolo, B. A., & Orellana Guayas, M. E., 2021)

**Figura 16**  
*Plataforma Ubidots*



Este es un servicio gratuito que permite a los usuarios crear un entorno de computación en la nube asequible, fiable y utilizable en las plataformas IoT. Ubidots está especializado en dar soluciones a hardware y software conectadas para monitorear, controlar y automatizar procesos. Los principales beneficios que ofrece la plataforma son:

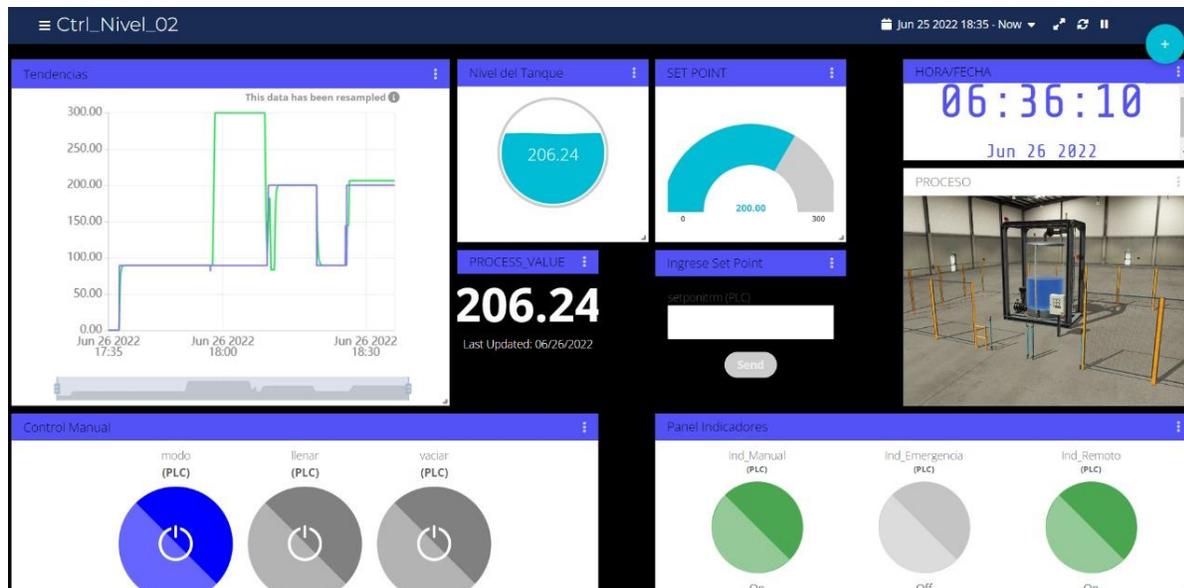
- La conexión del hardware con la nube y diversas bibliotecas.
- Configuración de varias formas automática, configurable en sus propiedades y su apariencia en los dispositivos para duplicar el proceso en nuevos dispositivos.
- Modificación de la API (Application Programming Interface).
- Supervisión para el análisis de datos de las aplicaciones con integraciones API.
- Conversión de datos de origen en información con variables sintéticas.
- Diseño para cuadros de mando analizados en el tiempo real para análisis de datos y control del dispositivo.

- Mejora la función de compartir datos con enlaces públicos o integrando cuadros de mando o widgets en aplicaciones web privadas y móviles.
- Los comandos “KIL SWICH” o “RESTART” se activan cuando el hardware ha estado inactivo durante mucho tiempo.
- La asignación de permisos y restricciones para cualquier usuario que interactue con los mandos o dispositivo.

Ubidots maneja cuatro términos necesarios para el manejo de los datos en la nube:

- Data Source: maneja un conjunto de información haciendo referencia a un dispositivo, donde cada data source utiliza una o más variables.
- Variable: maneja un conjunto actual de una variable en un momento determinado.
- Value: es un valor actual de una variable en un momento determinado.
- Event: permite tomar una acción en un instante determinado permitiendo una interacción con cada elemento de Ubidots a través de su API para poder modificarlo o eliminarlo. (Espinosa Apolo, B. A., & Orellana Guayas, M. E., 2021)

**Figura 17**  
*Plataforma Ubidots funcionando*



## Siemens y el internet de las cosas

Las organizaciones están experimentando una transformación como consecuencia de las tecnologías digitales. El análisis de grandes volúmenes de datos afecta los procesos empresariales y la forma en que se toman las decisiones; las redes sociales reflejan las acciones, interacciones y consumos de los clientes; las aplicaciones móviles y la computación en las nubes son esenciales para brindar servicios; el Internet de las Cosas (IoT por sus siglas del inglés Internet of Things) vincula productos a través de sensores creando nuevas posibilidades para crear valor o reducir costos; los robots, drones e impresión 3D, constituyen conductores clave en la evolución de las cadenas de suministros. Para Latino América, se estima un crecimiento del 21% en el tráfico de Internet; 6,7% en usuarios de Internet; un 8% en conexiones de dispositivos (CAGR de

2015 a 2020) (CISCO, 2016). En particular, el desarrollo de propuestas de valor basadas en el Internet de las Cosas está movilizandoo el sector de los fabricantes y desarrolladores de chips, sensores, software y las tecnologías utilizadas para transmitir datos desde y hacia dispositivos conectados. (Sánchez, 2018)

### **Librería LMQTT**

Message Queuing Telemetry Transport (MQTT) es un protocolo de mensajería estándar por comunicaciones M2M de publicación/suscripción o por sistemas IoT. Puede llegar a requerir una ampliada de banda de RAM mínima y funciona sobre el protocolo TCP/IP, en el día utiliza en una amplia variedad de protocolos que sirve en la fabricación y telecomunicaciones. Siemens proporciona la librería LMQTT, por tal manera implementa el protocolo MQTT en programas de PLCs de Siemens. La librería permite ser cliente en esquema del protocolo, además también permite publicar y suscribirse a un broker. En el propio PLC sería el publicador de los protocolos MQTT dando conectividad con el PLC a internet. (Guman López de Briñas, G. , 2021)

## **PLC Sim advanced**

El programa informático Simatic PLCSIM Advanced genera controladores virtuales para simular los que pertenecen al tipo **S7-1500** y comprobar su funcionamiento se pueden simular los del tipo 1500 para la generación de estos controladores.

Una de las características más importantes del S7 PLCSIM es que permite variar los tiempos en los procesos simulados; al acelerar las emulaciones se obtienen datos o resultados rápidamente, además es capaz de realizar simulaciones completas de lógica programada.

La instalación del programa S7 PLCSIM se puede hacer con TIA Portal, tal como si fuera un PLC Real. Primero se coloca el dispositivo, se configura la dirección IP y el programa que se quiere simular y se procede a la exportación. ( Gandhi, s.f.)

### ***Ventajas del programa S7 PLCSIM Advanced***

El SIMATIC S7-1500 puede ser simulado con el PLCSim Advanced para tener un modelo virtual más eficiente y en menor tiempo. Se presenta algunas de las ventajas de este programa como simulador de PLC.

- Emulación elevada de una gran cantidad de funciones. Permite visualizar las funciones tal como si se estuviese llevando a cabo físicamente en una instalación.
- Generación de varios controladores virtuales a la vez. Se pueden desarrollar distintos controladores con funciones independientes entre cada uno.
- Desarrollo de trabajos para la industria 4.0. En la digitalización, favorece el mantener una buena capacidad de comunicación entre diferentes servidores.

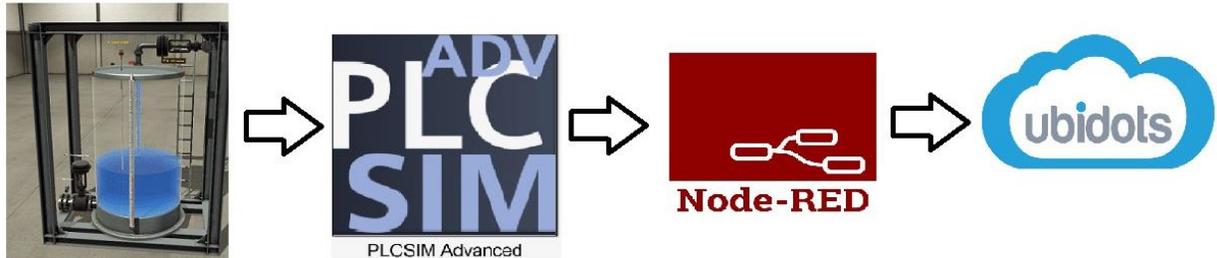
- Compatibilidad con otros productos de SIEMENS. Este programa ofrece distintas conexiones sin que se presenten fallos. ( Gandhi, s.f.)

## Capítulo III

### Desarrollo y Simulación

#### Descripción general del proyecto

Para el desarrollo del presente proyecto se ha utilizado varios softwares que permiten la implementación de una arquitectura IoT con un proceso virtual, la aplicación del protocolo MQTT permite brindar acceso remoto tanto al control como al monitoreo de sensores, actuadores y variables del proceso. La imagen 18 muestra una representación de la arquitectura desarrollada y el software aplicado. El proceso de nivel se lo implementa en Factory I/O, en el cual se contempla los botones de comando y señales de instrumentación necesarias para la implementación del control. El algoritmo de control y bloques de datos se lo desarrolla en Tia Portal de Siemens, el cual se ejecuta en un PLC virtual el cual se lo tiene gracias a PLC Sim Advanced (se simula un PLC 1500 debido a compatibilidad de software), la información necesaria para IoT es creada y encapsulada dentro de un bloque de datos de programa, este bloque de datos es accesible para Node red en atributos tanto de lectura como escritura. Finalmente, con la información que posee, Node red se comunica con el broker MQTT en cual se encuentra en la nube, para este caso en específico se usa la plataforma de desarrollo UBIDOTS, el cual provee un link de enlace mediante el cual se puede acceder al dashboard implementado.

**Figura 18***Arquitectura y la nube*

La tabla 8 muestra las versiones de software utilizados en el desarrollo del proyecto.

Los aspectos y requerimientos relacionados a cada parte de la implementación son descritos a continuación.

**Tabla 8***Tabla de las versiones de los softwares utilizados*

PROGRAMA	VERSIÓN
Tia Portal	V 16.0
Factory IO	V 2.5.0
PLCSim Advance	V 2.0
Nodo Red	V 3.0

## Descripción del proceso y componentes Factory IO

En la figura 19 se observa el proceso desarrollado, el cual consta de forma general de un tanque contenedor de líquido, un tablero de control, una válvula de carga, una válvula de descarga, un sensor de nivel, una baliza de indicación, una sirena. La interacción de los elementos está gobernada por el software mientras que el control está sujeto a los valores recibidos del controlador. A continuación, se describe los componentes del proceso.

### Figura 19

*Imagen General del proceso*



### ***Tanque de agua***

Es un tanque contenedor que simula una capacidad de 300 cm<sup>3</sup> de almacenamiento, por el frente presenta una escala de medición del nivel.

**Figura 20**

*Tanque de agua*



### ***Válvulas de Carga y Descarga.***

El sistema consta de dos válvulas proporcionales una ubicada en la parte superior cuya apertura controla el flujo de agua que ingresa al tanque y otra ubicada en la parte inferior la cual controla el flujo de agua que sale del tanque contenedor.

**Figura 21**  
*Válvulas de Carga y Descarga*



### **Baliza**

La baliza de indicación tiene 3 colores los cuales están relacionados a distintos modos de operación del sistema, el color rojo indica la activación del paro de emergencia, el color verde indica el funcionamiento en modo automático del control, mientras que el color amarillo indica la operación en modo manual.

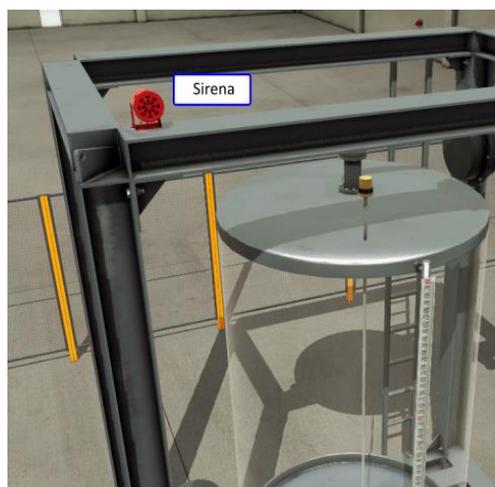
**Figura 22**  
*Baliza del sistema*



### **Sirena**

La sirena se activa y emite una señal sonora únicamente en caso de activarse el paro de emergencia.

**Figura 23**  
*Sirena de alerta*



### **Tablero de Control**

Dentro del tablero de control se cuenta con varios dispositivos cuyas acciones son las siguientes:

Pulsador Manual/ Automático, permite seleccionar si el sistema trabaja mediante un control manual y realizar tanto el llenado como el vaciado del tanque mediante la acción de los pulsadores correspondientes; si el modo de trabajo seleccionado es automático se ejecutan las acciones correctivas para mantener el nivel del agua en el valor de consigna ingresado.

Pulsador “Llenado”, está activado en el modo manual permite abrir la válvula de llenado al 100% y permitir el paso de agua hacia el tanque. Se la puede activar en el modo automático con el fin de realizar una perturbación al control y verificar su respuesta.

Pulsador “Vaciado”, está activado en el modo manual permite abrir la válvula de vaciado al 100% y permitir el paso de agua del tanque hacia el desfogue. Se la puede activar en el modo automático con el fin de realizar una perturbación al control y verificar su respuesta.

Potenciómetro “Set Point”, permite ingresar el valor consigna mediante una entrada analógica del controlador.

Selector “SP\_Local/SP\_Remoto”, selecciona el punto de consigna para el funcionamiento del proceso. Cuando se lo establece en modo local, el set point corresponderá al valor ingresado mediante el potenciómetro del tablero; mientras que, si establece el modo remoto, el set point será el valor receptado desde el dashboard remoto.

Displays, muestran tanto el valor del set point establecido y el valor de proceso en una escala de 0 al 100%.

Luces piloto, se activan de acuerdo al modo de operación del sistema, manual, automático o si se ha presionado el botón de paro de emergencia.

**Figura 24**  
*Tablero de control del proceso*



La relación de cada uno de los elementos descritos y las entradas/salidas del controlador se establecen como lo describe la tabla 9:

**Tabla 9**

*Tabla de direccionamiento de variables*

Descripción	Dirección	Tipo
	<b>Controlador</b>	
<b>Botón paro de emergencia</b>	I0.0	Booleano
<b>Selector modo</b>	I0.1	Booleano

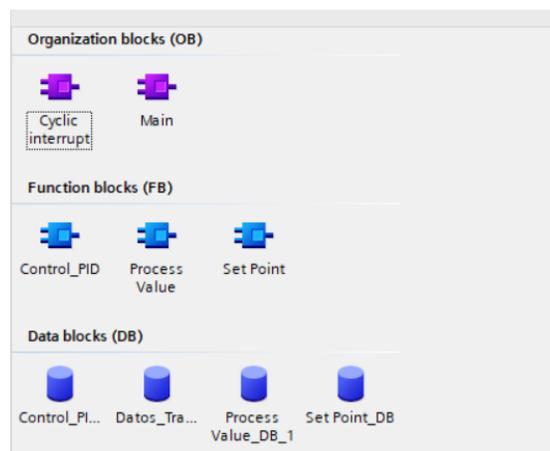
<b>Descripción</b>	<b>Dirección</b>	<b>Tipo</b>
	<b>Controlador</b>	
<b>Pulsador llenado</b>	I0.2	Booleano
<b>Pulsador vaciado</b>	I0.3	Booleano
<b>Selector SetPoint</b>	I0.4	Booleano
<b>Set Point Local</b>	IW100	Word
<b>Sensor de nivel</b>	IW102	Word
<b>Indicador modo manual</b>	Q0.0	Booleano
<b>Indicador modo automático</b>	Q0.1	Booleano
<b>Sirena</b>	Q0.2	Booleano
<b>Indicador Emergencia</b>	Q0.3	Booleano
<b>Baliza luz roja</b>	Q0.4	Booleano
<b>Baliza luz verde</b>	Q0.5	Booleano
<b>Baliza luz amarilla</b>	Q0.6	Booleano
<b>Válvula de descarga</b>	QW100	Word
<b>Válvula de llenado</b>	QW102	Word
<b>Display Set Point</b>	QW104	Word
<b>Display Proces</b>	QW106	Word
<b>Value</b>		

## Descripción de los bloques de control y el algoritmo implementado.

El control de la aplicación se lo desarrolla en el software Tia Portal V16 de Siemens, el programa consta de dos bloques de organización, el principal donde se ejecuta el barrido de entradas, ejecución del algoritmo y actualización de salidas y un bloque de interrupción cíclica que ejecuta el bucle de control PID. Adicionalmente se tiene 3 bloques de funciones en los cuales se realizan los escalamientos y conversión de tipos de datos para el PV, SP y control PID. Finalmente se implementaron 4 bloques de datos 3 de los cuales están asociados a los bloques de funciones descritos anteriormente y uno adicional el cual contiene las variables para intercambio de datos con node red. Estos componentes se pueden observar en la figura 25.

### Figura 25

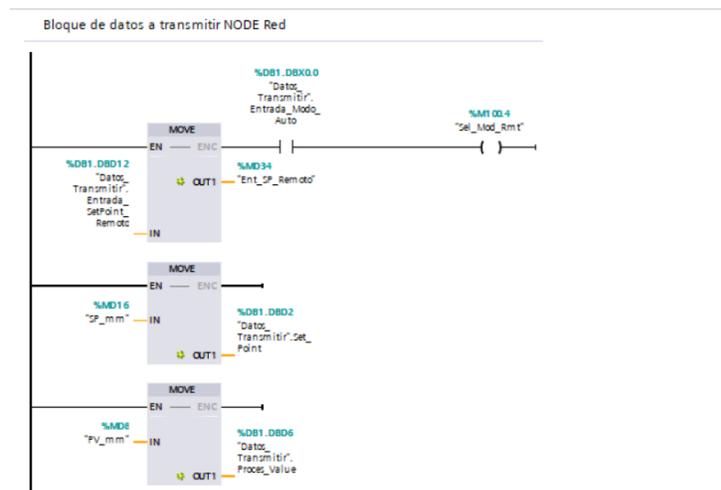
*Bloques de funciones creados para el control*



Dentro de la rutina Main se ejecuta varios subprocesos, cuyos segmentos se describen a continuación, el programa completo se registra en el Anexo 1 del presente documento.

El segmento 1 está relacionado al bloque de comunicaciones para la interacción con node red, la figura 26 muestra una parte de la programación del mismo, mientras que la figura 27 muestra los nombres, tipos de datos, atributos y direcciones creadas.

**Figura 26**  
Segmento de programación 1



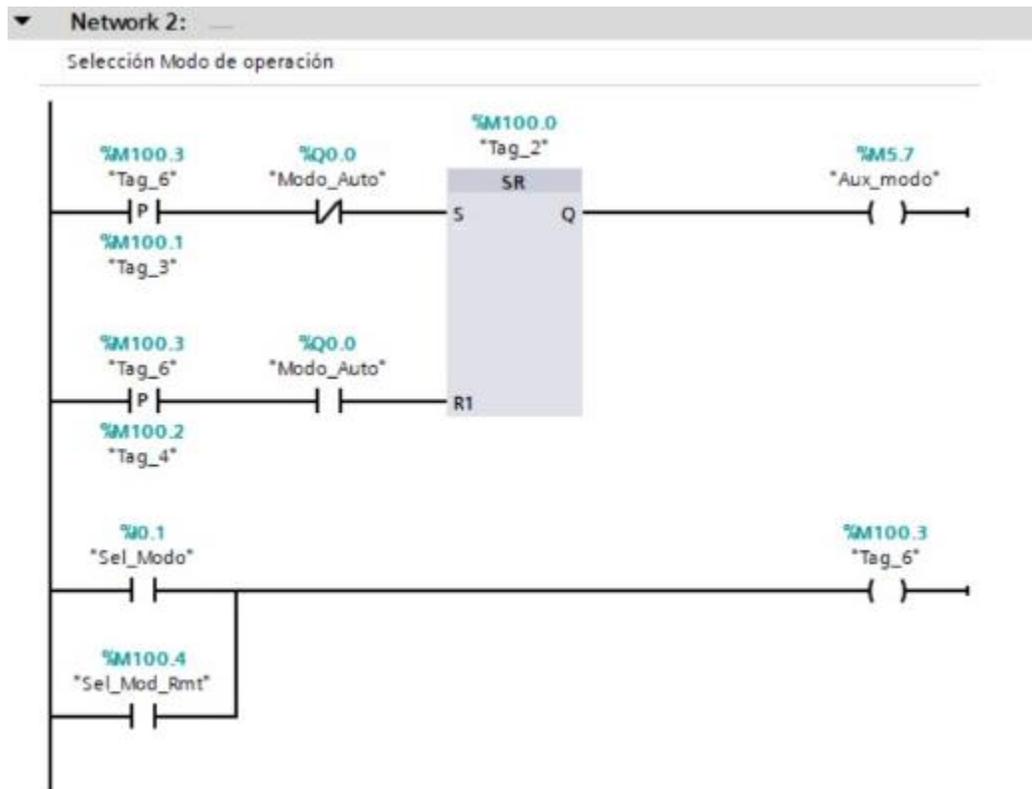
**Figura 27**  
Tipos de datos, atributos y direcciones creadas

Datos_Transmitir											
	Name	Data type	Offset	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	Supervis...	Cor
1	Static										
2	Entrada_Modo_Auto	Bool	0.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
3	Set_Point	Real	2.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
4	Proces_Value	Real	6.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
5	Entrada_Btn_Vaciado	Bool	10.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
6	Entrada_Btn_Llenado	Bool	10.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
7	Entrada_SetPoint_Re...	Real	12.0	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
8	Puls_emergencia	Bool	16.0	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
9	Ind_Manual	Bool	16.1	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
10	Ind_Remoto	Bool	16.2	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		

En el bloque 2 y 3 se utiliza un comando Set-Reset para desarrollar la lógica de selección del modo de operación manual o automática y que esta pueda ser cambiada desde el tablero de control y también desde el dashboard remoto. La figura 28 muestra la lógica implementada para las condiciones expuestas.

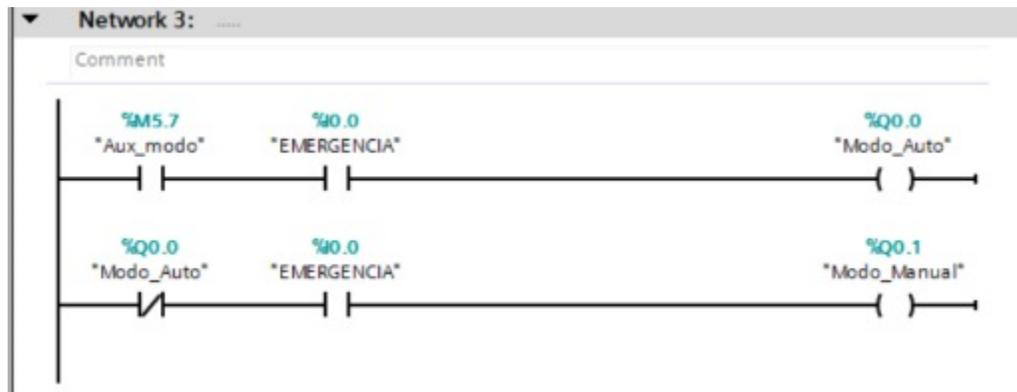
**Figura 28**

*Segmento de programación 2*



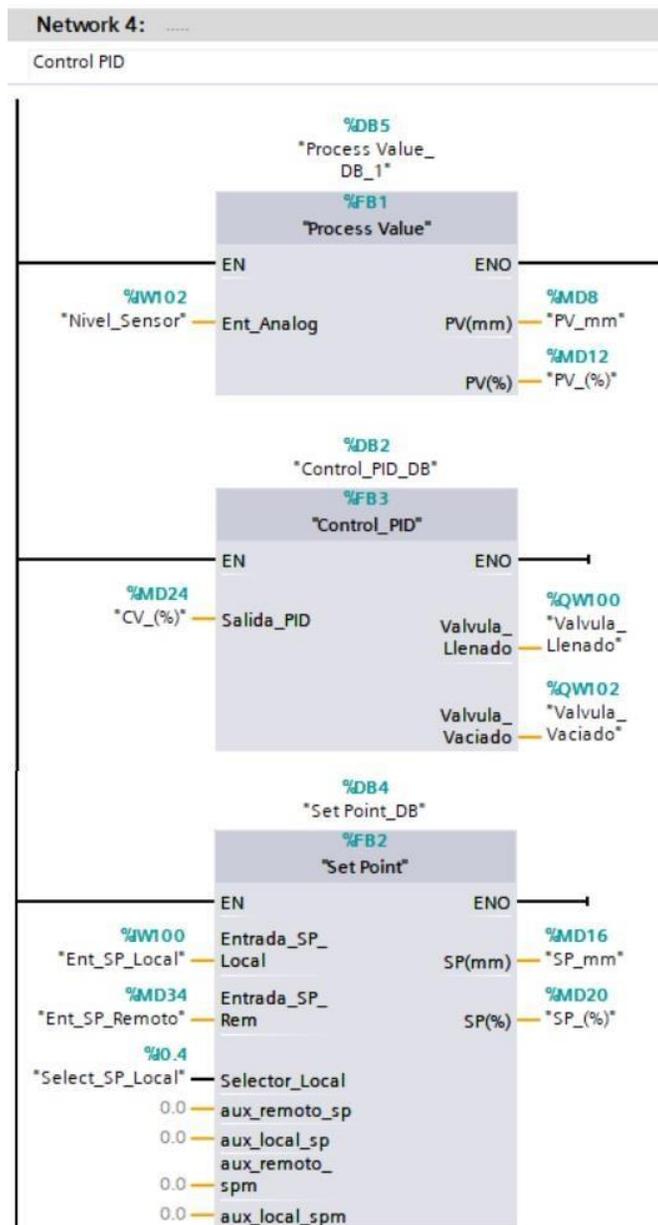
A continuación, en el bloque 4 se ejecuta el escalamiento tanto de SP y PV para enviar estos datos al bloque de control PID, como se observa en la figura 29.

**Figura 29**  
*Segmento de programación 3*



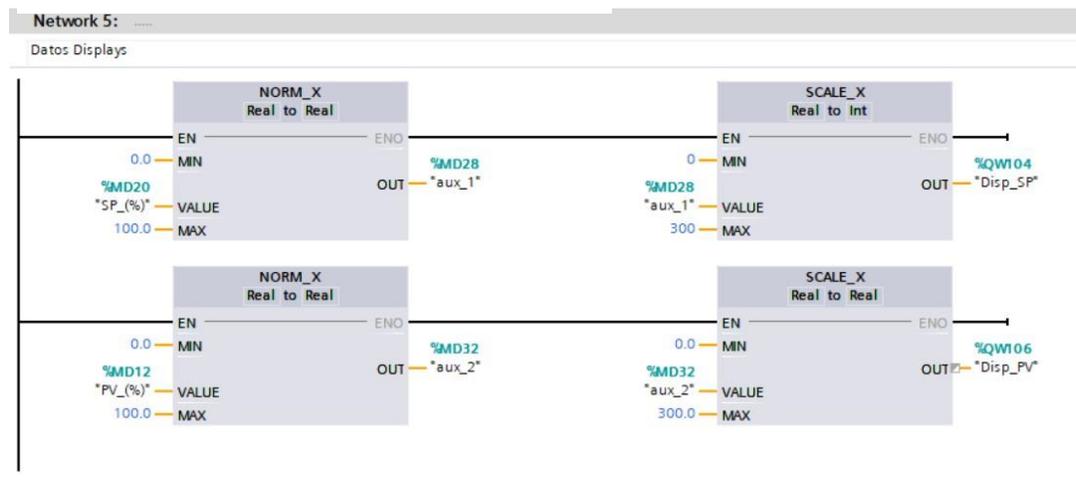
En el bloque 5 se realiza la normalización y escalamiento de valores para ser mostrados en los displays indicadores que están ubicados en el tablero de control. Ver figura 30.

**Figura 30**  
 Segmento de programación 4



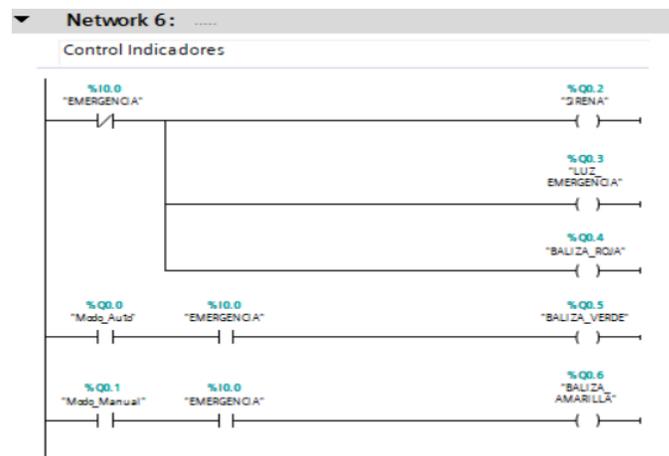
Dentro del bloque 6 se realiza las configuraciones de control de indicadores de modo de control y emergencia activa.

**Figura 31**  
Segmento de programación 5

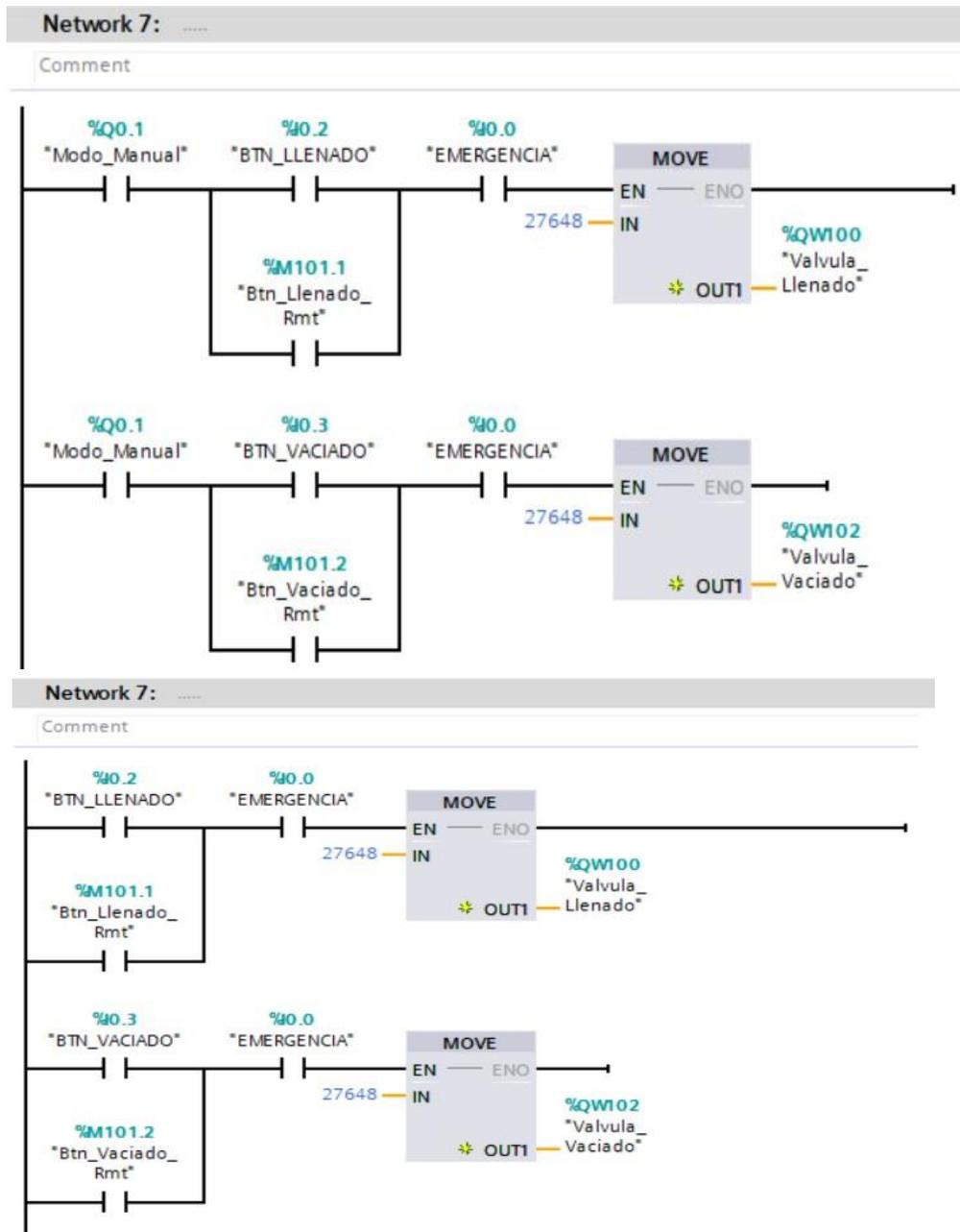


Finalmente, en el bloque 7 se activa la operación manual de las válvulas tanto de llenado como vaciado del proceso. Como se indica en la figura 32, esta acción no está permitida cuando el sistema opera en modo automático. Para la verificación del sistema de control se quiere realizar una perturbación al sistema por lo que se modifica este segmento como lo muestra la figura 33.

**Figura 32**  
Segmento de programación 6



**Figura 33**  
 Segmento de programación modificado



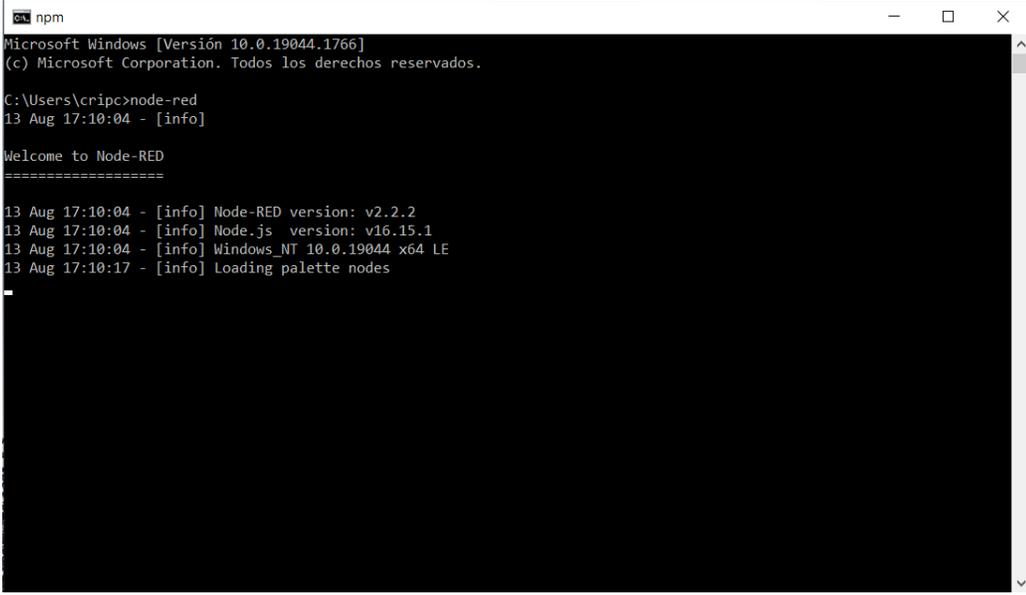
## Configuración y creación de variables en Node-Red.

Una vez instalado el software Node-Red es necesario ejecutar el comando:

```
Node-red
```

La figura 34 muestra el resultado de la ejecución del comando, a continuación, se puede acceder a la interfaz de configuración de node red, para ello se debe acceder a la dirección 127.0.0.1:1880, desde cualquier navegador web y se obtendrá un resultado como el que se muestra en la figura 35.

**Figura 34**  
*Ejecución de node Red*



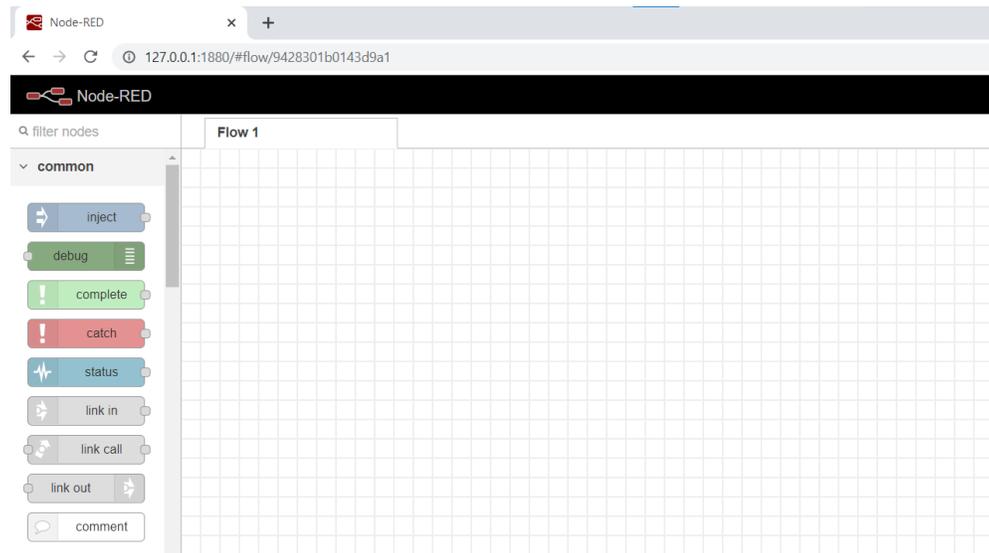
```
npm
Microsoft Windows [Versión 10.0.19044.1766]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\cripc>node-red
13 Aug 17:10:04 - [info]

Welcome to Node-RED
=====

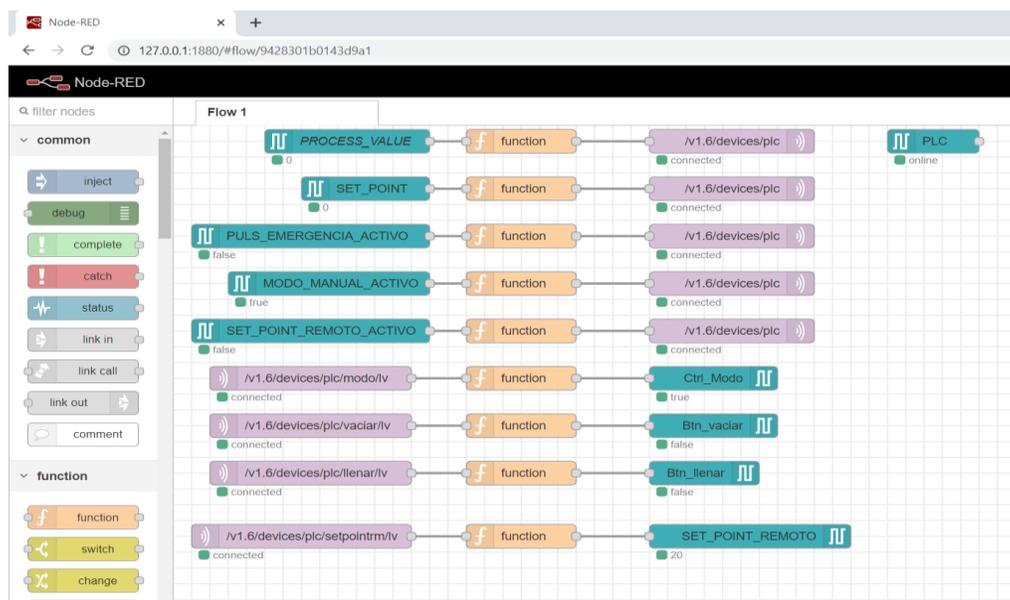
13 Aug 17:10:04 - [info] Node-RED version: v2.2.2
13 Aug 17:10:04 - [info] Node.js version: v16.15.1
13 Aug 17:10:04 - [info] Windows_NT 10.0.19044 x64 LE
13 Aug 17:10:17 - [info] Loading palette nodes
-
```

**Figura 35**  
Interfaz de Node Red



Una vez que se accede a la interfaz de configuración se procede a instalar las librerías y crear las variables necesarias. La figura 36 muestra las variables creadas para el desarrollo del presente proyecto.

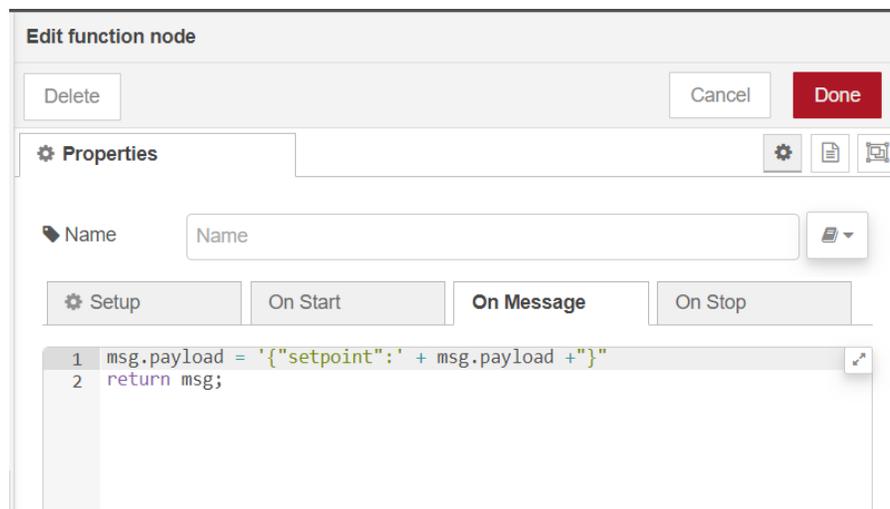
**Figura 36**  
Flujo del proceso Node Red



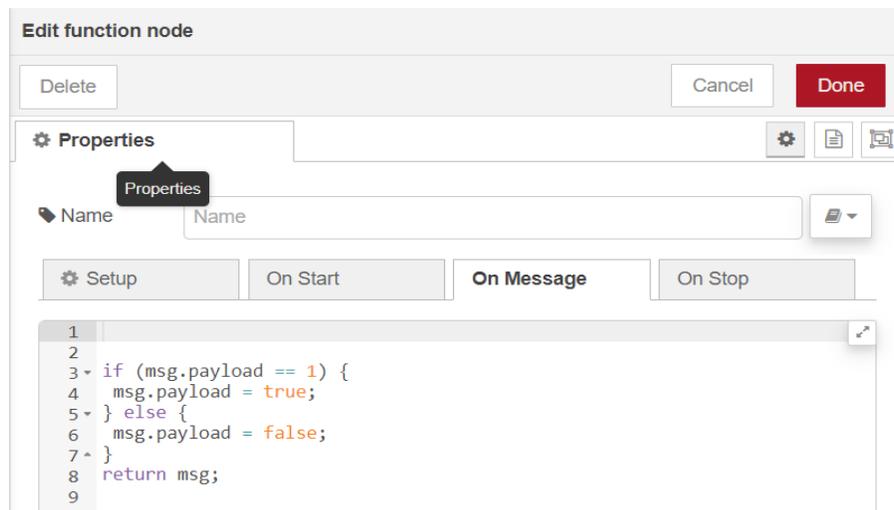
Como se observa en la imagen 36 se crearon 9 variables para el intercambio de información con el servidor MQTT, de las cuales 5 se escriben en el servidor mientras que 4 son datos que provienen del servidor y cuyos datos son escritos en la dirección de memoria asignada del PLC.

Para el intercambio de información es necesario estructurar los datos en los formatos requeridos tanto por el PLC como por el bróker MQTT por lo que es necesario incorporar bloques de funciones en los cuales se da la reestructuración de texto y formato de datos, la figura 37 indica la operación de la función de una variable del PLC para ser enviada al bróker. Mientras que la figura muestra el bloque de función de una variable enviada desde el broker hacia el PLC.

**Figura 37**  
Nodo de función



**Figura 38**  
Bloque de función Node Red



Dentro de las imágenes anteriores se identifican variables de color violeta que indica que son variables de comunicación MQTT, por lo que es necesario configurar tanto la dirección del broker (en este caso se utiliza la IP proporcionada por Ubidots para la comunicación) como el tópico al cual se suscriben, la figura 39 muestra los datos de configuración del broker, mientras que en la figura se indica la suscripción al tópico deseado.

**Figura 39**  
Configuración del servidor Ubidots

Edít mqtt out node > Edít mqtt-broker node

Delete Cancel Update

**Properties**

Name SER\_UDI

Connection Security Messages

Server 169.55.61.243 Port 1883

Connect automatically  
 Use TLS

Protocol MQTT V3.1 (legacy)

Client ID Leave blank for auto generated

Keep Alive 60

Session  Use clean session

**Figura 40**  
Configuración del Nodo MQTT

Edít mqtt out node

Delete Cancel Done

**Properties**

Server SER\_UDI

Topic /v1.6/devices/plc

QoS 1 Retain false

Name Name

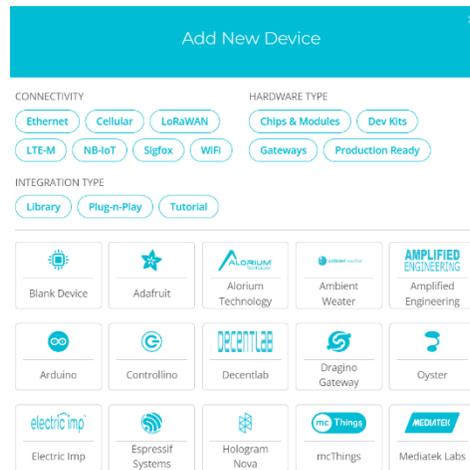
Tip: Leave topic, qos or retain blank if you want to set them via msg properties.

## **Creación de dispositivos y dashboard en Ubidots.**

Para el acceso a la información a través del internet es necesario contar con un servidor el cual aloje la información y además permita la interacción entre el sistema y el host del usuario; para ello se tienen diferentes alternativas como: arrendamiento de una de IP pública y asignar esta dirección a un servidor dentro del cual corra el software que brinde el servicio requerido. Otra opción sería el arrendamiento de una máquina virtual, la cual posee ya un a IP pública y las políticas de seguridad y acceso están sujetas al contrato de arrendamiento con el proveedor del servicio. Adicionalmente se puede contratar un servidor dedicado a funciones de IoT, sobre el cual se puede implementar el protocolo requerido y realizar los ajustes y configuraciones necesarias para el proyecto, en esta alternativa, los proveedores incluyen opciones de uso de manera gratuita los cuales incluyen restricciones como tiempos de prueba, cantidad de intercambio de información limitada, número de dispositivos accesibles limitado; sin embargo los mismos proveedores ofrecen opción de pago las cuales no tienen las restricciones mencionadas, es por ello que para el desarrollo de la presente investigación se optó por esta solución y se seleccionó la plataforma Ubidots.

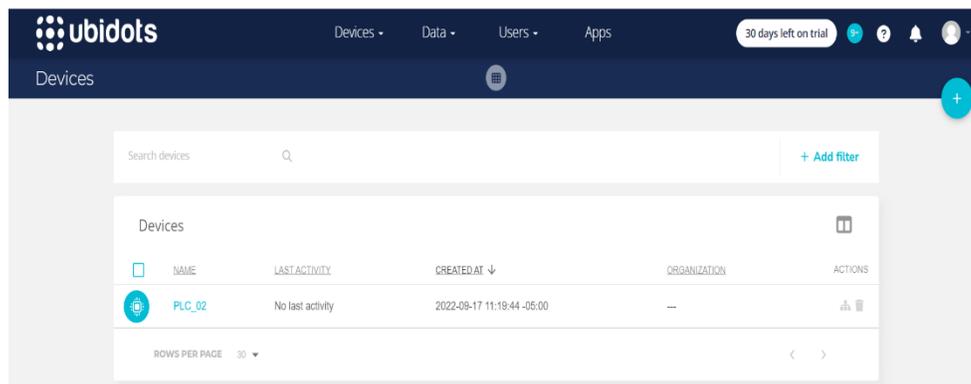
Inicialmente se debe configurar el tipo de comunicación a establecer como se indica en la figura 41, para este caso se utiliza la comunicación de protocolo Ethernet, y se configura un dispositivo en blanco.

**Figura 41**  
Creación del servidor Ubidots.



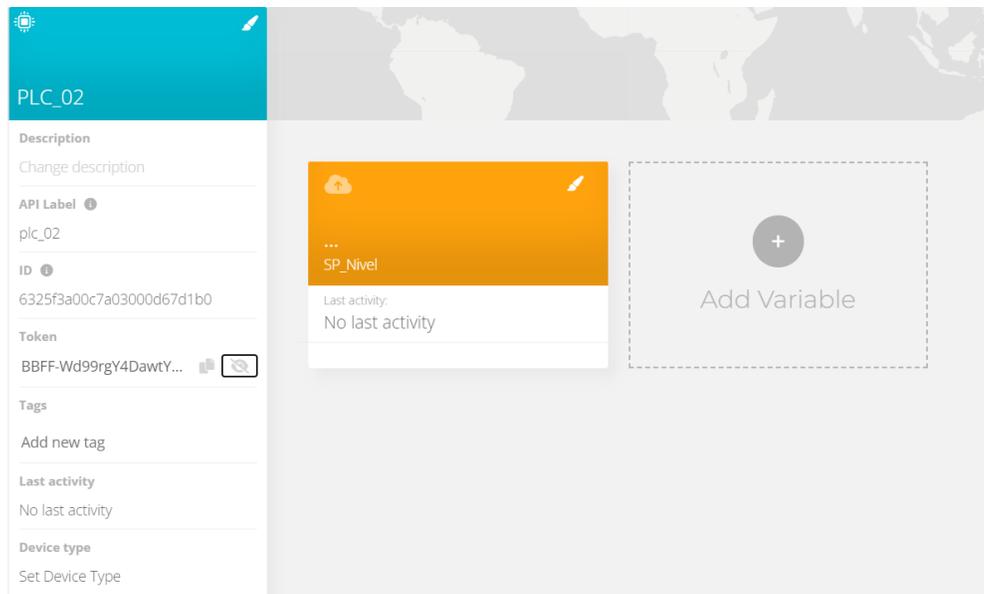
Una vez seleccionado tanto la conectividad como el tipo de dispositivo, se tiene una instancia como la que se muestra en imagen.

**Figura 42**  
Conectividad con el Dispositivo Ubidots.



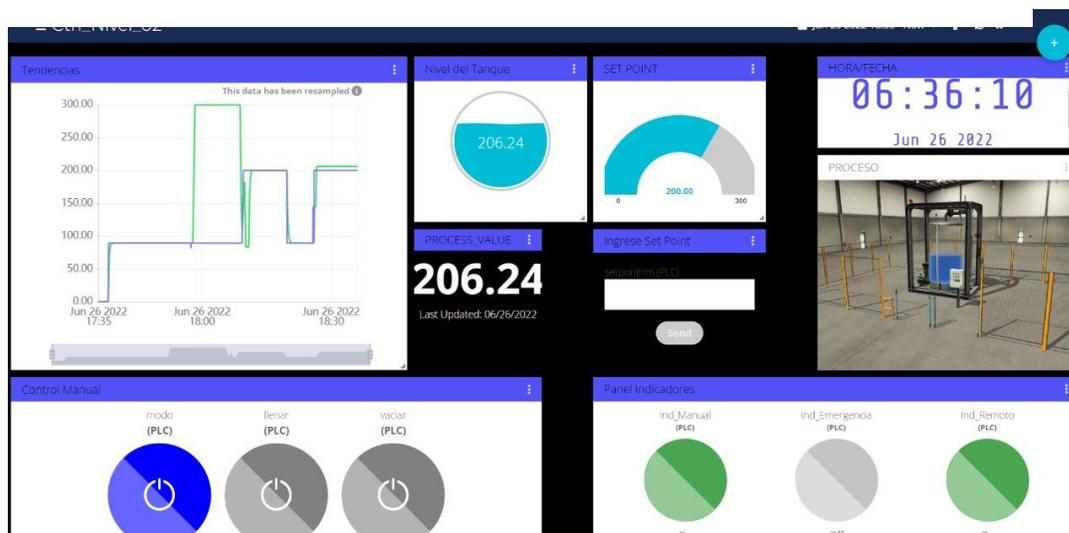
Dentro del dispositivo es necesario crear las variables las cuales se manejan como tópicos independientes para el intercambio de información. Como se puede observar en la figura. Una vez creadas las variables, éstas pueden asociarse a un widget del dashboard según los requerimientos.

**Figura 43**  
Creación de variables



Como se muestra en la figura 44, el dashboard desarrollado cuenta con botones, indicadores, grafico de tendencias e ingreso de valores para facilitar la interacción con el proceso. Los valores de tendencias son almacenados en el mismo servidor y se las puede exportar para análisis posteriores.

**Figura 44**  
Gráfico de tendencias e ingreso de valores



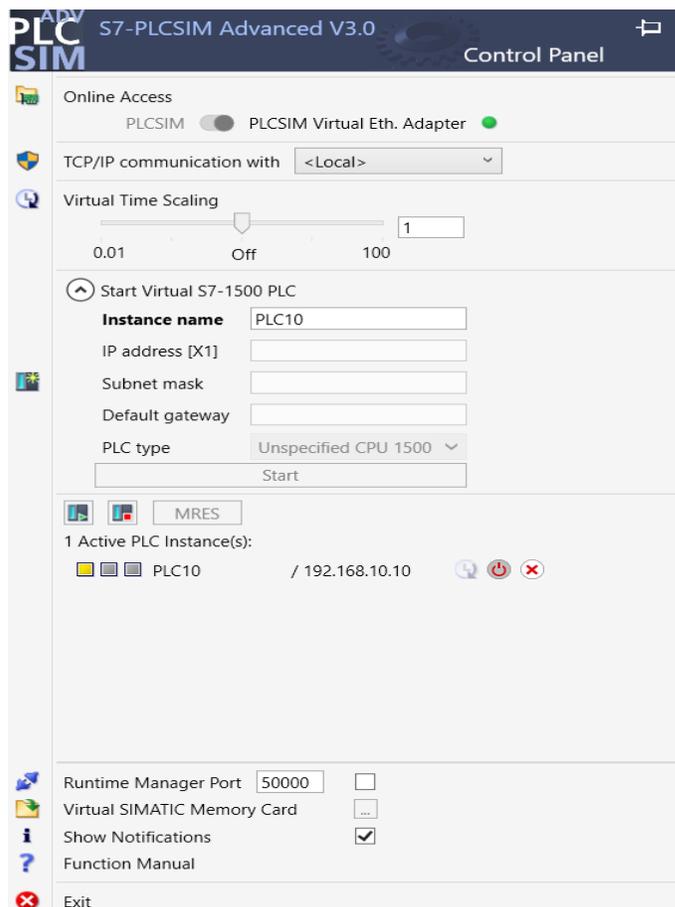
## Capítulo IV

### Experimentación y Resultados

#### Prueba de la Instancia PLC-Sim y Tia Portal

Para iniciar las pruebas de comunicación es necesario crear la instancia del PLC y asignar una IP, como se observa en la figura 45 la dirección asignada al PLC es la 192.168.10.10, mientras que como se observa en la imagen 45 la dirección IP asignada a la computadora es la 192.168.0.101, ambas tienen una máscara de subred 255.255.255.0

**Figura 45**  
Dirección Asignada



**Figura 46**  
Direccionamiento IPv4

```

C:\WINDOWS\system32\cmd.exe

Sufijo DNS específico para la conexión. . . :
Vínculo: dirección IPv6 local. . . . . : fe80::9dbc:911b:7f5:b0e2%10
Dirección IPv4. . . . . : 192.168.0.101
Máscara de subred . . . . . : 255.255.255.0
Puerta de enlace predeterminada . . . . . : 192.168.0.1

Adaptador de Ethernet Conexión de red Bluetooth:

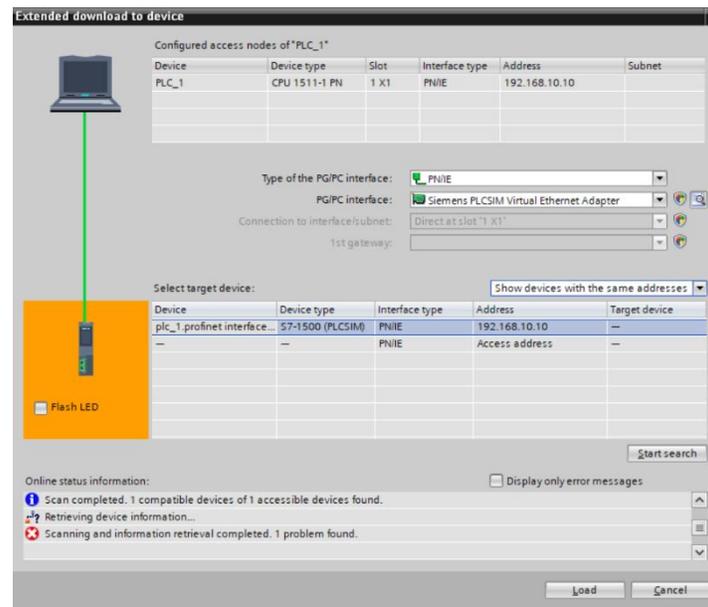
Estado de los medios. . . . . : medios desconectados
Sufijo DNS específico para la conexión. . . :

C:\Users\cripc>

```

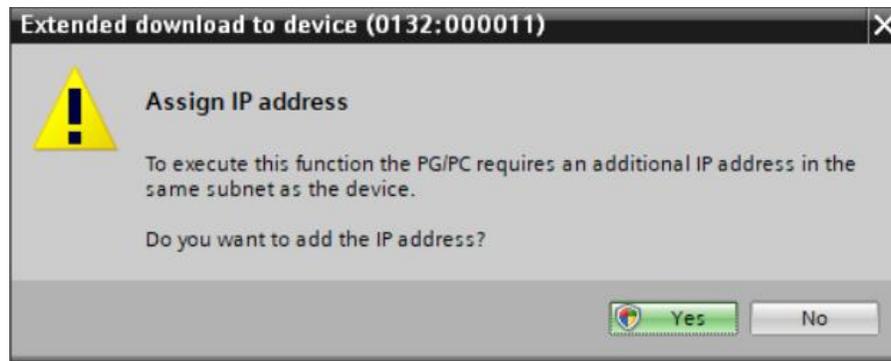
Para cargar el programa dentro de la instancia virtual se realiza el mismo procedimiento que para un PLC físico. Como se puede observar en la imagen 46 el programa encuentra el dispositivo disponible.

**Figura 47**  
Programación del PLC



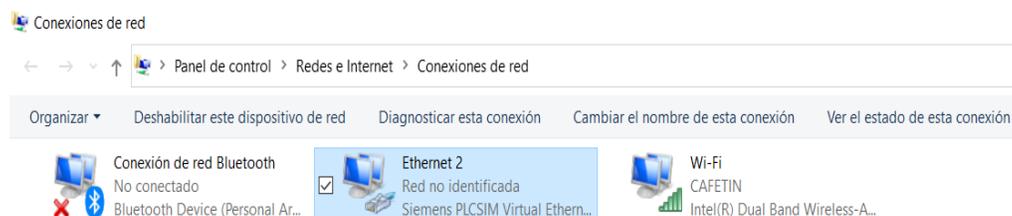
A continuación, se solicitará la autorización de la creación de una IP adicional la cual por defecto es la dirección 241 dentro de la misma subred del dispositivo, ver imagen 48, para este caso se añade la dirección 192.168.10.241.

**Figura 48**  
IP adicional



Esta dirección se asignará a una conexión de red virtual creada por PLC Sim como se muestra en la imagen 49. En caso de problemas en la descarga del programa, se debe verificar el estado de la conexión con la dirección añadida, como se muestra en la figura 49.

**Figura 49**  
Red virtual en el PLC



**Figura 50**  
Conexión con la dirección añadida

```

C:\WINDOWS\system32\cmd.exe
C:\Users\cripc>ping 192.168.10.241

Haciendo ping a 192.168.10.241 con 32 bytes de datos:
Respuesta desde 192.168.10.241: bytes=32 tiempo<1m TTL=128

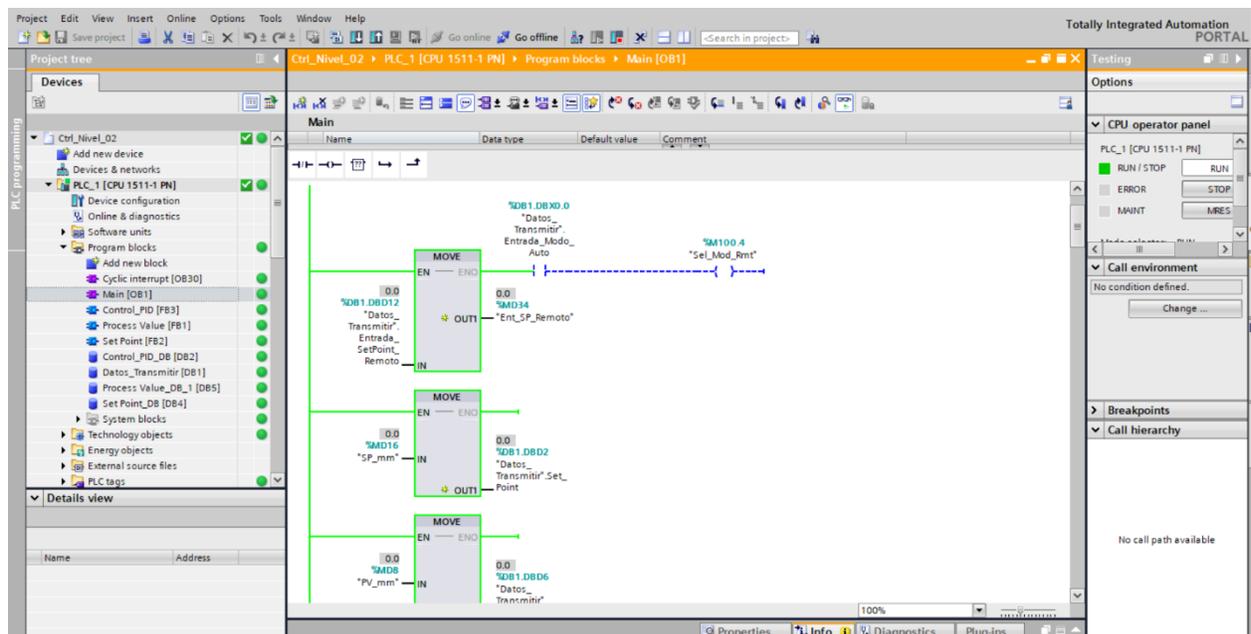
Estadísticas de ping para 192.168.10.241:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\cripc>

```

Una vez que se realiza el procedimiento se puede comprobar la conexión online entre Tia Portal y el dispositivo virtual, como se observa en la figura 51.

**Figura 51**  
Conexión online entre Tia Portal y el dispositivo virtual.

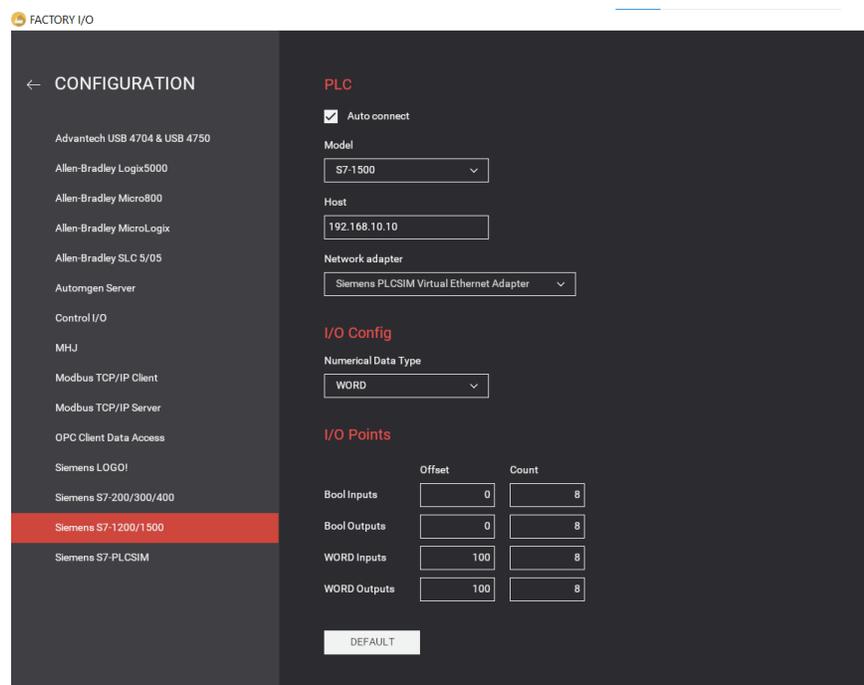


## Prueba de comunicación Factory IO

Luego de que se establece la comunicación del PLC, es necesario verificar la comunicación con Factory IO que es el programa donde se ejecuta el proceso a ser controlado. Como se muestra en la figura 52, se configura los parámetros asociados al controlador:

- Modelo: S7-1500
- Dirección IP: 192.168.10.10
- Adaptador de red: Adaptador Virtual PLC Sim
- Configuración de entradas y salidas.

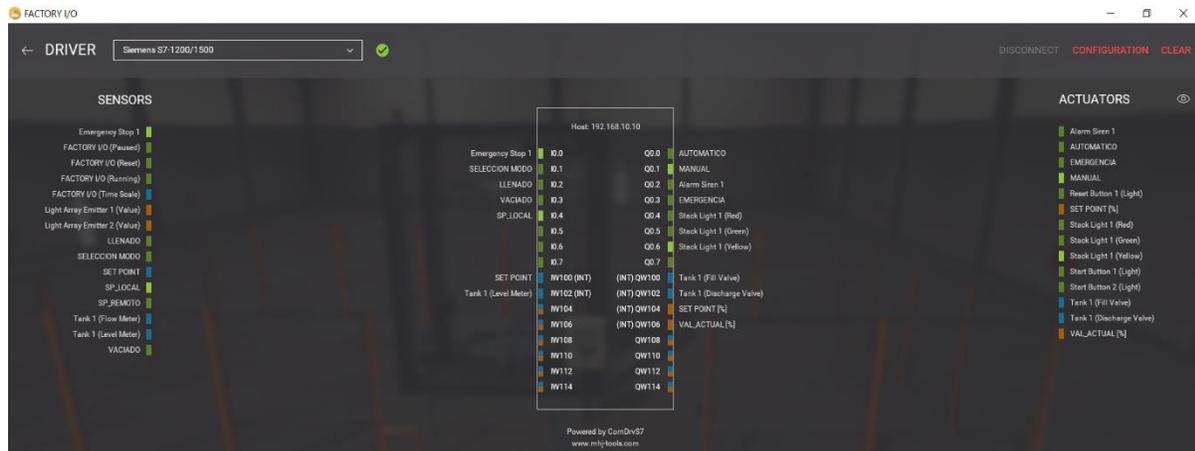
**Figura 52**  
Parámetros de control



Si la configuración fue ingresada correctamente el programa muestra un visto de color verde, como se muestra en la figura 53, lo que indica que la conexión es exitosa.

**Figura 53**

Conexión Exitosa



Finalmente se puede intercambiar el modo de operación del modo Edit al modo Run, figura 54.

**Figura 54**

Sistema en modo Run

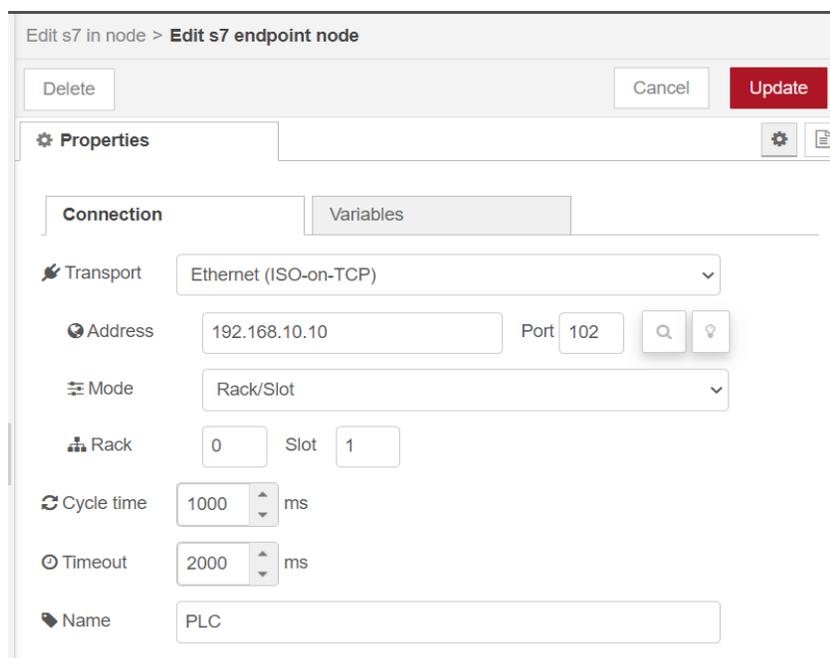


## Pruebas de comunicación Node Red

En el software Node Red también es necesario configurar los parámetros del PLC como dirección IP, rack y slot de los módulos, además se debe configurar la ip del broque MQTT asociado al proyecto.

En la figura 55, se verifica la configuración del nodo PLC con la información necesaria. Mientras que en las figuras 56, se observa la configuración del nodo del servidor Ubidots.

**Figura 57**  
Configuración del nodo PLC



The image shows the configuration interface for an 's7 endpoint node' in Node-RED. The window title is 'Edit s7 in node > Edit s7 endpoint node'. At the top, there are three buttons: 'Delete', 'Cancel', and 'Update'. Below this is a 'Properties' section with a settings icon and a document icon. The 'Connection' tab is active, and there is a 'Variables' tab. The configuration fields are as follows:

- Transport:** Ethernet (ISO-on-TCP)
- Address:** 192.168.10.10
- Port:** 102
- Mode:** Rack/Slot
- Rack:** 0
- Slot:** 1
- Cycle time:** 1000 ms
- Timeout:** 2000 ms
- Name:** PLC

**Figura 60**  
Configuración del nodo del servidor Ubidots.

Edit mqtt out node > **Edit mqtt-broker node**

Delete Cancel Update

**Properties**

Name SER\_UDI

**Connection** Security Messages

Server 169.55.61.243 Port 1883

Connect automatically  
 Use TLS

Protocol MQTT V3.1 (legacy)

Client ID Leave blank for auto generated

Keep Alive 60

**Figura 63**  
Ventana de ejecución del Node Red

Edit mqtt out node > **Edit mqtt-broker node**

Delete Cancel Update

**Properties**

Name SER\_UDI

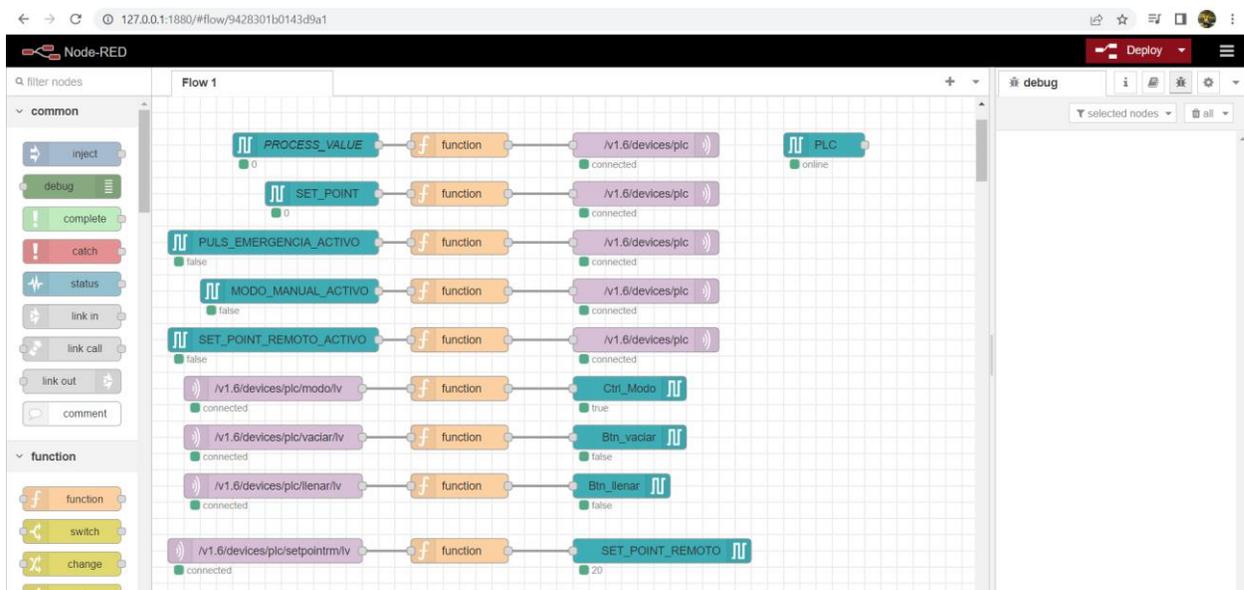
**Security** Connection Messages

Username BBFF-flaAeZB6BAH2cqCpKqbNFEUVPeSIfH

Password

Finalmente, si la comunicación se establece de manera correcta, en el flujo del programa se observa los indicadores de los nodos de comunicación con el PLC en color verde, mientras que los nodos de comunicación con el servidor Ubidots mostraran el mensaje “*connected*”, ver figura 58, además en la ventana de ejecución de node red se mostrará el estado de las conexiones, como en la figura 58

**Figura 66**  
Comunicación del servidor



**Figura 69**  
Estado de Conexión

```
node-red
Microsoft Windows [Versión 10.0.19044.1826]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\cripc>node-red
17 Sep 13:54:12 - [info]

Welcome to Node-RED
=====

17 Sep 13:54:13 - [info] Node-RED version: v2.2.2
17 Sep 13:54:13 - [info] Node.js version: v16.15.1
17 Sep 13:54:13 - [info] Windows_NT 10.0.19044 x64 LE
17 Sep 13:54:28 - [info] Loading palette nodes
17 Sep 13:55:08 - [info] Dashboard version 3.1.7 started at /ui
17 Sep 13:55:11 - [info] Settings file : C:\Users\cripc\.node-red\settings.js
17 Sep 13:55:11 - [info] Context store : 'default' [module=memory]
17 Sep 13:55:11 - [info] User directory : \Users\cripc\.node-red
17 Sep 13:55:11 - [warn] Projects disabled : editorTheme.projects.enabled=false
17 Sep 13:55:11 - [info] Flows file : \Users\cripc\.node-red\flows.json
17 Sep 13:55:11 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

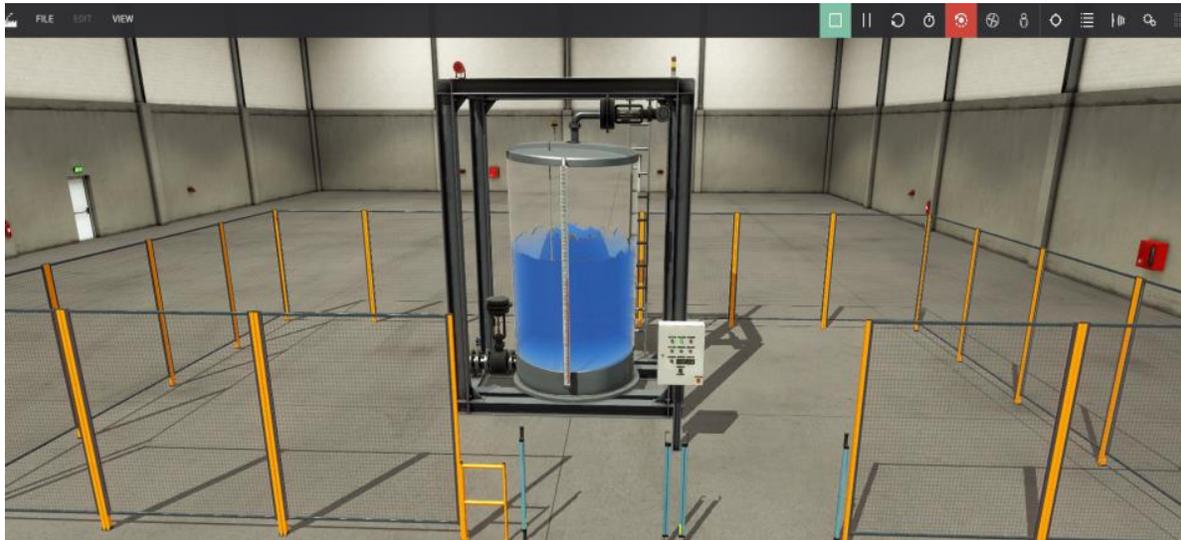
17 Sep 13:55:11 - [info] Server now running at http://127.0.0.1:1880/
17 Sep 13:55:11 - [info] Starting flows
17 Sep 13:55:12 - [info] Started flows
17 Sep 13:55:12 - [info] [mqtt-broker:SER_UDI] Connected to broker: mqtt://169.55.61.243:1883
```

## Sintonía del lazo de control

Cuando se ha conseguido la comunicación entre los diferentes componentes de la arquitectura, se puede ejecutar la comprobación de funcionamiento del proceso.

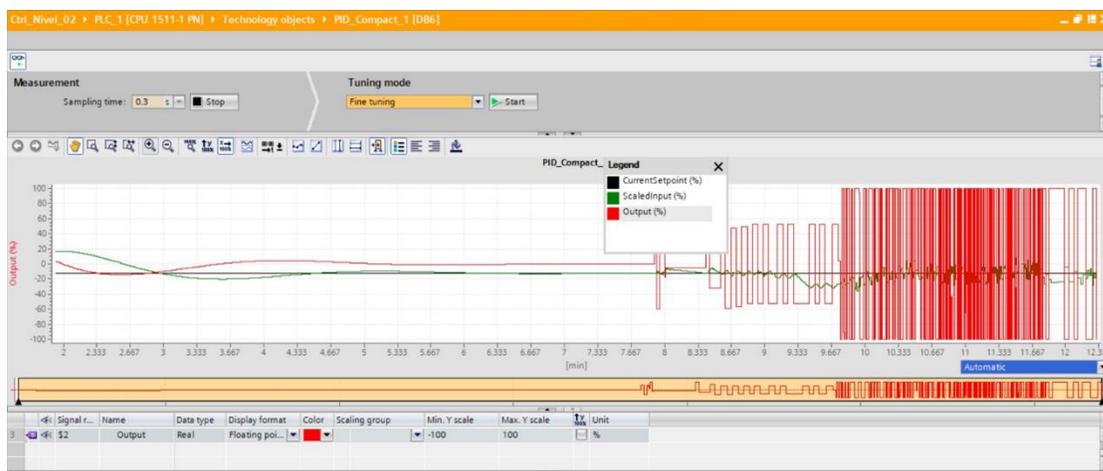
Como se puede observar en la figura 60, la respuesta del proceso es brusca y no se consigue la estabilidad del PV, por lo que es necesario llevar a cabo la sintonización del proceso.

**Figura 72**  
Respuesta del proceso



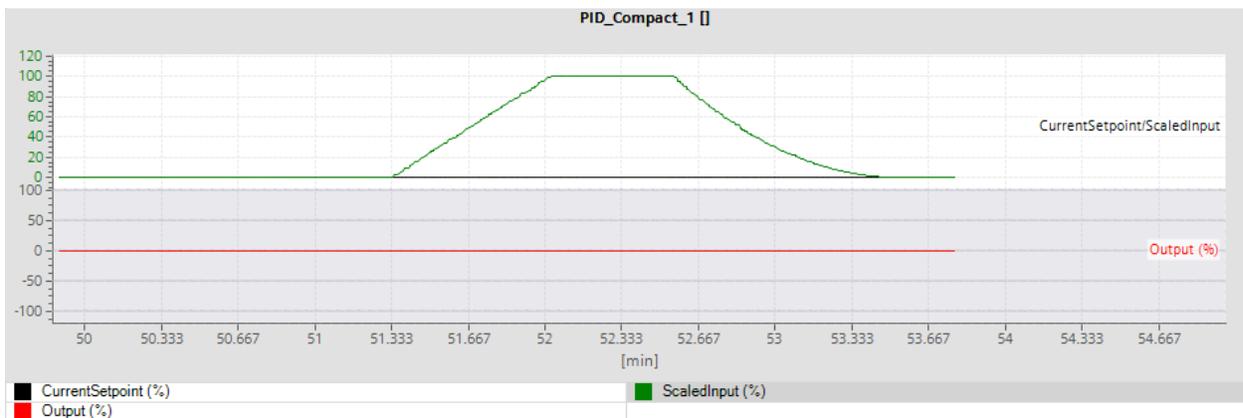
Una de las maneras de ejecutar la sintonización del lazo PID es el auto tuning, la figura muestra este proceso, sin embargo, al final del mismo se observa que el sistema aún se mantiene oscilante con picos tanto positivos como negativos. Por lo que es necesario ajustar el valor de las constantes mediante el uso de un método de otro método de sintonía.

**Figura 75**  
Sincronización del lazo PID



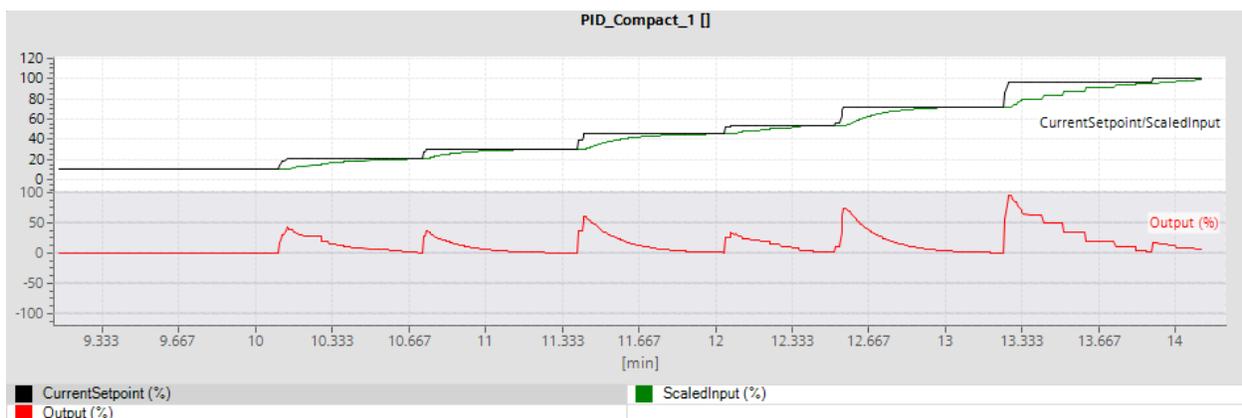
Para la sintonía del lazo se aplica el procedimiento del método lambda, como se puede observar en la figura, se analiza la respuesta del proceso tanto de forma ascendente como descendente. De tal manera que se establecen los parámetros explicados en el apartado 2.10 del presente documento.

**Figura 78**  
Procedimiento del método lambda

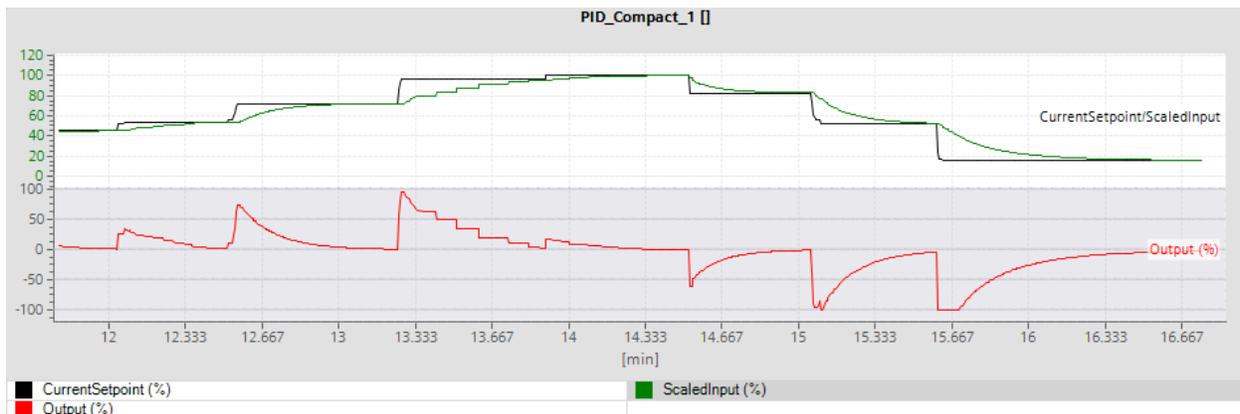


Las figuras, muestran la respuesta del sistema tanto para escalones positivos como negativos, además se puede apreciar que la respuesta de CV es estable lo que indica que la posición de la válvula no tiene oscilaciones permanentes

**Figura 81**  
Respuestas del sistema de CV

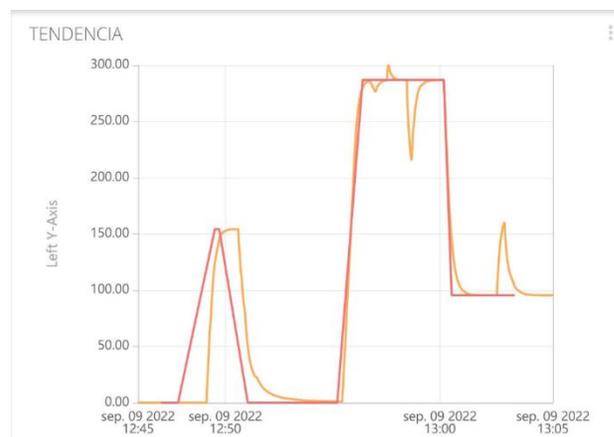


**Figura 84**  
Respuesta a escalones



Las pruebas anteriores se realizaron con la interfaz de TIA portal, y no se ha utilizado aun la interfaz de Ubidots debido a que la sintonía del proceso se lo debe realizar en tiempo real, una vez sintonizado el proceso se verifica el comportamiento del mismo de manera remota, la figura muestra la manipulación del SP y generación de perturbaciones ya desde el Dashboard de monitoreo IoT, la respuesta del controlador es buena y el tiempo de retardo es mínimo.

**Figura 87**  
Grafica de tendencias en el servidor



## Capítulo V

### Validación de la hipótesis

En los capítulos que componen el presente documento se revisó conceptos, plataformas de software y diversas interacciones orientadas a la implementación de una arquitectura de IoT, para monitorear y controlar una estación de nivel mediante un autómata simulado. A continuación, se describen algunos aspectos relevantes contenidos en los capítulos que anteceden:

De manera general el desarrollo del proyecto puede dividirse en dos partes, la primera en la cual se implementa el entorno virtual del proceso, así como el controlador y su algoritmo, se realizan las configuraciones y pruebas necesarias a fin de establecer la comunicación e interacción entre los diferentes componentes del proyecto.

La segunda parte consistió en el envío y recepción de datos a un broker MQTT alojado en la nube y el desarrollo de un dashboard que permite tanto la visualización como el ingreso de información relacionada con el proceso de control de nivel. Una vez que se consiguió la convergencia de la arquitectura desarrollada, se realizaron pruebas de acceso al servidor desde dispositivos remotos y se validó la modificación de parámetros, control y visualización desde el proceso.

En base a la información contenida en el presente documento se puede afirmar que: se diseñó un prototipo de arquitectura IoT para el monitoreo y control de una estación de nivel mediante un autómata simulado, con lo que se valida la hipótesis planteada en el presente trabajo y se pone a disposición esta herramienta como base para el desarrollo de futuros proyectos relacionados.

## Conclusiones

- Se diseñó un prototipo de arquitectura IoT para el monitoreo y control de una estación de nivel mediante un autómata simulado.
- Se implementó diferentes bloques de datos y comunicaciones para acceder y modificar condiciones de operación en el autómata simulado, a través de una plataforma de uso gratuito como node-red.
- Se implementó la comunicación mediante protocolo MQTT hacia un broker en la nube, para la manipulación y monitoreo de los sensores y actuadores de la Estación de nivel virtual desarrollada en el presente proyecto.
- Se incorporó el registro de tendencias de comportamiento del proceso dentro del servidor, con lo que se puede recomendar acciones de mantenimiento preventivas del sistema, siempre que se trabaje con un sistema físico.
- Como trabajos futuros se pueden desarrollar diferentes arquitecturas de IoT/IIoT enfocados en soluciones tanto domésticas como industriales, los cuales permitan incrementar la eficiencia tanto en consumo energético como el en aprovechamiento de materia prima e insumos.

## Recomendaciones

- La implementación de una arquitectura IoT virtualizada permite el análisis de respuesta del proceso y elementos, así como la interacción de los

mismos, pero se debe tener en cuenta que no se considera perturbaciones que pueden estar presentes en procesos físicos.

- Cuando se va a enviar datos a la nube se debe utilizar las herramientas de seguridad, encriptación, autenticación disponible en el protocolo utilizado en la implementación de proyectos de IoT, para prevenir o mitigar un posible ataque cibernético.
- El control local de los procesos debe tener mayor injerencia que las acciones permitidas de forma remota, ya que una falla en la comunicación con el servidor IOT debe permitir la función normal del proceso.

## Bibliografía

- Gandhi, M. (s.f.). *Autycom*. Obtenido de <https://www.autycom.com/s7-plcsim/#:~:text=El%20programa%20inform%C3%A1tico%20Simatic%20PLCSIM,la%20generaci%C3%B3n%20de%20estos%20controladores.>
- Almachi Chiguano, J. W., & Naranjo Barrera, E. A. . (2021). Desarrollo de un sistema SCADA para una estación de bombeo de agua a presión constante. *Bachelor's thesis, Ecuador: Latacunga: Universidad Técnica de Cotopaxi: UTC.*
- Baeza Martin, F. J. (2019). El contrato de hosting. *Barbaraiot*. (29 de Abril de 2021). Obtenido de [https://barbaraiot.com/blog/protocolos-iot-que-deberias-conocer/#:~:text=En%20el%20%C3%A1mbito%20inform%C3%A1tico%20y,la%20comunicaci%C3%B3n%20Machine2Machine%20\(M2M\).](https://barbaraiot.com/blog/protocolos-iot-que-deberias-conocer/#:~:text=En%20el%20%C3%A1mbito%20inform%C3%A1tico%20y,la%20comunicaci%C3%B3n%20Machine2Machine%20(M2M).)
- Cama-Pinto, A., Gil-Montoya, F., Manzano-Agugliaro, F., García-Cruz, A., & Gómez-López, J. . (2014). Sistema inalámbrico de monitorización para cultivos en invernadero. 81(184), 164-170.
- Cepymenews. (04 de Mayo de 2022). Obtenido de <https://cepymenews.es/caracteristicas-usos-internet-cosas/>
- Chuchico-Arcos, C., & Rivas-Lalaleo, D. (Diciembre 2020). Sensor Nodes and Communication Protocols of the Internet of Things Applied to Intelligent Agriculture. *n International Conference on Applied Technologies Springer, Cham.*, pp. 686-703.
- Espinosa Apolo, B. A., & Orellana Guayas, M. E. (2021). Desarrollo de aplicaciones de monitoreo y control basadas en IoT a través de la plataforma Ubidots.

Aplicaciones a sistemas de automatización bajo entornos de simulación.

*Bachelor's thesis.*

Guagchinga, F., Calvopiña, P., Alpúsig, S., & Cajas Buenaño, M. (2021). Control PID de procesos autoregulados e integrantes sintonizados con el método lambda.

*Revista Prespectivas.*

Guman López de Briñas, G. . (2021). Diseño e implementación de un sistema IoT para monitorear un dosificador de harina industrial. *Universidad Autonoma de Barcelona .*

Kerkez, W. &. (2016). Obtenido de

[https://www.researchgate.net/publication/307874031\\_Real-time\\_environmental\\_sensor\\_data\\_An\\_application\\_to\\_water\\_quality\\_using\\_web\\_services](https://www.researchgate.net/publication/307874031_Real-time_environmental_sensor_data_An_application_to_water_quality_using_web_services)

Mahedero Biot, F. (2020). Desarrollo de una aplicación IoT para el envío de imágenes mediante el protocolo MQTT. *Doctoral dissertation, Universitat Politècnica de València.*

Marín Sánchez, J. (2018). Diseño de un sistema de control distribuido usando Factory IO y Codesys V3. *comunication through OPC.*

*Mecalux.* (10 de Enero de 2020). Obtenido de <https://www.mecalux.es/blog/iiot-internet-de-las-cosas-industrial>

Moral-Pérez, M. A. (2021). Node-RED como herramienta visual de dispositivos IoT. *Universidad de Jaen.*

*Pickdata.* (11 de Mayo de 2020). Obtenido de [https://www.pickdata.net/es/noticias/node-red-programacion-visual-iot?gclid=CjwKCAjw0dKXBhBPEiwA2bmObYtHlfq\\_Hp0U029qMVDt5hLF1mLHZuMtzEd05wMD8owP0CegQ8OycBoCeHIQAvD\\_BwE](https://www.pickdata.net/es/noticias/node-red-programacion-visual-iot?gclid=CjwKCAjw0dKXBhBPEiwA2bmObYtHlfq_Hp0U029qMVDt5hLF1mLHZuMtzEd05wMD8owP0CegQ8OycBoCeHIQAvD_BwE)

- Rodríguez Molano, José Ignacio; Cueva Lovelle, Juan Manuel . (27). Metamodelo para la integración de la internet de las cosas y redes sociales. *Dialnet*.
- Sánchez Restrepo, F. (2020). Modelado farmacocinético y farmacodinámico de agentes anestésicos y su aplicación en el desarrollo de un sistema de control en lazo cerrado durante la anestesia general. *Plantilla\_UdeA*.
- Sánchez, M. y. (2018). Valor a partir de la Creación de Internet de las cosas: Estudio exploratorio en la Provincia de Buenos Aires. *Visión de futuro*.
- Satorres Mira, S. (2021). Automatización de un paletizador industrial. Programación con TIA Portal y virtualización con Factory IO. Doctoral dissertation, Universitat Politècnica de València.
- Semle, Aron. (2016). Protocolos IIoT para considerar. *Aadeca Revista*, 34.
- Sinadrives*. (Julio de 2021). Obtenido de <https://sinadrives.com/es/control-de-maquinaria-control-en-lazo-abierto-cerrado/#:~:text=Control%20en%20lazo%20cerrado%3A,sensor%20y%20actuar%20en%20consecuencia>.
- Thompson, Fernando. (31 de Mayo de 2022). *Fernando Thompson*. Obtenido de <https://soyfernando.com/2022/05/usos-internet-de-las-cosas/>
- Torres Capuz, J., & Chanoluisa Martinez, X. (2015). Diseño e implementación de un sistema de control en una planta de llenado de tanques a escala de un transmisor de nivel y algoritmo PID para pruebas en el laboratorio de automatización industrial.
- Trejo, Christian. (01 de Septiembre de 2021). *Sistemas Dinámicos y Control*. Obtenido de <https://www.youtube.com/watch?v=T5F89iH6ndU>

**Anexos**