



**Gestión de un Sistema de Control Distribuido utilizando Multi-Network TCP/IP para Procesos  
Industriales Virtuales, basado en IoT**

Cóndor Sinchiguano, Henry Wladimir

Departamento de Eléctrica y Electrónica

Carrera de Ingeniería en Electromecánica

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electromecánica

Ing. Sánchez Ocaña, Wilson Edmundo

10 de febrero del 2022

Latacunga



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA  
CARRERA DE INGENIERÍA EN ELECTROMECAÁNICA**

### **CERTIFICACIÓN**


Certifico que el trabajo de titulación, **“Gestión de un Sistema de Control Distribuido utilizando Multi-Network TCP/IP para procesos industriales virtuales, basado en IoT”** fue realizado por el señor **Cóndor Sinchiguano, Henry Wladimir** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

**Latacunga, 10 de febrero del 2022.**

.....  
**Ing. Sánchez Ocaña, Wilson Edmundo**

C. C.: 0501529937

## Reporte de verificación de contenidos



tesis ultima versión henry cóndor.pdf  
Scanned on: 3:14 February 13, 2022 UTC



Identical Words	370
Words with Minor Changes	61
Paraphrased Words	251
Ommited Words	0



Website | Education | Businesses

.....  
**Ing. Sánchez Ocaña, Wilson Edmundo**

**DIRECTOR**



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA  
CARRERA DE INGENIERÍA EN ELECTROMECÁNICA**

**RESPONSABILIDAD DE AUTORÍA**

Yo, **Cóndor Sinchiguano, Henry Wladimir**, con cédula de ciudadanía n° 0550079032, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Gestión de un Sistema de Control Distribuido utilizando Multi-Network TCP/IP para procesos industriales virtuales, basado en IoT”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

**Latacunga, 10 de febrero del 2022.**

A handwritten signature in blue ink is positioned above a horizontal dotted line. The signature is stylized and appears to read 'Henry Wladimir Sinchiguano'.

**Cóndor Sinchiguano, Henry Wladimir**

C.C.: 0550079032



**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA  
CARRERA DE INGENIERÍA EN ELECTROMECAÁNICA**

**AUTORIZACIÓN DE PUBLICACIÓN**

Yo **Cóndor Sinchiguano Henry Wladimir**, con cédula de ciudadanía n° **0550079032**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Gestión de un Sistema de Control Distribuido utilizando Multi-Network TCP/IP para procesos industriales virtuales, basado en IoT”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

**Latacunga, 10 de febrero del 2022.**

A handwritten signature in blue ink is shown above a horizontal dotted line. The signature is stylized and appears to read 'Córdor Sinchiguano'.

**Córdor Sinchiguano, Henry Wladimir**

C.C.: 0550079032

### **Dedicatoria**

Dedico este trabajo de titulación a mi yo del futuro, tener fe en ti y en las cosas que tú harás me ha dado fuerzas para culminar con éxito este proyecto. Te dedico sin haberte conocido, pero sé que en un par de años nos volveremos a encontrar, gracias por ser consiente, ser constante, ser valiente, ser prudente, ser sincero y decir todo lo que sientes. El camino fue duro, pero sé que mañana serás alguien admirable, no te rindas cumple tus metas talvez tardes, pero cúmpelas en esta vida o en otra. ¡CREO EN TI ¡

### **Agradecimiento**

Mi agradecimiento eterno a mi Alma Mater por brindarme la oportunidad de ser parte de ella, dentro de sus aulas y pasillos formé mi carácter, adquirí valores humanos que de una forma u otra me mejoran día a día como persona y profesional.

Quiero agradecer a los docentes de la Universidad de las Fuerzas Armadas ESPE, extensión Latacunga que a lo largo de todos estos años de estudio han luchado por proporcionar una educación pública de calidad en especial en estos tiempos difíciles, no tengo las palabras adecuadas y suficientes para agradecer todas sus enseñanzas, en especial al ing. Wilson Sánchez por ser un excelente tutor y gran líder como fuente de inspiración, gracias a todos por formar parte de los conocimientos que hoy poseo.

Gracias a mis amigos y compañeros por los buenos y malos momentos que hemos compartido en nuestra vida universitaria llena de estrés e incertidumbre, en especial a Xavier M. y Rolando H. por brindarme su apoyo cada vez que lo he necesitado.

Gracias a la Sra. María Hilda Sinchiguano Velasco por sus impresionantes esfuerzos para cumplir los roles de madre, cocinera, asesora, consejera, amiga y doctora. Gracias por su apoyo incondicional en todos estos años sin usted no sería posible finalizar este proyecto.

	8
<b>Tabla de Contenido</b>	
<b>Carátula</b> .....	<b>1</b>
<b>Certificación</b> .....	<b>2</b>
<b>Reporte de verificación de contenido</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>Tabla de contenido</b> .....	<b>8</b>
<b>Índice de figuras</b> .....	<b>13</b>
<b>Índice de tablas</b> .....	<b>12</b>
<b>Índice de ecuaciones</b> .....	<b>18</b>
<b>Resumen</b> .....	<b>19</b>
<b>Abstract</b> .....	<b>20</b>
<b>Problemática</b> .....	<b>21</b>
<b>Antecedentes</b> .....	<b>21</b>
<b>Justificación e Importancia</b> .....	<b>23</b>
<b>Alcance del proyecto</b> .....	<b>24</b>
<b>Objetivos</b> .....	<b>26</b>
<b>Objetivo general</b> .....	<b>26</b>



<i>Objetivos específicos</i> .....	26
<b>Marco Teórico</b> .....	<b>28</b>
<b>Pirámide de automatización CIM</b> .....	<b>28</b>
<b>Redes de comunicación industrial</b> .....	<b>31</b>
<b>Protocolos de comunicación</b> .....	<b>31</b>
<i>Profibus</i> .....	<b>31</b>
<i>Modbus</i> .....	<b>32</b>
<i>Comunicación S7</i> .....	<b>34</b>
<b>Sistema de control distribuido</b> .....	<b>34</b>
<b>Simulación de procesos industriales</b> .....	<b>35</b>
<b>Internet de las Cosas (IoT)</b> .....	<b>37</b>
<b>Gestión de procesos industriales</b> .....	<b>37</b>
<b>Modelos de optimización</b> .....	<b>38</b>
<b>Gestión de datos</b> .....	<b>40</b>
<b>Control estadístico de procesos</b> .....	<b>42</b>
<i>Gráficos de control</i> .....	<b>42</b>
<b>Metodología</b> .....	<b>45</b>
<b>Protocolos y comunicación</b> .....	<b>45</b>
<b>Estructura de comunicación del sistema de control distribuido</b> .....	<b>46</b>
<b>Procesos</b> .....	<b>47</b>

	10
<b>Comunicación web</b> .....	<b>47</b>
<b>Gestión de procesos</b> .....	<b>48</b>
<b>Diseño</b> .....	<b>51</b>
<b>Introducción</b> .....	<b>51</b>
<b>Selección de protocolos y comunicación</b> .....	<b>51</b>
<i>Profinet</i> .....	<b>51</b>
<i>Modbus</i> .....	<b>57</b>
<i>Comunicación S7</i> .....	<b>61</b>
<b>Estructura de comunicación del sistema de control distribuido</b> .....	<b>64</b>
<b>Simulación de procesos industriales</b> .....	<b>69</b>
<i>Proceso uno en FluidSIM: unidad de montaje y mecanizado</i> .....	<b>69</b>
<i>Proceso dos en Factory IO: mecanizado y ensamble</i> .....	<b>73</b>
<i>Proceso tres Interno: activación de generadores</i> .....	<b>74</b>
<b>Comunicación Web: control y monitoreo de procesos</b> .....	<b>78</b>
<b>Pre-procesamiento de datos</b> .....	<b>88</b>
<b>Almacenamiento de información</b> .....	<b>89</b>
<b>Visualización en la nube</b> .....	<b>93</b>
<b>Modelo de optimización</b> .....	<b>96</b>
<i>Planificación de la producción</i> .....	<b>96</b>
<i>Despacho económico</i> .....	<b>101</b>

	11
Gráficas de control .....	107
Aplicación del Telegrambot .....	108
Análisis de Resultados.....	111
Red de comunicación industrial.....	111
Resultados de los nodos programados en node-red .....	113
<i>Nodos de limpieza de datos</i> .....	113
<i>Nodos de visualización y monitoreo</i> .....	114
Resultados de los algoritmos matemáticos .....	117
<i>Optimización de la producción</i> .....	117
<i>Optimización de las operaciones de generación</i> .....	120
Seguimiento a la base de datos .....	122
Envío y recepción de peticiones a través del TelegramBot .....	126
Conclusiones y recomendaciones.....	129
Conclusiones.....	129
Recomendaciones .....	131
Bibliografía .....	135
Anexos.....	136

**Índice de Tablas**

<b>Tabla 1</b> <i>Componentes eléctricos y neumáticos empleados en el proceso uno .....</i>	69
<b>Tabla 2</b> <i>Distribución de las variables de restricción en Excel .....</i>	118
<b>Tabla 3</b> <i>Resultados y distribución de las variables de decisión en Excel .....</i>	118
<b>Tabla 4</b> <i>Unit Commitment de un día completo con cinco tecnologías de generación diferentes</i>	120
<b>Tabla 5</b> <i>Resultado de las operaciones de escalado y normalizado para el generador a gas.....</i>	121

## Índice de Figuras

<b>Figura 1</b> <i>Esquema sobre el alcance del proyecto</i> .....	26
<b>Figura 2</b> <i>Pirámide CIM de la automatización</i> .....	29
<b>Figura 3</b> <i>Esquema de comunicaciones de Profibus</i> .....	32
<b>Figura 4</b> <i>Ejemplo de arquitectura de comunicación con Modbus</i> .....	33
<b>Figura 5</b> <i>Arquitectura básica de un DCS</i> .....	35
<b>Figura 6</b> <i>Escenas de programación dentro del entorno de simulación de FACTORY I/O</i> .....	36
<b>Figura 7</b> <i>Cadena de valor de los datos masivos</i> .....	41
<b>Figura 8</b> <i>Ejemplo de gráfico de control</i> .....	44
<b>Figura 9</b> <i>Metodología propuesta, para el diseño del sistema de control distribuido</i> .....	45
<b>Figura 10</b> <i>Metodología propuesta para la gestión de procesos</i> .....	49
<b>Figura 11</b> <i>Ubicación del bloque TSEND_C en TIA portal</i> .....	52
<b>Figura 12</b> <i>Parámetros de configuración del bloque TSEND_C</i> .....	53
<b>Figura 13</b> <i>Configuración del bloque TSEND_C</i> .....	54
<b>Figura 14</b> <i>Configuración del bloque TRCV_C</i> .....	55
<b>Figura 15</b> <i>Parámetros de configuración del bloque TRCV_C</i> .....	56
<b>Figura 16</b> <i>Esquema de envío y recepción de datos entre dos controladores</i> .....	57
<b>Figura 17</b> <i>Ubicación del bloque MB_CLIENT en TIA Portal</i> .....	58
<b>Figura 18</b> <i>Configuración del bloque MB_CLIENT</i> .....	59
<b>Figura 19</b> <i>Configuración del bloque de datos TCON_IP_v4</i> .....	59

	14
<b>Figura 20</b> Configuración del bloque MB_SERVER del cliente .....	60
<b>Figura 21</b> Configuración del bloque de datos TCON_IP_V4 del servidor .....	60
<b>Figura 22</b> Ubicación de los bloques PUT y GET en TIA Portal.....	61
<b>Figura 23</b> Creación del enlace S7 cliente-servidor.....	62
<b>Figura 24</b> Conexiones de los dispositivos y redes.....	62
<b>Figura 25</b> Configuración del bloque PUT.....	63
<b>Figura 26</b> Propiedades del bloque PUT .....	63
<b>Figura 27</b> Configuración del bloque GET.....	64
<b>Figura 28</b> Estructura de comunicación del DCS.....	65
<b>Figura 29</b> Estructura de comunicación del cliente uno .....	66
<b>Figura 30</b> Estructura de Comunicación del Cliente dos.....	67
<b>Figura 31</b> Estructura de comunicación del cliente tres .....	68
<b>Figura 32</b> Estructura de comunicación del DCS con la nube y la base de datos .....	69
<b>Figura 33</b> Proceso neumático implementado en FluidSIM-P .....	71
<b>Figura 34</b> HMI del proceso neumático, implementado en DopSoft y TIA Portal .....	71
<b>Figura 35</b> Configuración de la comunicación FluidSIM-TIA Portal utilizando KepServer .....	72
<b>Figura 36</b> Configuración de un canal DDE Client en KepServer.....	72
<b>Figura 37</b> Proceso de mecanizado implementado en Factory IO.....	73
<b>Figura 38</b> Configuración del canal DDE client para el proceso del cliente uno .....	74
<b>Figura 39</b> Escalado de operaciones de E/S para un generador a gas en hora punta. ....	75

<b>Figura 40</b> Configuración del canal TCP para la comunicación con el cliente tres con KepServer ..	76
<b>Figura 41</b> Comunicación entre OPCRouter y KepServer .....	77
<b>Figura 42</b> Almacenamiento en Mongodb utilizando OPCRouter .....	77
<b>Figura 43</b> Escritura en Excel desde KepServer utilizando OPCRouter .....	78
<b>Figura 44</b> Configuración de Node-Red .....	79
<b>Figura 45</b> Node-Red nodos de comunicación S7 .....	80
<b>Figura 46</b> Configuración de conexiones del nodo S7.....	80
<b>Figura 47</b> Variables de entrada y salida del controlador .....	81
<b>Figura 48</b> Lectura de las variables de forma individual y grupal utilizando los nodos S7.....	82
<b>Figura 49</b> Instalación de los nodos del servidor compartido en la nube FRED.....	83
<b>Figura 50</b> Creación de un nuevo cliente FRED.....	83
<b>Figura 51</b> Nodos para el control del proceso Uno.....	84
<b>Figura 52</b> Nodos de control de restricciones para IBM .....	85
<b>Figura 53</b> Nodos de monitoreo de producción .....	86
<b>Figura 54</b> Nodos para el control del proceso implementado en FluidSIM-P .....	87
<b>Figura 55</b> Nodos de monitoreo de producción del proceso dos .....	87
<b>Figura 56</b> Nodos de envío de potencia demandada por hora.....	88
<b>Figura 57</b> Nodo para limpieza de datos .....	89
<b>Figura 58</b> Nodo para insertar datos en una instancia Mongo .....	90
<b>Figura 59</b> Configuración de una instancia en Mongodb2 .....	90

<b>Figura 60</b> <i>Conexión de MongoDB Atlas con node-red</i> .....	91
<b>Figura 61</b> <i>Usuarios con acceso a la base de datos</i> .....	92
<b>Figura 62</b> <i>Direcciones IP con acceso a la Red</i> .....	92
<b>Figura 63</b> <i>Colecciones y documentos en MongoDB Atlas</i> .....	93
<b>Figura 64</b> <i>Conexión básica de los nodos Dashboard</i> .....	93
<b>Figura 65</b> <i>Programación del nodo función para lectura y conversión de datos</i> .....	94
<b>Figura 66</b> <i>Configuración de plantilla utilizando el nodo format template</i> .....	95
<b>Figura 67</b> <i>Nodo template para la visualización de gráficos</i> .....	95
<b>Figura 68</b> <i>Programación de los nodos para lectura de datos y visualización en tablas.</i> .....	96
<b>Figura 69</b> <i>Declaración de variables en CPLEX Optimization Studio</i> .....	99
<b>Figura 70</b> <i>Función objetivo y restricciones en CPLEX Optimization Studio</i> .....	100
<b>Figura 71</b> <i>Lectura y escritura en Excel utilizando CPLEX Optimization Studio</i> .....	100
<b>Figura 72</b> <i>Declaración de variables en un bloque de datos</i> .....	100
<b>Figura 73</b> <i>Parámetros de los generadores y variables de decisión en FICO XPRESS</i> .....	103
<b>Figura 74</b> <i>Lectura de parámetros desde Excel hacia Fico Xpress</i> .....	104
<b>Figura 75</b> <i>Función objetivo y restricción de no negatividad en Fico Xpress</i> .....	104
<b>Figura 76</b> <i>Restricciones de potencia mínima y máxima</i> .....	105
<b>Figura 77</b> <i>Optimización del problema lineal de variables enteras</i> .....	105
<b>Figura 78</b> <i>Resultados de la optimización escritura en consola</i> .....	106
<b>Figura 79</b> <i>Resultados de la optimización escritura en Excel</i> .....	106



<b>Figura 80</b> <i>Cálculo de los límites de control utilizando un nodo de función</i> .....	107
<b>Figura 81</b> <i>Configuración del nodo SPC en node-red</i> .....	108
<b>Figura 82</b> <i>Configuración de los nodos del TelegramBot</i> .....	109
<b>Figura 83</b> <i>Lectura de las peticiones del TelegramBot utilizando un nodo de función</i> .....	110
<b>Figura 84</b> <i>Intercambio de información entre servidor-cliente 2</i> .....	112
<b>Figura 85</b> <i>Intercambio de información entre servidor-cliente 1 cliente 3</i> .....	113
<b>Figura 86</b> <i>Diferencia en la visualización de datos almacenados utilizando el nodo de limpieza</i> .	114
<b>Figura 87</b> <i>Control y monitoreo de los procesos uno y dos desde la nube</i> .....	115
<b>Figura 88</b> <i>Monitoreo del proceso tres desde la nube y almacenamiento en MongoDB</i> .....	116
<b>Figura 89</b> <i>Trazado de curvas en Excel y la nube resultados del Unit Commitment</i> .....	117
<b>Figura 90</b> <i>Monitoreo de las bases de datos en MongoDB</i> .....	122
<b>Figura 91</b> <i>Conexiones activas del servidor MongoDB</i> .....	123
<b>Figura 92</b> <i>Gráfico de barra de producción, demanda y costos</i> .....	124
<b>Figura 93</b> <i>Gráfica de medias utilizando las variables de producción</i> .....	125
<b>Figura 94</b> <i>Grafica de barras producción total de potencia</i> .....	125
<b>Figura 95</b> <i>Consultas del proceso uno y dos utilizando TelegramBot</i> .....	127
<b>Figura 96</b> <i>Errores comunes en las peticiones con telegramBot</i> .....	128
<b>Figura 97</b> <i>Consultas del proceso tres utilizando telegramBot</i> .....	128

**Índice de Ecuaciones**

<b>Ecuación 1</b> <i>Vector de actividades con <math>n</math> variables de decisión</i> .....	39
<b>Ecuación 2</b> <i>Vector de precios con <math>n</math> componentes</i> .....	39
<b>Ecuación 3</b> <i>Vector de disponibilidad de recursos de <math>n</math> componentes</i> .....	39
<b>Ecuación 4</b> <i>Función objetivo</i> .....	39
<b>Ecuación 5</b> <i>Restricciones del problema</i> .....	40
<b>Ecuación 6</b> <i>Condiciones de no-negatividad</i> .....	40
<b>Ecuación 7</b> <i>Límite de control superior</i> .....	43
<b>Ecuación 8</b> <i>Límite de control inferior</i> .....	43
<b>Ecuación 9</b> <i>Costo total de un sistema con <math>n</math> estaciones de producción</i> .....	96
<b>Ecuación 10</b> <i>Suma de los niveles de producción</i> .....	97
<b>Ecuación 11</b> <i>No negatividad</i> .....	99
<b>Ecuación 12</b> <i>Costo total de un sistema con <math>n</math> generadores</i> .....	101
<b>Ecuación 13</b> <i>Suma de los niveles de generación</i> .....	101
<b>Ecuación 14</b> <i>Balance energético</i> .....	102

## Resumen

El presente proyecto de investigación plantea una metodología dividida en dos partes para implementar y gestionar un sistema de control distribuido basado en el estándar TCP/IP utilizando procesos industriales virtuales, los ejes principales que conforman esta metodología son, estructurar una red de comunicación para el SCD, comunicar al SCD con la Web, gestionar una base de datos no SQL en la nube, para finalmente culminar con la ejecución y seguimiento que permita validar los resultados. El SCD está conformado por un PLC servidor y tres PLCs cliente de la marca SIEMENS modelo S7-1500, cada uno controla un proceso virtual a través de TIA Portal configurados con los protocolos de bus de campo Profibus, Modbus TCP y la comunicación S7. El control y monitoreo de los procesos se lo hace de manera remota utilizando el servidor compartido en la nube Fred, para enlazar al SCD con la Web se utiliza node-red. Una vez que el control y monitoreo así ejecutado de manera correcta, se plantea dos modelos de optimización, el primer modelo se encarga de optimizar la producción descentralizando la demanda total requerida entre el proceso uno realizado en Factory IO y el proceso dos realizado en FluidSIM, para resolver el problema matemático se utiliza el software CPLEX Optimization Studio. El segundo modelo consiste en un problema de optimización basado en el Unit Commitment por día, el problema se resuelve utilizando FICO Xpress, se simula las consignas de entrada ideales que deberían ajustarse a los generadores para producir la energía solicitada. Para finalizar se almacenan los datos más relevantes en MongoDB Atlas.

Palabras clave:

- **MONGODB ATLAS**
- **UNIT COMMITMENT**
- **MODELO DE OPTIMIZACIÓN**

**Abstract**

This research project proposes a methodology divided into two parts to implement and manage a distributed control system based on the TCP/IP standard using virtual industrial processes, the main axes that make up this methodology are, structure a communication network for the SCD, communicate the SCD with the Web, manage a non-SQL database in the cloud, to finally culminate with the implementation and monitoring to validate the results. The SCD consists of a PLC server and three client PLCs of SIEMENS model S7-1500, each one controls a virtual process through TIA Portal configured with Profibus fieldbus protocols, Modbus TCP and S7 communication. The control and monitoring of the processes is done remotely using the shared server in the cloud Fred, to link the SCD with the Web is used node-network. Once the control and monitoring is correctly executed, two optimization models are proposed, the first model is in charge of optimizing the production by decentralizing the total demand required between process one performed in Factory IO and process two performed in FluidSIM, to solve the mathematical problem the CPLEX Optimization Studio software is used. The second model consists of an optimization problem based on the Unit Commitment per day, the problem is solved using FICO Xpress, the ideal input setpoints that should be adjusted to the generators to produce the requested energy are simulated. Finally, the most relevant data are stored in Mongoddb Atlas.

Key words:

- **MONGODB ATLAS**
- **UNIT COMMITMENT**
- **OPTIMIZATION MODEL**

## Capítulo I

### 1. Problemática

#### 1. 1. Antecedentes

En la actualidad las tecnologías de la comunicación e información de arquitectura abierta se han convertido de vital importancia en la industria 4.0, la influencia de herramientas tecnológicas orientadas a esta revolución ha permitido una mayor flexibilidad dentro de los sistemas. Ofreciendo nuevas maneras de organizar e integrar los procesos, permitiendo cada vez obtener sistemas más eficientes y con una mejor gestión, a través de tecnologías como la integración vertical y horizontal, análisis de datos, optimización, nube y simulación, entre otros.

En (Carrera, 2021) se menciona que, ante la necesidad de soluciones para el control y mantenimiento de los sistemas de producción, se ha producido un cambio de pensamiento en la industria la cual busca la manera de actualizar sus conocimientos para lograr involucrarse a los cambios dentro de la matriz productiva enfocadas a automatizar los procesos desde la nube. Por ello el desarrollo y aplicación de plataformas basadas en arquitectura IoT para la supervisión y control de los procesos de fabricación en tiempo real de forma local y remota, a través de la nube fue de vital importancia para proveer soluciones estables y seguras. Estas soluciones permiten conectar de manera remota y ver o actuar sobre el estado de los dispositivos que se encuentren conectados a las máquinas.

Con este enfoque (Mora, 2017) menciona que, en el mercado existen varias plataformas que facilitan el desarrollo de sistemas basados en IoT ofreciendo servicios del tipo Back-end como analítica avanzada, Big Data, Machine Learning, entre otros. Algunos ejemplos de

plataformas de este tipo tenemos a: Azure IoT hub, Cloud IoT Core, Amazon Web Service IoT, entre otros.

Además en (Llanos, 2018) se expone que, las empresas dedicadas al monitoreo de procesos han identificado problemas de gestión de datos, presente en su adquisición desde sensores. La falta de una planificación, desarrollo y selección de herramientas de sistemas de almacenamiento en base a las necesidades y limitantes provoca que se presenten problemas con la información y el másico volumen de datos que este involucra. Además, varios estudios han surgido con la finalidad de ofrecer alternativas de comunicación por la nube para la recepción y envío de datos en los procesos industriales, cada una ofreciendo alternativas de solución industrial a nivel empresarial con ventajas y desventajas dependiendo del área de implementación.

Para finalizar según (Liu et al., 2017), las tecnologías IoT han permitido optimizar productos, servicios y operaciones dentro de la industria, a través de IoT es posible la recopilación de datos e información de tipo estructurada y no estructurada proviniendo dicha información de dispositivos que se encuentren conectados a internet como sensores climáticos, de temperatura, nivel, velocidad, etc. Con las soluciones de IoT conjuntamente con softwares y herramientas dedicados a la gestión y análisis que utilicen como base modelos estadísticos o matemáticos, ha permitido incrementar sus aplicaciones en múltiples áreas como en el área eléctrica para planificar la producción de generación, además de permitir el monitoreo de forma remota del uso de la energía. En el área de manufactura los sensores y la recopilación de sus datos han permitido mejorar la eficiencia en la producción, proporcionando un flujo constante de información para así tomar mejores decisiones, en la logística para acortar rutas, en el comercio minorista, la agricultura, etc.

## **1. 2. Justificación e importancia**

Los crecientes desarrollos de nuevas tecnologías han facilitado el análisis, comunicación y transmisión de información en diferentes áreas de la ingeniería por ejemplo a las dedicadas al área de control y automatización que buscan mejorar y optimizar los procesos de producción, ya sea diseñando o rediseñando los sistemas de control o programando modelos que permitan comprender de manera más eficiente el comportamiento de los sistemas. Por lo que en el presente proyecto se hace uso de las mismas aplicando herramientas que permitan el control y monitoreo de los procesos de manera local y remota a través de un servidor compartido en la nube, aprovechando las redes de comunicación industrial se plantea una arquitectura basada en el estándar Ethernet TCP/IP que permita integrar los protocolos de bus de campo que han presentado un notable crecimiento en aplicaciones industriales como: MODBUS, PROFINET y COMUNICACIÓN S7.

Considerando lo descrito en las secciones anteriores, el proyecto busca escalar a un nivel superior de la pirámide CIM más allá del control, monitoreo y adquisición de datos de forma local. Se desarrolla dos metodologías: la primera metodología recoge técnicas, softwares y plataformas IoT que permitan controlar y monitorear los procesos industriales virtuales de manera eficiente y sencilla. La segunda metodología está dedicada al nivel de planificación, ejecución, gestión de datos y gestión de operaciones para mejorar la producción, considerando parámetros y variables que se encuentren estrechamente relacionadas con los procesos como: la demanda, costos de producción, disponibilidad de dispositivos y máquinas, tiempos de producción, disponibilidad de suministros y calidad de los productos desde la etapa de transformación hasta su ensamble final.

Además, el proyecto incluye el uso de diversos softwares de: resolución de problemas de optimización a través de programación lineal y lineal entera, programación visual en flujos, lenguajes de programación PLC, interfaces de comunicación OPC, bases de datos orientados a documentos y tablas, todo esto con el fin de aprovechar las tecnologías presentes en nuestro entorno que permita obtener resultados lo más cercanos a la realidad, y que sirva como referencia y contribuya para una futura implementación en procesos físicos reales.

### **1. 3. Alcance del proyecto**

La finalidad del trabajo de titulación presentado, es la gestión de un sistema de control distribuido utilizando una Multi-Network TCP/IP para procesos industriales virtuales, basado en IoT. Para definir el alcance que tendrá el proyecto se toma como referencia la pirámide CIM. El diseño está conformado por 4 etapas como se muestra en la Figura 1, proceso, control básico, supervisión y control de operaciones de fabricación.

En la primera etapa se eligen e implementan en los softwares de simulación de procesos del tipo 2D y 3D, aquellos dispositivos y equipos necesarios para simular las condiciones y operaciones de los procesos industriales procurando que las secuencias de operación sean lo más cercano a la realidad posible, en esta etapa se agregan los sensores, actuadores, válvulas, robots, bandas transportadoras, entre otros.

La segunda etapa corresponde al nivel del control, aquí se eligen los controladores lógicos programables PLC's y se programan las secuencias de operaciones utilizando ya sea lenguaje de texto o lenguaje por gráficos y Grafcet. Además, se seleccionan los protocolos de buses de campo basados en TCP disponibles en TIA PORTAL V16.



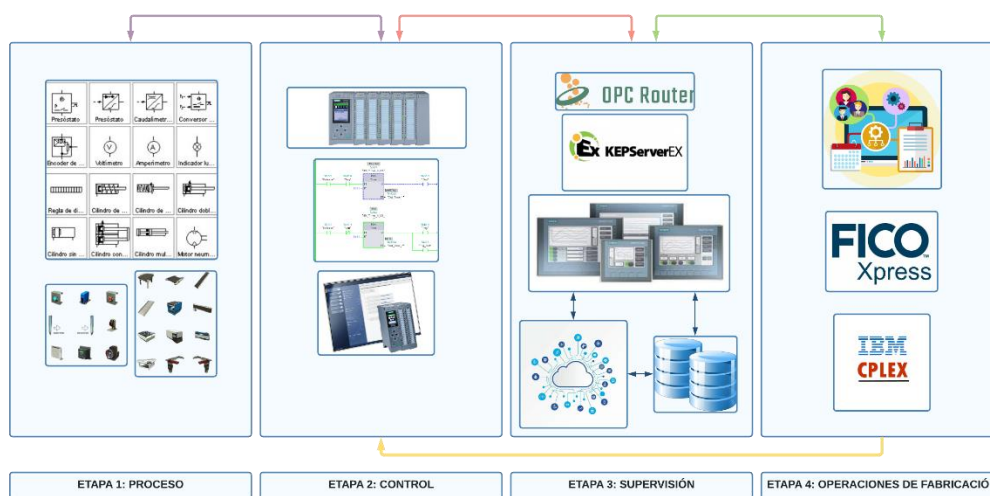
En la etapa de supervisión y control, se diseñan SCADA e interfaces HMI según sea la necesidad de cada proceso, además se utilizan servidores OPC como KepServer y OPCRouter para comunicar los procesos con TIA PORTAL y permitir la transferencia e intercambio de información con el servidor en la nube y la base de datos.

Finalmente, en la etapa de control de operaciones de fabricación se plantea un algoritmo matemático que permita optimizar la producción minimizando sus costos, para ello se utilizan softwares dedicados a solucionar problemas de optimización comercial basados en programación lineal o lineal de enteros mixtos. Los resultados obtenidos de esta etapa definirán las nuevas condiciones de operación que se deberán asignar a cada uno de los procesos.

Adicional los datos que se obtenga en cada etapa serán alojados en una base de datos, después de pasar por un preprocesamiento que elimine datos erróneos, valores nulos, valores no definidos o información irrelevante para posteriormente utilizando interfaces gráficas se permita su visualización e interpretación aplicando técnicas estadísticas como SPC para determinar si los procesos se encuentran dentro de los límites de control permitidos.

Figura 1

Esquema sobre el alcance del proyecto



## 1. 4. Objetivos

### 1.4.1. Objetivo General

Gestionar una red de comunicación industrial para el control, monitoreo y almacenamiento de información de un sistema de control distribuido basado en TCP/IP en procesos industriales virtuales, utilizando plataformas IoT.

### 1.4.2. Objetivos Específicos

- Realizar el monitoreo y control de un sistema de control distribuido empleando protocolos de comunicación industrial, basados en TCP/IP.
- Analizar las diferentes plataformas IoT que permitan la visualización y almacenamiento de información.
- Manipular y Configurar los diferentes nodos de la red de comunicación industrial en el local host, utilizando node-red.

- Programar en la plataforma IoT seleccionada, los eventos que se visualizarán, controlarán, y almacenarán del Sistema de Control Distribuido.
- Procesar y Almacenar la información más relevante que permita optimizar la toma de decisiones.
- Desarrollar un algoritmo matemático que permita optimizar los procesos de producción del Sistema de Control Distribuido.
- Presentar la información almacenada de forma visual y sencilla de comprender a través de interfaces gráficas utilizando un servidor Web que sea amigable con el usuario desde cualquier dispositivo.

## Capítulo II

### 2. Marco Teórico

#### 2. 1. Pirámide de automatización CIM

Como se muestra en la Figura 2, CIM suele ser representado con una pirámide que puede considerarse como un modelo lógico conformado por 5 niveles que integran desde los procesos de producción hasta los sistemas de automatización y tecnologías de la información a nivel de empresa. Este modelo es altamente jerárquico, por ejemplo: las actividades de apoyo encargadas de gestionar el flujo de información se encuentran a un nivel superior que el nivel de producción y que, a su vez se encuentra a un nivel inferior que el nivel de automatización influyendo directamente en el diseño y la planificación de la producción. Sin embargo, este modelo no puede ser considerado como un tipo de técnica de diseño que permita la construcción de factorías con sistemas automatizados altamente seguros, flexibles y robustos (Veneri & Capasso, 2018).

Además, debido a los altos costos y complejidad que representa implementar la pirámide CIM solo las grandes empresas lo emplean en su totalidad. CIM no pretende ser un modelo a seguir más bien se considera como una técnica de referencia que facilite la implementación de la automatización industrial basándose en la recopilación, coordinación, intercambio y transmisión de datos e información entre los diferentes niveles y subniveles mediante aplicaciones de software con arquitectura abierta y redes de comunicación (Morgan & Haley, 2019).

**Figura 2***Pirámide CIM de la automatización*

*Nota.* La figura representa la jerarquía por niveles de la pirámide CIM mostrando los dispositivos, herramientas y servicios que intervienen en cada nivel. Tomado de *Ciberseguridad industrial e infraestructuras críticas* (p. 25), por F. Sevillano et al., 2021, Ra-Ma.

Nivel 0: Está conformado por los instrumentos de campo que interactúan directamente con el proceso como sensores y actuadores que permiten convertir las variables físicas como el caudal, presión, distancia, energía, peso, etc. en variables neumáticas, hidráulicas o eléctricas (Sevillano et al., 2021).

NIVEL 1: El siguiente nivel conocido como el sistema de mando y control se encuentra conformado por aquellos dispositivos que se encuentran interactuando de manera directa con los sensores y actuadores. Los controladores ya sea de tipo microcontrolador, PLC's, controladores PID, controladores robots o controladores numéricos por ordenador, son algunos ejemplos que permiten controlar las entradas y salidas de los dispositivos ubicados en el NIVEL 0. Además, poseen la capacidad de comunicarse con otros dispositivos a través de protocolos de

comunicación estándar, considerándose este nivel uno de los más importantes de la automatización industrial (Sevillano et al., 2021).

NIVEL 2: Ascendiendo en la pirámide este nivel está conformado por los sistemas de control y supervisión (SCADA) que permiten recibir y enviar información del nivel 1 para poder visualizar, controlar y operar de manera remota los procesos. Algunas funciones que se cumplen en este nivel son la recepción de instrucciones procedentes de los niveles superiores, transformación de los dispositivos del nivel inferior en acciones y comandos, recopilación de información de niveles inferiores para enviarlos al nivel superior, y sobre todo facilitar información para el supervisor. Equipos como el HMI ( interfaces hombre-máquina) en ocasiones suele incluirse en este nivel debido que permite visualizar el proceso a pie de máquina (Morgan & Haley, 2019).

Los niveles superiores correspondientes al nivel 4 conocido como supervisión y planificación de planta y el nivel 5 considerado el nivel de gestión empresarial incluyen las compras, la planificación de recursos, planificación de operaciones, el almacenamiento de información, la venta y el marketing integrando los sistemas de soporte. La información se filtra de los datos a medida que pasa de un nivel a otro. La base de datos de producción recopila y almacena los principales parámetros del proceso productivo, y la coordinación entre las células se lleva a cabo para implementar todo el proceso productivo (Morgan & Haley, 2019). Además, en el ámbito de empresa, la información recopilada de los niveles inferiores, sirven de apoyo al momento de tomar decisiones, ayudan al área gerencial a planificar de manera más efectiva los flujos de materia prima, la mano de obra calificada y el recurso financiero necesario para el mantenimiento de equipos (Veneri & Capasso, 2018).

## **2. 2. Redes de comunicación industrial**

Como se menciona en (Belomonte, 2018), a medida que los sistemas industriales se volvieron más complejos se empezó a requerir inmensas cantidades de cableado para lograr el intercambio y control de información, y debido al gran avance tecnológico, las empresas con el fin de mejorar su producción y rendimiento han implementado redes de comunicación a nivel industrial capaces de resistir condiciones ambientales hostiles, cada una dedicada a cubrir ciertas necesidades de intercomunicación en tiempo real por lo tanto, requiriendo de técnicas adecuadas para su aplicación. Estas redes de comunicación presentan ventajas, por ejemplo: la posibilidad de descentralizar la producción utilizando una o varias plantas, mejorando así el control de los procesos con unos costes de implementación bajos, permitiendo a la industrial aumentar la producción e incrementando su calidad, haciéndolo más competitivo en el mercado.

## **2. 3. Protocolos de comunicación**

### **2.3.1. Profibus**

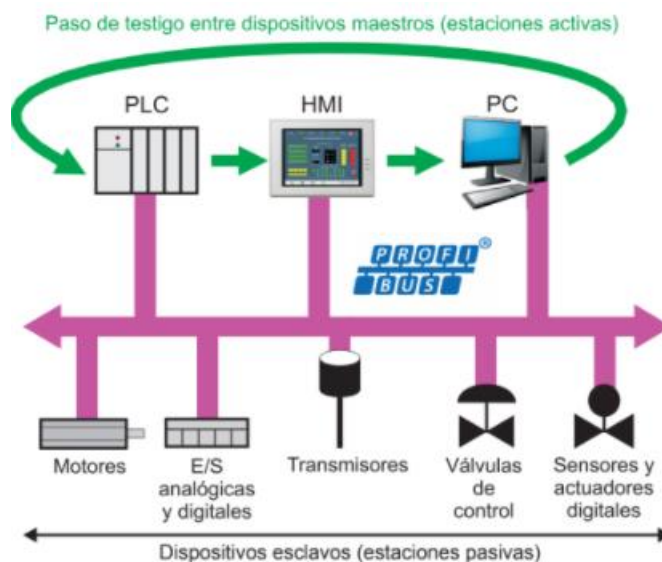
Profibus es un bus de campo de estándar abierto con una predominante aplicación industrial debido a que tiene la capacidad de solucionar los requerimientos de comunicación de los procesos industriales utilizados en fabricación y producción, permitiendo la conexión de los controladores descentralizados desde el nivel de campo al nivel de control, tomando en cuenta la jerarquía de la pirámide CIM se podría decir que Profibus es muy utilizado para aplicaciones que se encuentren en los niveles más bajos de la jerarquía piramidal ubicándose en los niveles 0, 1 y 2 (Antúnez, 2016).

Profibus puede ser usado para transmisiones críticas de alta velocidad con cantidad de datos pequeños o medianos ya sea en tiempo real o para ejecutar tareas de comunicación

extensas y complejas, utilizando técnicas asíncronas orientadas a caracteres, cada uno formado por 11 bits. Algunas características importantes de este bus de campo es que su medio de acceso es híbrido en otras palabras permite entablar una comunicación del tipo maestro-esclavo (también conocido como dispositivos activos-pasivos) y de paso de testigo entre dispositivos maestros (Valdivia, 2019). Profibus está referido al modelo OSI en este modelo cada capa realiza actividades definidas compartiendo parámetros entre la capa 1 (capa física) y la capa 2 (capa de enlace de datos) diferenciándose en la capa 7 correspondiente a la capa de aplicación (Bonilla, 2016).

**Figura 3**

*Esquema de comunicaciones de Profibus*



*Nota.* Tomado de Comunicaciones Industriales (p. 103), por C. Valdivia, 2019, Paraninfo.

### **2.3.2. Modbus**

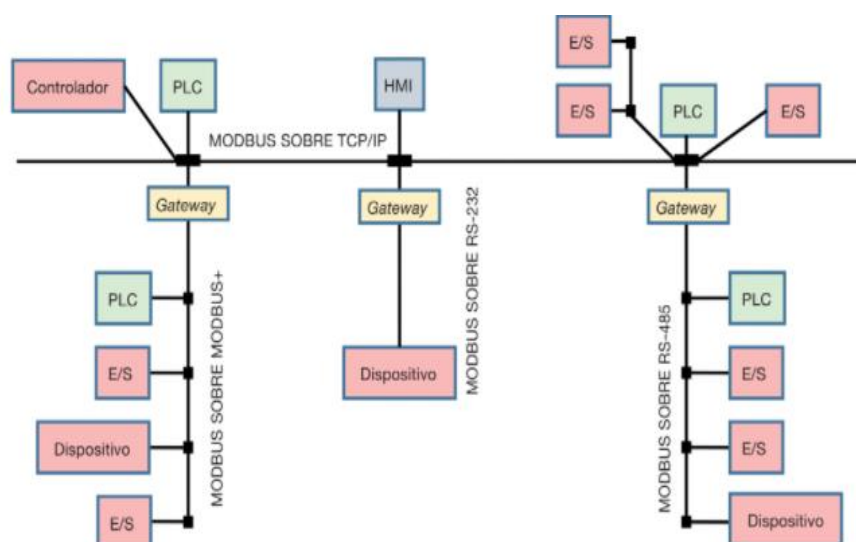
Modbus es considerado uno de los primeros buses de campo con la mayor disponibilidad de conexiones, que proporciona una arquitectura de comunicación cliente-



servidor o entre equipos maestro-esclavo incluso en la modalidad de punto a punto, independientemente del tipo de conexión, ya sea en diferentes redes o buses permitiendo así su fácil integración mediante pasarelas (Gateway). Como se menciona en (Pardo, 2012), Modbus posee versiones para puerto serie y para Ethernet sobre ISO on TCP, lo cual es una gran ventaja a la hora de combinar con los sistemas informáticos de oficina totalmente adaptados a la tecnología Ethernet. Al ser un protocolo de comunicación de estándar abierto, Modbus es ampliamente utilizado en la industria y al estar basado en el modelo OSI permite la interconexión entre dispositivos provenientes de diferentes fabricantes, cada uno con su tipo de conexión física y software de aplicación propio para parametrizar sus productos de esta manera el intercambio de información es posible sin presentar problemas de compatibilidad (J. Gómez et al., 2021).

**Figura 4**

*Ejemplo de arquitectura de comunicación con Modbus*



*Nota.* Tomado de Simulación de Sistemas Mecatrónicas (p. 126), por J. Gómez et al., 2021,

Paraninfo.

### **2.3.3. Comunicación S7**

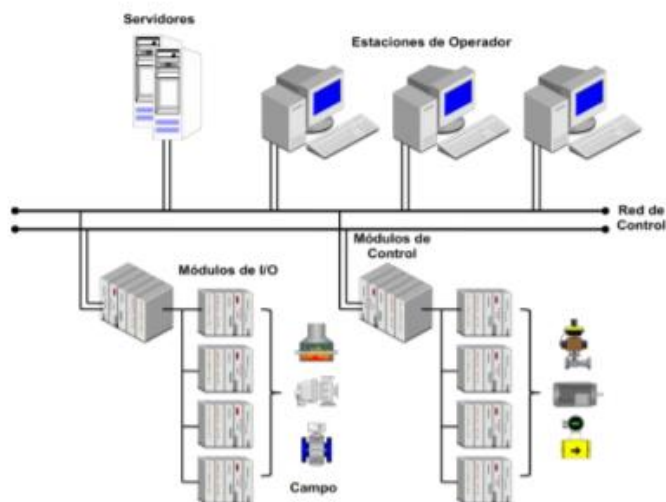
S7 es un enlace de comunicación propio de los controladores S7 de Siemens que permite la transferencia de datos entre Cliente-Servidor o Cliente-Cliente en cantidades pequeñas de forma dinámica y variable a través del bus MPI o Profibus. Los bloques PUT y GET por lo general son las instrucciones más utilizadas para permitir la transferencia de datos entre dos controladores S7 ofreciendo un servicio unidireccional de escritura y lectura.

### **2. 4. Sistema de Control Distribuido**

Un DCS se puede definir como una colección de controladores (esclavos) conectados en serie dentro de un área local, siendo el sistema de control más utilizado en la actualidad por sus ventajas como, integración de diferentes equipos con controladores capaces de realizar funciones específicas, cada uno con sus respectivos componentes de hardware y software, compuestos por instrumentos de campo como sensores y actuadores, instrumentos de acondicionamiento, centros de supervisión, control y adquisición de datos, hasta por sistemas ERP para la gestión de empresa. Además, un DCS es capaz de adquirir grandes volúmenes de datos y es capaz de canalizar los datos recogidos a un controlador central (maestro) a través de redes de comunicación de alta velocidad, para posteriormente poner a disposición de los operadores de planta. A diferencia de un sistema de control centralizado que concentra todos los procesos y tareas en un solo controlador, un DCS reparte las tareas y funciones a ejecutar en dos o más controladores repartidos en el mismo espacio permitiendo controlar los diferentes procesos de forma paralela (Godoy & Veloso, 2019).

**Figura 5**

*Arquitectura básica de un DCS*



*Nota.* Tomado de Implementación de un Sistema de Control Distribuido, integrando Virtualización y el entorno de labVIEW (p. 13), por S. Bravo, 2019, Universidad Técnica Federico Santa María.

## 2. 5. Simulación de procesos industriales

La simulación de procesos nos permite construir una representación simbólica de los procesos reales, utilizando técnicas de simulación como construcción de prototipos y programación por ordenador. Además como se menciona en (Pardo, 2012), las tareas de simulación son casi indispensables pues facilitan la implementación de procesos de automatización muy complejos mediante otros menos complejos, facilitando la programación de tareas y mejorando los sistemas de control.

Con el avance tecnológico se han presentado cada vez dispositivos más complejos dificultando la automatización a través de la simulación. En la actualidad existentes proveedores

de software de simulación de procesos y equipos cada vez más realistas, incluso con la capacidad de simular sensores como es la visión artificial, un claro ejemplo de este tipo de programas es FACTORY IO, en cuyo entorno permite al usuario construir procesos tan complejos con la capacidad de transferir información desde un ordenador. Algunos proveedores de equipos para el control como Allen-Bradley, Siemens, Analog Devices, Texas instruments, entre otros, disponen de programas de simulación de procesos que permiten entrenar a los usuarios a través de entornos de programación específicos y flexibles con variables predefinidas, o con conjuntos de subprogramas organizados en bloques (Creus, 2007).

### Figura 6

*Escenas de programación de distintos equipos de automatización dentro del entorno de simulación de FACTORY I/O*



Utilizando herramientas de programación se puede lograr que los modelos diseñados presenten un comportamiento lo más real posible tomando en cuenta todas las variables que influyen en el proceso permitiendo emular las condiciones y/o comportamiento de los sistemas, confiriendo valores a los instrumentos de medición como sensores y comprobando la reacción que estos tiene sobre los actuadores evaluando, depurando y corrigiendo o ajustando el

programa de control antes de cargar la programación en los autómatas programables o robots, evitando de esta manera, posibles errores que dificulten el correcto funcionamiento de los procesos, y servicios como la adquisición de datos, generación de alarmas, optimización de los diseños y sobre todo dificultando la integración del nivel de producción con el nivel empresarial.

## **2. 6. Internet de las Cosas (IoT)**

IoT es un sistema de dispositivos físicos y virtuales interrelacionados con identificadores únicos que se encuentran conectados a internet con la capacidad de recopilar y transferir datos a través de una red sin requerir la presencia humana, estos datos pueden ser utilizados para optimizar productos, servicios y operaciones, facilitando la implementación de aplicaciones inteligentes en múltiples áreas ya sea en el ámbito energético, comercial, agricultura, control industrial, entre otros (Bahga & Madiseti, 2014).

El internet de las cosas a empezado a ser una realidad cotidiana cobrando impulso rápidamente no solamente en el sector industrial, gracias a los avances tecnológicos las redes IoT poseen mayores capacidades que las redes actuales. El objetivo de IoT es que esté presente en todo, facilitando las configuraciones, el control y la conexión en red entre dispositivos y máquinas, no necesariamente limitado a ellos. Actualmente junto con el Big Data y el internet industrial de las cosas se han convertido en la espina dorsal en la que se sustenta la nueva era industrial conocida como industria 4.0 (Joyanes, 2021).

## **2. 7. Gestión de procesos industriales**

La gestión de procesos permite definir tareas para mejorar los procesos de fabricación de los productos hasta su ensamble final, ahorrar costos de producción a la vez que genera más rentabilidad utilizando técnicas o herramientas que se encuentren presentes desde, el diseño

hasta la ejecución de la producción, definiendo secuencias de operación que optimicen los procesos de producción, acortando tiempos de fabricación, reduciendo tiempos muertos de máquinas, a la vez que se aprovecha la materia prima y los costos que adquisición que esto implica, cumpliendo la función de puente entre los niveles de planificación, diseño y programación de los recursos e inventarios (Wu et al., 2018).

Con la presencia del internet de las cosas (IoT) y del internet industrial de las cosas (IIoT) la conectividad y comunicación entre los dispositivos físicos presentes en el mundo y entre las máquinas presentes en el área industrial ha sido posible, con IIoT se ha integrado tecnologías que han estado presentes desde hace años como el Big Data, Data Analytics, Machine Learning, entre otros permitiendo dar un mejor uso a los datos recopilados de los instrumentos de medición (sensores), utilizando diferentes tipos de análisis o modelos se ha logrado aplicar estas tecnologías para encontrar tendencias ocultas entre los procesos de fabricación, costos de producción, calidad de los productos y servicios, de esta manera los gerentes de operación han logrado encontrar y seleccionar condiciones de operación que puedan mejorar las secuencias de fabricación y reducir los costos, ahorrando no solo dinero sino tiempo en la identificación de los problemas como la ineficiencia en los procesos de fabricación mostrando un gran potencial con prácticas sostenibles que mejoran la cadena de suministro y la eficiencia operativa (Sharma, 2016).

## **2. 8. Modelos de optimización**

Uno de los modelos de optimización más aplicados para la toma de decisiones dentro de una organización es un tipo de modelo matemático basado en programación lineal o lineal de enteros, este tipo de programación se ocupa de aquellos problemas de optimización que están compuestos por expresiones lineales en una función objetivo la cual se pretende maximizar o

minimizar por ejemplo maximizar los beneficios o minimizar los costos de producción, cuentan con un conjunto de restricciones que pueden estar ligadas a una frontera superior o inferior que deben satisfacerse para encontrar la solución más óptima y factible que permita su implementación en la realidad. Este tipo de optimización es conocida como una herramienta con facilidad de aplicación de fácil de entender y explicar, permitiendo dar solución a numerosos problemas relacionados con la gestión, desde la selección de personal, distribución de recursos, diseño de redes de distribución, despacho económico, planificación de la producción, entre otros (Céspedes, 2021). Un problema de programación lineal puede definirse de la siguiente manera:

$$X = (x_1, x_2, x_3, x_4 \dots \dots x_n)^T$$

**Ecuación 1**

*Vector de Actividades con n Variables de Decisión*

*Nota. (Céspedes, 2021).*

$$C = (c_1, c_2, c_3, c_4 \dots \dots c_n)$$

**Ecuación 2**

*Vector de precios con n componentes*

*Nota.(Céspedes, 2021).*

$$B = (b_1, b_2, b_3, b_4 \dots \dots b_n)^T$$

**Ecuación 3**

*Vector de disponibilidad de recursos de n componentes*

*Nota. (Céspedes, 2021).*

$$Z = C * X$$

**Ecuación 4**

*Función objetivo*

*Nota.* (Céspedes, 2021).

Donde:

$Z$ = Optimizar máximo o mínimo.

$C$ = Vector de precios con  $n$  componentes.

$X$ = Vector de actividades con  $n$  variables de decisión.

$$A * X \leq B$$

### **Ecuación 5**

*Restricciones del problema*

*Nota.* (Céspedes, 2021).

Donde:

$B$ = Vector de disponibilidad de recursos con  $n$  componentes.

$A$ = matriz de coeficientes con  $n$  columnas y  $m$  filas  $a_{ij}$ .

$$X \geq 0$$

### **Ecuación 6**

*Condiciones de no-negatividad*

*Nota.* (Céspedes, 2021).

Donde:

$n$ = Cantidad de recursos  $j$  que se necesita por unidad de actividad  $i$ .

$0$ = Vector columna de  $n$  ceros.

## **2. 9. Gestión de datos**

La gestión de datos se ha convertido en una estructura clave en muchas organizaciones conjuntamente con las nuevas tecnologías que han mejorado la recolección y transformación de datos en información que ha permitido la aplicación de sistemas de información como el "Big

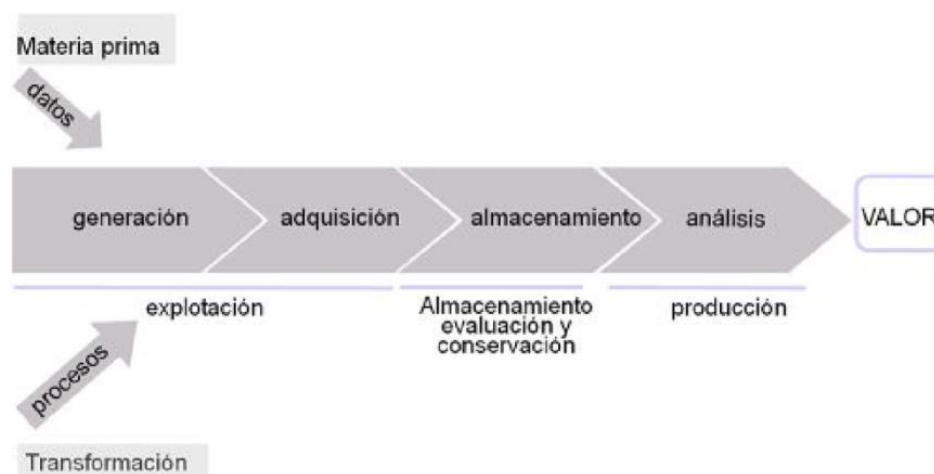


Data” con datos estructurados y no estructurados con un amplio alcance, llegando a tener la capacidad de mejorar la toma de decisiones, diseñar acciones para las organizaciones, generar conocimiento y facilitar el despliegue y rendimiento de las bases de datos, una correcta gestión de la información puede aportar valor y mayor claridad ofreciendo una visión clara sobre los productos, servicios y clientes ayudando a innovar, mejorar y optimizar los servicios (International, 2020).

Para una gestión de datos se involucran profesionales con diferentes roles técnicos como, administración de base de datos, programación y administración de las redes, hasta estrategias de negocios como data Stewards, estregas de datos y CDOs (oficial en jefe de datos) involucrando tecnologías capaces de capturar datos, procesar, analizar e interpretar de manera rápida y eficiente.

### Figura 7

*Cadena de valor de los datos masivos*



*Nota.* Tomado de Big Data: Gestión y Explotación de Grandes Volúmenes de Datos, (p. 23), por M. García, 2017, UOC.

La gestión de datos se basa en la cadena de valor: explotación, almacenamiento y producción, el almacenamiento de información debe incluir procesos específicos para la evaluación y mantenimiento que pueda garantizar la calidad y veracidad de los datos almacenados para su recuperación que permita obtener o crear conocimiento y plantear acciones (Alsina, 2017).

## **2. 10. Control estadístico de procesos**

El control estadístico de procesos al igual que el muestreo de aceptación y la estadística descriptiva forma parte del control estadístico de la calidad permitiendo conocer el funcionamiento de los procesos aplicando técnicas estadísticas, involucrando a la información y a los datos que se adquieren en los procesos de fabricación, distribución o en la prestación de servicios. Es una herramienta muy poderosa para verificar la estabilidad e implementar una posible mejora de los procesos para mantenerlos más estables bajo control estadístico y con la mínima variabilidad (Gómez & Gómez, 2019). Las herramientas más útiles y mayormente utilizadas en el control de procesos estadísticos son las gráficas de control y cartas de control.

### ***2.10.1. Gráficos de control***

Una gráfica de control consiste en una línea central con dos límites de control, uno superior a la línea central y uno inferior con valores característicos al azar o bajo causas asignables, normalmente son utilizados para vigilar los procesos de producción, cuando los valores registrados en la gráfica se encuentran fuera de los límites se puede interpretar como un indicador de falta de control y cuando se encuentren los valores dentro de los límites se dice que el proceso está bajo control, además debe operar por un largo tiempo sin producir fallas o señales falsas por lo general causados por fallas en la maquinaria, por la falta de calidad en la materia prima, errores de programación en las etapas de producción o debido a un error

humano (Kume & Vasco, 1992). Todos los datos que se obtengan de los procesos de fabricación deben dividirse en subgrupos, después se debe calcular la media de cada uno de los subgrupos y, cuando uno de ellos no se encuentre dentro de los límites de control, el gestor concluye que el proceso está fuera de control y necesita un reajuste o acción correctiva que suele tomarse en base a la experiencia y de un juicio altamente desarrollado (Ross, 2018). Para determinar los límites que tendrá la gráfica se utiliza las siguientes ecuaciones:

$$LCS = \bar{\bar{X}} + Z * \sigma_{\bar{X}}$$

**Ecuación 7**

*Límite de control superior*

*Nota.* (Heizer & Render, 2004)

$$LCI = \bar{\bar{X}} - Z * \sigma_{\bar{X}}$$

**Ecuación 8**

*Límite de control inferior*

*Nota.* (Heizer & Render, 2004)

Donde:

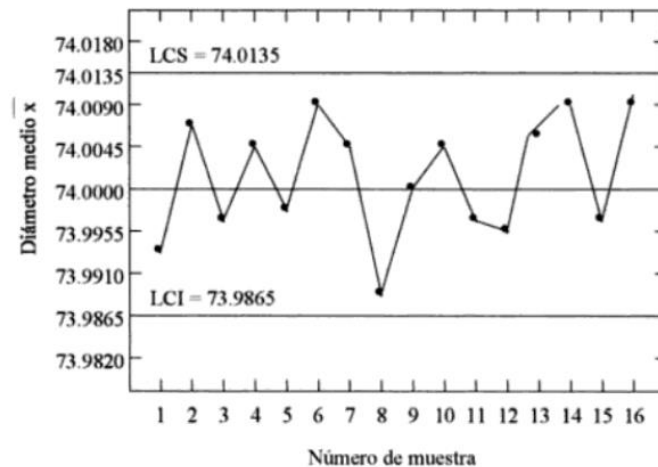
$\bar{\bar{X}}$ =media de la media de las muestras o el valor meta establecido.

Z= número de desviaciones estándar.

$\sigma_{\bar{X}}$ = Desviación estándar de la media de las muestras

Figura 8

*Ejemplo de gráfico de control*



*Nota.* La figura representa la media del diámetro tomando cada hora una muestra de cinco diámetros. Tomado de Sistema de indicadores para la mejora y el control integrado de la calidad de los procesos (p. 94), por J. Heredia, 2000, Universidad de Jaume.

Antes de diseñar el gráfico de control es necesario considerar tres decisiones críticas que tienen relación con el tamaño de la muestra, el periodo base y el nivel de significancia que son explicados mediante gráficos  $\bar{X} - R$ . En donde el tamaño de la muestra indica la cantidad de elementos que se incluirán en cada muestra, el periodo base incluye una prueba de hipótesis que en caso de no ser rechazada indica que los procesos se encuentran bajo control, la probabilidad máxima de error viene definida por el nivel de significancia que dependerá del tamaño de la muestra (Heredia, 2001).

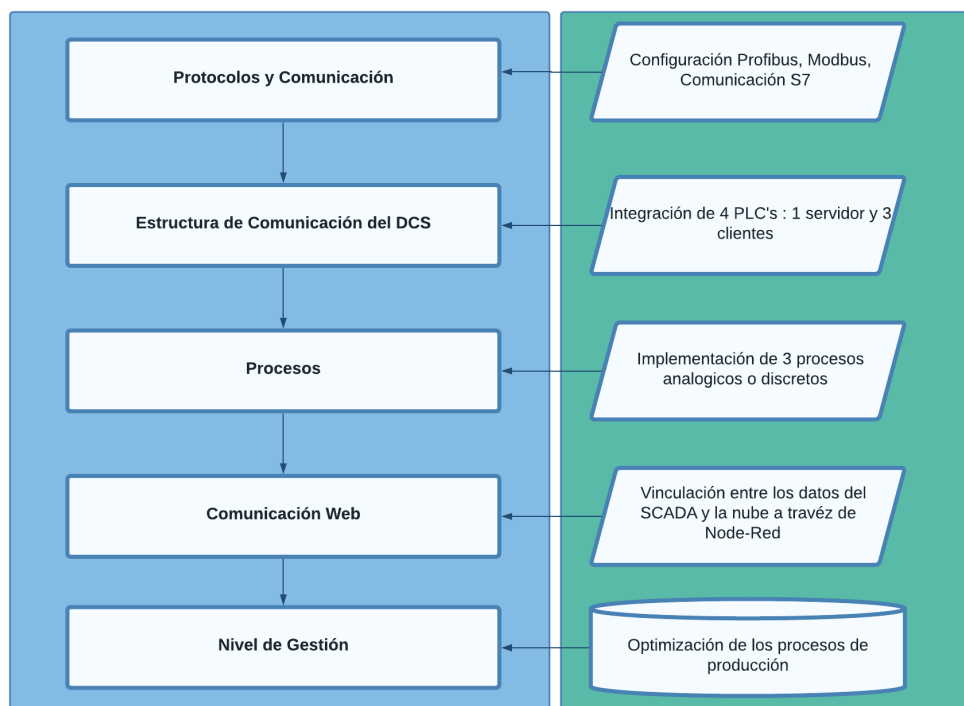
## Capítulo III

### 3. Metodología

La propuesta metodológica se dividió en dos partes: la primera metodología se puede observar en la Figura 9, en donde se considera como etapa inicial la selección de los protocolos que conforman la red de comunicación industrial que posteriormente a través de los bloques de función que posee la biblioteca de TIA PORTAL permita su configuración y validación.

**Figura 9**

*Metodología propuesta, para el diseño del Sistema de Control Distribuido*



#### 3. 1. Protocolos y Comunicación

En esta etapa se seleccionan los protocolos como Profibus, Modbus, y el protocolo propio de Siemens S7, debido a que TIA PORTAL proporciona bloques de función como:

TSEND\_C, TRCV\_C, MB\_CLIENT, MB\_SERVER, PUT/GET es posible una fácil configuración y sobre todo mantener una conexión estable durante el funcionamiento de los procesos.

### **3. 2. Estructura de comunicación del Sistema de Control Distribuido**

En la etapa siguiente se integra 4 PLC's de la marca Siemens modelo S7 con CPU 1511-1 PN cada uno con sus respectivos módulos de entradas y salidas del tipo analógico o digital según lo requiera. La función de cliente será repartida en 3 controladores quienes realizan las peticiones necesarias al controlador servidor para el control o el monitoreo de los procesos.

El cliente 1 será configurado con el bloque TSEND\_C para permitir enviar los datos adquiridos de las variables más relevantes del proceso al PLC 4 que cumple el rol de servidor. Adicional se configura el bloque TRCV\_C para que el PLC 1 pueda recibir la información proveniente del PLC servidor, que a su vez recibe datos e información desde la nube a través de Node-red.

A continuación, el siguiente controlador con el rol de cliente 2 tendrá la configuración Modbus TCP, utilizando el bloque MB\_CLIENT y el bloque MB\_SERVER para enviar y recibir la información, este último bloque será configurado en el bloque de programación del controlador servidor.

Para finalizar, el cliente 3 será configurado con la comunicación S7 con los bloques PUT y GET que permiten el envío y recepción de la información entre el cliente y el servidor. Al ser un protocolo propio de Siemens los bloques pueden ser configurados dentro del entorno de programación del propio servidor o si prefiere se agrega en el área de trabajo del controlador cliente.

### **3. 3. Procesos**

El proceso que estará vinculado al cliente 1 se programa en el software de simulación 3D creado por Real Games llamado FACTORY IO, que permite construir y controlar procesos industriales con señales analógicas o digitales.

El proceso vinculado al cliente 2 será implementado en el software FluidSIM P que facilita la simulación de circuitos del tipo neumático o hidráulico. Además, para permitir la comunicación bidireccional entre TIA PORTAL y FluidSIM se utilizará un servidor OPC desarrollado por la compañía KepWare conocido como KepServer.

El proceso implementado en el Cliente 3 corresponde a un proceso interno dentro de TIA PORTAL, que consiste en el normalizado y escalado de condiciones de operación para la activación de los generadores involucrados en el unit commitment desarrollado en el software de optimización FICO XPRESS, cabe recalcar que FICO XPRESS no permite la simulación de procesos, más bien facilita la resolución de problemas de optimización grandes y complejos a través de programación lineal, programación lineal entera mixta, programación cuadrática convexa, entre otros.

### **3. 4. Comunicación Web**

Para permitir el intercambio de datos entre la nube y el controlador servidor, se utiliza la herramienta de programación visual basado en flujos, node-red es una herramienta que facilita la comunicación entre hardware, Apis, servicios en línea, entre otros. una vez que node-red fue instalado en la PC, los nodos necesarios a utilizar se los puede descargar directamente de node-red o a través de consola escribiendo el siguiente comando: `npm install node-red-`

contrib-s7 el nodo instalado permite la comunicación con los dispositivos S7 de Siemens, además node-red-node-red-dashboard facilita la creación de interfaces de visualización.

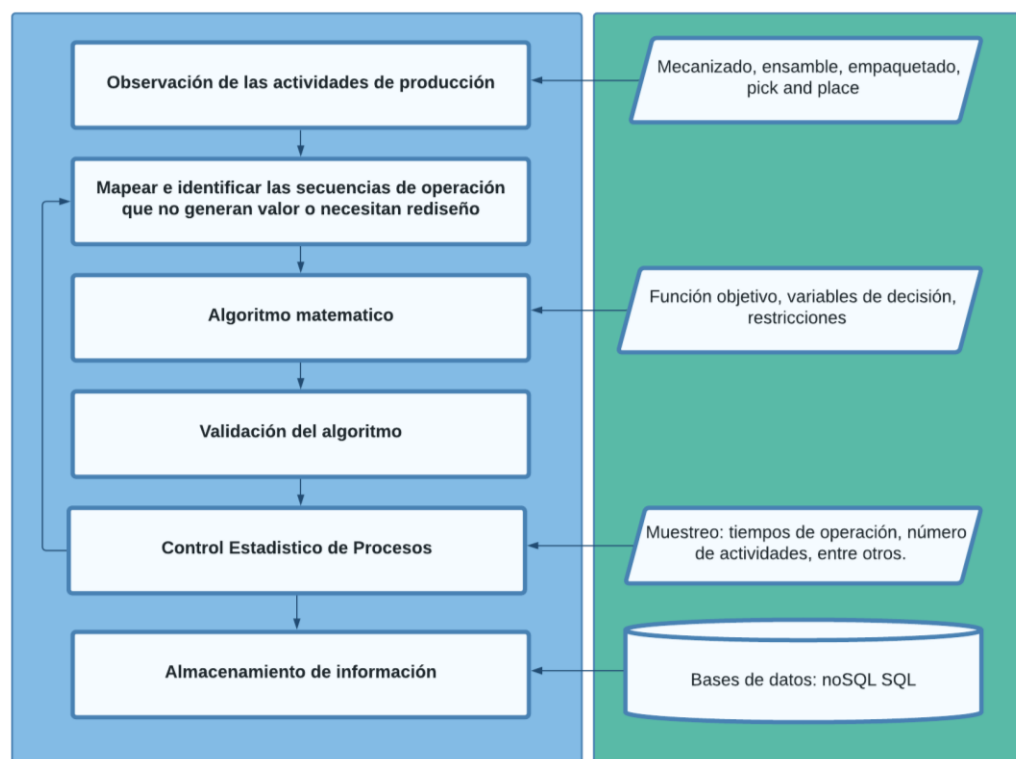
### **3. 5. Gestión de procesos**

La segunda parte de la metodología se detalla en la Figura 10, la cual corresponde al nivel de gestión de datos y de la producción, como etapa principal se realizará una observación de las actividades de producción que permita determinar cuáles actividades o etapas de cada uno de los procesos no genera valor a la producción o necesitan ser mejoradas, a través de restricciones como disponibilidad de materia prima, tiempo de producción, costes de producción, disponibilidad de mano de obra y maquinaria se desarrolla un algoritmo matemático basado en programación lineal que permita optimizar la producción reduciendo sus costos, para así mejorar la toma de decisiones dentro de los procesos de producción, lo cual permita diseñar o rediseñar un marco de planificación de la producción.



Figura 10

*Metodología propuesta para la gestión de procesos*



Tal y como se menciona en (Tripathi et al., 2022), “ *la planificación de los marcos de producción es un tema crítico en el entorno de fabricación, en las industrias de todo el mundo, se gasta una cantidad muy alta en la planificación de la producción*”. Por lo tanto, se requiere una mejora en la estrategia y las técnicas que se utilizan para optimizar la producción.

Para validar el funcionamiento de las operaciones se aplica un control de procesos estadísticos utilizando gráficas de control que permita la visualización e interpretación de los resultados, las variables que intervienen en esta etapa involucran a los tiempos de operación total e individual de cada estación de trabajo, las actividades realizadas y a los tiempos de paro.

Para aprovechar las herramientas de la industrial 4.0 la información recopilada de los procesos industriales, así como los resultados del algoritmo matemático serán almacenados en una base de datos no SQL en la nube utilizando MongoDB ATLAS permitiendo de esta manera una mejor gestión de los datos, con documentos recolectados y almacenados de manera ordenada con un acceso permanente a los mismos para su posterior utilización.

## Capítulo IV

### 4. Diseño

#### 4. 1. Introducción

La metodología propuesta en el presente proyecto de investigación fue implementada para el control y monitoreo de procesos industriales que simulan operaciones de fábrica como traslado de piezas, mecanizados de piezas a medida, empaquetado y almacenamiento de los productos. Dichos procesos no cuentan con una gestión de producción ni mucho menos con una gestión de los datos. Definida las secuencias de operaciones que tendrá cada proceso y conociendo las etapas que necesitan ser optimizadas se procede con la aplicación de la primera fase de la metodología:

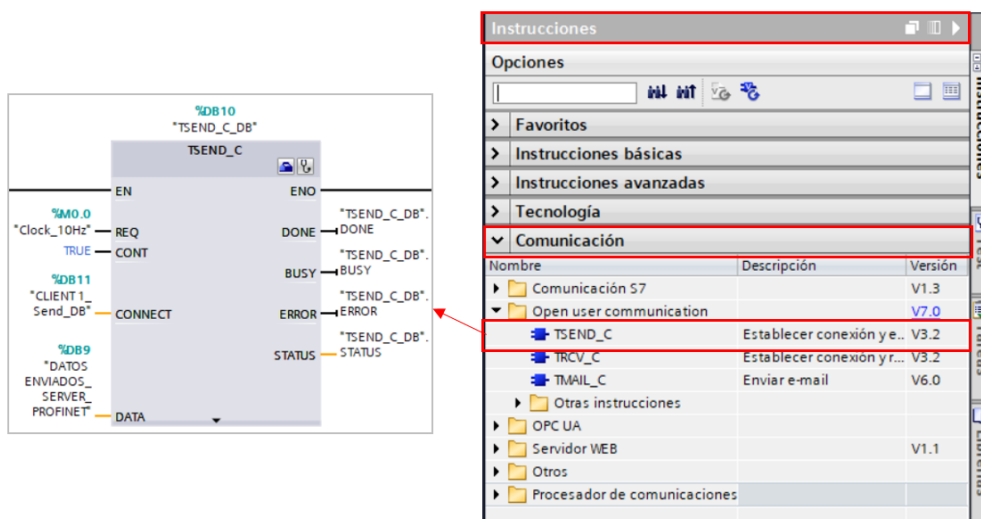
#### 4. 2. Selección de protocolos y comunicación

##### 4.2.1. *Profinet*

Para poder simplificar la comunicación, siemens a través del software TIA PORTAL, permite agregar al área de trabajo las instrucciones TSEND\_C y TRCV\_C cuyas funciones son: enviar y recibir información de manera unidireccional o bidireccional, como también permite configurar y establecer una comunicación. Como se muestra en la Figura 11 para agregar estas instrucciones al área de trabajo basta ingresar a la ventana de instrucciones buscar la carpeta de comunicación abierta, se elige la versión y se agrega primero el bloque TSEND\_C para enviar los datos al servidor, permitiendo así crear una comunicación del tipo TCP o ISO sobre TCP con un interlocutor.

**Figura 11**

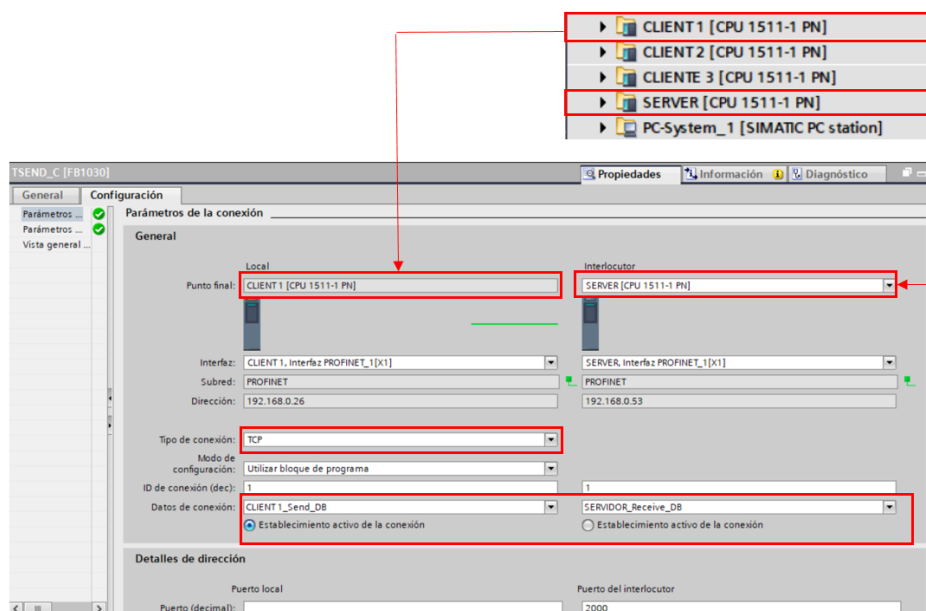
*Ubicación del bloque TSEND\_C en TIA portal*



El bloque TSEND\_C ya integra de manera interna las instrucciones de comunicación TSEND, T\_DIAG, T\_RESET, T\_DISCON y TCON facilitando al usuario enviar y recibir datos, como también presenta la facilidad para establecer o deshacer una conexión. Como se muestra en la Figura 12, para configurar la instrucción TSEND\_C se debe ingresar a las propiedades de instrucción, se selecciona que controlador cumplirá el rol de interlocutor se completa todos los parámetros solicitados y para comprobar que la configuración fue exitosa al lado izquierdo se puede observar dos checks verdes.

**Figura 12**

*Parámetros de configuración del bloque TSEND\_C*



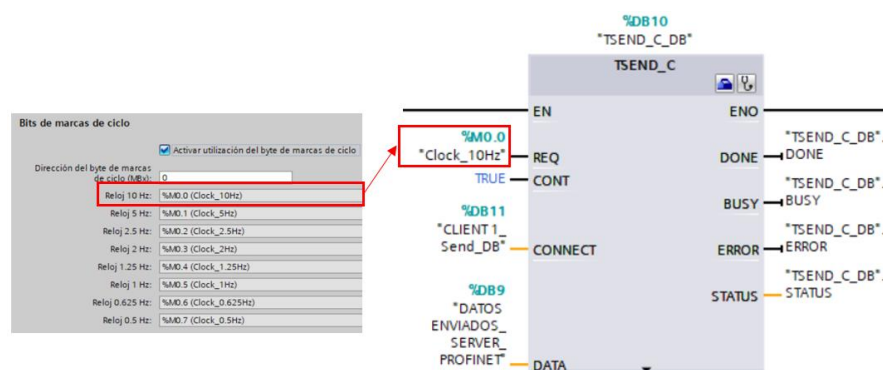
Al seleccionar al interlocutor se debe verificar que ambos controladores se encuentren dentro de la misma subred, además se puede comprobar que las direcciones I.P. sean las correctas, el tipo de conexión por defecto será TCP, para el modo de configuración se elige utilizar bloques de programa. Es muy importante que la ID de comunicación establecida sea única entre el interlocutor y el controlador local.

Los datos de conexión por defecto se encuentran vacíos, por tal motivo se crea uno nuevo, TIA Portal asigna de manera automática el nombre del bloque de datos de conexión, en caso de ser requerido se puede ingresar a las propiedades del DB creado y modificar el nombre para evitar posibles confusiones o equivocaciones. Además, se debe asegurar que solamente uno de los dos dispositivos se encuentre estableciendo una comunicación.

En la Figura 13 se muestra la configuración externa de la instrucción. Una vez establecida la comunicación se envía al parámetro REQ la orden de transmitir los datos, cada vez que detecte un flanco ascendente, se elige una marca de ciclo M0.0 aunque se puede seleccionar otra área de memoria como tipo: [I], [Q], [D], [L], [T], [C]. Los bits de memoria de ciclo se activan ingresando a las propiedades del controlador en la ventana de marcas del sistema y de ciclo.

**Figura 13**

*Configuración del bloque TSEND\_C*



*Nota.* El siguiente bloque está configurado solamente para enviar los datos desde el controlador local a un controlador interlocutor no se permite la recepción de datos.

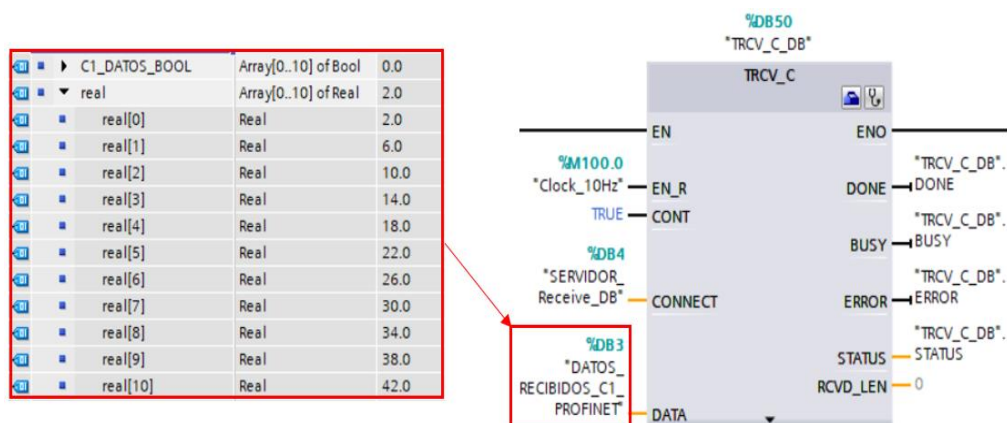
El parámetro CONT se encarga del control de la conexión, el tipo de dato admitido es solamente del tipo Booleano, se ingresa True para establecer y mantener la conexión o se ingresa False para deshacer la conexión. De acuerdo a la versión de la instrucción que se esté utilizando se mostrará el parámetro LEN que se encarga de establecer el número máximo de bytes a enviar. El siguiente parámetro es CONNECT cuya estructura de conexión puede ser de conexión programada o conexión configurada. Para finalizar se tiene el parámetro DATA, que no es más que el área de transmisión que contiene las direcciones y las longitudes de los datos que

se enviarán al interlocutor. Se debe tener en cuenta que, en el área de recepción el parámetro DATA debe tener la misma estructura y longitud que el área de envi6.

Una vez finalizada la configuraci6n de la instrucci6n TSEND\_C, se ingresa nuevamente a la carpeta de comunicaci6n y se configura la instrucci6n TRCV\_C que se encargará de recibir los datos del controlador local, para la configuraci6n de este bloque se sigue los mismos pasos anteriormente expuestos. Al finalizar la configuraci6n tanto del establecimiento de la conexi6n y de los parámetros, el resultado obtenido ser igual o semejante al mostrado en la Figura 14.

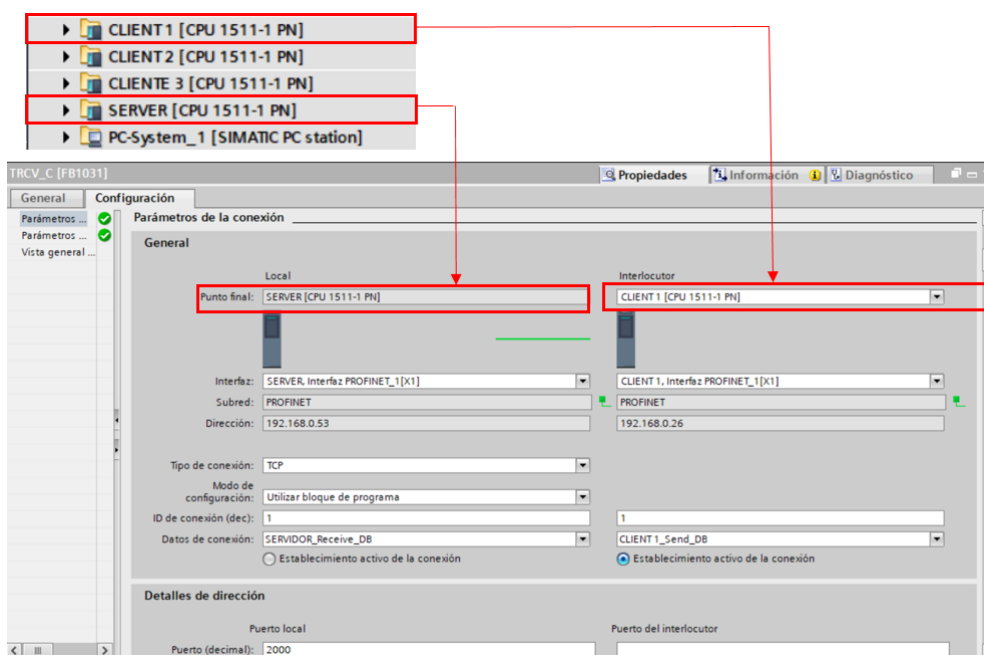
**Figura 14**

*Configuraci6n del bloque TRCV\_C*



**Figura 15**

*Parámetros de configuración del bloque TRCV\_C*

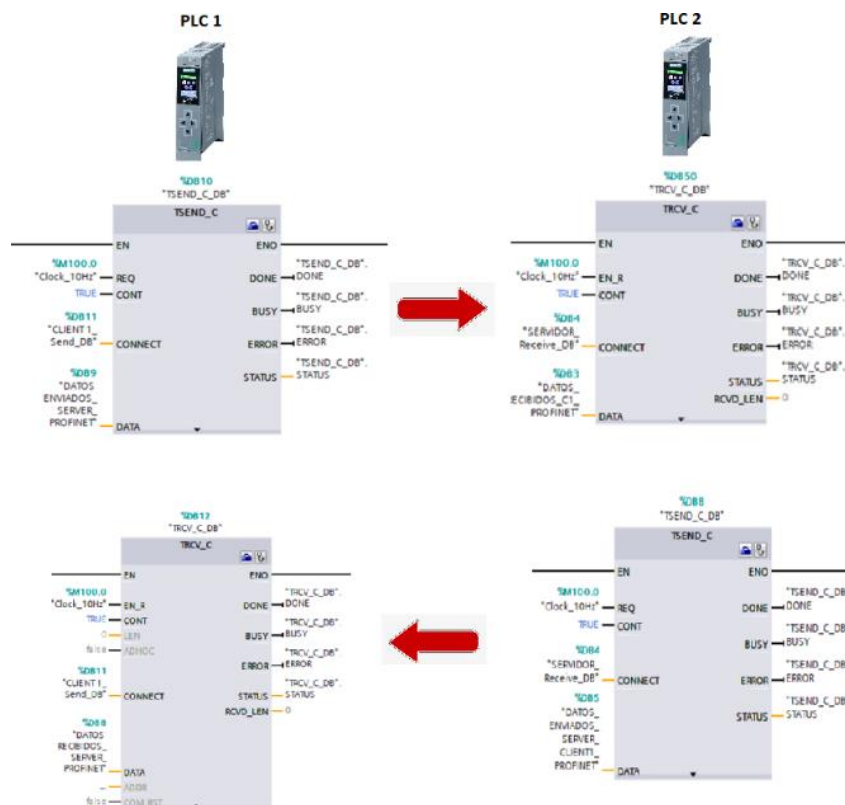


Cabe destacar que la configuración realizada hasta el momento solamente permite escribir datos desde el cliente 1 al servidor. Para lograr el envío y recepción de datos de manera bidireccional se debe agregar y configurar las instrucciones TSEND\_C y TRCV\_C de manera contraria tal y como se muestra en la Figura 16.



Figura 16

Esquema de envío y recepción de datos entre dos controladores

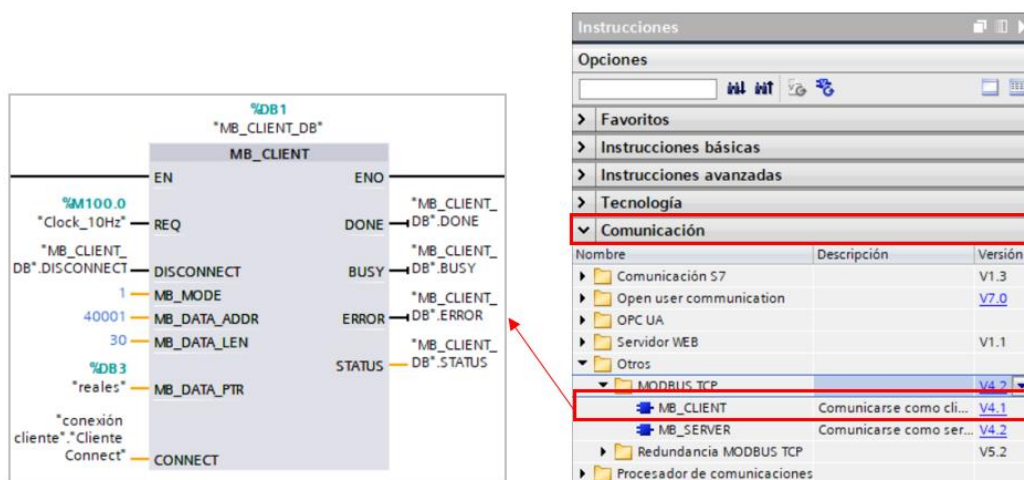


#### 4.2.2. Modbus

Para configurar la comunicación Modbus TCP se utiliza los bloques MB\_CLIENT y MB\_SERVER, la instrucción MB\_CLIENT cumple el rol de comunicarse como cliente a través de Profinet, permitiendo establecer una conexión cliente-servidor, de acuerdo a la configuración que se realice, se puede enviar órdenes y recibir respuestas, así como controlar la conexión o desconexión del cliente. Como se muestra en la Figura 17 se ingresa a la ventana de instrucciones en la carpeta de otro tipo de comunicaciones se elige la versión y se arrastra el bloque MB\_CLIENT al área de trabajo.

Figura 17

Ubicación del bloque MB\_CLIENT en TIA Portal

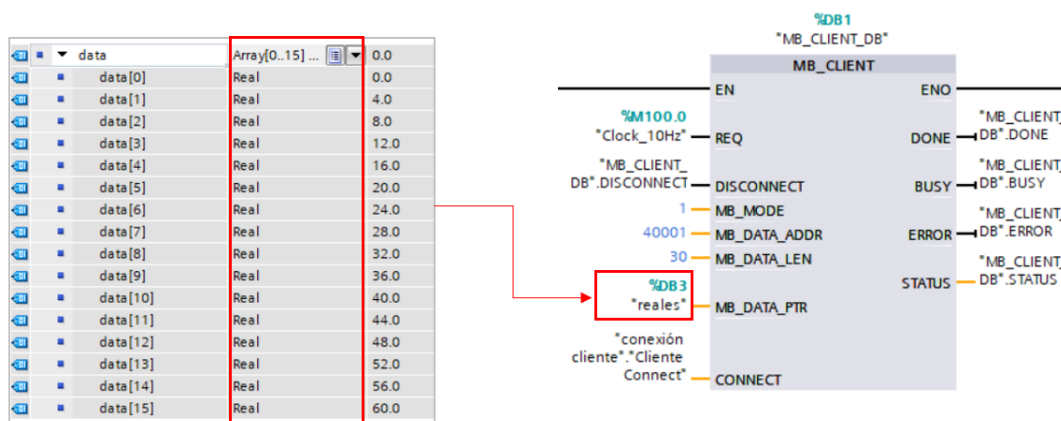


El siguiente paso es configurar cada uno de los parámetros del bloque MB\_CLIENT, al igual que en el apartado 4.2.1 REQ envía la orden para controlar la comunicación cuando su entrada este activa con un flanco ascendente, DISCONNECT controla el establecimiento de la conexión cuando se encuentra en False y, en caso de querer deshacer la conexión se cambia False por True, MB\_MODE define la orden que tendrá Modbus ya sea de escritura, lectura o diagnóstico, en este caso se ingresa el valor 1 correspondiente al modo de escritura, en otras palabras ahora el controlador cliente podrá solamente escribir datos en el controlador servidor y al configurar el servidor este podrá leer la información del cliente más no escribir. El parámetro MB\_DATA\_ADDR contiene información de la dirección a partir de la cual la instrucción MB\_CLIENT pretende leer o escribir los datos, se puede seleccionar valores desde 40001 hasta el 49999. MB\_DATA\_LEN define la longitud que tendrán los datos ya sean bits o Word, MB\_DATA\_PTR se configura con un bloque de datos DB, donde se pretende enviar o recibir la información desde el servidor Modbus, por último, el parámetro CONNECT contiene la

estructura que se utilizará para establecer la comunicación ya sea por la instrucción TCON\_IP\_v4 o TCON\_configured.

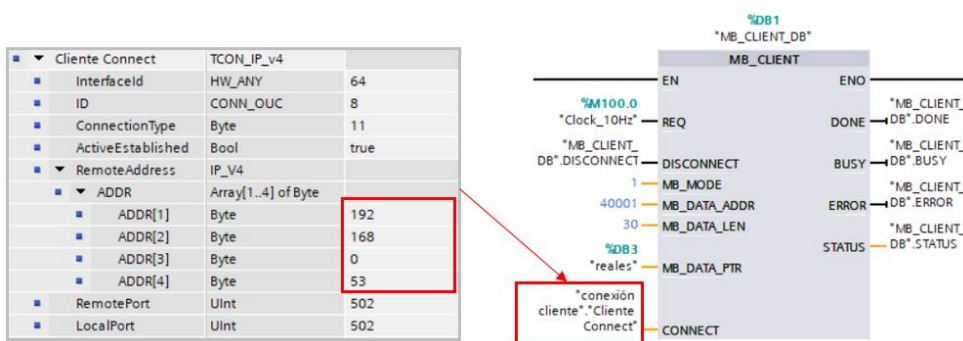
**Figura 18**

*Configuración del bloque MB\_CLIENT*



**Figura 19**

*Configuración del bloque de Datos TCON\_IP\_v4*

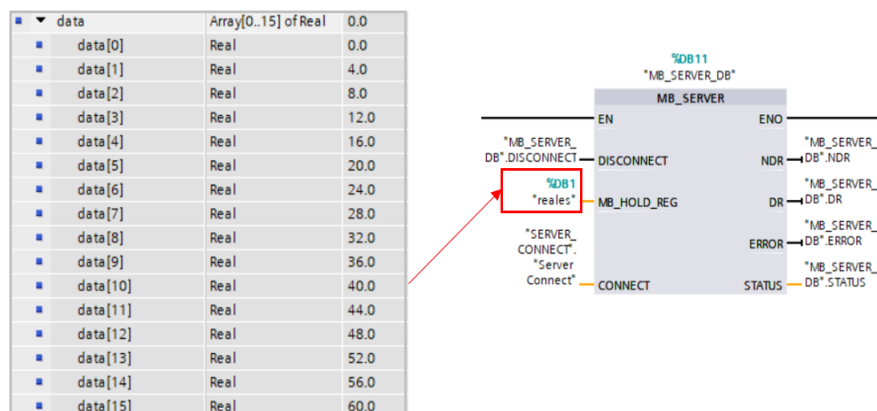


Como se muestra en la Figura 20, el bloque MB\_SERVER se configura dentro del bloque de programas del PLC servidor, de igual manera que MB\_CLIENT. El parámetro DISCONNECT contiene una marca de memoria M100.0 para controlar la conexión, MB\_HOLD\_REG contiene el registro con los datos que puede acceder el cliente Modbus ya sea para leer o escribir y por

último el parámetro CONNECT contiene la estructura para establecer la comunicación configurada con TCON\_IP\_v4.

**Figura 20**

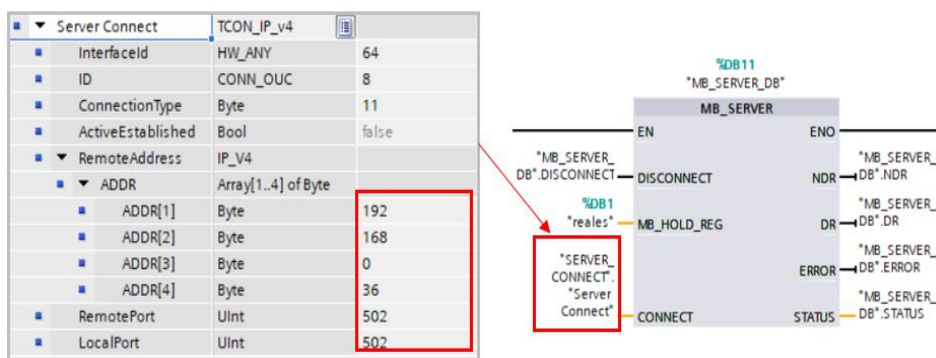
*Configuración del bloque MB\_SERVER del cliente*



Como se muestra en la Figura 21 al configurar el bloque MB\_SERVER, la dirección I.P. que debe agregarse corresponde a la I.P. del controlador cliente, y el puerto por el cual se realiza el intercambio de información debe ser el mismo tanto para el bloque MB\_SERVER y MB\_CLIENT.

**Figura 21**

*Configuración del bloque de datos TCON\_IP\_V4 del servidor*



### 4.2.3. Comunicación S7

Para la comunicación S7 se utiliza los bloques PUT-GET para enviar y recibir información. Como se muestra en la Figura 22 para agregar los bloques PUT y GET se ingresa a la ventana instrucciones en la carpeta de comunicaciones S7 se elige la versión y se suelta en el área de trabajo, al ser una comunicación abierta propia de Siemens se puede agregar directamente los dos bloques dentro del controlador cliente.

**Figura 22**

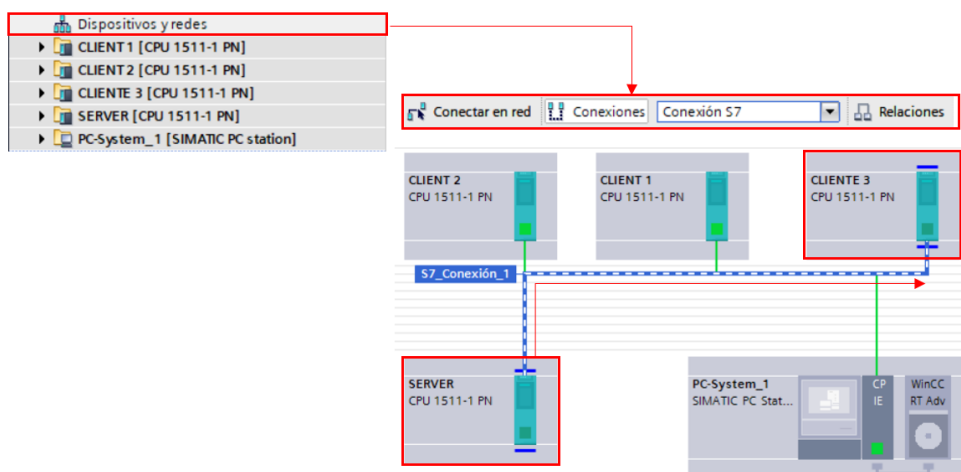
*Ubicación de los bloques PUT y GET en TIA Portal*



Para establecer la comunicación se ingresa a dispositivos y redes, se cambia a la opción de conexiones, se elige conexión s7 y se crea un nuevo enlace arrastrando desde el controlador cliente al controlador servidor, tal y como se muestra en la Figura 23.

Figura 23

Creación del enlace S7 cliente-servidor



Como se muestra en la Figura 24 para comprobar que la conexión fue realizada, se ingresa a la ventana vista de redes, conexiones y se visualiza el nombre de la conexión local creada, en este caso el nombre es S7\_Conexión 1.

Figura 24

Conexiones de los dispositivos y redes

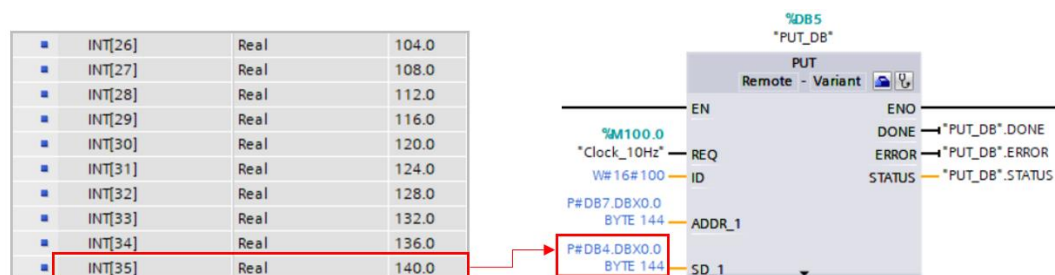
Vista de redes						
Vista general de la red	Conexiones	Relaciones	Comunicación E/S	VPN		
Nombre de conexión local	Punto final local	ID local (hex)	ID del interloc...	Interlocutor	Tip...	
S7_Conexión_2	SERVER [CPU 1511-...	101	100	CLIENTE 3 [CPU 15...	Co...	
S7_Conexión_2	CLIENTE 3 [CPU 15...	100	101	SERVER [CPU 1511-...	Co...	
S7_Conexión_1	SERVER [CPU 1511-...	100	101	CLIENTE 3 [CPU 15...	Co...	
S7_Conexión_1	CLIENTE 3 [CPU 15...	101	100	SERVER [CPU 1511-...	Co...	
HMI_Conexión_1	PC-System_1 \ HMI...			CLIENTE 3 [CPU 15...	Co...	
HMI_Conexión_3	PC-System_1 \ HMI...			CLIENTE 1 [CPU 151...	Co...	

La configuración establecida para el bloque PUT se muestra en la Figura 25, esta instrucción permite escribir en el PLC servidor. El parámetro REQ contiene una marca de memoria M100.0 para controlar el intercambio de datos cuando este se encuentre con un flanco ascendente. ADDR\_1 se encuentra en el puntero del bloque de datos DB4 posición X0.0 y

con una longitud de 144 bytes, el parámetro SD\_1 se encuentra en el bloque de datos (DB4) propia de la CPU desde donde se transmitirán los datos.

**Figura 25**

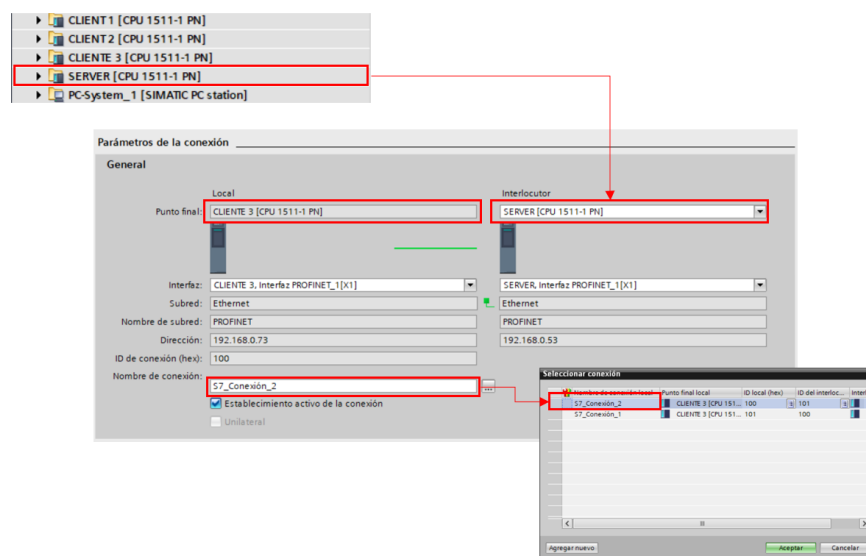
*Configuración del bloque PUT*



Para establecer el parámetro ID se debe ingresar a las propiedades del bloque PUT, como se puede observar en la Figura 26, se debe elegir cual será el controlador que cumple el rol de interlocutor, automáticamente el resto de parámetros serán completados.

**Figura 26**

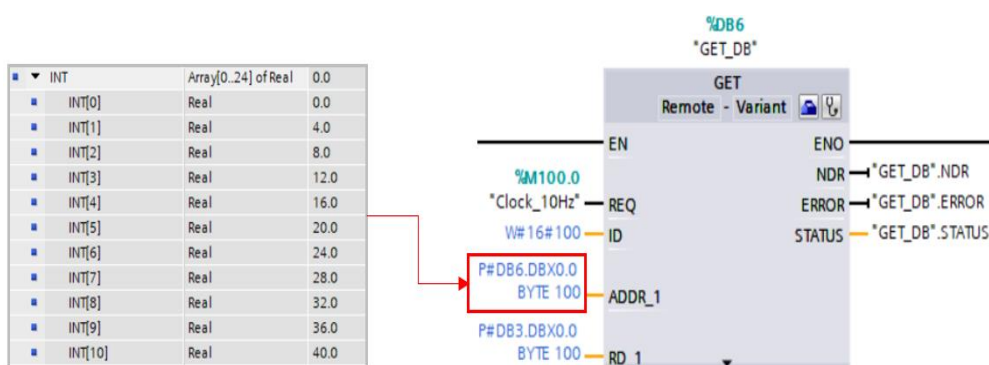
*Propiedades del bloque PUT*



Para leer los datos provenientes del cliente, el bloque GET lleva la configuración mostrada en la Figura 27, la configuración es similar al bloque PUT, el parámetro ADDR\_1 contiene un bloque de datos DB que será el área de donde se leerán los datos y el parámetro RD\_1 contiene un bloque de datos DB en donde se almacenarán los datos recibidos.

**Figura 27**

*Configuración del bloque GET*



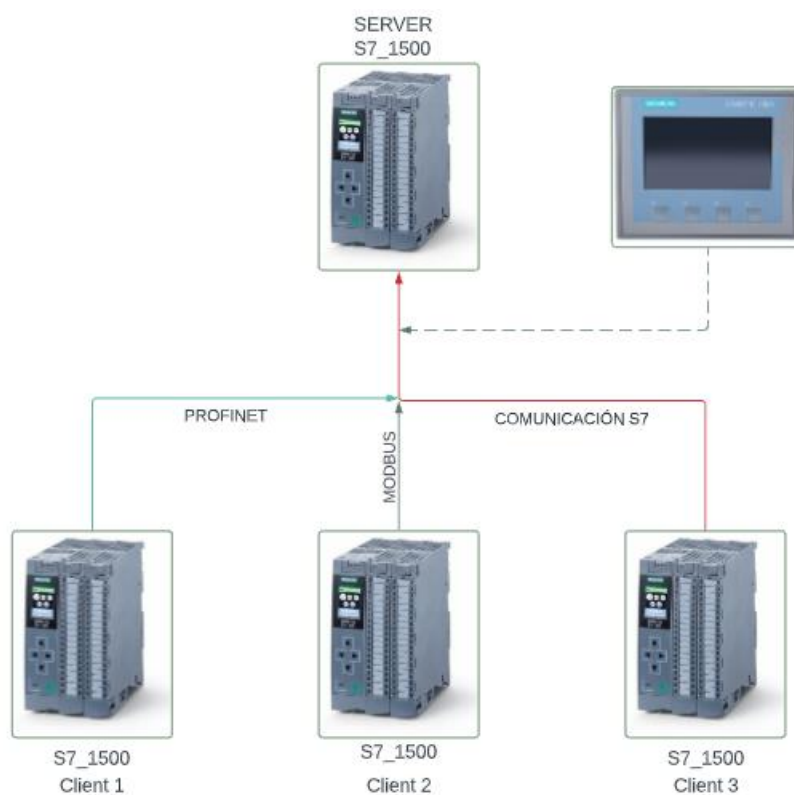
### 4. 3. Estructura de comunicación del Sistema de Control Distribuido

La arquitectura de la red de comunicación industrial consta de 4 PLC's de la marca Siemens S7 CPU 1511-1 PN, como se muestra en la Figura 28, se tiene 3 clientes y un servidor, cada cliente contará con un proceso virtual industrial y con sus respectivos bloques de comunicación configurados previamente en el apartado 4. 2, además se agrega un PC System para el control y monitoreo.



**Figura 28**

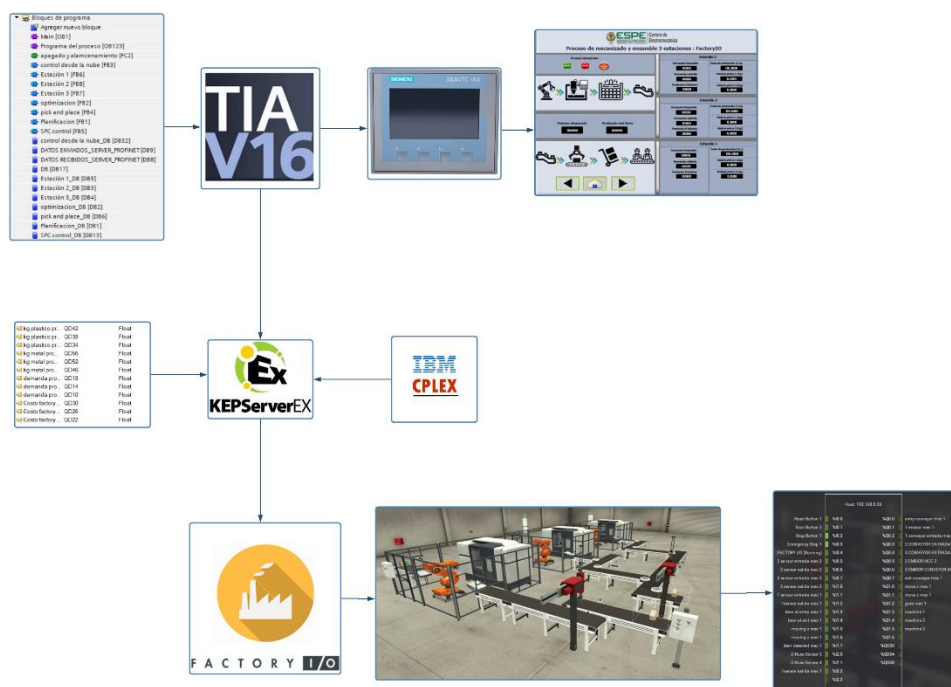
*Estructura de comunicación del DCS*



Como se muestra en la Figura 29, el PLC 1 contiene el proceso de fábrica realizado en FACTORY IO, se agrega a una subred Profinet utilizando los bloques TSEND\_C y TRCV\_C, además se agrega una pantalla HMI creada en un PC SYSTEM para el control y monitoreo del proceso.

Figura 29

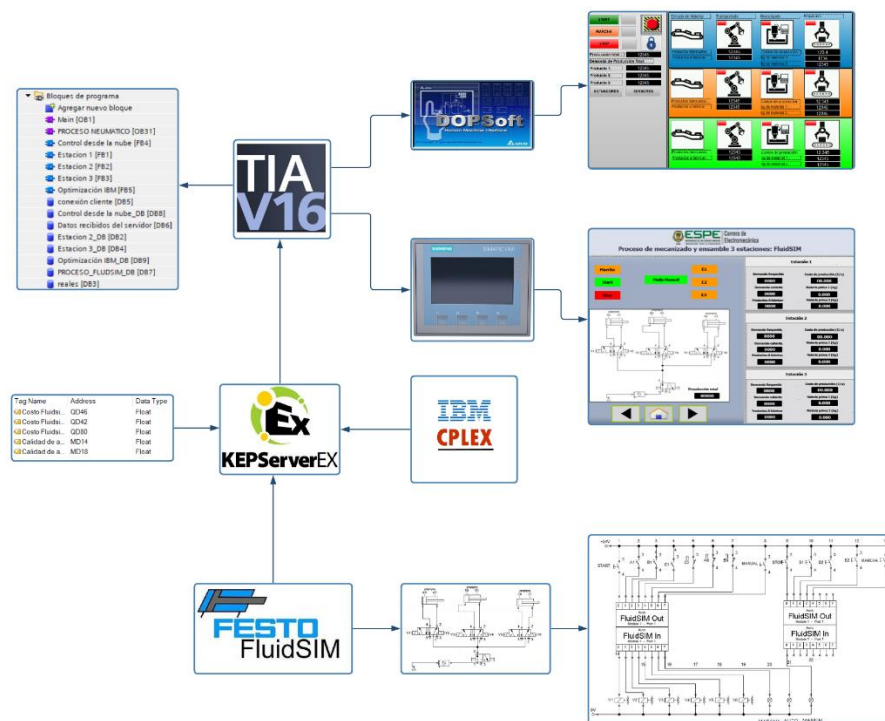
Estructura de comunicación del cliente uno



A continuación, en la Figura 30, el PLC 2 se configura con el protocolo Modbus TCP, utilizando los bloques MB\_CLIENT y MB\_SERVER, el proceso que se controla y monitorea corresponde a un proceso neumático, conformado por 9 actuadores de doble efecto. Para permitir la comunicación entre TIA Portal y FluidSIM se utiliza el servidor OPC KepServer para crear, configurar y modificar los tags necesarios para la comunicación. Además, se agrega un HMI en una pantalla simulada de la marca Delta utilizando DopSoft como también en el PC System.

Figura 30

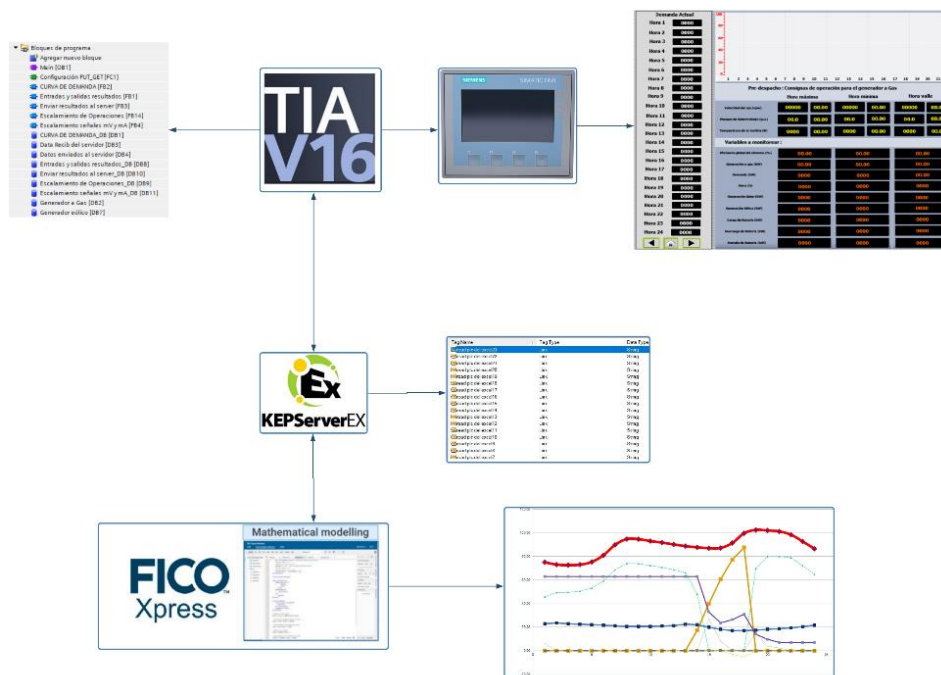
Estructura de comunicación del cliente dos



La estructura de comunicación del cliente 3 se muestra en la Figura 31, se utiliza el enlace s7 para la comunicación, el proceso corresponde a la parametrización de las consignas de operación para n generadores involucrados en un problema de optimización relacionado con el despacho económico. Para permitir el envío y recepción de datos entre TIA Portal y FICO XPRESS se utilizó el servidor OPC KepServer configurando un DDE CLIENT, el control y monitoreo se realiza desde TIA Portal utilizando un PC System.

Figura 31

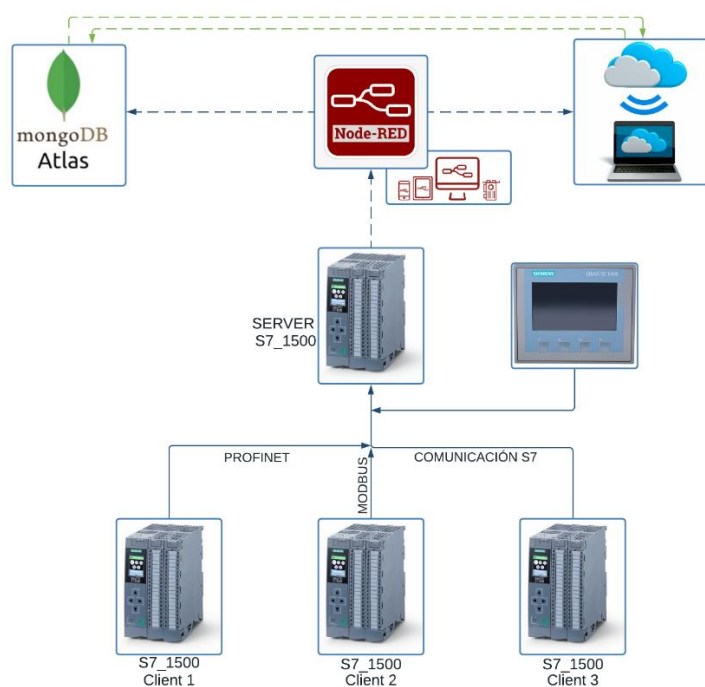
Estructura de comunicación del cliente tres



Para finalizar se utiliza Node-Red para permitir el intercambio de datos con la Web utilizando Fred-Sensetecnic. Además, desde Node-Red se establece una comunicación con una base de datos NoSQL local y en la nube utilizando MongoDB Atlas, que posteriormente permitirá almacenar y visualizar la información más relevante de cada uno de los procesos.

Figura 32

*Estructura de comunicación del DCS con la nube y la base de datos*



#### 4. 4. Simulación de procesos industriales

##### 4.4.1. Proceso uno en FluidSIM: unidad de montaje y mecanizado

Para simular los componentes neumáticos se utilizó el software FluidSIM-P, cuyo entorno permite agregar componentes eléctricos, así como interruptores y puertos que permiten establecer una comunicación con otros servidores mediante EASYPOR, OPC o por medio de un DDE. En la Tabla 1 se muestra los componentes empleados para la simulación.

**Tabla 1**

*Componentes eléctricos y neumáticos empleados en el proceso uno*

Denominación de Componentes	Cantidad
Fuente de tensión (0V)	1

Denominación de Componentes	Cantidad
Fuente de tensión (24V)	1
Fuente de aire comprimido	3
Indicador Luminoso	3
Puerto de entrada FluidSIM	3
Unidad de mantenimiento	3
Válvulas de 3/n vías	3
Puerto de salida FluidSIM	4
Pulsador (Obturador)	7
Cilindro doble efecto	9
Regla de distancia	9
Válvula de 5/n vías	9
Obturador	18
Solenoides de válvula	18
Total	90

Como se muestra en la Figura 33, el proceso consta de 9 cilindros de doble efecto que simulan el rol de 3 estaciones de mecanizado y ensamble, cada estación está conformada por 3 actuadores de doble efecto, el funcionamiento de cada estación se describe a continuación: El actuador “A” activa la entrada de la materia prima, cuando el material a ingresado, se activa el actuador “B” del centro de mecanizado, pasado cierto tiempo se activa el actuador “C” correspondiente al ensamble de la pieza y posterior salida, finalizando el proceso. El proceso trabaja en ciclo continuo como también en régimen manual y automático, además se agregan lámparas que indican al operario en que régimen se está trabajando. En la Figura 34 se observa el HMI implementado en una pantalla de la marca Delta y en un PC System de TIA portal.

Figura 33

Proceso neumático implementado en FluidSIM-P

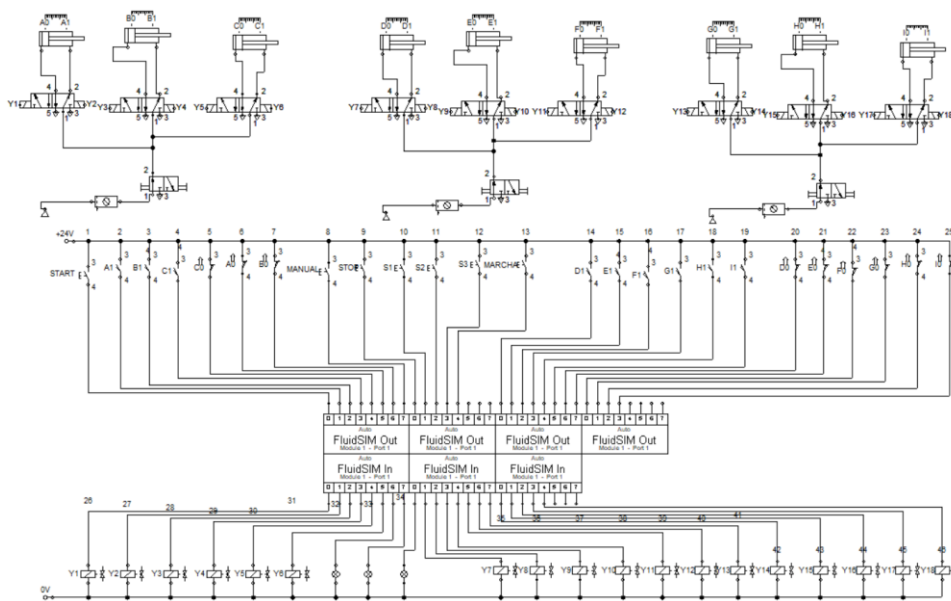
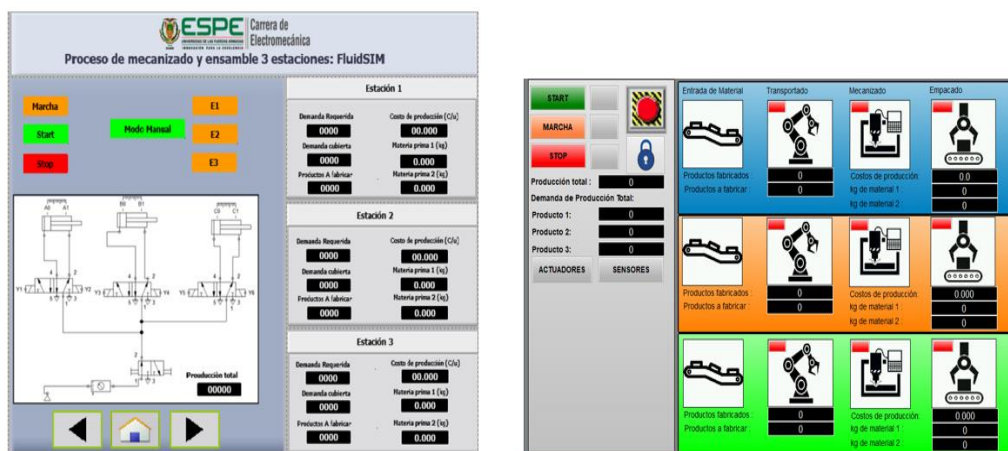


Figura 34

HMI del proceso neumático, implementado en DopSoft y TIA Portal

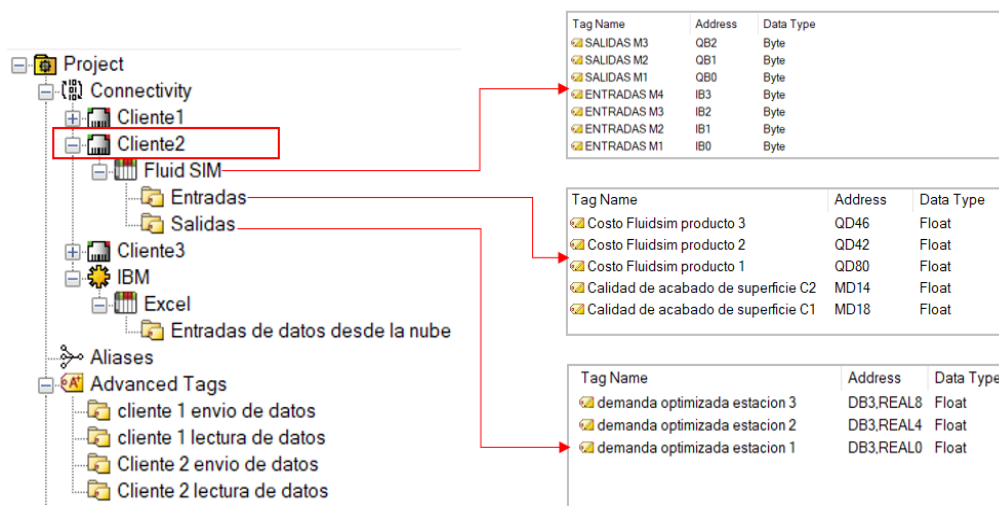


Como se observa en la Figura 35, para la comunicación entre FluidSIM con TIA Portal se utiliza el servidor OPC KepServer, se ha creado un canal que contiene los tags tipo byte para el

control del proceso. Además, se añade dos grupos para leer las restricciones y escribir los resultados del algoritmo matemático en la hoja de Excel como se muestra en la Figura 36.

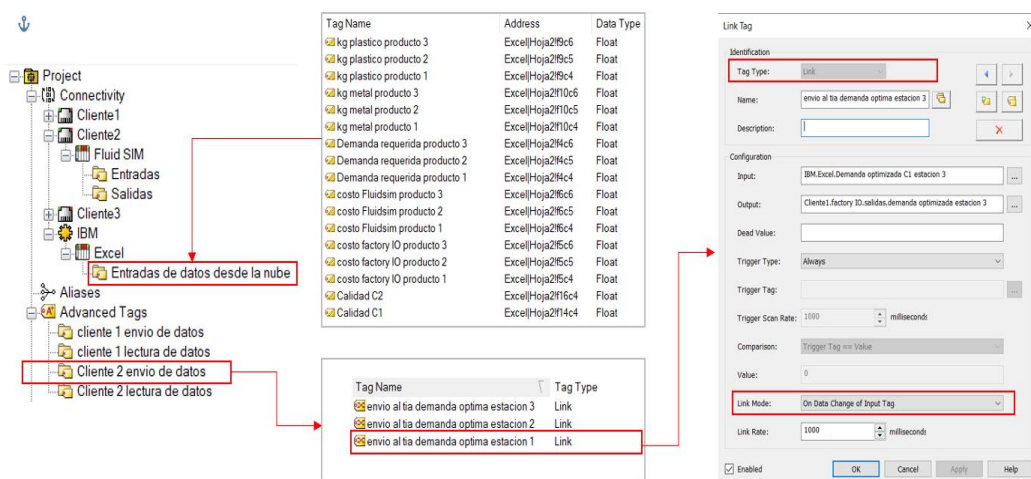
**Figura 35**

*Configuración de la comunicación FluidSIM-TIA Portal utilizando KepServer*



**Figura 36**

*Configuración de un canal DDE Client en KepServer*



Tag Name	Address	Data Type
kg plastico producto 3	Excel Hoja2 B6c5	Float
kg plastico producto 2	Excel Hoja2 B6c5	Float
kg plastico producto 1	Excel Hoja2 B6c4	Float
kg metal producto 3	Excel Hoja2 I0c6	Float
kg metal producto 2	Excel Hoja2 I0c5	Float
kg metal producto 1	Excel Hoja2 I0c4	Float
Demanda requerida producto 3	Excel Hoja2 H4c6	Float
Demanda requerida producto 2	Excel Hoja2 H4c5	Float
Demanda requerida producto 1	Excel Hoja2 H4c4	Float
costo Fluidsim producto 3	Excel Hoja2 K6c6	Float
costo Fluidsim producto 2	Excel Hoja2 K6c5	Float
costo Fluidsim producto 1	Excel Hoja2 K6c4	Float
costo factory IO producto 3	Excel Hoja2 K6c6	Float
costo factory IO producto 2	Excel Hoja2 K6c5	Float
costo factory IO producto 1	Excel Hoja2 K6c4	Float
Calidad C2	Excel Hoja2 I16c4	Float
Calidad C1	Excel Hoja2 I14c4	Float

Tag Name	Tag Type
envio al tia demanda optima estacion 3	Link
envio al tia demanda optima estacion 2	Link
envio al tia demanda optima estacion 1	Link

**Link Tag**

Identification

Tag Type:

Name:

Description:

Configuration

Input:

Output:

Dead Value:

Trigger Type:

Trigger Tag:

Trigger Scan Rate:  milliseconds

Comparison:

Value:

Link Mode:

Link Rate:  milliseconds

Enabled

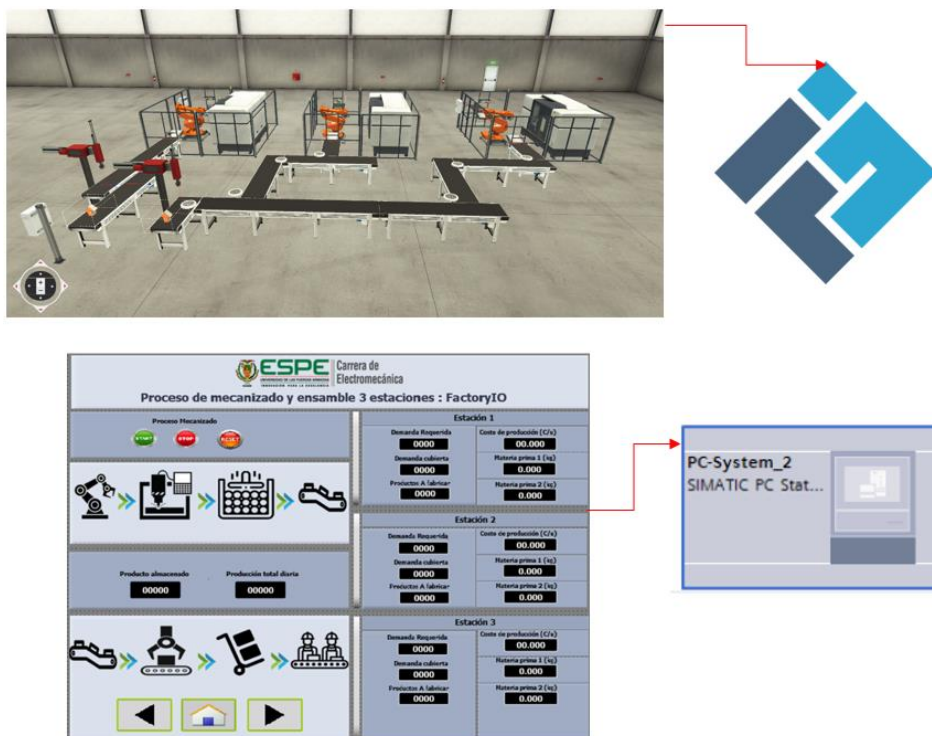


#### 4.4.2. Proceso dos en Factory IO: mecanizado y ensamble

En Factory IO se ha implementado 3 estaciones de mecanizado y ensamble, cada estación posee un brazo robot, cintas transportadoras para el ingreso materia prima y para la salida del producto mecanizado, al final de cada estación se agrega un brazo cartesiano de 2 ejes que simula el ensamble de las piezas. El proceso implementado se muestra en la Figura 37 conjuntamente con el HMI realiza en el PC System. La secuencia que cumple este proceso es similar al mencionado en el apartado 4.4.1.

**Figura 37**

*Proceso de mecanizado implementado en Factory IO*

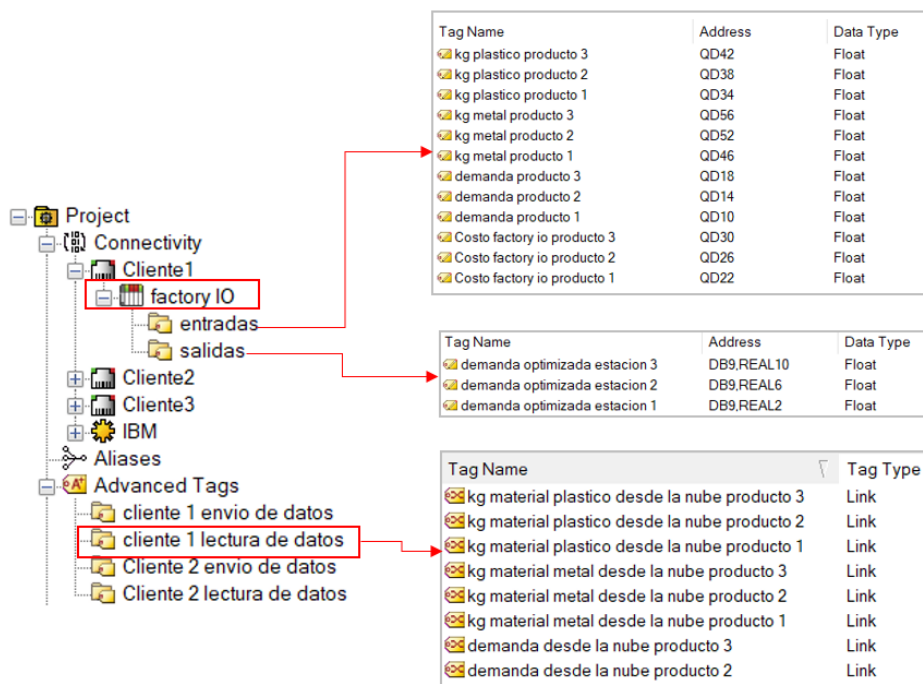


Al ser un proceso de producción similar al presentado en el apartado 4.4.1, los centros de mecanizados simulados en Factory IO también se ven involucrados en el algoritmo matemático para optimizar la producción. Por este motivo como se muestra en la Figura 38, se

configura un nuevo canal en KepServer con los tags necesarios para leer las restricciones para posteriormente escribir sus valores en una hoja de Excel.

**Figura 38**

*Configuración del canal DDE Client para el proceso del cliente uno*



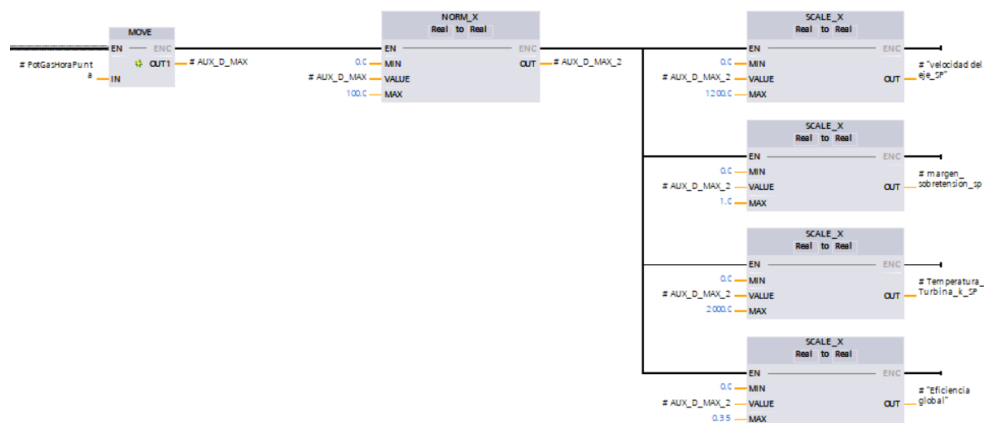
#### 4.4.3. Proceso tres Interno: activación de generadores

El proceso 3 está relacionado con el problema fundamental de operación de los Sistemas Eléctricos de Potencia, repartir la demanda total de un día entre los generadores disponibles procurando que los costos de operación sean mínimos y se respeten los niveles de calidad y las limitaciones técnicas. Cada unidad de generación tiene sus propias restricciones y un costo variable, relacionado con la eficiencia de los GS en función de la generación, debido a la presencia de generaciones mínima se realiza un análisis de predespacho utilizando variables enteras para representar el estado operativo o detenido de las unidades.

Para simular el encendido o apagado de las unidades se agrega un bloque de función con las operaciones de escalado y normalizado teniendo como entrada la potencia generada en cierta hora resultante del problema de optimización y, a la salida los valores de entrada necesarios para la puesta en marcha de un generador a Gas por ejemplo: en la Figura 39 se muestra las operaciones para obtener los valores de la velocidad del eje, margen de sobre Voltaje y temperatura de la turbina que debería ajustarse a las condiciones de operación del generador.

**Figura 39**

*Escalado de Operaciones de E/S para un generador a gas en hora punta.*

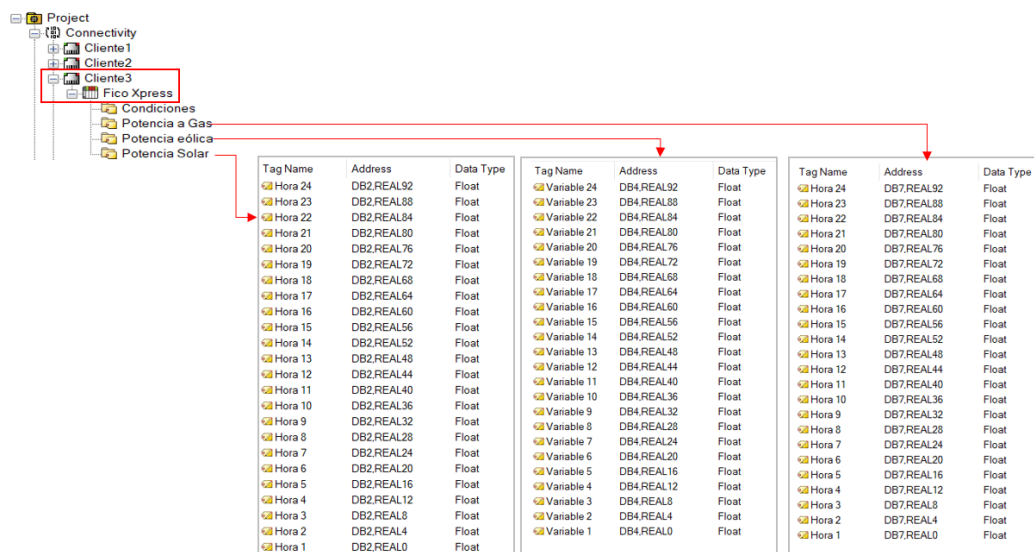


Como no se dispone de un generador físico realizar un monitoreo se hace imposible, por este motivo se realiza las mismas operaciones de escalado y normalizado para obtener los valores de salida como, eficiencia del sistema, temperatura de la turbina y potencia generada.

Los valores de entrada que se necesitan para las operaciones de escalado y normalizado se encuentran en una Hoja de Excel como resultado del algoritmo matemático para poder enviar estos datos al TIA Portal como se muestra en la Figura 40, se crea un canal en KepServer con las variables de generación para cada hora de las unidades de generación más relevantes.

Figura 40

Configuración del canal TCP para la comunicación con el Cliente tres con KepServer



Por la gran cantidad de variables que involucra este proceso y debido a que, se necesita leer y enviar la mayor cantidad de variables a la nube para su monitoreo y almacenamiento se decide no utilizar el canal DDE Client de KepServer en su defecto se utiliza el servidor OPCRouter. Como se muestra en la Figura 41 se crea una nueva conexión para leer los datos de la hoja de Excel y escribirlos en los tags del KepServer de esta manera se simplifico el uso de tags en un 75% de los necesarios si se configurase con KepServer. En la Figura 42 se muestra las conexiones para almacenar los resultados de la optimización en una base de datos de manera local en MongoDB.

Figura 41

Comunicación entre OPCRouter y KepServer

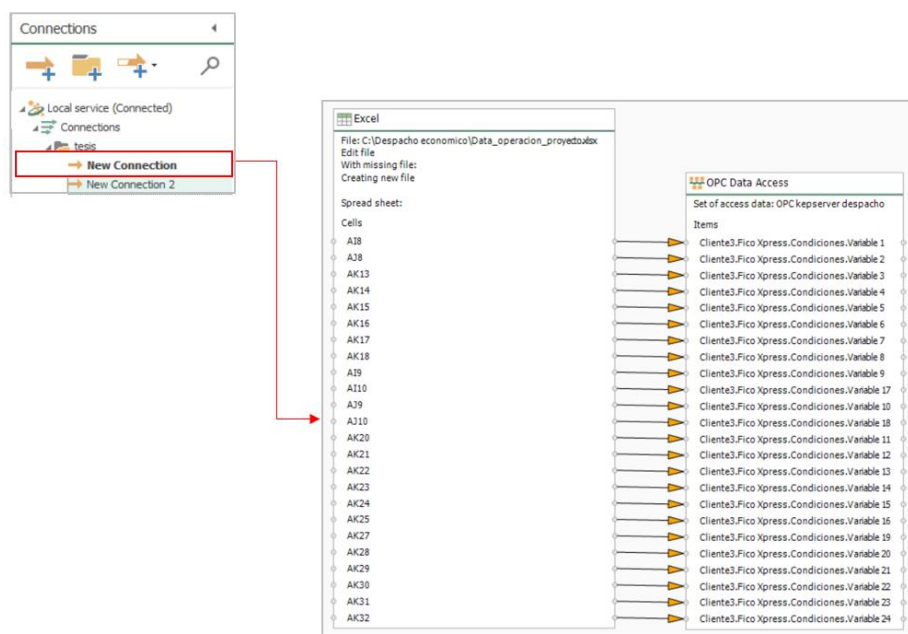
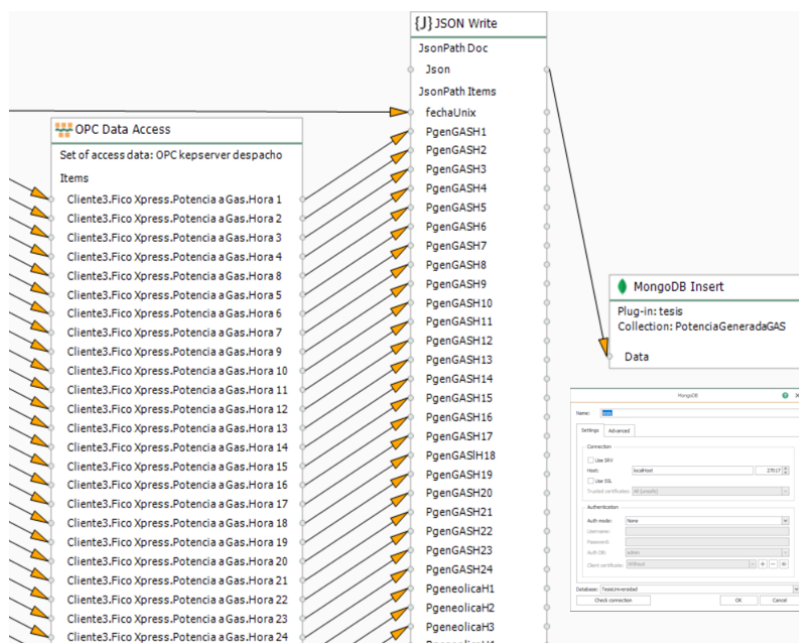


Figura 42

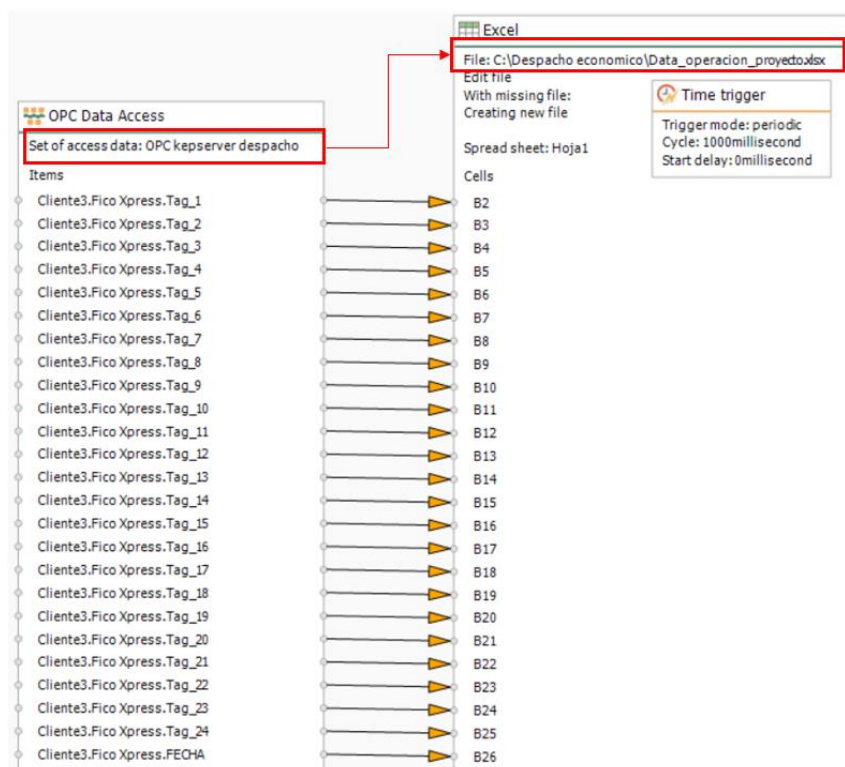
Almacenamiento en MongoDB utilizando OPCRouter



Adicional se configura dos bloques en OPCRouter para escribir en la hoja de Excel, como se muestra en la Figura 43 los datos ingresan desde KepServer y se escriben en la fila columna de la demanda diaria, 24 variables una para cada hora.

**Figura 43**

*Escritura en Excel desde KepServer utilizando OPCRouter*



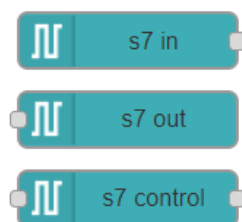
#### 4. 5. Comunicación Web: control y monitoreo de procesos

Para permitir la comunicación entre el controlador servidor y la nube se utiliza la herramienta de programación visual Node-Red, una vez instalado node-red en el computador se ejecuta una consola en PowerShell de Windows como se muestra en la Figura 44, para abrir el entorno de desarrollo de node-red, es necesario copiar la dirección <http://127.0.0.1.1880/> y pegar en el navegador predeterminado.



Figura 45

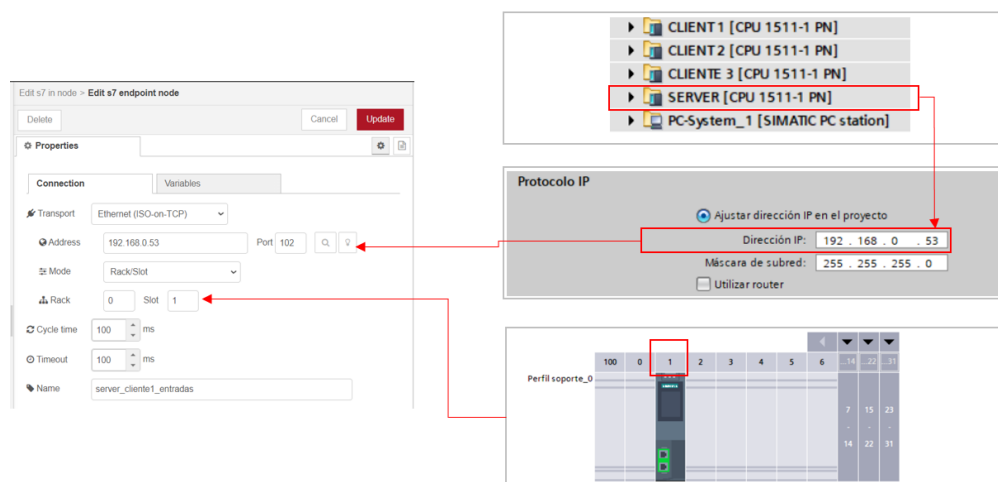
*Node-Red nodos de Comunicación S7*



Para configurar un nodo S7 primero se debe configurar el tipo de conexión, como se muestra en la Figura 46 la información será transportada a través de ethernet (ISO on TCP), la dirección I.P. pertenece al controlador en el cual se aloja la información, se ingresa la I.P. 192.168.0.53 en el puerto 102, en mode se elige rack-slot que identifica la ubicación del controlador dentro de TIA Portal, por defecto se encuentra en el rack 0 slot 1.

Figura 46

*Configuración de conexiones del nodo S7*



Después de configurar el tipo de conexión, se procede a ingresar las variables para el control y monitoreo de cada uno de los procesos, estas variables se encuentran alojadas dentro



de un bloque de datos, por tal motivo la sintaxis debe empezar con DBX, seguido por el tipo de datos, ya sea entero o Real.

**Figura 47**

*Variables de entrada y salida del controlador*

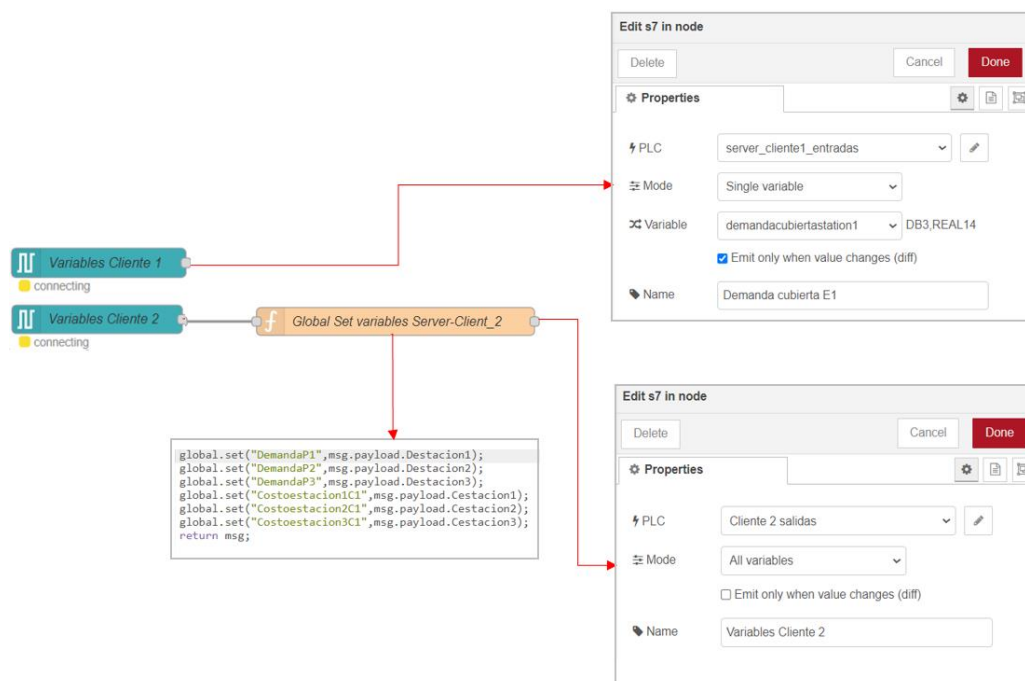
real	Array[0..10] of Real	2.0
real[0]	Real	2.0
real[1]	Real	6.0
real[2]	Real	10.0
real[3]	Real	14.0
real[4]	Real	18.0
real[5]	Real	22.0
real[6]	Real	26.0
real[7]	Real	30.0
real[8]	Real	34.0
real[9]	Real	38.0
real[10]	Real	42.0

Como se muestra en la Figura 48 para leer la información del controlador se realiza de 2 modos, utilizando un nodo de entrada para cada variable y un nodo general para leer todas las variables, posteriormente utilizando los comandos de GET y SET se llama a las variables desde cualquier área de desarrollo dentro del entorno de node-red.

Figura 48

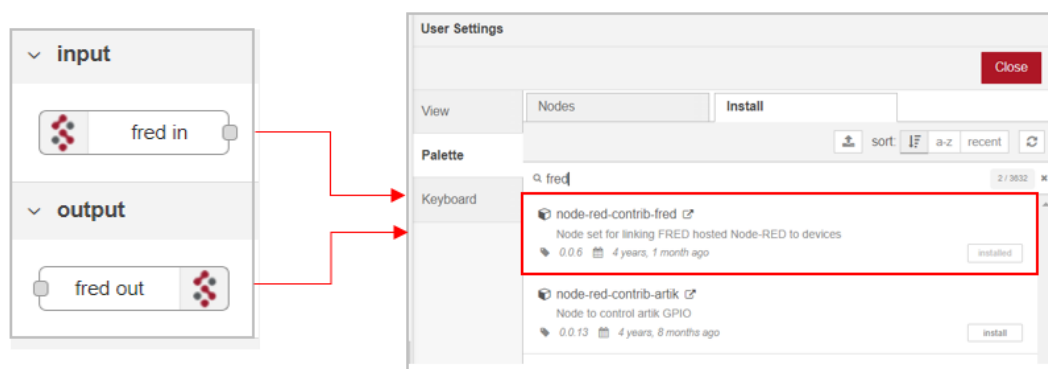
*Lectura de las variables de forma individual y grupal utilizando los nodos S7*



Para la comunicación con la nube se necesita instalar: node-red-contrib-fred, este nodo se puede instalar desde consola por el cmd o directamente de node-red ingresando a la ventana de nodos como se muestra en la Figura 49. Cuando se ha instalado el paquete de nodos de Fred en la parte izquierda en vista de nodos general, se podrá observar dos nodos uno de entrada y otro de salida.

**Figura 49**

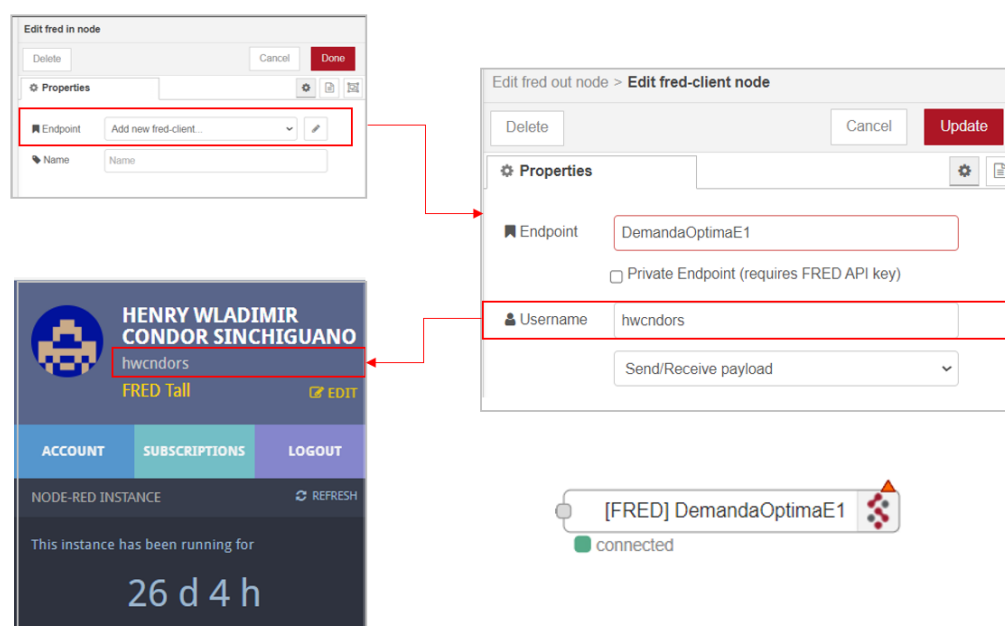
*Instalación de los nodos del servidor compartido en la nube FRED*



Como se muestra en la Figura 50, para configurar los nodos de Fred se añade un nuevo Endpoint que será el identificador de entrada o salida entre Fred y node-red, se debe tener en cuenta que el nombre que se asigne a cada nodo no se debe repetir. El nombre de usuario Fred, se encuentra debajo del nombre de registro en la cuenta de Fred-Sensetecnic.

**Figura 50**

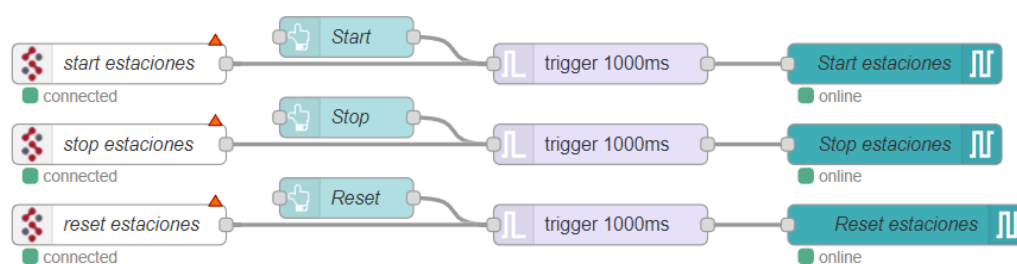
*Creación de un nuevo cliente FRED*



En la Figura 51 se observa los nodos implementados para el control desde la nube del proceso 1 construido en Factory IO. Un nodo Start que permite la activación de los 3 centros de mecanizados, un nodo stop que detiene el proceso en su totalidad y un botón para el reseteo de los contadores.

**Figura 51**

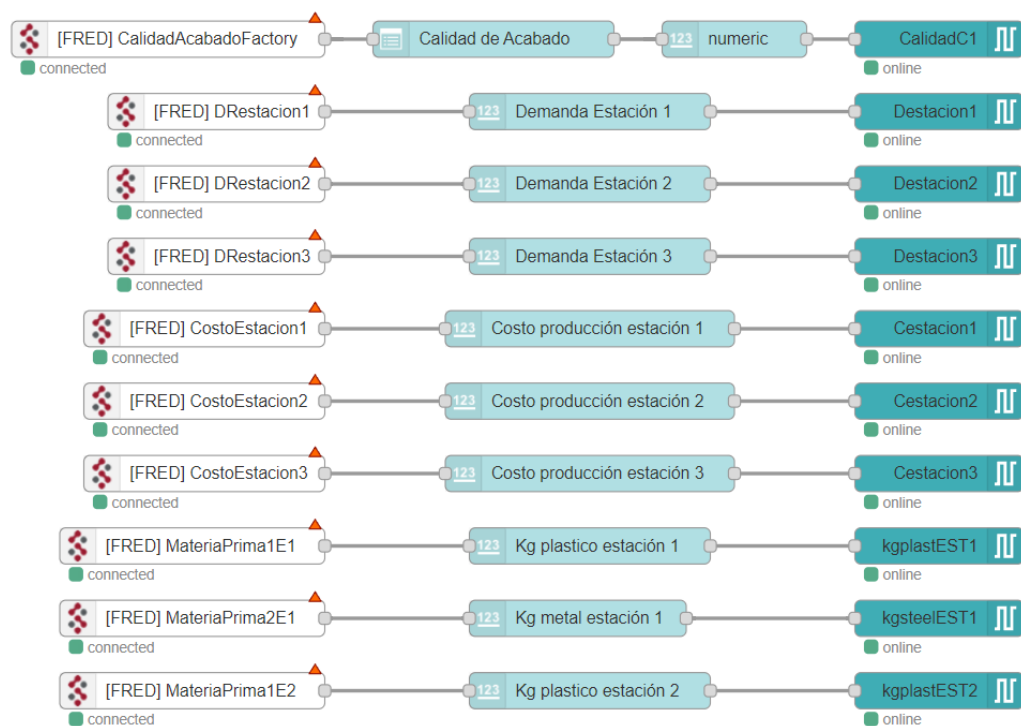
*Nodos para el control del proceso uno*



Además, se envían datos del tipo REAL que corresponden a los valores de las restricciones para el algoritmo matemático que se resuelve en el software CPLEX Optimization Studio de IBM, considerando la demanda día para cada producto, el costo de producción, la cantidad de materia prima, y los tiempos máximos de operación tal y como se muestra en la Figura 52.

Figura 52

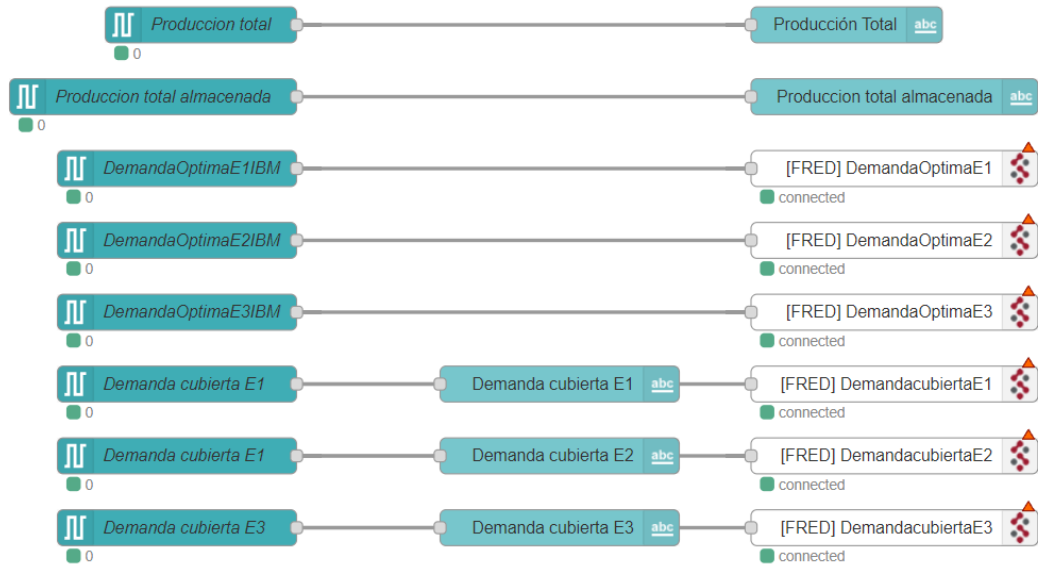
*Nodos de control de restricciones para IBM*



Como se muestra en Figura 53, los nodos configurados para el monitoreo del proceso, corresponde a las piezas mecanizadas y ensambladas por cada estación del proceso uno, además se agregan gauges para visualizar los resultados obtenidos en CPLEX Optimization Studio.

Figura 53

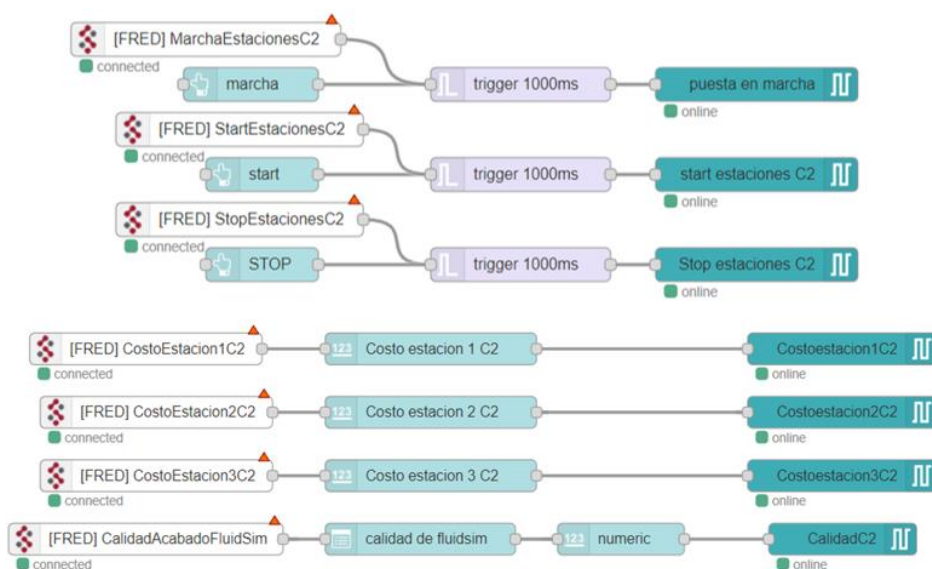
## Nodos de monitoreo de producción



Para el control y monitoreo del proceso dos implementado en FluidSIM se agrega los nodos mostrados en la Figura 54, corresponden a la activación y desactivación de todos los centros de mecanizado, al igual que el proceso uno se envían datos del tipo Real correspondientes a los costos de producción y calidad de acabado de superficie, estas variables ingresarán al algoritmo matemático desarrollado en CPLEX Optimization Studio.

**Figura 54**

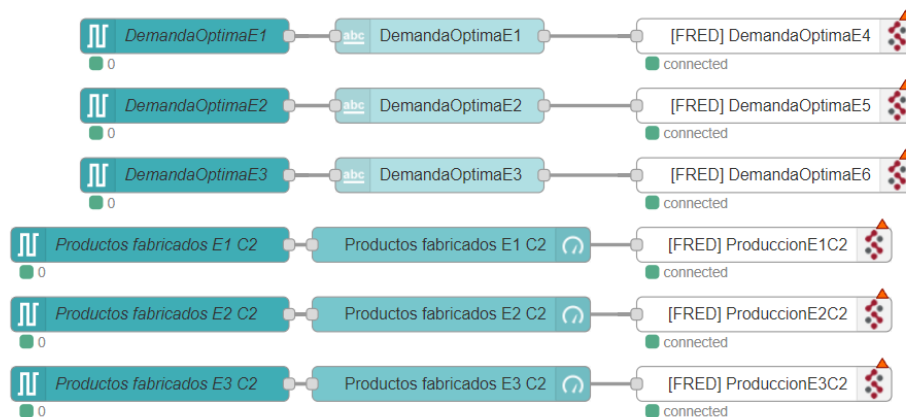
*Nodos para el control del proceso implementado en FluidSIM-P*



Como se muestra en la Figura 55, las variables a monitorear corresponden a las piezas mecanizadas en cada estación como también a los resultados del problema de optimización.

**Figura 55**

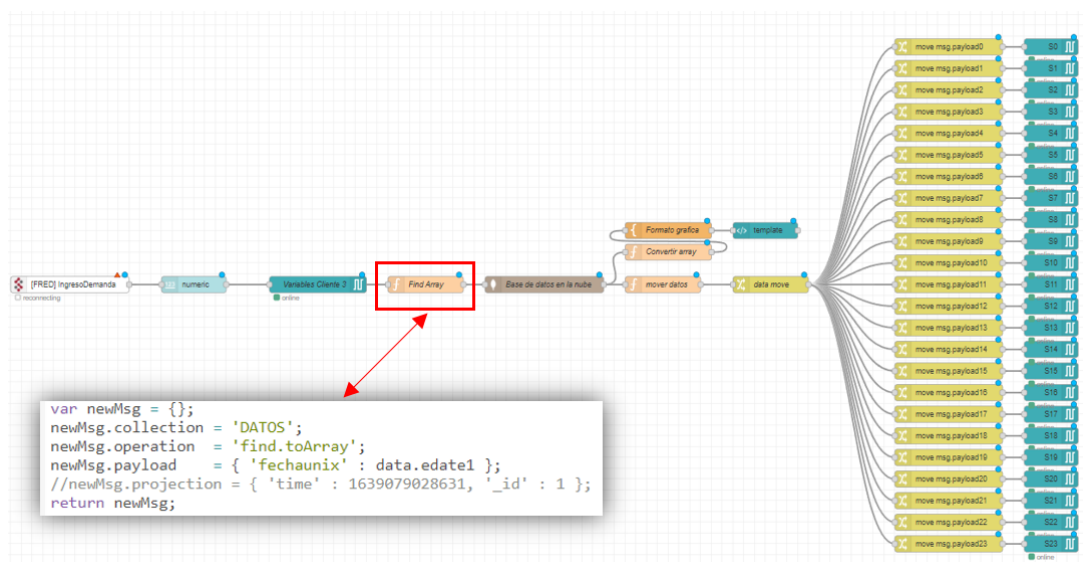
*Nodos de monitoreo de producción del proceso dos*



Como se muestra en la Figura 56, se programan nodos función para enviar la información necesaria para el algoritmo matemático desarrollado en FICO XPRESS, los datos que se envían corresponden a la demanda en kW en cada hora de cierto día. Para saber que datos se debe enviar al servidor, se agrega un nodo Fred de entrada que contiene la fecha, posteriormente ingresa a un nodo función (Find Array), que envía un nuevo mensaje con el comando de buscar array por fecha dentro de la base de datos en MongoDB.

**Figura 56**

*Nodos de envió de potencia demandada por hora*



#### 4. 6. Preprocesamiento de datos

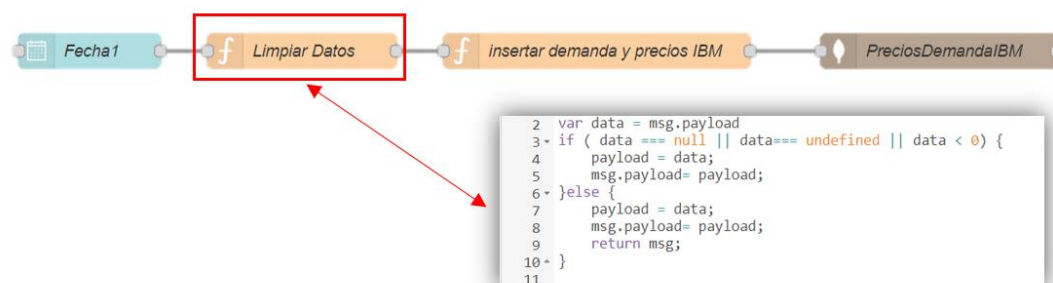
Previo a enviar los datos obtenidos de cada uno de los procesos a la base de datos, es necesario aplicar un preprocesamiento para limpiar los datos, descartando valores inconsistentes que puedan complicar la visualización e interpretación de los mismos. Para ello se agrega un nodo de función antes de ingresar al nodo de MongoDB. Como se muestra en la



Figura 57, el nodo función está programado con condiciones de restricción no envía ninguna respuesta a la salida, en caso de tener a la entrada valores nulos, negativos o indefinidos.

**Figura 57**

*Nodo para limpieza de datos*



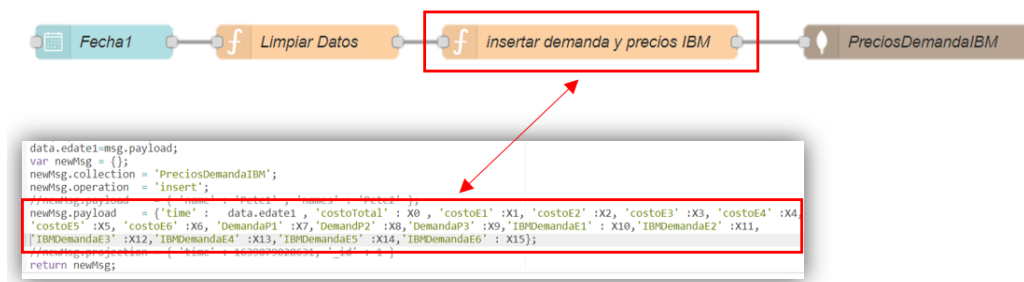
#### 4. 7. Almacenamiento de información

Los datos a almacenar corresponden a variables como consignas de operación, tiempos de ejecución, cantidad de demanda, resultados de optimización, y todos los valores de las variables de decisión que puedan aportar algún valor para el análisis e interpretación por medio de gráficas de control.

Tomando como referencia los nodos mostrados en la Figura 57, los datos para ser almacenados en una instancia de MongoDB, deben ser enviados en un msg.payload, los datos deben ser del tipo Objeto, caso contrario MongoDB no permitirá su almacenamiento. En la Figura 58 se muestra la estructura del código utilizado para insertar datos en una nueva colección de documentos.

Figura 58

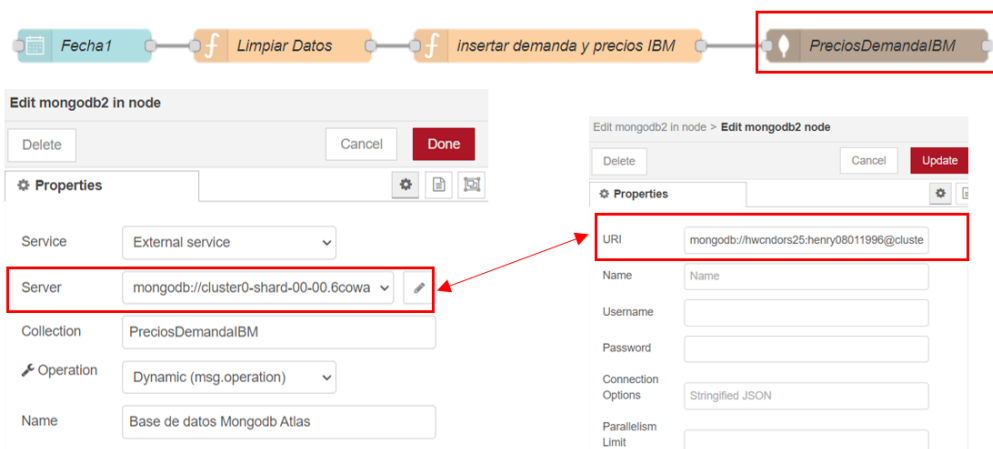
Nodo para insertar datos en una instancia Mongo



Una vez que los datos fueron limpiados y organizados, se procede a configurar el nodo MongoDB 2, el cual aloja todos los datos insertados en una base de datos NoSQL en la nube. Como se muestra en la Figura 59, es necesario definir un nuevo servidor que contendrá la URI ya sea de tipo local, Remoto o Clúster. Se asigna una colección que puede o no estar creada con anterioridad, la operación será dinámica debido a que los datos están estructuras como mensaje.

Figura 59

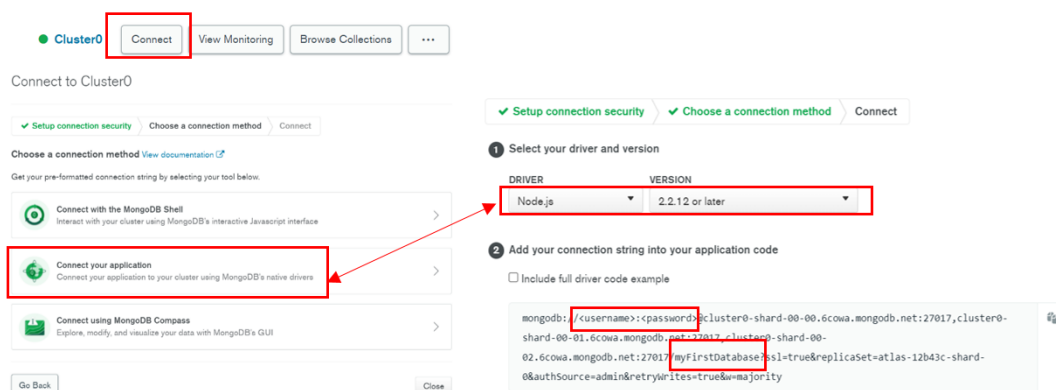
Configuración de una Instancia en MongoDB2



Para obtener la URI se debe crear una base de datos con una cuenta gratis o de paga en la siguiente dirección: <https://cloud.mongodb.com/> como se pretende conectar la base de datos con node-red y a la vez con Fred como se muestra en la Figura 60, en tipo de conexión se elige conectar con una aplicación, el driver será Node.js y la versión seleccionada será la más antigua con el código más extenso, se cambia el nombre de usuario y se define la contraseña como también el nombre de la base de datos a cual se tendrá acceso.

**Figura 60**

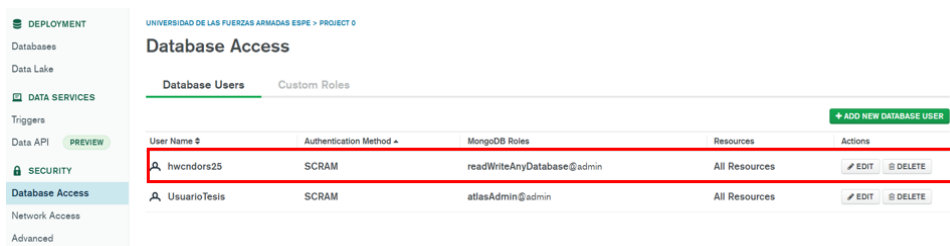
### Conexión de MongoDB Atlas con node-red



En la pestaña de seguridad y acceso a la base de datos se puede añadir a los usuarios que tendrán acceso de lectura y escritura. Además, para tener acceso libre desde cualquier dispositivo se añade la I.P. 0.0.0.0/0 aunque por seguridad se recomienda definir direcciones I.P. de confianza.

Figura 61

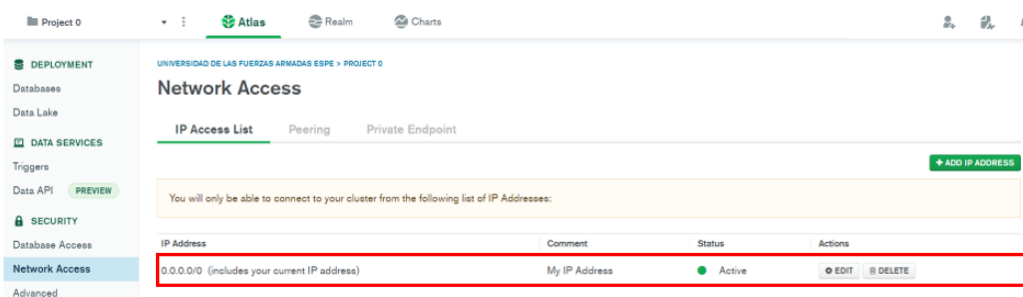
*Usuarios con acceso a la base de datos*



User Name	Authentication Method	MongoDB Roles	Resources	Actions
hwcndors25	SCRAM	readWriteAnyDatabase@admin	All Resources	<a href="#">EDIT</a> <a href="#">DELETE</a>
UsuarioTesis	SCRAM	atlasAdmin@admin	All Resources	<a href="#">EDIT</a> <a href="#">DELETE</a>

Figura 62

*Direcciones IP con acceso a la Red*



IP Address	Comment	Status	Actions
0.0.0.0/0 (includes your current IP address)	My IP Address	Active	<a href="#">EDIT</a> <a href="#">DELETE</a>

Como se muestra en la Figura 63 se creó una sola base de datos con el nombre de Tesis Universidad con siete colecciones, en la columna dos se observa la cantidad de documentos guardados hasta el momento en cada colección y en la columna tres nos indica la cantidad de espacio que están ocupando.

Figura 63

### Colecciones y documentos en MongoDB Atlas

The screenshot shows the MongoDB Atlas interface for a database named 'TesisUniversidad'. The interface includes a sidebar with a search bar and a list of namespaces. The main area displays a table of collections with the following data:

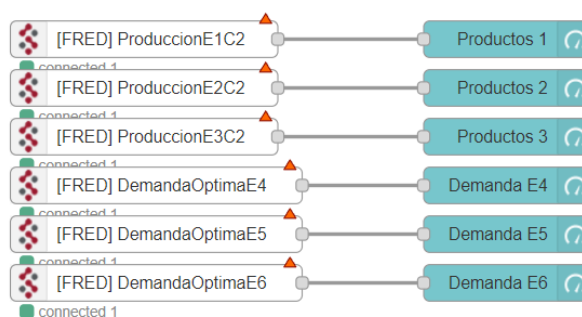
Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
CurvaDemanda	408	69.07KB	174B	1	20KB	20KB
DATOS	408	34.25KB	86B	1	20KB	20KB
GraficadeControl	1	356B	356B	1	36KB	36KB
PLC	408	124.11KB	312B	1	36KB	36KB
PotenciaGenerada	192	19.76KB	106B	1	36KB	36KB

## 4. 8. Visualización en la nube

Para el monitoreo desde la nube se utiliza medidores, gráficas y visores de texto ya establecidos en los nodos de cuadro de mando de las librerías de node-red y Fred, muy útiles cuando la variable a monitorear es continua e individual. Para visualizar dos o más variables, es recomendable utilizar los nodos template debido a que, facilita el uso de expresiones incrustadas en más de una línea utilizando JavaScript como lenguaje de programación y HTML para estructurar y desplegar los documentos.

Figura 64

### Conexión básica de los nodos Dashboard



A continuación, se presenta un resumen de la programación de los nodos de función que contienen el código para convertir los datos a valores leíbles. En la Figura 65, el nodo de función se encarga de convertir a datos estructurados a aquellos datos JSON provenientes del nodo Mongodb 2, cada variable leída se almacena temporalmente en un vector, además se agrega 12 líneas de código para convertir los datos de fecha Unix a formato de día, mes, año, hora, minutos y segundos.

**Figura 65**

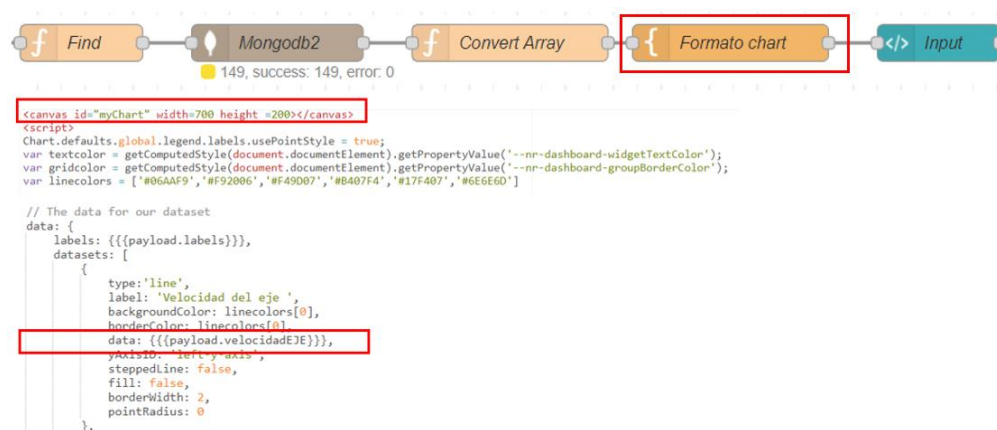
*Programación del nodo función para lectura y conversión de datos*



Una vez que los datos fueron guardados en el vector se envía a un nodo de plantilla, este nodo contiene las configuraciones como colores, tipo de línea, ubicación de datos por sección, entre otros elementos que puedan enriquecer a una mejor visualización de los datos.

Figura 66

Configuración de plantilla utilizando el nodo format template



Como se muestra en la Figura 67, se añade un nodo template para mostrar una interfaz dinámica que cambia su apariencia en función del formato de entrada, el resultado será una interfaz que combina múltiples graficas con las respectivas leyendas.

Figura 67

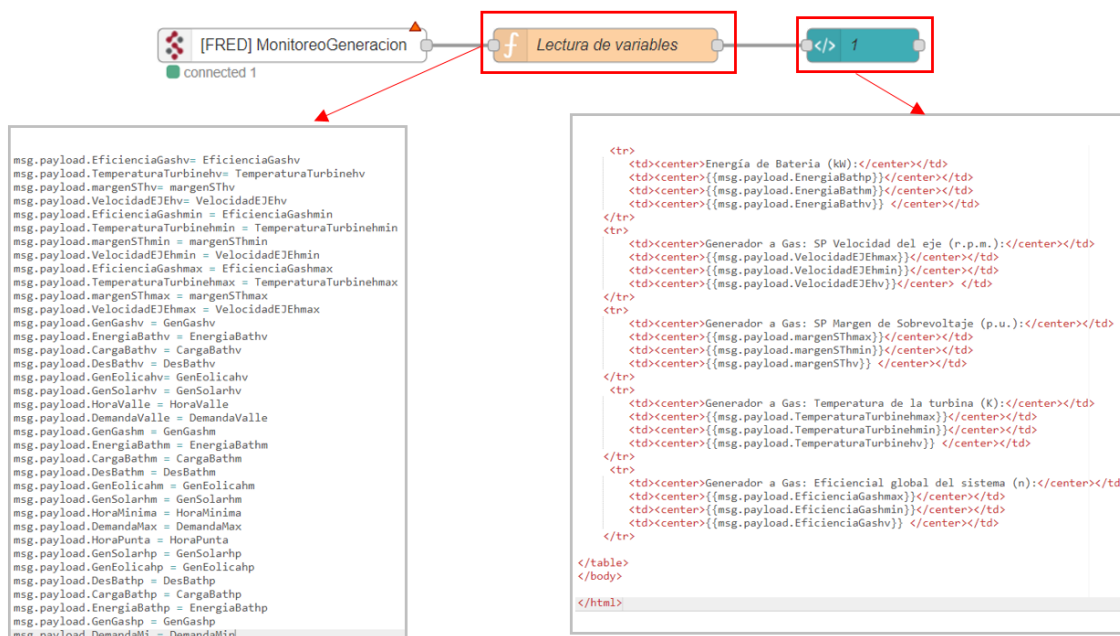
Nodo template para la visualización de gráficos



Además, en los procesos implementados se presentan variables que no necesitan ser graficadas, por este motivo se añaden plantillas de tablas. En la Figura 68, se muestra la programación de un nodo función encargado de leer y separar de manera individual los datos provenientes de node-red para posteriormente enviarlos a un nodo plantilla que muestra los resultados en tablas dinámicas.

Figura 68

Programación de los nodos para lectura de datos y visualización en tablas.



## 4. 9. Modelo de optimización

### 4.9.1. Planificación de la producción

El siguiente modelo de optimización fue planteado con la finalidad de descentralizar la producción entre las seis estaciones de trabajo, distribuidos entre los procesos construidos en Factory IO y FluidSIM. Para ello se considera los costos de fabricación que incluyen mano de obra, adquisición de materia prima, uso de energía y costos de mantenimiento.

$$\min C(U_E) = \sum_{i=1}^n C_E(U_{Ei})$$

Ecuación 9

Costo total de un sistema con n estaciones de producción

$$\min C(U_E) = \sum_{t=1}^3 [C_{EF}(t) * U_{EF}(t) + C_{EFS}(t) * U_{EFS}(t)]$$



Donde:

$C_{EF}$ : Costos de producción en Factory IO

$C_{EFS}$ : Costos de producción en FluidSIM

$U_{EF}$ : Unidades a fabricar en Factory IO

$U_{EFS}$ : Unidades a fabricar en FluidSIM

Sujeto a:

$$s. a. \sum_{i=1}^n U_{Ei} = U_D^{Total} + U_{def}$$

**Ecuación 10**

*Suma de los niveles de producción*

$$U_{Ei}^{min} < U_{Ei} \leq U_{Ei}^{max} \quad i \in \{1, \dots, n\}$$

La primera restricción muestra que los productos fabricados deben consumir menos o igual cantidad que la disponibilidad total de los recursos en la fábrica.

$$\sum_{t=1}^3 [A_{EF}(t) * U_{EF}(t)] \leq B_{EF}(t)$$

Donde:

$A_{EF}$ : Cantidad de material a consumir por pieza en Factory IO

$B_{EF}$ : Disponibilidad total de materia prima en Factory IO

La segunda restricción indica que el tiempo total de fabricación no debe ser mayor a los tiempos de operación diario en la fábrica.

$$\sum_{t=1}^3 [T_{EF}(t) * U_{EF}(t) + T_{EFS}(t) * U_{EFS}(t)] \leq T_{diario}(t)$$

Donde:

$T_{EF}$ : Tiempo de fabricación por pieza en Factory IO

$T_{EFS}(t)$ : Tiempo de fabricación por pieza en FluidSIM

$T_{diario}$ : Tiempo maximo de operación día

La siguiente restricción indica que la cantidad de productos fabricados en ambos procesos (Factory IO y FluidSIM) debe ser igual o mayor a la demanda diaria requerida.

$$\sum_{t=1}^3 [U_{EF}(t) + U_{EFS}(t)] \geq D_{total}$$

Donde:

$D_{total}$ : Demanda diaria requerida

Suponiendo que la fábrica construida en Factory IO posee mejores equipos de mecanizado y para evitar paros innecesarios de maquinaria, para aprovechar a lo máximo su disponibilidad, se agrega una restricción de cantidad mínima a producir de la demanda total.

$$\sum_{t=1}^3 [U_{EF}(t)] \geq D_{min}$$

Donde:

$D_{min}$ : Producción mínima en Factory IO

La última restricción corresponde a la no negatividad en ambas fabricas (Factory IO y FluidSIM).

$$U_{EF}, U_{EFS} \geq 0$$

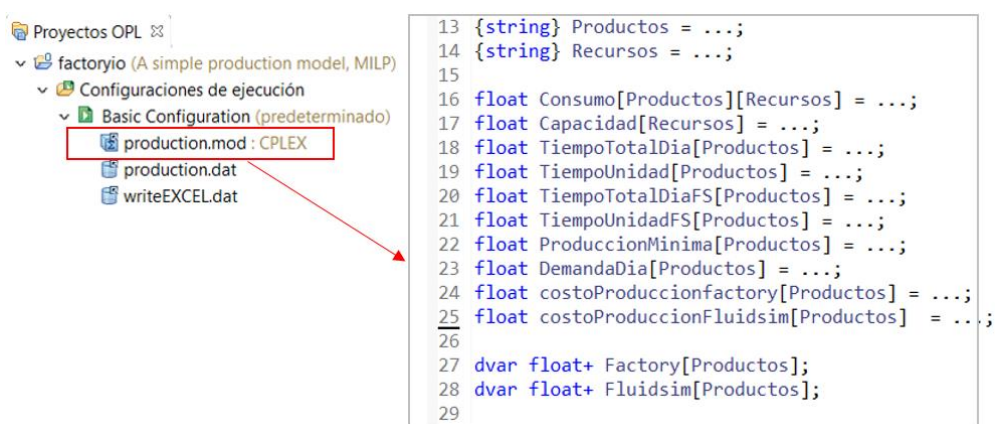
**Ecuación 11**

*No negatividad*

Para resolver el problema de optimización se utiliza el software CPLEX Optimization Studio aplicando el método Simplex utilizando lenguaje de programación OPL. Como se muestra en la Figura 69 se agrega un nuevo modelo en donde se desarrolla la función objetivo y las restricciones, una segunda y tercera ventana leerá los datos agregados en CPLEX y Excel.

**Figura 69**

*Declaración de variables en CPLEX Optimization Studio*



```

13 {string} Productos = ...;
14 {string} Recursos = ...;
15
16 float Consumo[Productos][Recursos] = ...;
17 float Capacidad[Recursos] = ...;
18 float TiempoTotalDia[Productos] = ...;
19 float TiempoUnidad[Productos] = ...;
20 float TiempoTotalDiaFS[Productos] = ...;
21 float TiempoUnidadFS[Productos] = ...;
22 float ProduccionMinima[Productos] = ...;
23 float DemandaDia[Productos] = ...;
24 float costoProduccionfactory[Productos] = ...;
25 float costoProduccionFluidsim[Productos] = ...;
26
27 dvar float+ Factory[Productos];
28 dvar float+ Fluidsim[Productos];
29

```

Figura 70

*Función objetivo y restricciones en CPLEX Optimization Studio*

The screenshot shows the CPLEX Optimization Studio interface. On the left, the project tree is expanded to show the 'production.mod: CPLEX' file. On the right, the code editor displays the following code:

```

36 minimize
37   sum( p in Productos )
38   ( costoProduccionfactory[p] * Factory[p] + costoProduccionFluidsim[p] * Fluidsim[p] );

42 subject to {
43   // restricción de consumo
44   forall( r in Recursos )
45     ctCapacidad:
46       sum( p in Productos )
47         Consumo[p][r] * Factory[p] <= Capacidad[r];
48
49   // restricción de tiempo de fabricación
50   forall( p in Productos )
51     ctTiempodeProduccion:
52       TiempoUnidadFS[p]*Fluidsim[p] + TiempoUnidad[p]*Factory[p] <= TiempoTotalDiaFS[p];
53
54   forall( p in Productos )
55     // restricción de demanda requerida
56     ctDemandaDia:
57       Factory[p] + Fluidsim[p] >= DemandaDia[p];
58
59   // restricción de producción mínima
60   forall( p in Productos ) {
61     ctDemandaMinima:
62       Factory[p] >= ProduccionMinima[p];
63
64   }

```

Figura 71

*Lectura y escritura en Excel utilizando CPLEX Optimization Studio*

The screenshot shows the CPLEX Optimization Studio interface. On the left, the project tree is expanded to show the 'writeEXCEL.dat' file. On the right, the code editor displays the following code:

```

7 SheetConnection excelSheet("DATOS.xlsx");
8 // Escribir Resultados
9 Factory to SheetWrite(excelSheet,"Hoja2!I4:I7");
10 Fluidsim to SheetWrite(excelSheet,"Hoja2!J4:J7");
11 // Lectura de variables
12 DemandaDia from SheetRead(excelSheet,"Hoja2!D4:F4");
13 costoProduccionfactory from SheetRead(excelSheet,"Hoja2!D6:F6");
14 costoProduccionFluidsim from SheetRead(excelSheet,"Hoja2!D5:F5");
15 Capacidad from SheetRead(excelSheet,"Hoja2!I8:J8");
16
17 TiempoUnidad from SheetRead(excelSheet,"Hoja2!D14:F14");
18 TiempoTotalDia from SheetRead(excelSheet,"Hoja2!D15:F15");
19 TiempoUnidadFS from SheetRead(excelSheet,"Hoja2!D16:F16");
20 TiempoTotalDiaFS from SheetRead(excelSheet,"Hoja2!D17:F17");
21 ProduccionMinima from SheetRead(excelSheet,"Hoja2!D18:F18");

```

Figura 72

*Declaración de variables en un bloque de datos*

The screenshot shows the CPLEX Optimization Studio interface. On the left, the project tree is expanded to show the 'production.dat' file. On the right, the code editor displays the following code:

```

12 Productos = { "TAPASverdes", "TAPASazules", "TAPASmetalicas" };
13 Recursos = { "Metal", "Plastico" };

```

#### 4.9.2. Despacho económico

Para el despacho económico se utiliza programación de variables enteras para minimizar el costo total de operación sujeto al balance de potencia y a los límites de producción de cada generador, considerando en las restricciones costos de operación, potencias mínimas y máximas de cada unidad, disponibilidad de recursos, en donde la suma de los niveles de generación debe cumplir:

Función objetivo.

$$\min C(P_G) = \sum_{i=1}^n C_i(P_{Gi})$$

**Ecuación 12**

*Costo total de un sistema con n generadores*

*Nota. (Ortíz, 2019).*

$$\min C(P_G) = \sum_{t=1}^{24} (C_{GD} * P_{GD}(t) + C_{GG} * P_{GG}(t) + C_{NS} * P_{NS}(t) + 0.8 * P_{VS}(t) + 0.8 * P_{VE}(t))$$

Sujeto a:

$$s. a. \sum_{i=1}^n P_{Gi} = P_D^{Total} + P_{perd}$$

**Ecuación 13**

*Suma de los niveles de generación*

*Nota. (Ortíz, 2019).*

$$P_{Gi}^{min} < P_{Gi} < P_{Gi}^{max} \quad i \in \{1, \dots, n\}$$

Donde:

$P_D^{Total}$  = Potencia total demandada.

$P_{perd}$  = Perdida de potencia.

$P_{Gi}$ =Potencia de generación.

$P_{Gi}^{min}$ =Potencia de generación mínima.

$P_{Gi}^{max}$  =Potencia de generación máxima.

$$\sum_{t=1}^{24} = (P_{GD}(t) + P_{GG}(t) + P_{GS}(t) + P_{GE}(t) + P_{BD}(t))$$

$$= D(t) + P_{VE}(t) + P_{VS}(t) - P_{NS}(t) - P_{BC}(t)$$

#### Ecuación 14

*Balance energético*

$$P_{GD}^{min} \geq P_{GD}(t) \leq P_{GD}^{max}$$

$$P_{GG}^{min} \geq P_{GG}(t) \leq P_{GG}^{max}$$

$$P_{VE}(t) \leq P_{GE}(t)$$

$$P_{VS}(t) \leq P_{GS}(t)$$

$$P_{BC}(t) \geq -P_{BC}^{inst}$$

$$P_{BC}(t) \leq P_{BC}^{inst}$$

Donde:

$P_{GD}(t)$ = Potencia de generación a Diésel.

$P_{GG}(t)$ = Potencia de generación a Gas.

$P_{GS}(t)$ = Potencia de generación Solar.

$P_{GE}(t)$ = Potencia de generación Eólica.

$P_{NS}(t)$ =Potencia no suministrada.

$P_{VS}(t)$ =Potencia de vertimiento Solar.

$P_{VE}(t)$ =Potencia de vertimiento Eólico.

$P_{BC}(t)$ =Potencia de carga de Baterías.

$P_{BD}(t)$ = Potencia de descarga de Baterías.

Para resolver el problema planteado se utilizó FICO XPRESS, que al igual que CPLEX Optimization Studio permite resolver problemas de optimización complejos que involucran varias variables utilizando programación lineal de enteros.

Como se muestra en la Figura 73 primero se debe declarar las variables de decisión del tipo lineal y entera, además se declara los parámetros necesarios para calcular la potencia de las unidades de generación eólica y solar. En la Figura 74 se muestra la programación para leer el valor que le corresponde a cada parámetro desde Excel.

**Figura 73**

*Declaración de parámetros de los generadores y variables de decisión en FICO XPRESS*

```

declarations
P_GD, P_GG, X_GD, X_GG, P_VS, P_VE, P_NS, P_BD, P_BC, X_BD, X_BC, E, SOC : array(1..24) of mpvar
objective:linctr
balance:array(1..24) of linctr
!parametros
D:array(1..24)of real
I:array(1..24)of real
v:array(1..24)of real
Cp:array(1..24)of real

Costo: real
Cost_GD=0.15 ![usd/kwh] Costos de Generación Diesel
Cost_GG=0.06 ![usd/kwh] Costos de Generación Gas
!Datos para Generación Solar [100 kW]
A_total=684.4662 !m^2
A_panel=0.99198 !m^2 (1485 * 668)mm^2
n_panel=0.1592 ! rendimiento del panel
n_inst=0.85 ! rendimiento de la instalación
P_GS_nom=100 ![kW]
P_GS_panel=0.15 ![kW]
!Datos para Generación Eólica [20 kW]
P_GE_nom=20 !kW 20
P_GE_turbina=3 !Kw
den=1.2754 ![kg/m^3]; densidad a nivel mar
A_barrido=11.34 !m^2
Cost_NS= 0.8 !Costo de energía no suministrado
!Baterías
Eo= 70 ! Puede cambiar, es un punto de operación
n_BC=0.9 ! Rendimiento de La Batería en Carga
n_BD=0.9 ! Rendimiento de La Batería en Descarga
P_B_inst=70 !Potencia máxima de La Bat. Instalada
!P_GD_max=100 !kW, P_GG_max=80 !kW; Generador Diesel y Gas
P_GD_max:real
P_GG_max:real
P_GD_min:real
P_GG_min:real
end-declarations

```

**Figura 74**

*Lectura de parámetros desde Excel hacia Fico Xpress*

```

initializations from 'mmsheet.xlsx':+Data
U as "skip;noindex;D"
Ir as "skip;noindex;Ir"
v as "skip;noindex;v"
Cp as "skip;noindex;Cp"
P_GD_max as '[Hoja1$B29]'
P_GG_max as '[Hoja1$C29]'
P_GD_min as '[Hoja1$D29]'
P_GG_min as '[Hoja1$E29]'
end-initializations

```

En la Figura 75 se declara la función objetivo que involucra a las variables de decisión correspondiente a la producción de potencia de cada unidad de generación por cada hora, además se añade la restricción de no negatividad para valores menores a cero.

**Figura 75**

*Función objetivo y restricción de no negatividad en Fico Xpress*

```

!Funcion Objetivo
objective:= sum(t in 1..24)(Cost_GD*P_GD(t))+sum(t in 1..24)(Cost_GG*P_GG(t))+sum(t in 1..24)(Cost_NS*P_NS(t))+ sum(t in 1..24)(0.8*P_VS(t))+sum(t in 1..24)(0.8*P_VE(t))

forall (t in 1..24) do
P_BC(t) is_free !Cuando va a tener valores negativos
if t=1 then
E(1)=Eo-n_BC*P_BC(1)-(P_BD(1)/n_BD)
E(1)<=P_B_inst
SOC(1)=E(1)/P_B_inst
else
E(t)=E(t-1)-(n_BC*P_BC(t))-(P_BD(t)/n_BD)
E(t)<=P_B_inst
SOC(t)=E(t)/P_B_inst
end-if
end-do

```

En la Figura 76 se declara las restricciones a las que estará sujeta las variables de decisión, estas restricciones incluyen a la potencia mínima que debe generar cada unidad, así también a la potencia máxima que puede producir cada unidad de generación.



Figura 76

## Restricciones de potencia mínima y máxima

```

!Sujeto a:
forall (t in 1..24) do
  P_GS(t) := n_panel*n_inst*A_panel*(P_GS_nom/P_GS_panel)*Ir(t) !Modelo Generador Solar
  P_GE(t) := (0.5*(den*A_barrido*v(t)^3*(P_GE_nom/P_GE_turbina))*Cp(t))/1000 !Modelo Generador Eólico

  balance(t) := P_GD(t)+P_GG(t)+P_GS(t)+P_GE(t)+P_BD(t)-D(t)+P_VE(t)+P_VS(t)-P_NS(t)-P_BC(t)

  X_GD(t) is_binary
  X_GG(t) is_binary
  P_GD(t) <= P_GD_max*(X_GD(t))
  P_GG(t) <= P_GG_max*(X_GG(t))
  P_GD(t) >= P_GD_min*(X_GD(t))
  P_GG(t) >= P_GG_min*(X_GG(t))
  P_VE(t) <= P_GE(t)
  P_VS(t) <= P_VS(t)
  X_BC(t) is_binary
  X_BD(t) is_binary
  X_BC(t)+X_BD(t) <= 1
  0 <= P_BC(t)
  0 <= P_BD(t)
  P_BC(t) >= -(P_B_inst)*(X_BC(t))
  P_BD(t) <= (P_B_inst)*(X_BD(t))
  SOC(t) >= 0.1
  SOC(t) <= 0.9
end-do

```

Para dar solución a la función objetivo se optimiza cada variable de decisión involucrada procurando cumplir las restricciones anteriormente planteadas. En la Figura 77 se muestra las líneas de código para la optimización.

Figura 77

## Optimización del problema lineal de variables enteras

```

minimise(XPRS_BAR, objective)
forall (t in 1..24) do
  P_GD_sol(t) := getsol(P_GD(t))
  P_GG_sol(t) := getsol(P_GG(t))
  X_GD_sol(t) := getsol(X_GD(t))
  X_GG_sol(t) := getsol(X_GG(t))
  P_VS_sol(t) := getsol(P_VS(t))
  P_VE_sol(t) := getsol(P_VE(t))
  P_NS_sol(t) := getsol(P_NS(t))
  P_BC_sol(t) := getsol(P_BC(t))
  P_BD_sol(t) := getsol(P_BD(t))
  E_sol(t) := getsol(E(t))
  X_BC_sol(t) := getsol(X_BC(t))
  X_BD_sol(t) := getsol(X_BD(t))
  SOC_sol(t) := getsol(SOC(t))
end-do

```

Para saber que valores son los más óptimos se debe imprimir los resultados, para ello en la Figura 78 se muestra la estructura y la programación para escribir los resultados en la consola de Fico Xpress.

Figura 78

Resultados de la optimización escritura en consola

```
writeln("Costo total: ", getobjval, "USD")
writeln("Potencia Generada_Diesel: ", P_GD_sol, "kW")
writeln("Potencia Generada_Gas: ", P_GG_sol, "kW")
writeln("Potencia Generada_Solar: ", P_GS, "kW")
writeln("Potencia Generada_Eolica: ", P_GE, "kW")
writeln("Potencia Vertimiento_Solar: ", P_VS_sol, "kW")
writeln("Potencia Vertimiento_Eolico: ", P_VE_sol, "kW")
writeln("Potencia No_Suministrada: ", P_NS_sol, "kW")
writeln("Potencia de carga de la Bateria_Consumo: ", P_BC_sol, "kW")
writeln("Potencia de descarga de la Bateria_Generador: ", P_BD_sol, "kW")
writeln("Energía de la Bateria (Generador): ", E_sol, "kW")
writeln("modo Carga: ", X_BC_sol, "")
writeln("modo Descarga: ", X_BD_sol, "")
writeln("SOC: ", SOC_sol, "")
writeln("EN/AP Generador Diesel: ", X_GD_sol, "kW")
writeln("EN/AP Generador GAS: ", X_GG_sol, "kW")
Costo:= getobjval
```

Valor

String

Además, en la Figura 79 se muestra la programación para escribir los mismos resultados, pero en una hoja de Excel, se debe tener en cuenta que los nombres de las variables deben ser las mismas caso contrario no se mostrará los resultados para verificar este último punto se puede ingresar al administrador de nombres y comprobar que los cuadros de nombre en Excel sean los mismos de Fico.

Figura 79

Resultados de la optimización escritura en Excel

```
initializations to "mmsheet.xlsx":"+Data
P_GD_sol as 'skip;grows;P_GD_sol'
P_GG_sol as 'skip;grows;P_GG_sol'
X_GG_sol as 'skip;grows;X_GG_sol'
X_GD_sol as 'skip;grows;X_GD_sol'
P_GS as 'skip;grows;P_GS'
P_GE as 'skip;grows;P_GE'
P_VS_sol as 'skip;grows;P_VS_sol'
P_VE_sol as 'skip;grows;P_VE_sol'
P_NS_sol as 'skip;grows;P_NS_sol'
P_BC_sol as 'skip;grows;P_BC_sol'
P_BD_sol as 'skip;grows;P_BD_sol'
E_sol as 'skip;grows;E_sol'
Costo as "skip;grows;noindex;Total_Cost"
SOC_sol as 'skip;grows;SOC_sol'
end-initializations
end-model
```

	V	Im	Cp
Cp	0	12.27	0.26
D	0	9.99	0.26
E_sol	0	12.37	0.26
Ir	0	12.93	0.26
P_BC_sol	0	12.26	0.26
P_BD_sol	0	11.83	0.27
P_Diesel_KW	0	11.71	0.27
P_GD_sol	0	11.52	0.28
P_GE	0	11.52	0.28
P_GG_sol	0	11.61	0.28
P_GS	0	11.90	0.27
P_NS_sol	0	12.28	0.26
P_VE_sol	0	12.08	0.26
P_VS_sol	0	11.23	0.28
Q	0	10.57	0.32
SOC_sol	0	10.20	0.34
Total_Cost	0	10.20	0.34
v	0	10.24	0.33
X_GD_sol	0	10.57	0.32
X_GG_sol	0	10.76	0.31
	0	11.94	0.30
	0	11.42	0.28
	0	11.83	0.27

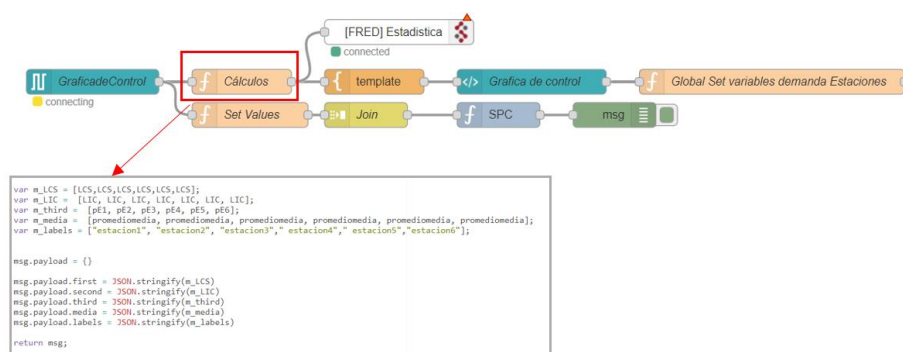
#### 4. 10. Gráficas de control

Para vigilar los procesos de producción se utiliza las gráficas de control, estas gráficas determinarán que procesos se han salido de control y cuales siguen manteniéndose entre la media. En el proyecto de investigación se utiliza este tipo de gráfica para validar los resultados obtenidos del problema de optimización mostrados en el apartado 4. 9. se utiliza dos herramientas la primera consiste en programar un nodo función que contenga los cálculos de los límites superior, inferior y central considerando los tiempos de producción total de cada estación (3 estaciones construidas en FluidSIM y 3 en Factory IO), tiempos de producción en ciclo único, actividades realizadas y tiempos de inactividad.

En la Figura 80 se muestra la programación que realiza los cálculos de los límites de control, además para permitir una correcta visualización e interpretación los valores calculados se guardan temporalmente en un vector para finalmente ser enviados y leídos por los nodos template encargados de trazar la gráfica de medias.

**Figura 80**

*Cálculo de los límites de control utilizando un nodo de función*

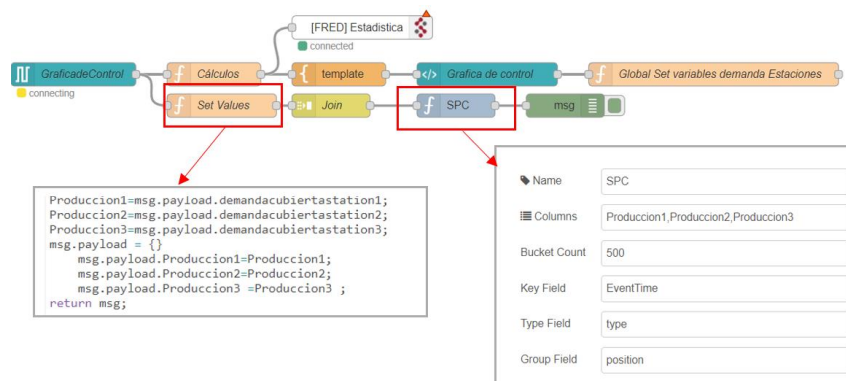


Para aprovechar al máximo las herramientas que ofrece node-red, para el cálculo de la varianza se utiliza el nodo de control estadístico de procesos (SPC) propio de node red, este tipo

de nodo está basado en la gráfica media-varianza, a diferencia de los cálculos manuales mostrados en la Figura 80 el nodo SPC realiza los cálculos de manera interna, requiriendo solamente como entrada los valores de las variables en formato JSON.

**Figura 81**

*Configuración del nodo SPC en node-red*

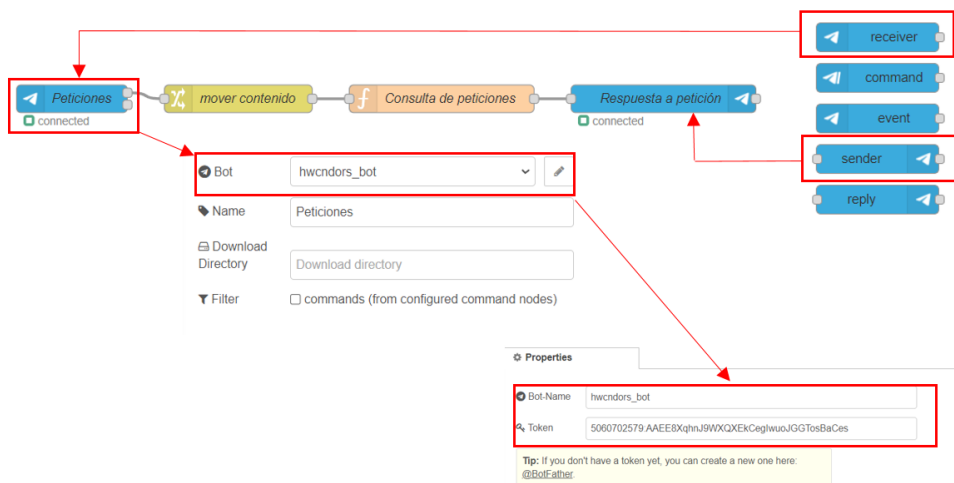


#### 4.11. Aplicación del Telegram Bot

Además de realizar un monitoreo desde la nube, HMI y SCADA se configura los nodos de un Bot creado en Bot Father a través de telegram para realizar consultas en “tiempo real” sobre el estado actual de los procesos. Como se muestra en Figura 82, se utilizan los nodos de enviar y recibir, se descarta el nodo de comandos, se añade un nuevo Bot que debe estar previamente creado y con el Token que permitirá la comunicación entre telegram y node-red.

Figura 82

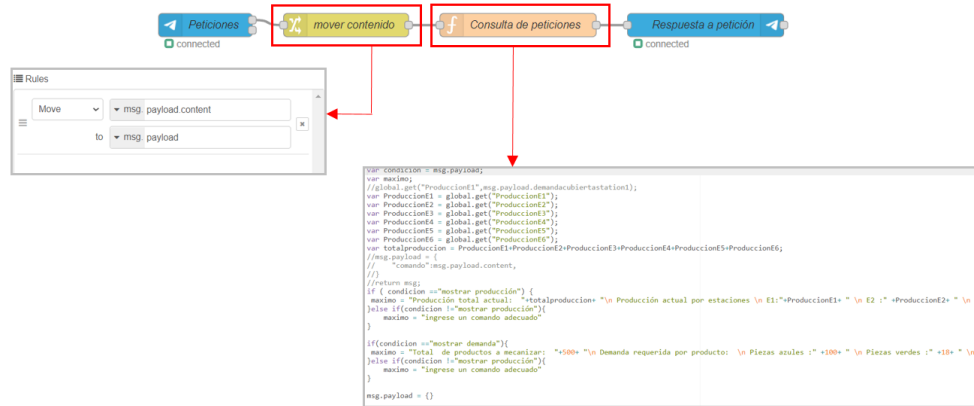
## Configuración de los nodos del Telegram Bot



Las peticiones solicitadas desde telegram llegan en un `msg.payload.content` por comodidad se utiliza un nodo de reglas para mover las peticiones a un `msg.payload` después la petición ingresa a un nodo de función programado con condiciones simples. Como se muestra en la Figura 83 si la petición de entrada es igual a la condición se envía una respuesta que contendrá, por ejemplo, la producción actual de los centros de mecanizado, o la demanda solicitada, los costos actuales de producción, los resultados de la optimización, entre otros.

Figura 83

Lectura de las peticiones del Telegram Bot utilizando un nodo de función.



## Capítulo V

### 5. Análisis de Resultados

Dentro de este capítulo se presenta los resultados de la gestión que se realiza al sistema de control distribuido basado en el estándar TCP/IP para procesos industriales virtuales utilizando una topología tipo bus con un solo canal de comunicación, poniendo en marcha una planificación de producción de fábrica y producción energética, además se muestra los resultados de las estrategias utilizadas para mantener una gestión de datos.

#### 5. 1. Red de comunicación industrial

Para validar la comunicación cliente-servidor y servidor-cliente en el sistema de control distribuido es necesario realizar un intercambio de información ya sea de manera local o remota, se utiliza un único servidor y 3 clientes enlazando así a los controladores con el sistema SCADA, en la Figura 84 el intercambio de información se lo realiza utilizando el bus de campo Modbus TCP, al ser de estándar abierto su arquitectura es más compleja y su configuración requiere más tiempo, además se presenta una latencia de 500 ms aproximadamente para enviar y recibir información.

Figura 84

*Intercambio de información entre servidor-cliente 2*

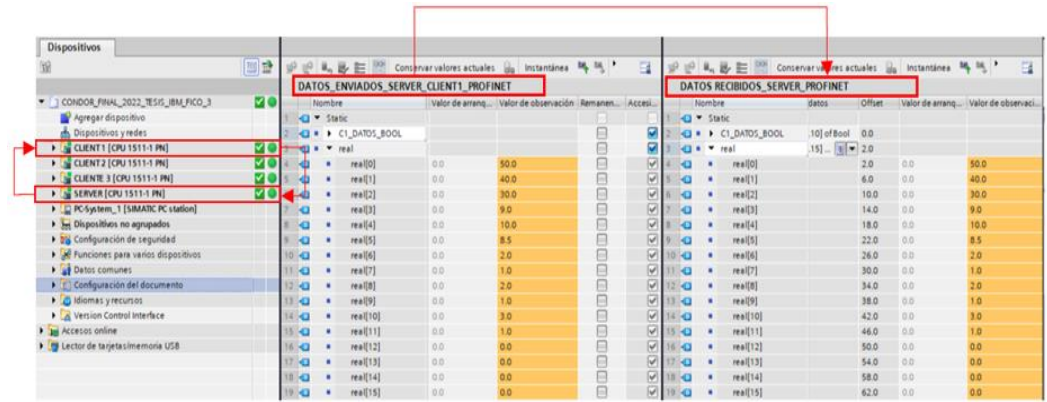
Nombre	Tipo de datos	Offset	Valor de arran...	Valor de ...
Static	Array[0..15]	0.0	0.0	0.0
data[0]	Real	0.0	0.0	45.0
data[1]	Real	4.0	0.0	35.0
data[2]	Real	8.0	0.0	25.0
data[3]	Real	12.0	0.0	45.0
data[4]	Real	16.0	0.0	35.0
data[5]	Real	20.0	0.0	25.0
data[6]	Real	24.0	0.0	0.0
data[7]	Real	28.0	0.0	105.0
data[8]	Real	32.0	0.0	0.0
data[9]	Real	36.0	0.0	0.0
data[10]	Real	40.0	0.0	0.0
data[11]	Real	44.0	0.0	0.0
data[12]	Real	48.0	0.0	0.0
data[13]	Real	52.0	0.0	0.0
data[14]	Real	56.0	0.0	0.0
data[15]	Real	60.0	0.0	0.0

Con respecto al proceso uno y proceso tres el intercambio de información, se realiza utilizando el bus de campo Profibus y en el enlace s7. En la Figura 85 se muestra que el envío y recepción de datos se realiza correctamente, a diferencia de Modbus TCP, a) Profibus y b) Comunicación S7 presentan una arquitectura fácil de implementar además no existe latencia en el intercambio de información.

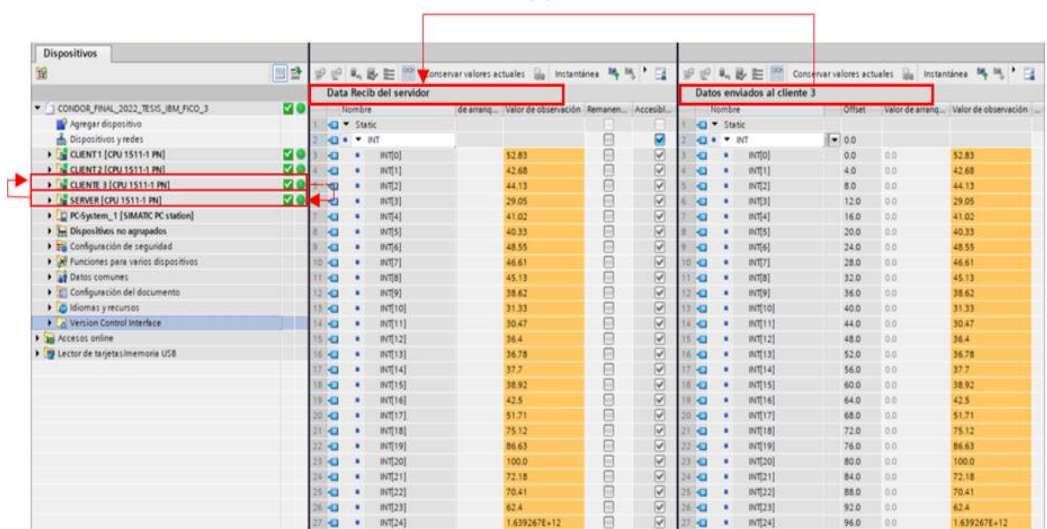


Figura 85

Intercambio de información entre servidor-cliente 1 cliente 3



(a)



(b)

## 5. 2. Resultados de los nodos programados en node-red

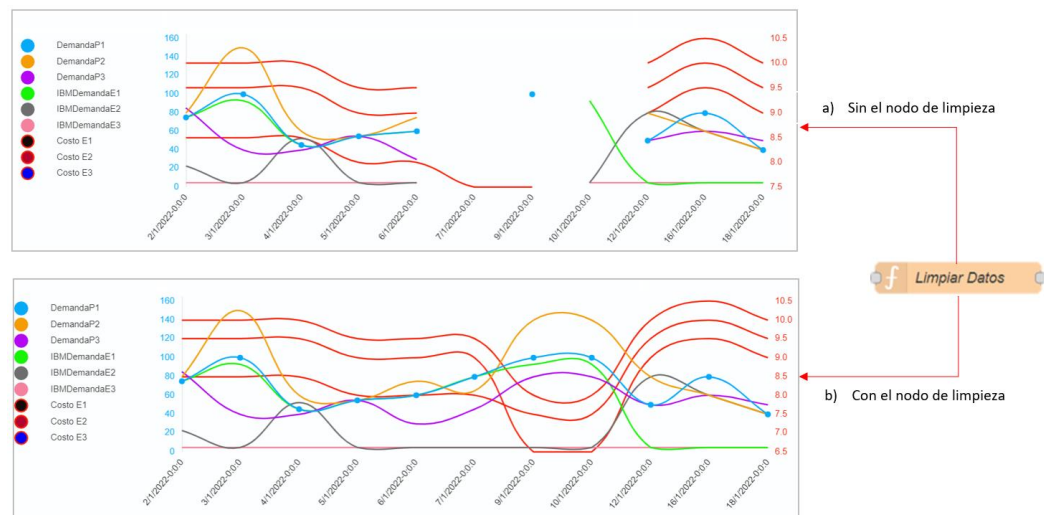
### 5.2.1. Nodos de Limpieza de Datos

Los resultados de la programación del apartado 4. 8 se muestra en la Figura 86, un adecuado uso de las técnicas de limpieza y secuencias de programación ahorra espacios de disco y sobre todo ahorra tiempo en caso de querer eliminar los datos erróneos, nulos o incompletos que hayan sido almacenados sin un tratamiento previo. En a) se han guardado los datos de

producción de los procesos uno y dos sin utilizar un nodo de limpieza, en b) se han vuelto a repetir las mismas operaciones en cada proceso, pero esta vez se almacenan los datos utilizando un nodo de limpieza como se mencionó anteriormente este paso no solo ahorra espacio en el disco, además permite visualizar los datos de manera rápida sin tener que realizar una eliminación manual antes de volver a graficar.

**Figura 86**

*Diferencia en la visualización de datos almacenados utilizando el nodo de limpieza*

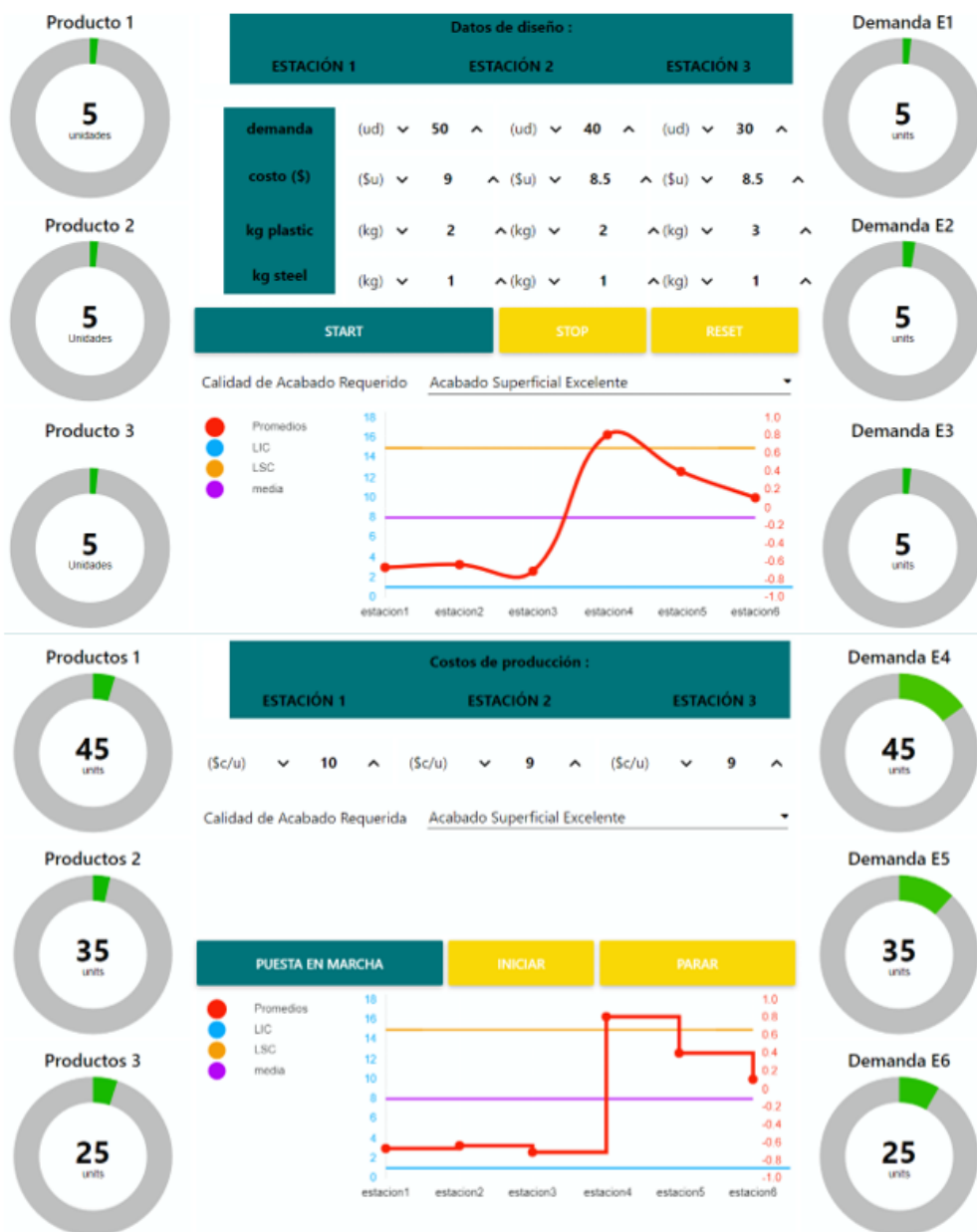


### 5.2.2. Nodos de Visualización y Monitoreo

En la Figura 87 se muestra los resultados obtenidos del algoritmo matemático estableciendo que cada estación de trabajo no puede o no debería fabricar más productos de los recomendados.

Figura 87

Control y monitoreo de los procesos uno y dos desde la nube



Con respecto al proceso tres y como se muestra en la Figura 88 los resultados obtenidos del unit commitment y las operaciones del apartado 4. 8 se visualizan por medio de tablas facilitando en, a) el monitoreo de las variables más relevantes en “tiempo real” para en, b)

almacenar para posteriormente recuperar y visualizar por medio de tablas los valores de las variables.

**Figura 88**

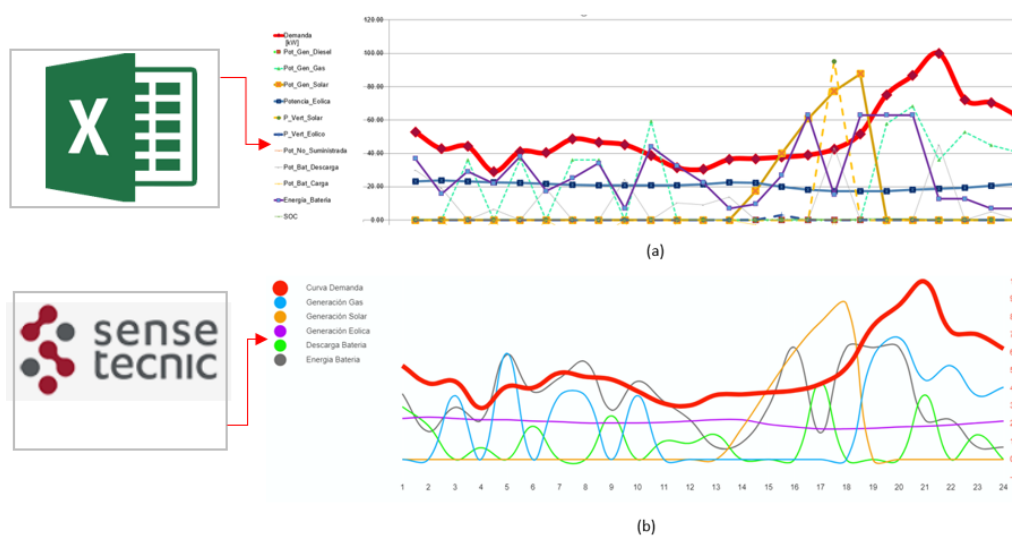
*Monitoreo del proceso tres desde la nube y almacenamiento en MongoDB*



Tomando en consideración que, para una interpretación de datos más eficiente y rápida, la utilización de tablas dificulta esta tarea, en la Figura 89 en a) se traza las curvas de los resultados del Unit Commitment en la hoja de Excel que contiene los resultados de Fico Xpress, en b) se traslada estas curvas a la nube procurando mantener las mismas propiedades de origen. Los resultados en a y b muestran ser muy semejantes cualitativamente y cuantitativa, de esta manera se reduce la posibilidad de obtener interpretaciones negativas basados en la observación.

Figura 89

Trazado de curvas en Excel y la Nube resultados del Unit Commitment



### 5.3. Resultados de los algoritmos matemáticos

A continuación, se presenta los resultados obtenidos de los modelos de optimización del apartado 4.9, un primer modelo resuelto en CPLEX Optimization Studio se encarga de optimizar la producción procurando obtener los costos mínimos, los resultados muestran la cantidad de productos a fabricar en cada estación, descentralizando la demanda total entre el proceso uno y dos. Un segundo modelo se encarga de buscar las condiciones de operación más óptimas para la generación de energía eléctrica para cada hora de un día, los resultados del modelo son utilizados para encontrar las consignas óptimas que debería tener un generador a Gas para poder producir la energía que el Unit Commitment considera óptima.

#### 5.3.1. Optimización de la producción

Los parámetros involucrados en el modelo son ingresados desde la nube utilizando una interfaz gráfica como se mostró en el apartado 4.9, los datos son enviados a una hoja de Excel, en la Tabla 2 se muestra la ubicación de fila-columna que se asigna a cada parámetro.

**Tabla 2***Distribución de las variables de restricción en Excel*

	Fila	Producto 1	Producto 2	Producto 3
Columna		D	E	F
Demanda por día	4	50	80	40
Costo de Producción $X_1$	5	10	9.5	9
Costo de Producción $X_2$	6	10	10	9.5
Consumo de Materia prima 1	9	2	2	3
Consumo de Materia prima 2	10	1	1	1
Minutos de trabajo c/u en $X_1$	14	1.5	1.5	1.5
Minutos de trabajo c/u en $X_2$	15	1.5	1.5	1.5
Máximo tiempo de trabajo	16	480	480	480
Producción mínima en $X_1$	17	5	5	5

Al resolver las ecuaciones planteadas en el apartado 4.9.1 con los datos de la Tabla 2 por medio de software se obtienen los resultados mostrados en la Tabla 3 considerando una disponibilidad para la materia prima 1 de 175 kg y de 130 kg para la materia prima 2 por día.

**Tabla 3***Resultados y distribución de las variables de decisión en Excel*

	Fila	Factory IO	FluidSIM
Columna		I	J
Estación 1	4	5	45
Estación 2	5	80	0
Estación 3	6	5	45
Disponibilidad de Recursos	8	175	130

Para validar que los resultados del modelo sean óptimos se aplica una igualdad de ecuaciones al apartado 4.9.1 con los resultados de la Tabla 3.

$$\min C(U_E) = \sum_{t=1}^3 [C_{EF}(t) * U_{EF}(t) + C_{EFS}(t) * U_{EFS}(t)]$$

$$\sum_{t=1}^3 [A_{EF}(t) * U_{EF}(t)] \leq B_{EF}(t)$$

$$[(A_{EF1} * U_{EF1} + A_{EF2} * U_{EF2} + A_{EF3} * U_{EF3}) (A_{EF1} * U_{EF1} + A_{EF2} * U_{EF2} + A_{EF3} * U_{EF3})] \leq B_{EF}$$

$$[(2 * 5 + 2 * 75 + 3 * 5) (1 * 45 + 1 * 5 + 1 * 35)] \leq [175 \ 130]$$

$$[(10 + 150 + 15) (45 + 5 + 35)] \leq [175 \ 130]$$

$$[175 \ 85] \leq [175 \ 130]$$

$$\sum_{t=1}^3 [T_{EF}(t) * U_{EF}(t) + T_{EFS}(t) * U_{EFS}(t)] \leq T_{diario}(t)$$

$$T_{EF1} * U_{EF1} + T_{EF2} * U_{EF2} + T_{EF3} * U_{EF3} + T_{EFS1} * U_{EFS1} + T_{EFS2} * U_{EFS2} + T_{EFS3} * U_{EFS3} \leq 480$$

$$1.5 * 5 + 1.5 * 75 + 1.5 * 5 + 1.5 * 45 + 1.5 * 5 + 1.5 * 35 \leq 480$$

$$7.5 + 112.5 + 7.5 + 67.5 + 7.5 + 52.5 \leq 480$$

$$255 \leq 480$$

$$\sum_{t=1}^3 [U_{EF}(t) + U_{EFS}(t)] \geq D_{total}$$

$$[U_{EF1} + U_{EF2} + U_{EF3}] + [U_{EFS1} + U_{EFS2} + U_{EFS3}] \geq [50 \ 80 \ 40]$$

$$[5 \ 75 \ 5] + [45 \ 5 \ 35] \geq [50 \ 80 \ 40]$$

$$[50 \ 80 \ 40] \geq [50 \ 80 \ 40]$$

$$\sum_{t=1}^3 [U_{EF}(t)] \geq D_{min}$$

$$[U_{EF1} + U_{EF2} + U_{EF3}] \geq [5 \ 5 \ 5]$$

$$[5 \ 75 \ 5] \geq [5 \ 5 \ 5]$$

### 5.3.2. Optimización de las operaciones de generación

El problema del despacho busca generar la potencia óptima con costos mínimos de operación los resultados del algoritmo se muestran en la Tabla 4, cada columna muestra la generación óptima que debería producir cada unidad de generación se utiliza las tecnologías a Diésel, Gas, Solar, Eólica, Baterías se resuelve el problema para la demanda de cada hora por un día.

**Tabla 4**

*Unit Commitment de un día completo con cinco tecnologías de generación diferentes*

<b>Demanda</b>	<b>Diésel</b>	<b>Gas</b>	<b>Solar</b>	<b>Eólico</b>	<b>Descarga Bat.</b>	<b>Carga Bat.</b>
52.83	0.00	0.00	0.00	23.04	29.79	0.00
42.68	0.00	0.00	0.00	23.64	19.04	0.00
44.13	0.00	36.0	0.00	23.04	0.00	-14.91
29.05	0.00	0.00	0.00	22.48	6.57	0.00
41.02	0.00	36.7	0.00	22.21	0.00	-17.96
40.33	0.00	0.00	0.00	21.69	18.64	0.00
48.55	0.00	36.0	0.00	21.18	0.00	-8.63
46.61	0.00	36.0	0.00	20.69	0.00	-10.08
45.13	0.00	0.00	0.00	20.69	24.44	0.00
38.62	0.00	59.1	0.00	20.69	0.00	-41.18
31.33	0.00	0.00	0.00	20.93	10.40	0.00
30.47	0.00	0.00	0.00	21.43	9.04	0.00
36.40	0.00	0.00	0.00	22.48	13.92	0.00
36.78	0.00	0.00	17.4	22.21	0.00	-2.88
37.70	0.00	0.00	40.0	19.96	0.00	-19.28
38.92	0.00	0.00	60.7	18.22	0.00	-40.07
42.50	0.00	0.00	77.2	17.18	43.07	0.00
51.71	0.00	0.00	87.7	17.18	0.00	-53.17
75.12	0.00	57.6	0.00	17.45	0.00	0.00
86.63	0.00	68.4	0.00	18.22	0.00	0.00
100.0	0.00	36.0	0.00	18.73	45.27	0.00
72.18	0.00	52.7	0.00	19.47	0.00	0.00
70.41	0.00	44.8	0.00	20.44	5.13	0.00
62.40	0.00	40.7	0.00	21.69	0.00	0.00



Como datos de entrada se utiliza la curva de demanda mostrada en la Figura 88 a continuación se analiza la ecuación de balance para la hora punta, hora mínima y hora valle.

Ecuación de balance par la hora punta:  $t=21$

$$P_{GD}(t) + P_{GG}(t) + P_{GS}(t) + P_{GE}(t) + P_{BD}(t) = D(t) + P_{VE}(t) + P_{VS}(t) - P_{NS}(t) - P_{BC}(t)$$

$$0 + 36 + 0 + 18.73 + 45.27 = 100 + 0 + 0 - 0 - 0$$

$$100 (kW) = 100(kW)$$

Ecuación de balance par la hora mínima:  $t=4$

$$P_{GD}(t) + P_{GG}(t) + P_{GS}(t) + P_{GE}(t) + P_{BD}(t) = D(t) + P_{VE}(t) + P_{VS}(t) - P_{NS}(t) - P_{BC}(t)$$

$$0 + 0 + 0 + 22.48 + 6.57 = 29.05 + 0 + 0 - 0 - 0$$

$$29.05 (kW) = 29.05(kW)$$

Ecuación de balance par la hora valle:  $t=5$

$$P_{GD}(t) + P_{GG}(t) + P_{GS}(t) + P_{GE}(t) + P_{BD}(t) = D(t) + P_{VE}(t) + P_{VS}(t) - P_{NS}(t) - P_{BC}(t)$$

$$0 + 36 + 0 + 22.21 - 17.19 = 41.02 + 0 + 0 - 0 - (-0)$$

$$41.02 (kW) = 41.02(kW)$$

**Tabla 5**

*Resultado de las operaciones de escalado y normalizado para el generador a Gas*

	Hora Punta	Hora Mínima	Hora Valle
Velocidad del Eje (rpm)	538.06	0	718.52
Margen de Sobre tensión (p.u)	0.45	0	0.6
Temperatura de la turbina (K.)	896.76	0	1197.53
Eficiencia global del Sistema	0.16	0	0.21
Demanda (kW)	100	29.05	41.21

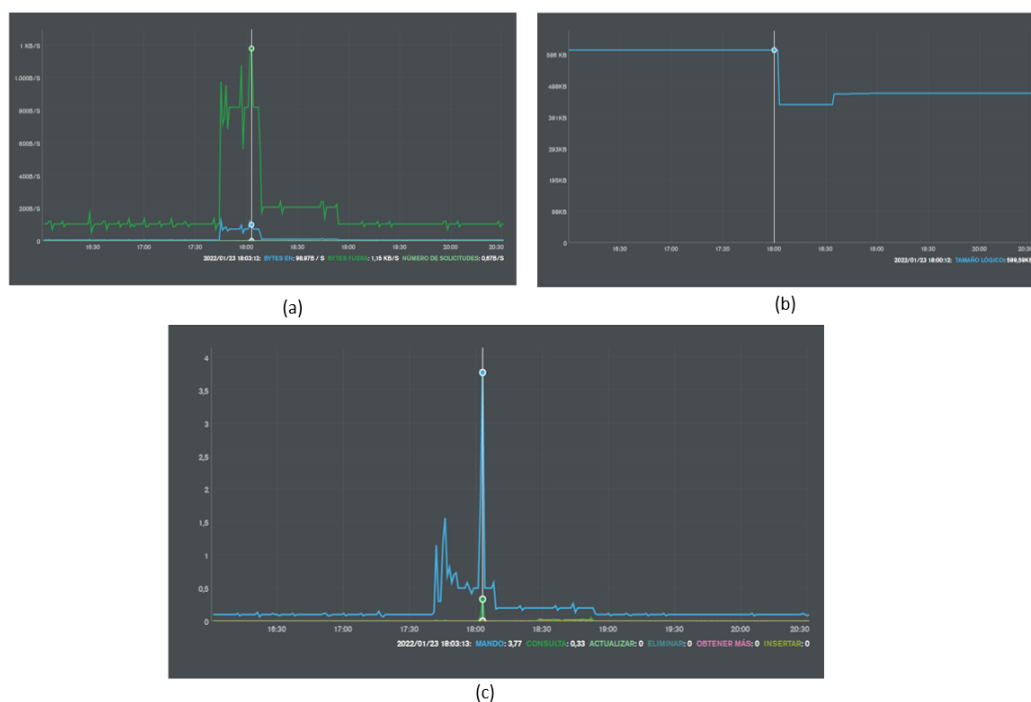
En la Tabla 5 se muestra los resultados de las operaciones para determinar las consignas que debería asignarse al generador de Gas para generar la potencia adecuada para cierta hora.

#### 5. 4. Seguimiento a la base de datos

En la Figura 90 en a) se muestra la cantidad de Bytes por segundo que son enviados al servidor de MongoDB a través de node-red representado de color azul, la cantidad de bytes por segundo que son leídos de este servidor desde node-red y la nube se representa con color verde.

**Figura 90**

*Monitoreo de las bases de datos en MongoDB*



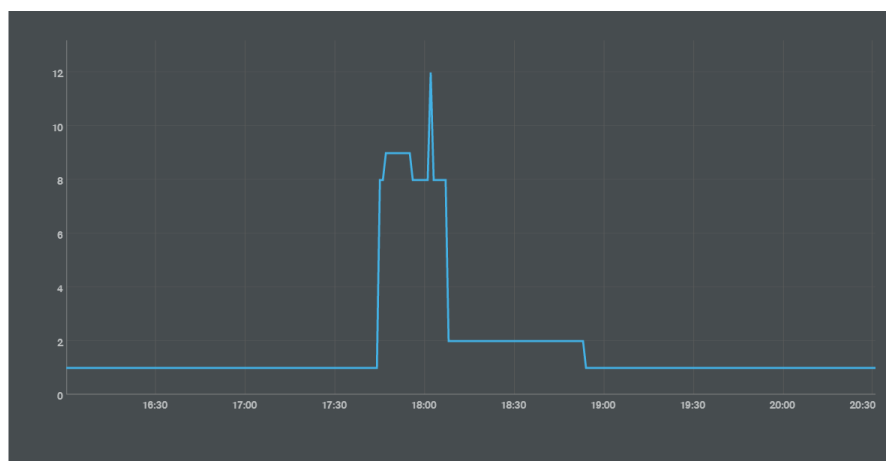
En b) se muestra el tamaño total de los documentos alojados en el servidor, el tamaño total no es constante presentando una disminución alrededor de las 18:00 y con un nuevo incremento a las 18:30, esto debido a que en esos lapsos de tiempo se vaciaron dos colecciones

con datos demasiados antiguos que no aportaban valor alguno y su eliminación permitió ahorrar espacio que podría utilizarse para guardar nuevos datos con mayor relevancia.

En c) se muestra la tasa promedio de comandos ejecutados por segundo durante el periodo de muestra. El gráfico representa la velocidad y el tipo de las operaciones realizadas en la instancia. En la Figura 91 se muestra el número de conexiones activas del servidor de MongoDB con los clientes durante el periodo de la muestra.

### Figura 91

*Conexiones activas del servidor MongoDB*

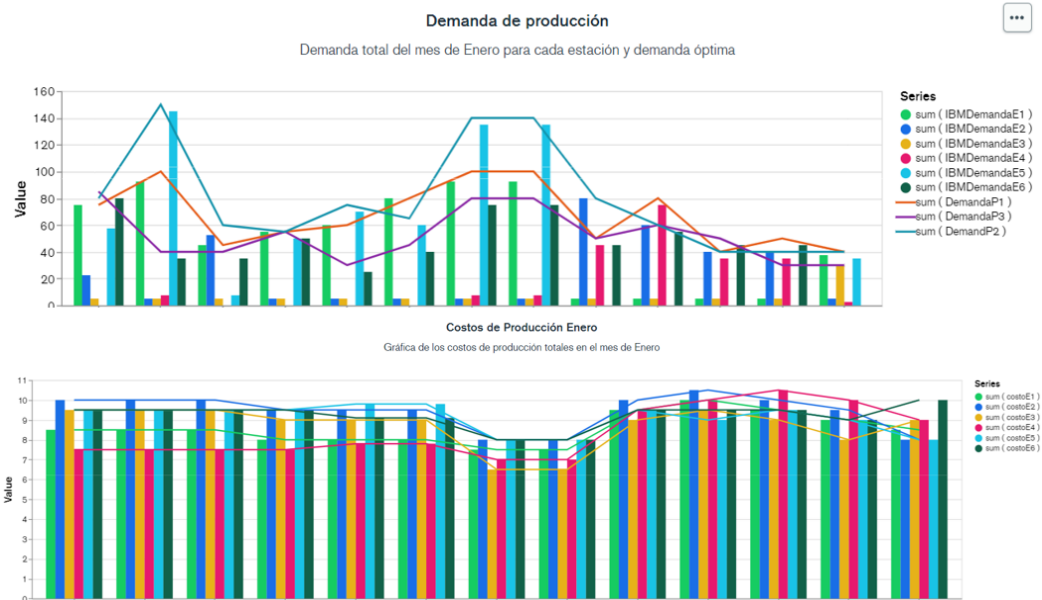


También se han trazado gráficos de barras y líneas para mostrar los resultados óptimos del problema de optimización sobre la producción y la demanda total de cada producto durante el mes de enero. Como se muestra en la Figura 92 en los primeros días de implementación del algoritmo matemático del apartado 4.9.1, en ciertas estaciones de mecanizado simuladas, la producción se ha mantenido en nulo por más de un día, obligando de esta manera a mejorar el algoritmo reajustando las restricciones para aprovechar al máximo la disponibilidad de los

equipos, es por ello que para días con grandes cantidades de demanda, los centros de mecanizado se encuentran operativos en su totalidad limitándose a cierta cantidad por día.

**Figura 92**

*Gráfico de barra de producción, demanda y costos*



Para aplicar el control estadístico de procesos se utiliza la gráfica de control de medias, se tomaron muestra cada minuto desde que inicia el proceso hasta que finaliza, los límites de control, como LCS, LIC y limite central se calculan utilizando las variables del número de actividades, tiempos de operación, tiempos de paro y tiempos de operación de ciclo único. En la Figura 93 los resultados muestran que los proceso se mantienen en control presentando un crecimiento gradual hasta alcanzar un punto de estabilidad.

**Figura 93**

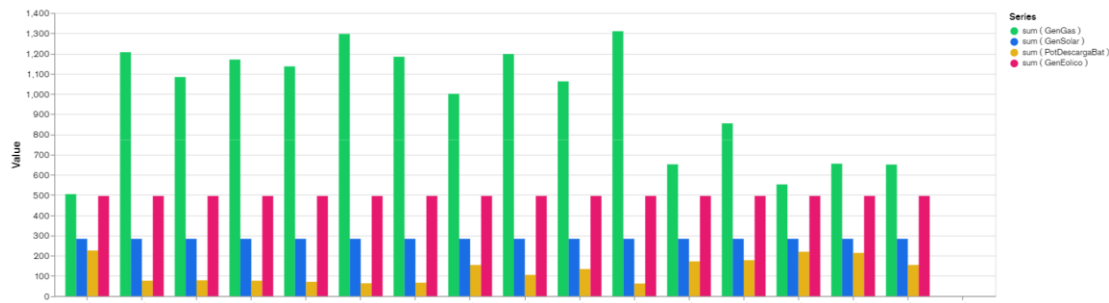
*Gráfica de medias utilizando las variables de producción*



Con respecto al proceso tres se utiliza un gráfico de barras para mostrar el total de potencia generada, los datos utilizados para trazar el grafico de la Figura 94 son los resultados obtenidos del algoritmo matemático almacenados en la base de datos, se observa que la potencia solar y eólica se mantiene constante para todos los días, debido a que las condiciones de irradiancia y de viento que fueron utilizados para el modelo fueron los mismos para todos los días, el único valor que cambio para cada día fue la curva de la demanda, esto se refleja en las potencias de las unidades a Gas, Diésel, y baterías.

**Figura 94**

*Grafica de barras producción total de potencia*



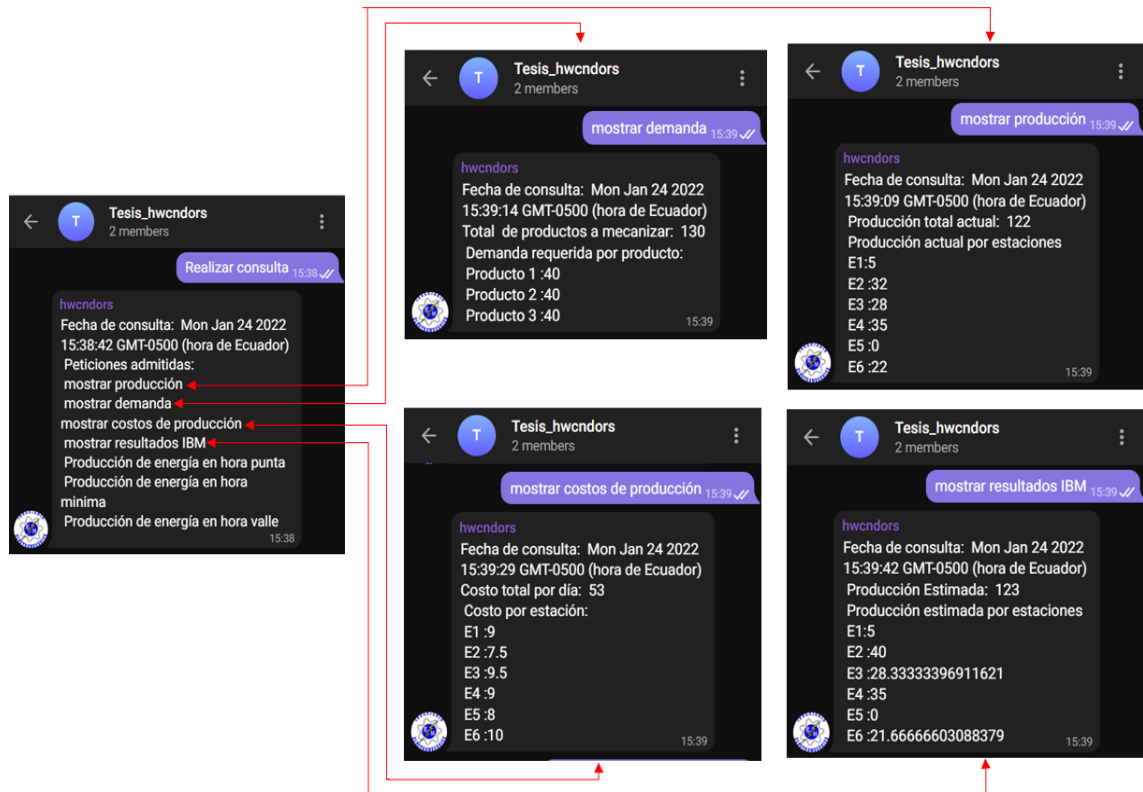
La información almacenada en MongoDB se obtuvo durante dos meses insertando datos de prueba hasta el mes de enero, mes en el cual los procesos fueron implementados en su totalidad seleccionando las variables y parámetros más relevantes, utilizando las herramientas de monitoreo de MongoDB se realiza un seguimiento a la base de datos para conocer su comportamiento, se elige una muestra de 5 horas del día 23 de enero del 2022, lapso en el cual se ejecutaron los tres procesos del sistema de control distribuido

### **5. 5. Envío y recepción de peticiones a través del Telegram Bot**

Con respecto a la configuración del telegram Bot mostrado en el apartado 4.11 es notorio que existe un adecuado intercambio de información y sobre todo presenta una ventaja al momento de realizar consultas ya sea de variables en ficheros o en tiempo real. Además, la ventaja de utilizar telegram para realizar consultas es que no se necesitó ninguna configuración previa para dar permisos o acceso a terceros.

Figura 95

Consultas del proceso uno y dos utilizando Telegram Bot



Al estar utilizando node-red de manera local en el computador, como se muestra en la Figura 96 en a), en caso de errores en la escritura el usuario no recibe respuesta a su petición, además en b) en caso de no estar ejecutando node-red de manera continua el usuario no recibe ningún tipo respuesta en el periodo de tiempo que se realiza la petición.

Figura 96

*Errores comunes en las peticiones con Telegram Bot*

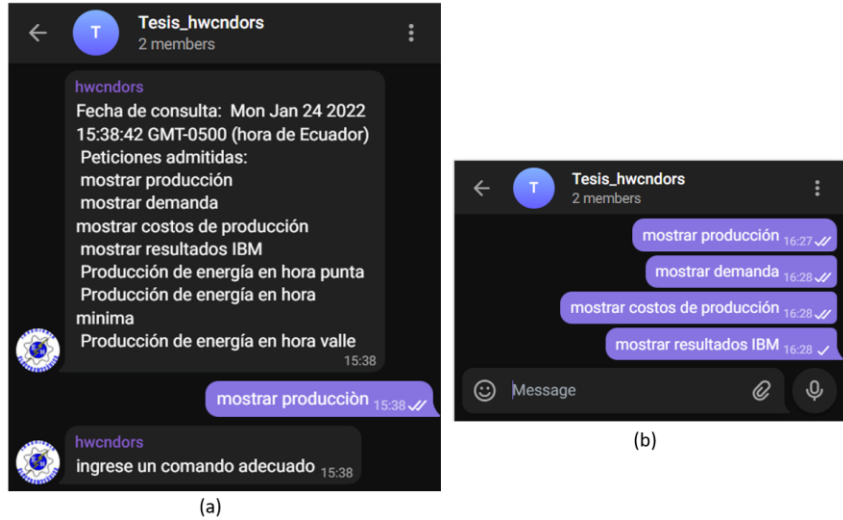
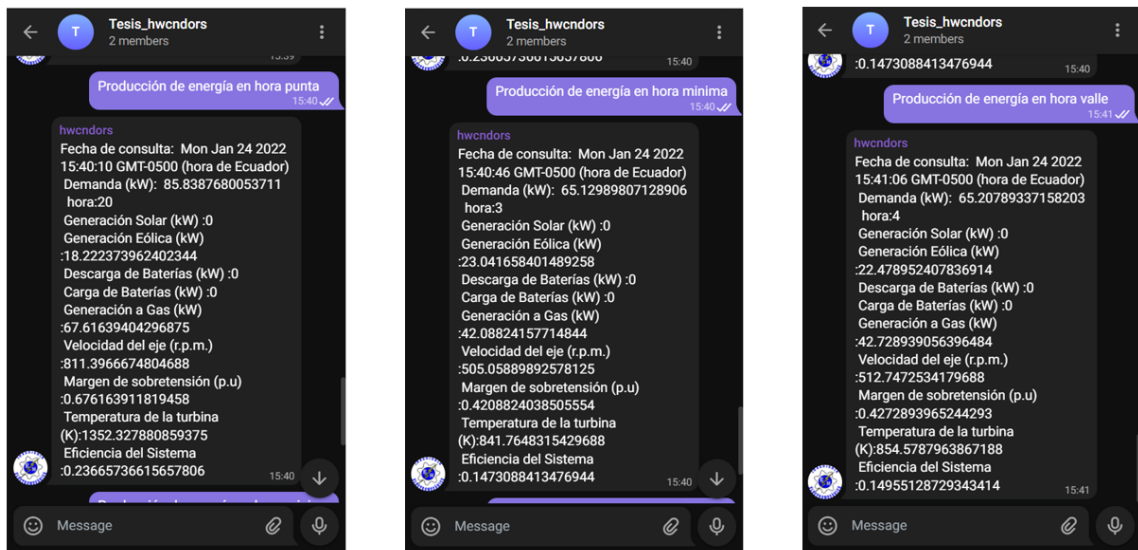


Figura 97

*Consultas del proceso tres utilizando Telegram Bot*





## Capítulo VI

### 6. Conclusiones y recomendaciones

#### 6. 1. Conclusiones

- Aprovechando los recursos de informática que posee la nube se ha logrado integrar el sistema de control y monitoreo local del SCD con un sistema remoto en la nube, fácil de usar y accesible desde cualquier parte del mundo a través de internet, presentando un mínimo de interrupciones con un alto grado de automatización permitiendo a los usuarios controlar, monitorear, almacenar datos, limpiar y visualizar los procesos en tiempo real logrando de esta manera una mayor interconectividad de los procesos industriales con las tecnologías emergentes de la industria 4.0.
- Los resultados de este trabajo muestran que la velocidad de transmisión de datos entre el controlador servidor y los clientes es rápida y eficiente, al utilizar los protocolos de bus de campo Profibus y la comunicación s7 existe una latencia nula al intercambiar datos a diferencia de utilizar, Modbus TCP que presenta una latencia de 500 ms en el intercambio, aun así, enviar y recibir información desde la nube a través del PLC servidor se lleva de manera exitosa sin perder ningún dato.
- En el referente a la virtualización de procesos industriales se ha identificado las ventajas e importancia que tienen los softwares de simulación dentro de la academia, permitiendo enseñar y aprender de manera continua a pesar de la crisis sanitaria que impide el uso y acceso a los laboratorios, además con la virtualización el consumo energético es menor, así como el espacio físico que se necesita concentrando múltiples máquinas y equipos virtuales en un solo equipo físico.

- Con respecto a la programación y configuración de los nodos se ha utilizado las librerías de node-red y de la plataforma IoT, con un mínimo grado de conocimiento en JavaScript se logró crear, mejorar e incorporar nuevas herramientas que permitan almacenar y visualizar los datos de manera local y desde la nube.
- Para minimizar los costos de producción se ha planteado un algoritmo matemático que se lo resuelve aplicando técnicas de programación lineal considerando factores como la demanda, tiempos máximos de operación, calidad del producto, consumo y disponibilidad de materia primera. Además, Para resolver los modelos planteados los entornos de herramientas informáticas de cálculo han permitido tener resultados óptimos, y conjuntamente con los softwares de simulación y programación como Factory IO, FluidSIM y TIA portal se pudo evaluar los resultados.
- Utilizar técnicas estadísticas como trazar graficas de control ha permitido analizar el comportamiento de los procesos como también identificar posibles fallos, saber cuándo el proceso está a punto de salirse de control, y conocer las posibles causas es de vital importancia en la ejecución. La correcta interpretación de estas graficas podría mejorar los procesos ajustando o reajustando los parámetros de producción.
- Para finalizar, la base de datos en la nube presenta mayor ventaja que una base de datos local al usar una base de datos en la Web los tiempos de configuración se redujeron y no se necesitó ninguna instalación para su funcionamiento. Al estar utilizando MongoDB almacenar los datos de cada proceso fue sencillo y con una programación mínima utilizando formato de texto JSON.

## 6. 2. Recomendaciones

- Al estar utilizando una base de datos no estructura como MongoDB es recomendable tener una mayor variedad de datos no solo del tipo numérico, podría ser textos, correos o alarmas que se puedan generar en los procesos. Al tener solo datos numéricos es recomendable utilizar una base de datos SQL, debido a que el servidor compartido en la nube no posee los nodos para conectar con una base de datos SQL de manera sencilla y dinámica se optó por usar una base de datos en MongoDB. Además de ofrecer un gran rendimiento y escalabilidad MongoDB formar parte de las herramientas más usadas del Big data y por ende de la industria 4.0.
- Al ser un proyecto con procesos virtualizados, construir procesos más complejos y con una mayor cantidad de variables es necesario si se quiere aprovechar al máximo las herramientas informáticas utilizadas en el proyecto, además esto podría mejorar o replantear un modelo más completo de optimización en donde se consideren más parámetros, como por ejemplo disponibilidad del espacio, trazado de rutas, tiempo de traslado del material y envíos, entre otros.
- Al estar trabajando con la última versión de node-red se recomienda investigar y configurar nuevas plataformas IoT que se encuentren en constante desarrollo debido a que, las actualizaciones de la plataforma utilizada en el presente proyecto no están al nivel actual de node-red dificultando la aplicación de ciertas técnicas y aumentando el tiempo de desarrollo.

**Bibliografía**

Alsina, M. G. (2017). *Big Data: Gestión y explotación de grandes volúmenes de datos* (Primera).

Editorial UOC. <https://books.google.com.ec/books?id=SFgtEAAAQBAJ>

Antúnez, F. (2016). *Puesta en marcha de sistemas de automatización industrial. ELEM0311*

(Primera). ic Editorial. <https://books.google.com.ec/books?id=31EpEAAAQBAJ>

Bahga, A., & Madiseti, V. (2014). *Internet of Things: A Hands-On Approach* (first). VPT Arshdeep

Bahga y Vijay Madiseti. <https://books.google.com.ec/books?id=JPKGBAAAQBAJ>

Belomonte, L. P. (2018). *Comunicaciones industriales y WinCC* (Primera). Marcombo.

<https://books.google.com.ec/books?id=UExOEAAAQBAJ>

Bonilla, J. (2016). *Prácticas de redes de datos e industriales*. Universidad de La Salle, Facultad de

Ingeniería, Programa de Ingeniería de Diseño y Automatización Electrónica.

<https://books.google.com.ec/books?id=U1z5DwAAQBAJ>

Carrera, A. (2021). *Desarrollo de una plataforma IoT para la supervisión y control de procesos*

*industriales de fabricación inteligente en tiempo real a través de la nube* [Universidad

Politécnica Salesiana]. <http://dspace.ups.edu.ec/handle/123456789/20134>

Céspedes, E. (2021). *Métodos de solución y análisis de programación lineal* (R. Céspedes (ed.);

Primera). Céspedes Esteban Roger. <https://books.google.com.ec/books?id=EioXEAAAQBAJ>

Creus, A. (2007). *Simulación y Control de Procesos por Ordenador* (Segunda). Marcombo.

<https://books.google.com.ec/books?id=udlcNQAACAAJ>

García, A. (2007). *Cim: el computador en la automatización de la producción* (Tercera). Ediciones

de la Universidad de Castilla-La Mancha.

<https://books.google.com.pe/books?id=Ook9Ec9n2ZcC>

Godoy, S., & Veloso, I. (2019). *Implementación de un Sistema de Control Distribuido, integrando virtualización y el entorno LabView* [Universidad Técnica Federico Santa María].

<https://hdl.handle.net/11673/48952>

Gómez, A., & Gómez, K. (2019). *El Control Estadístico de la Calidad de Los Procesos*. Amazon Digital Services LLC - KDP Print US. <https://books.google.com.ec/books?id=3rE9yAEACAAJ>

Gómez, J., Gómez, J., & Sánchez del Pozo, A. (2021). *Simulación de sistemas mecatrónicos* (Primera). Paraninfo. <https://books.google.com.ec/books?id=enl-EAAAQBAJ>

Heizer, J., & Render, B. (2004). *Principles of Operations Management* (Fifth). Pearson/Prentice Hall. <https://books.google.com.ec/books?id=jVlWssVHUFAC>

Heredia, J. (2001). *Sistema de indicadores para la mejora y el control integrado de la calidad de los procesos*. Publicacions de la Universitat Jaume I.

<https://books.google.com.ec/books?id=uLlt7WeQ7N4C>

International, D. (2020). *DAMA-DMBOK: Guía Del Conocimiento Para La Gestión De Datos* (Spanish Edition) (Segunda). Technics Publications.

<https://books.google.com.ec/books?id=5fnvDwAAQBAJ>

Joyanes, L. (2021). *Internet de las cosas: Un futuro hiperconectado: 5G, inteligencia artificial, Big Data, Cloud, BlockChain, Ciberseguridad* (Primera). Marcombo.

<https://books.google.com.ec/books?id=HE1OEAAAQBAJ>

Kume, H., & Vasco, E. (1992). *Herramientas estadísticas básicas para el mejoramiento de la calidad*. Norma. <https://books.google.com.ec/books?id=x4PnjSZYzMEC>

- Liu, X., Jun, P., Lin, L., Hao, C., Mi, Z., & Panos, P. (2017). Dynamic Coordinated Supply Chain Scheduling in an IoT Environment. *Optimization and Management in Manufacturing Engineering*, 126(3), 63–90. <https://doi.org/https://doi.org/10.1007/978-3-319-64568-1>
- Llanos, J. (2018). *Implementación de un sistema de almacenamiento de datos masivos para monitoreo estructura* [Universidad de Concepción].  
<http://repositorio.udec.cl/jspui/handle/11594/3420>
- Mora, A. (2017). *Diseño e implementación de una aplicación IoT en la nube de Azure para el análisis de imagen* [Universidad Politécnica de Madrid]. <https://oa.upm.es/48064/>
- Morgan, M., & Haley, T. (2019). Design of Food Process Controls Systems. *Handbook of Farm, Dairy and Food Machinery Engineering, Third*, 533–591.  
<https://doi.org/https://doi.org/10.1016/B978-0-12-814803-7.00022-1>
- Ortíz, D. (2019). Despacho Económico. *Sistemas Eléctricos de Potencia*, 25.
- Pardo, J. (2012). *Montaje y puesta en marcha de sistemas robóticos y sistemas de visión en bienes de equipo y maquinaria industrial* (Primera). ic Editorial.  
<https://books.google.com.ec/books?id=6VQpEAAAQBAJ>
- Ross, S. (2018). *Introducción a la estadística* (Segunda). Reverte.  
<https://books.google.com.ec/books?id=Ed3eDwAAQBAJ>
- Sánchez, W., Alvarado, E., Torres, E., & Toapanta, A. (2021). Design and Implementation of an Industrial Multinetwork TCP/IP of a Distributed Control System with Virtual Processes Based on IOT. *Proceedings of Sixth International Congress on Information and Communication Technology*, 236(Conference paper), 797–812.

[https://doi.org/https://doi.org/10.1007/978-981-16-2380-6\\_70](https://doi.org/https://doi.org/10.1007/978-981-16-2380-6_70)

Sevillano, F., Marta, B., Rodriguez, A., Valencia, A., Echave, D., Sánchez, S., Matilla, E., & Martínez, E. de P. (2021). *Ciberseguridad industrial e infraestructuras críticas*. Ediciones de la U. <https://books.google.com.ec/books?id=lqY5EAAAQBAJ>

Sharma, K. (2016). *Overview of Industrial Process Automation* (Second). Elsevier Science. <https://books.google.com.ec/books?id=hVEkDAAAQBAJ>

Tripathi, V., Saraswat, S., & Gautam, G. D. (2022). Development of a Systematic Framework to Optimize the Production Process in shop Floor Management. *Recent Trends in Industrial and Production Engineering, First*, 57–66. [https://doi.org/https://doi.org/10.1007/978-981-16-3135-1\\_7](https://doi.org/https://doi.org/10.1007/978-981-16-3135-1_7)

Valdivia, C. (2019). *Comunicaciones industriales* (Primera). Ediciones Paraninfo, S.A. <https://books.google.com.ec/books?id=znuYDwAAQBAJ>

Veneri, G., & Capasso, A. (2018). The CIM pyramid. In *Hands-On Industrial Internet of Things: Create a powerful Industrial IoT* (First, pp. 42–46). Packt Publishing. <https://books.google.com.ec/books?id=VN18DwAAQBAJ>

Wu, Y., Fei, H., Min, G., & Zomaya, A. (2018). *Big Data and Computational Intelligence in Networking* (First). CRC Press. <https://books.google.com.ec/books?id=pGpQDwAAQBAJ>

## **ANEXOS**