



POSGRADOS I

MAESTRÍA EN **ELECTRÓNICA Y AUTOMATIZACIÓN**

RPC-SO-30-No.507-2019

OPCIÓN DE
TITULACION:

PROYECTO DE DESARROLLO

TEMA:

DESARROLLO DE UNA PLATAFORMA WEB BASADA EN
INDUSTRIA 4.0 PARA EL CONTROL Y MONITOREO REMOTO
DE VARIABLES DENTRO DEL PROCESO DE EMPACADO DE
CAMARÓN

AUTOR:

JULIO CÉSAR LEMA PANDO

DIRECTOR:

BYRON XAVIER LIMA CEDILLO

GUAYAQUIL-ECUADOR
2022

Autor:



Julio César Lema Pando.

Ingeniero Electrónico con mención en Sistemas Industriales.

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

jlemap@est.ups.edu.ec

Dirigido por:



Byron Xavier Lima Cedillo.

Ingeniero Electrónico con mención en Sistemas Industriales.

Magíster en Automatización y Control Industrial.

blimac@ups.edu.ec

Todos los derechos reservados.

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

DERECHOS RESERVADOS

©2022 Universidad Politécnica Salesiana.

GUAYAQUIL – ECUADOR – SUDAMÉRICA

JULIO CÉSAR LEMA PANDO.

DESARROLLO DE UNA PLATAFORMA WEB BASADA EN INDUSTRIA 4.0 PARA EL CONTROL Y MONITOREO REMOTO DE VARIABLES DENTRO DEL PROCESO DE EMPACADO DE CAMARÓN.

Índice general

Índice de Figuras	V
Índice de Tablas	XI
Resumen	XII
Abstract	XII
1. Introducción	1
1.1. Descripción general del problema	2
1.2. Objetivos	2
1.2.1. Objetivo general	2
1.2.2. Objetivos específicos	3
1.3. Contribuciones	3
1.4. Organización del manuscrito	4
2. Marco Teórico	5
2.1. Estado del Arte	6
2.2. Definiciones Previas	7
2.2.1. Red Industrial Profinet	7
2.2.2. Bloques Get-Put	8
2.2.3. Desarrollo Industrial con Visual Studio	10
2.2.4. IOT Industria 4.0	37
3. Marco Metodológico	41
3.1. Desarrollo del Proyecto	42
3.1.1. Fase-1 Migración de Paneles Eléctricos.	42
3.1.2. Fase-2 Fase 2: Comunicaciones S7, configuración en la PC.	52
3.1.3. Fase-3 Desarrollo y aplicación HMI WEB	60

<i>ÍNDICE GENERAL</i>	IV
3.1.4. Fase 4: Desarrollo HMI industrial Visual Studio	65
3.1.5. Fase-5 Integración Node-Red y datos a la Nube	77
4. Resultados	109
5. Conclusiones	121
6. Recomendaciones	123
7. Glosario	124

Índice de Figuras

2.1. Red profinet Potabilizadora Omarsa	8
2.2. Bloque Get Simatic	9
2.3. Bloque Put Simatic	10
2.4. Ventana error-advertencia Visual Studio.	11
2.5. Consola de inicio Visual Studio.	13
2.6. Navegador de proyectos Visual Studio.	14
2.7. Estructura CSharp Proyecto Visual Studio.	14
2.8. Escribir en la Consola Visual Studio.	15
2.9. Comando Try-Catch.	16
2.10. Estructura del programa.	17
2.11. Uso de variables Visual Studio.	17
2.12. Convertir variables.	18
2.13. Leer datos por consola.	18
2.14. Condición IF.	19
2.15. Ciclo de repetición While.	19
2.16. Ciclo de repetición For.	20
2.17. Matrices o Arreglos.	20
2.18. Variables globales.	20
2.19. Permitir acceso Put/Get.	21
2.20. Acceso optimizado tia portal.	22
2.21. Enlace Sharp7.	23
2.22. Enlace S7.NET.	23
2.23. Control on-off/control analógico	23
2.24. Arquitectura cliente servidor [565]	24
2.25. Arreglo en Visual Studio.	25
2.26. Arreglo en Tia Portal.	25
2.27. Funciones para leer datos.	25
2.28. Crear proyecto (Window Forms).	26
2.29. Explorador de soluciones.	26

2.30. LiveCharts WinForms.	27
2.31. Plataforma LiveCharts.	28
2.32. Entorno Runtime.	28
2.33. Publicar un proyecto-Paso1.	29
2.34. Publicar un proyecto-Paso2.	29
2.35. Publicar un proyecto-Paso3.	30
2.36. Publicar un proyecto-Paso4.	30
2.37. Icono Setup de instalación.	31
2.38. Ruta de instalación.	31
2.39. Pasos 1 para creación de un HMI.	32
2.40. Pasos 2 para creación de un HMI.	32
2.41. Pasos 3 para creación de un HMI.	33
2.42. Pasos 4 para creación de un HMI.	33
2.43. Pasos 5 para creación de un HMI.	34
2.44. Variables booleanas Tia Portal.	34
2.45. Librería Power Packs.	35
2.46. Plataforma Symbol Factory.	36
2.47. Ejecución en Hilo.	36
2.48. Librería Threading.	37
2.49. Icono Node-Red	38
2.50. Icono NodeJs.	38
2.51. Nodos comunidad Node-Red	39
2.52. Nodos para envío de mensajes instantáneos.	39
2.53. Aplicaciones en la nube.[826]	40
3.1. Configuración de dispositivos Planta Agua.	43
3.2. Pantallas sistema de bombeo.	43
3.3. Pantallas planta de agua.	44
3.4. Pantallas Ptar.	44
3.5. Pantallas sistema de bombeo.	45
3.6. Pantallas sistema Ptar.	45
3.7. Paneles eléctricos obsoletos Tornado 1-2.	46
3.8. Paneles eléctricos actuales Tornado1-2.	46
3.9. Configuración dispositivos tornados.	47
3.10. Pantallas Tornados.	47
3.11. Panel eléctrico obsoleto Palinox-2.	47
3.12. Paneles eléctricos actuales Palinox-2.	48
3.13. Panel Eléctrico Dióxido de Cloro.	49
3.14. Paneles eléctricos Galpón 5.	49
3.15. Configuración de dispositivos Pelado.	50

3.16. Pantallas Área Galpón 5.	50
3.17. Plano de ubicación Túneles de congelación.	51
3.18. Tablero Túneles de congelación.	51
3.19. Configuración de dispositivos túneles y compresores.	52
3.20. Pantallas túneles de congelación.	52
3.21. Configuración direcciones IP.	53
3.22. Subprocesos bloques FC.	54
3.23. Direcciones IP dispositivos Omarsa.	55
3.24. Permitir comunicación Put/Get.	56
3.25. Activar marcas de ciclo.	56
3.26. Configuraciones S7-Simatic.	57
3.27. Variables Get PLC Máster.	57
3.28. Configuración de bloques DB.	57
3.29. Bloque Put PLC Sistema de Bombeo.	58
3.30. Bloque Get PLC Máster.	58
3.31. Parámetros conexión Put.	59
3.32. Parámetros conexión Get.	59
3.33. Traslado de datos entre PLCs.	60
3.34. Carpetas de trabajo.	61
3.35. Bloque de envío de datos Web Tia Portal.	62
3.36. Páginas de Usuarios HTML.	62
3.37. Servidor Web administrar usuarios.	63
3.38. Diseño recirculación de agua.	63
3.39. Análisis costo beneficio.	64
3.40. Implementación del sistema de recirculacion kjendal	64
3.41. Códigos del programa paso 1.	65
3.42. Códigos del programa paso 2.	66
3.43. Códigos del programa paso 3.	67
3.44. Página de usuario estación Siemens	68
3.45. HMI Web equipos kjendal.	68
3.46. Variables DB16 PLC Máster.	69
3.47. Crear una clase paso 1.	69
3.48. Crear una clase paso 2.	70
3.49. Pantallas del proceso Omarsa.	70
3.50. Crear el objeto PLC.	71
3.51. Clase Variables.	72
3.52. Función para establecer conexión al PLC.	73
3.53. Función de lectura de variables.	74
3.54. Función del Hilo.	74
3.55. Llamado al control PLC.	75

3.56. Función para navegar entre pantallas.	76
3.57. Pantalla Menús del Proceso.	76
3.58. Pantalla Sistema de Bombeo.	77
3.59. Pantalla Ptar-Potabilizadora 1-2.	77
3.60. Pantalla Succionadora de Cascaras.	78
3.61. Pantalla Sistema Dióxido de cloro.	78
3.62. Pantalla Palinox 2.	79
3.63. Pantalla Tornado 1-2.	79
3.64. Descargar Nodejs versión LTS.	80
3.65. Versión Nodejs.	80
3.66. Versión npm.	81
3.67. Instalación Node-Red.	81
3.68. Ejecución Node-Red.	82
3.69. Ventana Manage Palette.	82
3.70. Gestor de paquetes librería Node-Red.	83
3.71. Creación de grupos y subgrupos en Node-Red.	83
3.72. Crear conexión Simatic con Node-Red.	84
3.73. Creación de variables en Node-Red.	84
3.74. Nodos lectura de datos Reales.	85
3.75. Configuración Nodo S7 in.	85
3.76. Comandos para lectura de variables.	86
3.77. Configuración Nodo Gauge.	86
3.78. Configuración Nodo text in.	86
3.79. Configuración nodo text variables.	87
3.80. Configuración Nodo S7 out.	87
3.81. Configuración Nodos de escritura booleanos.	87
3.82. Comandos booleanos.	88
3.83. Configuración Nodos de lectura Booleanos.	88
3.84. Configuración led booleano.	89
3.85. Configuración led booleano.	89
3.86. Comandos de tiempo.	90
3.87. Integración de Nodos para envío de correos.	90
3.88. Configuración correo electrónico	90
3.89. Acceso aplicaciones google.	91
3.90. Programación Java envío de texto	91
3.91. Intervalo de tiempo correo electrónico.	92
3.92. Alertas desde Node-Red al correo electrónico	92
3.93. Alertas desde Node-Red.	92
3.94. Paquete TelegramBot.	93
3.95. Crear un nuevo bot.	93

3.96. Crear token para enlace.	94
3.97. Insertar token en nodo telegram.	94
3.98. Función envió de mensajes Telegram.	95
3.99. Estructura envió de mensajes Telegram.	95
3.100 Instalación paquete WampServer.	95
3.101 Consola WampServer.	95
3.102 Ingresar en phpMyAdmin.	96
3.103 Creación base de datos Omarsa.	97
3.104 Base de datos temperaturas Túneles.	97
3.105 Terminología CRUD.	97
3.106 Inserta variables a la base de datos.	98
3.107 Extracción de datos temperaturas túneles.	98
3.108 Curva de temperaturas túneles.	98
3.109 Exportar datos a excel.	99
3.110 Eliminar historial base de datos.	99
3.111 Nodos base de datos.	99
3.112 Credenciales Api Ubidots.	100
3.113 Copiar token en nodo Ubidots.	101
3.114 Crear dispositivos en Ubidots.	101
3.115 Variables Ubidots.	101
3.116 Programación nodo función Ubidots.	102
3.117 Agregar Widget.	102
3.118 Editor de estilos.	103
3.119 Dashboards planta Omarsa.	103
3.120 Menú Node Red.	104
3.121 Dashboard Sistema de Bombeo.	104
3.122 Dashboard Planta Potabilizadora.	105
3.123 Dashboard Temperatura de los túneles.	105
3.124 Dashboard Temperaturas tiempo real.	106
3.125 Dashboard Variables Galpón 6.	106
3.126 Administrador root.	107
3.127 Configuración IP Putty.	107
3.128 Activación librerías Putty.	108
3.129 Instalación de nodos por consola.	108
3.130 Instalación IOT-2040 Siemens.	108
4.1. Alertas Telegram.	110
4.2. Nodos envió de correos electrónicos.	111
4.3. Lógica de programación conteo SCL.	111
4.4. Bloque S7 Contador.	112

4.5. Instrucciones tía portal ciclos de envío alertas.	112
4.6. Avisos de alertas por correo electrónico.	113
4.7. Resumen libras congeladas.	113
4.8. Temperatura de las cajas.	114
4.9. Coches ingreso a túneles de congelación.	114
4.10. Curva de temperatura Túneles.	115
4.11. Curva de temperatura Túnel-1.	115
4.12. Curva de temperatura Túnel-2.	116
4.13. Curva de temperatura Túnel-3.	116
4.14. Curva de temperatura Túnel-4.	116
4.15. Curva de temperatura Túnel-5.	117
4.16. Curva de temperatura Túnel-6.	117
4.17. Curva de temperatura Túnel-7.	117
4.18. Cuadro de eficiencias túneles.	118
4.19. Nodos base de datos temperatura.	118
4.20. Editor fecha y hora parte 1.	119
4.21. Editor fecha y hora parte 2.	119
4.22. Función preparar tendencias parte 1.	119
4.23. Función preparar tendencias parte 2.	120
4.24. Función preparar tendencias parte 3.	120
4.25. Edición de seteos barra de temperatura.	120
5.1. Curva de Niveles Cisternas.	122

Índice de Tablas

2.1. Plantillas Visual Studio.	12
2.2. Tipos de variables CSharp	15
2.3. Descripción S7 Client.	24

Dedicatoria

El presente manuscrito lo dedico a Dios por darme la fortaleza y la valentía de asumir nuevos retos y sobre concederme la salud en estos tiempos de pandemia.

A mi familia cercana ellos son lo que me inspira a seguir creciendo profesionalmente y como ser humano.

A mis amigos que me han contribuido con ideas es este proyecto y me han dado el soporte necesario para culminar con éxito.

Agradezco a la Planta Omarsa que me ha facilitado las herramientas y equipos necesarios para realizar los diferentes proyectos de automatización.

Agradezco a mi tutor el Máster Byron Lima que me ha brindado todo su respaldo durante la elaboración del proyecto.

Resumen

El presente desarrollo pretende dar a conocer las diferentes aplicaciones HMI ya sean de escritorio (Windows) o a través de la Web y que se comuniquen directamente con los PLCs en este caso de la familia Simatic Siemens por medio de Profinet, todo esto sin OPC o licencias y con la gran ventaja de incluir una infinidad de tags.

Las aplicaciones y los programas se integran en el lenguaje de programación Visual Studio C Sharp, que a nivel de empresa se ha venido implementando soluciones a bajos costos y manteniendo la calidad y la fiabilidad en los sistemas, estas variables pueden ser monitoreadas desde cualquier parte del mundo solo necesitan los permisos empresariales y una conexión a internet.

Una de las ventajas de trabajar con los Plcs de la familia simatic es que dan la facilidad de trabajar con su servidor Web embebido, en este proyecto se realiza una aplicación a través del lenguaje HTML, donde se utiliza cualquier navegador y en la ventana de búsqueda se coloca la dirección Ip del Plc y se puede monitorear y controlar variables del proceso.

Otro método de integración de variables vista en este manuscrito es por la plataforma Node-red, se configuran los nodos a conveniencia de las necesidades de la planta y se reduce notablemente el esquema de programación, de lenguaje nativo a lenguaje Json que es auto descriptivo y fácil de entender.

Gracias a la programación en nodos se puede enviar alertas configurables en el tiempo mediante vía mensajes de texto por la plataforma Telegram, o por correo electrónico.

La recopilación de las variables más importantes del proyecto se almacenan en una base de datos phpMyAdmin donde se puede exportar los datos a través de una tabla en excel.

Finalmente se crea un Dashboard dinámico en la nube por medio de la plataforma UBIDOTS, para monitorear los procesos en tiempo real ya sean en una PC o desde cualquier dispositivo móvil.

Abstract

This development aims to present the different HMI applications either desktop (Windows) or through the Web and communicate directly with the PLCs in this case of the Simatic Siemens family through Profinet, all this without OPC or licenses and with the great advantage of including an infinite number of tags.

The applications and programs are integrated in the Visual Studio C Sharp programming language, which at the enterprise level has been implementing solutions at low costs and maintaining quality and reliability in the systems, these variables can be monitored from anywhere in the world only need business permissions and an internet connection.

One of the advantages of working with the Plcs of the simatic family is that they give the facility to work with its embedded Web server, in this project an application is made through HTML language, where any browser is used and in the search window the IP address of the Plc is placed and it is possible to monitor and control variables of the process.

Another method of variable integration seen in this manuscript is through the Node-network platform, the nodes are configured to suit the needs of the plant and the programming scheme is significantly reduced, from native language to Json language, which is self-descriptive and easy to understand.

Thanks to the node programming, it is possible to send configurable alerts over time via text messages through the Telegram platform, or by e-mail.

The collection of the most important variables of the project are stored in a phpMyAdmin database where the data can be exported through an excel table.

Finally, a dynamic Dashboard is created in the cloud through the UBIDOTS platform, to monitor processes in real time either on a PC or from any mobile device.

Capítulo 1

Introducción

El planteamiento de titulación que conlleva a este proyecto, se desarrolló en un entorno en la nube aprovechando el servidor Web embebido en los PLCS, recopilando todas las señales importantes, y llevarlo a un entorno más amigable y que permita a diferentes usuarios tener acceso a la información que brindan los PLCS y transformarlos en indicadores de producción.

Se creó una aplicación WEB Industrial con un entorno de visualización de los procesos y que involucran las últimas tecnologías utilizando los servicios de la nube en este caso UBIDOTS. Se desarrolló un entorno de visualización de ciertas variables para dispositivos inteligentes como teléfonos y Tablets que permiten a los operadores controlar el proceso desde un lugar remoto o por lo contrario directamente en las líneas de producción.

Se creó un sistema de supervisión de alarmas gracias a la pasarela de comunicación IOT2040 SIEMENS donde se notifica vía mensaje de texto, correo electrónico a los diferentes usuarios los avisos o alertas por averías mecánicas, o mantenimientos programados.

Al ser esto una tecnología innovadora a bajo costo tiene un protocolo abierto para seguir integrando proyectos a futuro. Los aspectos positivos de este proyecto es que puede ser replicado en cualquier proceso industrial, utilizando tecnología de punta a bajo a costo.

1.1. Descripción general del problema

En la planta Omarsa dedicada al sector camaronero se ha desarrollado en los últimos años diferentes proyectos de automatización en diversas áreas y sectores como clasificadoras, túneles de congelación, planta de tratamiento de agua, Ptar, área de valor agregado, etc.

Todas estas automatizaciones se han compilado con la finalidad de disminuir las paradas de producción por deficiencia en la detección y análisis de fallos.

Se ha testeado desperdicios de agua, químicos, floculantes y se ha logrado optimizar y minimizar los índices de mermas.

Con estas implementaciones se ha optimizado el funcionamiento de los equipos electrónicos debido al correcto diseño y control de las protecciones eléctricas.

Los Plcs instalados en la planta Omarsa están enlazados en una red Profinet y direccionados a una IP específica que actualmente se puede monitorear y realizar cambios ONLINE únicamente a través del software TIA PORTAL de SIEMENS, si existe algún evento y se requiere forzar alguna señal o cargar un nuevo dispositivo fuera de las horas laborables el departamento de sistemas ha configurado una herramienta informática FORTICLIENT a limitados usuarios de mantenimiento para poder ver los programas en línea y dar una solución sin tener que ir a la planta. Esto es un avance tecnológico, pero no al nivel de las soluciones actuales que ofrece los servidores Web.

Con estas implementaciones se ha optimizado el funcionamiento de los equipos electrónicos debido al correcto diseño y control de las protecciones eléctricas.

Adquirir una SCADA profesional para muchas de las empresas es demasiado costoso por el tema de licencias y el limitado número de TAGS que no permiten tener un control minucioso y con expectativas de crecimiento, la propuesta que se plantea es diseñar una plataforma con diseños de HMI'S Web y con un numero infinito de TAGS

1.2. Objetivos

1.2.1. Objetivo general

Desarrollar una plataforma Web basada en la industria 4.0 para el control y monitoreo de variables dentro del proceso de empacado de camarón.

1.2.2. Objetivos específicos

- Analizar el estado del arte sobre el desarrollo de una plataforma Web basada en la industria 4.0 mediante la revisión sistemática de documentos científicos, en aras de sentar las bases conceptuales y metodológicas para el desarrollo del proyecto.
- Definir los requerimientos del diseño mediante el análisis del funcionamiento de una plataforma web con la finalidad de sentar las bases necesarias para la implementación de variables en la Nube.
- Implementar la página web, mediante el uso de herramientas informáticas especializadas y en función de los requerimientos de diseño, con la finalidad de generar pantallas dinámicas y que recopilen información de los PLCs instalados.
- Validar la programación implementada en la página Web, mediante pruebas de campo y recolección de datos, con la finalidad de medir la eficiencia del sistema autónomo generado.

1.3. Contribuciones

En la mayoría de empresas dedicadas al sector camaronero, carecen de sistemas informáticos que ayuden a optimizar los procesos que en su mayoría son manuales y que generan grandes pérdidas por la falta de control y monitoreo.

La contribución en este proyecto tubo varias etapas las mas relevantes son:

- Invocación de paneles eléctricos centralizados en lugares estratégicos de la planta y en zonas secas libres de humedad.
- Implementación de una red de PLCs de la familia Simatic S71200-S71500 ,conservando el estándar en marca y tecnología.
- Desarrollo de una aplicación Web utilizando el servidor Web embebidos en los PLCs Simatic S7.
- Desarrollo de un HMI de escritorio sin un limite de usuarios con cero costo para la empresa, que a diferencia de implementar un SCADA profesional los costos serian elevados y con un numero de tags limitados.

- Creación de una base de datos en un servidor local para el monitoreo constante de variables y eso exportarlo a la nube mediante dashboards, reduciendo notablemente el costo por averías de los equipos y aumentando el rendimiento en las líneas de producción.

1.4. Organización del manuscrito

El presente documento se encuentra organizado de la siguiente manera:

- Capítulo 1: Introducción, Descripción del problema, Objetivo General, Objetivo Específico 1, Objetivo Específico 2, Objetivo Específico 3, Objetivo Específico 4, Contribuciones.
- Capítulo 2: Marco Teórico, Estado del Arte, Definiciones previas.
- Capítulo 3: Marco Metodológico, Desarrollo del Proyecto.
- Capítulo 4 Resultados.
- Capítulo 5 Conclusiones.
- Capítulo 6 Recomendaciones.
- Finalmente la respectiva Bibliografía .

Capítulo 2

Marco Teórico

En esta sección se presenta el estado del arte donde se realiza un estudio de las investigaciones que transcriben otros autores y se valida con lo realizado en este desarrollo. Además se detallan conceptos y definiciones previas concernientes a los suscritos en el objetivo general, relacionado con las teorías de funcionamiento y temáticas como son las comunicaciones S7 Simatic , desarrollo industrial con Visual Studio CSharp, servidores web embebidos en los Plcs, programación estructurada Iot.

2.1. Estado del Arte

En la presente manuscrito, se realiza una investigación previa sobre el desarrollo de una plataforma Web basada en la industria 4.0 mediante la revisión sistemática de documentos científicos, y que utilicen Plcs de la familia Simatic.

En [Tapia, 2017].describe a la expresión Industria 4.0 como una fábrica inteligente y que es un proceso que apunta a la revolución industrial y que lo fundamental es la fusión de la fabrica o empresa con el internet implementando componentes inteligentes para reparaciones y monitoreo a distancia.

En [Corral González et al., 2021] menciona que empresas especialistas en el análisis del consumo de Internet a través de la web, concluyen que más de 4 billones y medio de personas en el mundo consumen este tipo de servicio. Y que las estadísticas del uso del internet y de la web ya es obligatorio causando un gran impacto en las aplicaciones servidas en la plataforma.

En [923] con la evolución de las soluciones en la nube, todos los principales proveedores de servicios en la nube hoy en día tienen una plataforma informática sin servidor en su oferta. Por ejemplo, Amazon Web Services (AWS) tiene AWS Lambda que permite a los consumidores ejecutar su código sin aprovisionar la infraestructura.

Por su parte [854] utiliza un sistema de monitoreo y alertas a través de Internet de las cosas muy utilizado es (IoT) Ubidots Cloud que es una de las plataformas para monitorear variables en tiempo real ya sea en un teléfono móvil o computadora portátil. El panel de Ubidots proporciona una interfaz para mostrar datos como nivel, temperatura, etc, los mismos que envía una alertas al usuario a través de un telegrama o correo electrónico en el teléfono.

En temas de seguridad indica [534] que el sistema de correo electrónico es una de las aplicaciones basadas en Internet más ubicuo en la actualidad y facilita a los consumidores enviar y recibir mensajes de correo electrónico entre el entorno y fuera de la red de área local.

El sistema de correo electrónico se utiliza todos los días en casi todas las organizaciones como una herramienta de comunicación entre gerentes, empleados, clientes y socios para un mejor flujo de información y realizar negocios que requieren comunicación con personas ajenas a la organización o de diferentes ubicaciones geográficas. El sistema de correo electrónico ofrece una solución rápida, confiable y fácil para dicha comunicación.

La migración de datos a la nube [750] se ha impuesto con fuerza en el área de la investigación tanto para la industria como para los requisitos

académicos. Los proveedores de la nube se están moviendo o planean mover sus aplicaciones al entorno de la nube logrando infraestructuras y diferentes organizaciones de servicios en la nube.

Esta [750] tarea incluye una gran cantidad de detalles que deben ser seleccionados por académicos para abordar una tarea de este tipo que plantea muchas preguntas que deben abordarse con un enfoque científico y práctico. Gran parte de las investigaciones sobre este tema enfatizan los desafíos del proceso de migración a la nube. Sin embargo, pocos estudios hacen hincapié en la vida útil de las aplicaciones prácticas en el entorno de la nube supervisando algunas medidas de rendimiento seleccionadas para compararlas antes y después de la migración. El objetivo de este proceso podría ayudar a tomar una decisión crítica con respecto a mantener o modificar la aplicación práctica migrada.

Una herramienta de programación para conectar dispositivos en hardware, API es Node-Red, es un servicio en línea con nuevas e interesantes formas visuales dentro del internet de las cosas, en [Ollora Zaballa et al., 2021] desarrollan un sistema de programación visual que recibe la información de sensores y plataformas IoT heterogéneas, la data la almacenan en una base de datos en la nube, en particular, se utilizan diversas fuentes de datos que provienen de actuadores, sensores, dispositivos inteligentes, estos datos se envían sobre diferentes protocolos de aplicación y el servidor desplegado usando Node-RED recibirá todos los datos. También se almacenan los datos en InfluxDB [Naqvi et al., 2017] a corto o medio plazo y como base de datos históricos.

2.2. Definiciones Previas

2.2.1. Red Industrial Profinet

Dentro de las comunicaciones Simatic que se destaca en este proyecto [con] **PROFINET** es un patrón Ethernet abierto que cumple la especificación IEC 61158 dentro del campo de la automatización industrial. Este tipo de red permite conectar equipos desde el nivel de hardware (Plcs, actuadores, sensores, otros dispositivos) hasta el nivel de control y gestión corporativa (sistemas de internet e informáticos).

PROFINET brinda una comunicación uniforme con la ingeniería cubriendo toda la planta industrial a distancias no mayores a los cien metros y gestionando apoyo a nuevas tecnologías de la información hasta el nivel de control como muestra la figura 2.1.

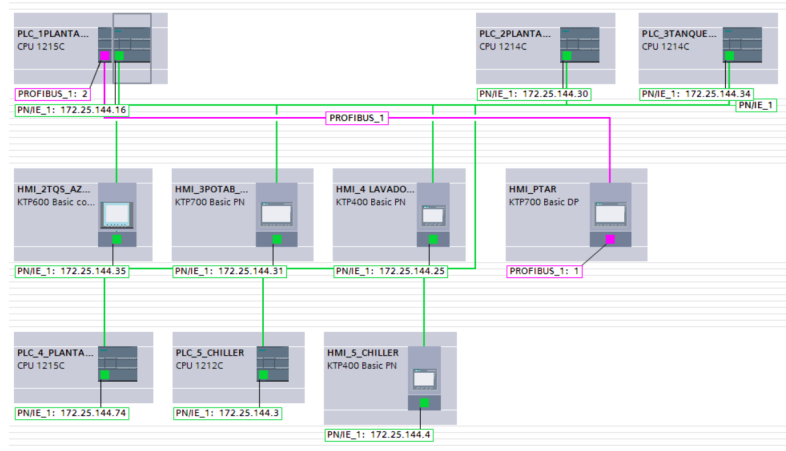


Figura 2.1: Red profinet Potabilizadora Omarsa

En [con] una red Ethernet, Profinet utiliza una totalidad de protocolos TCP/IP para el traspaso de datos en toda industrial y a todos los estándares. Podría decirse entonces que Profinet es una Ethernet Industrial, y sobrelleva tres tipos de tráfico.

- Tráfico de baja prioridad (TCP, UDP) [Oviedo et al., 2019]
- Tráfico real time (RT) [Coronel et al., 2004]
- Tráfico isochronous real time (IRT) [Rivalta]

2.2.2. Bloques Get-Put

Para que las CPU's puedan transferir datos [Rodríguez De La Rosa et al., 2021], se debe utilizar en el TIA Portal los bloques GET y PUT, propios de las comunicaciones tipo S7.

Las comunicaciones S7 son enlaces entre las CPU's a través de conexiones PROFINET y PROFIBUS.

En primer lugar, se debe activar la pestaña de "Permitir acceso vía comunicación PUT/GET" en las opciones de protección local de la CPU en la pestaña de propiedades del dispositivo. Esta opción en los PLC's de versiones superiores de firmware a la 4.0 por defecto se encuentra desactivada.

El bloque GET [Rodríguez De La Rosa et al., 2021] da la facilidad de leer datos desde una CPU interlocutora, en cambio que el bloque PUT

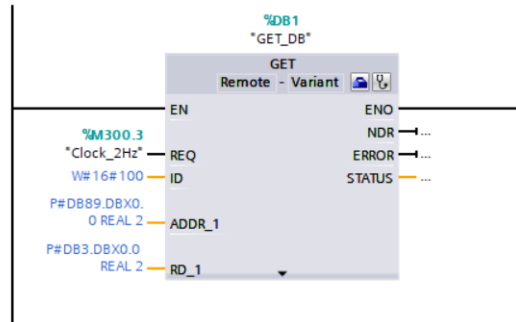


Figura 2.2: Bloque Get Simatic

permite escribir datos en una CPU interlocutora. Estos bloques son tratados por el Tia Portal como bloques de datos.

El bloque GET permite leer datos desde una CPU interlocutora. De esta manera, la CPU que utiliza el bloque obtiene datos de una CPU remota.

En el bloque Get se encuentran cuatro elementos a configurar:

REQ: Prepara la transferencia de datos en el flanco de subida de la variable asociada. Este parámetro cuando está activo establece la conexión con la CPU interlocutora para recoger los datos.

ID: El identificador hace relación a la conexión establecida entre dos PLC's. Esta conexión viene indicada mediante un código que simboliza la unión establecida

Normalmente es un formato de tipo Word. Este ID de conexión debe ser único para el enlace entre dos PLC's

ADDR: Es un puntero que necesita la dirección absoluta de la CPU remota para recoger los datos. En la dirección asignada los datos serán leídos cuando se active el REQ

La figura 2.2. muestra la estructura del bloque Get.

El bloque Put [Rodríguez De La Rosa et al., 2021] El bloque PUT tolera escribir datos del PLC local en un PLC remoto. Dentro de este bloque se alojan cuatro elementos a configurar:

REQ: Habilita la transferencia de datos en el flanco de subida de la variable asociada. Este parámetro cuando está activo establece la conexión con la CPU interlocutora y le envía los datos. El valor de este parámetro deberá ser del mismo tipo y longitud que el ADDR para que no se produzca una pérdida de información durante el envío.

ID: El identificador hace referencia a la conexión establecida entre dos

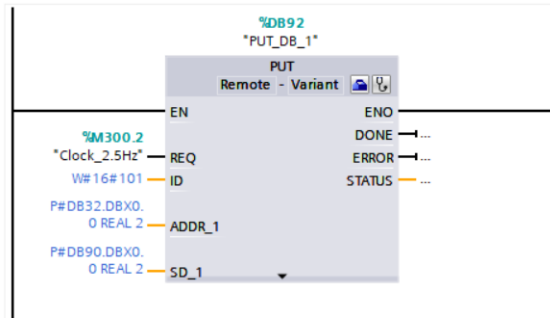


Figura 2.3: Bloque Put Simatic

PLC's. Esta conexión viene indicada mediante un código que simboliza la unión establecida. Normalmente es un formato de tipo Word. Este ID de conexión debe ser único para el enlace entre dos PLC's.

ADDR: Es un puntero que necesita la dirección absoluta de la CPU remota para escribir los datos. Los datos son enviados a la dirección remota indica en el ADDR.

SD: Es un puntero que necesita la dirección absoluta de un espacio de memoria de la propia CPU local. El envío de datos se efectúa mandando el valor de este parámetro cuando se inicia el REQ. El valor de este parámetro deberá ser del mismo tipo y longitud que el ADDR para que no se produzca una pérdida de información durante el envío.

La figura 2.3 detalla la estructura del bloque Put.

2.2.3. Desarrollo Industrial con Visual Studio

Entre las características Visual Studio proporciona un ambiente de desarrollo que permite diseñar, implementar, compilar, desplegar, y probar efectivamente varios tipos de aplicaciones y componentes utilizando diversos lenguajes de programación. Algunas de estas características son:

- **Un entorno de desarrollo integrado intuitivo (IDE).** Permite desarrollar, implementar, probar aplicaciones y componentes.
- **Desarrollo efectivo de aplicaciones.** Visual Studio brinda componentes gráficos que permite crear fácilmente interfaces de usuarios complejas.
- **Acceso a datos y servidores.** permite crear, consultar, modificar

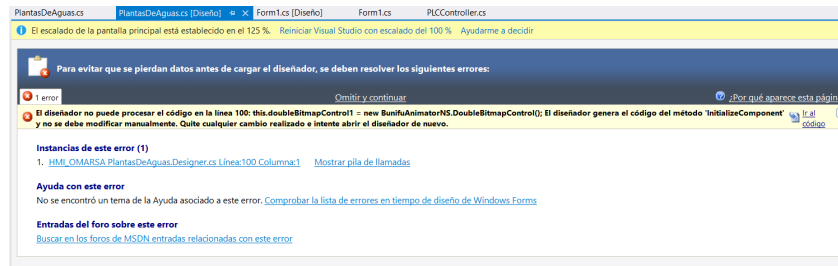


Figura 2.4: Ventana error-advertencia Visual Studio.

base de datos que las aplicaciones utilizan a través del diseñador de tablas.

- **Documentación y ayuda.** Visual Studio ofrece ayuda y orientación a través de Microsoft IntelliSense, fragmentos de códigos y del sistema de ayuda integrada que contiene la documentación y ejemplos.
- **Manejo de errores.** en las ventanas de errores se muestra las advertencias o mensajes que se manifiestan al crear o editar un código errado como se representa en la figura 2.4.

Plantillas de Visual Studio. en un entorno de desarrollo se permite crear aplicaciones Windows, Web, Servicios, Bibliotecas o aplicaciones empresariales para la nube. Visual Studio proporciona un serie de plantillas;

- Proporciona el código inicial para construir y crear rápidamente aplicaciones funcionales.
- Incluyen asistencia a componentes según el tipo de proyecto seleccionado.
- Configuran el IDE de Visual Studio desde el tipo de aplicación que se desea desarrollar.
- Agregan referencias a los ensambladores que el tipo de aplicación requiera.

En la siguiente tabla 2.1 se describe alguna de las plantillas mas comunes que se puede utilizar cuando se desarrolle aplicaciones NET Framework.

Microsoft Net. El marco Microsoft .NET y su familia de lenguajes se enfocan en el desarrollo multilinguaje para admitir la interoperabilidad

Tabla 2.1: Plantillas Visual Studio.

Plantilla	Detalle
Console Application	ligera interfaz de usuario gráfica
Windows Forms Application	gráfica en Windows Form
WPF Application	Aplicacion rica en interfaz usuario
Blank APP	Codigo inicial plataforma universal
Blank App 8.1	aplicaciones Windows Phone y Runtime
Class Library	Genera bibliotecas de clase
ASP.NET Web	Crea aplicaciones Web Forms, MVC,API

entre varios lenguajes de programación. El marco permite el desarrollo de aplicaciones similares en diferentes lenguajes a través de la reutilización de bibliotecas principales. Como resultado de este desarrollo multilingüe, la identificación y la capacidad de rastreo de fragmentos de código similares (clones) se convierte en un desafío clave [638].

NET Framework el NET Framework brinda un entorno de ejecución de las aplicaciones denominadas Common Language Runtime o CLR, esta plataforma es un componente de software cuya función es la de ejecutar las aplicaciones (.NET).

Servidores Empresariales los servidores empresariales (.NET) proporcionan escalabilidad, con fiabilidad, administración e integración dentro de la empresa o incluso entre distintas empresas. Dentro de los servidores (.NET) se puede encontrar plataformas como lo es Windows Server, Bistalk Server ,SQL Server, Exchange Server, Sharepoint Server, etc.

Bloque de servicios están integrados por Servicios Web que brindan diversas funcionalidades. Una aplicación de estos bloques de servicio se puede mencionar el servicio MicrosoftAccount(antes conocido como Windows Life) o al servicio One Drive (antes conocido como SkyDrive) entre otros.

Visual Studio .NET provee una plataforma para desarrollar aplicaciones para el (.NET Framework). Visual Studio disminuye el proceso entero de desarrollo de software, desde el diseño hasta la implementación, tiene soporte de varios monitores, desarrollo de aplicaciones para Sharepoint, suministra el desarrollo de aplicaciones para distintas versiones del (.NET Framework), Soporte intellinse y mucho más.

Según [PULIDO, 2003] Visual Studio .NET es un conjunto de herramientas de desarrollo para elaborar aplicaciones Web en ASP, Servicios Web XML, aplicaciones para computadores personales y aplicaciones para dispositivos android móviles. Todos los lenguajes apoyados por la plataforma

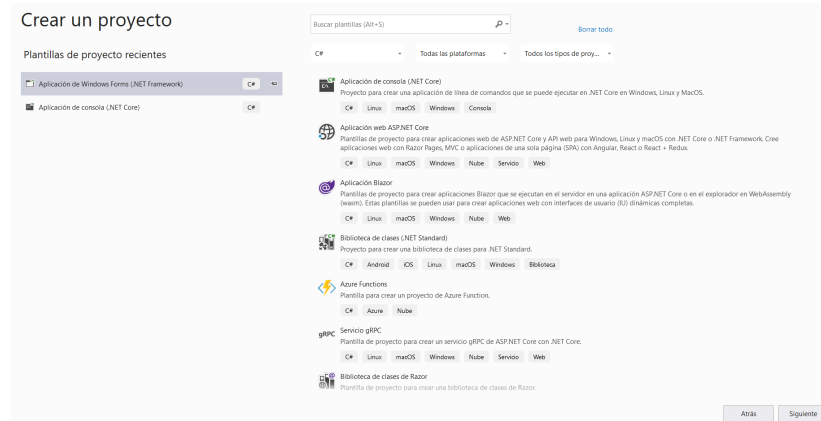


Figura 2.5: Consola de inicio Visual Studio.

.NET Framework, tales como: Visual Basic .NET, Visual C++ .NET, Visual CSharp .NET hacen uso de la misma plataforma de desarrollo integrado , el cual permite compartir herramientas y agrega valor a la creación de soluciones informáticas que requieran de una mezcla de lenguajes

.NET Core es un entorno de desarrollo de Microsoft mas actualizado, y no tiene este tipo de dependencia del sistema Windows y es modular, usando el sistema de paquetes NuGet.

.NET Core se lo usa en aplicaciones Web en plataformas como Windows, Linux, Mac Os. Se puede extender aplicaciones usando contenedores Docker en distintos entornos de Nube como son Azure, Amazon, GCP.

Se puede usar aplicaciones de escritorio UWP que permite ejecutar la aplicación en WINDOWS 10, XBOX, y Hololens compartiendo el código y sin tener que reescribir las bibliotecas.

CSharp es el lenguaje de programación desarrollado por Microsoft diseñado para compilar varias aplicaciones en .NET Framework que es un lenguaje elemental,eficaz y seguro. Las cuantiosas innovaciones de CSharp permiten desarrollar aplicaciones en el menor tiempo posible y mantener la expresividad de los lenguajes de estilo C.

Plantillas necesarias para el Proyecto

En el presente manuscrito se usan aplicaciones de consola de Visual Studio y posterior de Windows Forms conocidos también como formularios, estas plantillas se deben seleccionar al iniciar el proyecto como lo indica en la figura 2.5.

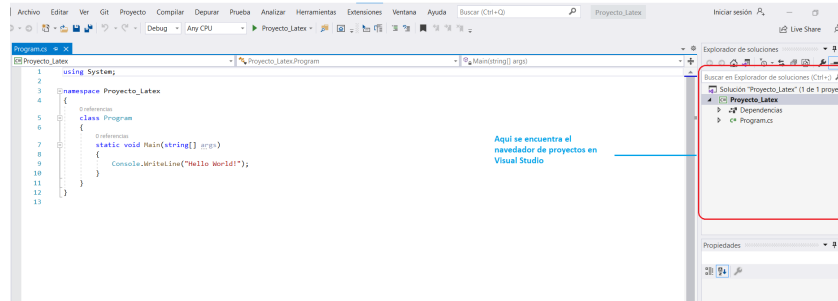


Figura 2.6: Navegador de proyectos Visual Studio.

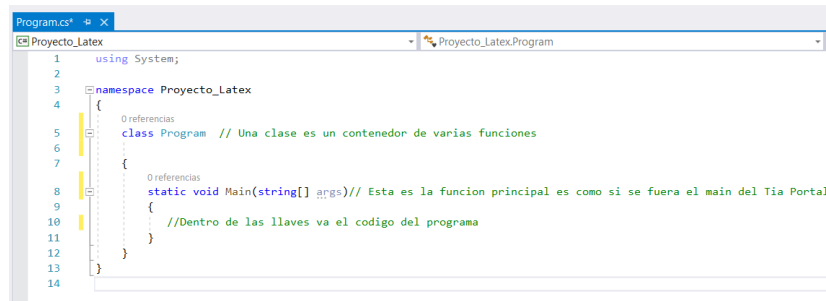


Figura 2.7: Estructura CSharp Proyecto Visual Studio.

Trabajar con soluciones y proyectos

Las soluciones son un grupo o conjunto de proyectos , el proyecto en si será el programa. Dentro de Visual Studio, en la parte derecha del escritorio se encuentra el explorador de soluciones en donde se podrá administrar los proyectos como lo indica en la figura 2.6.

Estructura de programa CSharp en consola

Al iniciar un proyecto el programa tiene un aspecto como en la figura 2.7.

Tipos de variables Visual Studio

Las variables de mayor uso y que nos sirve para enlazarnos con un PLC Simatic se detallan en la siguiente tabla 2.2

Escribir datos en Consola

La instrucción Console.WriteLine permite escribir datos en consola,

Tabla 2.2: Tipos de variables CSharp

Tipo de dato	Dimension
byte	1 byte
char	2 byte
short	2 bytes
int	4 bytes
long	8 bytes
float	4 bytes
double	8 bytes
bool	1 bit

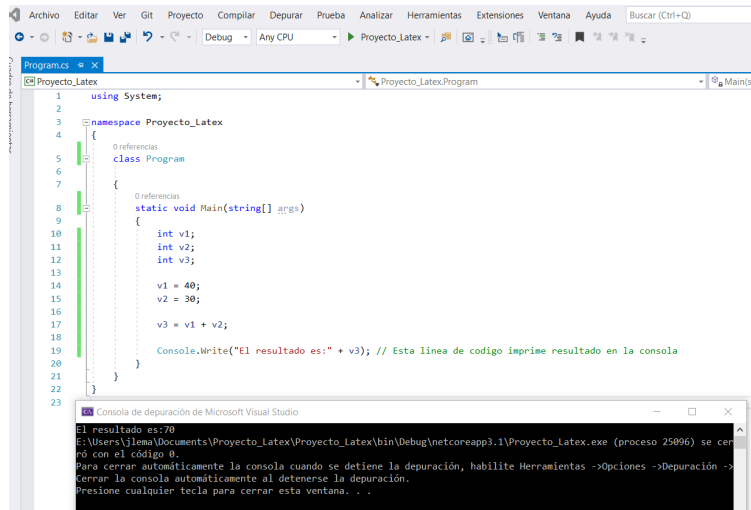


Figura 2.8: Escribir en la Consola Visual Studio.

CSharp distinguen las letras mayúsculas o minúsculas como lo muestra la figura 2.8.

Reconocer errores en el ingreso de datos Try-Catch

Con la instrucción Try-Catch es posible detectar cuando el usuario ha ingresado un dato incoherente o no deseado como se lo muestra en la figura 2.9.

Bases de programación en C

La terminología de esta investigación se relaciona con los siguientes

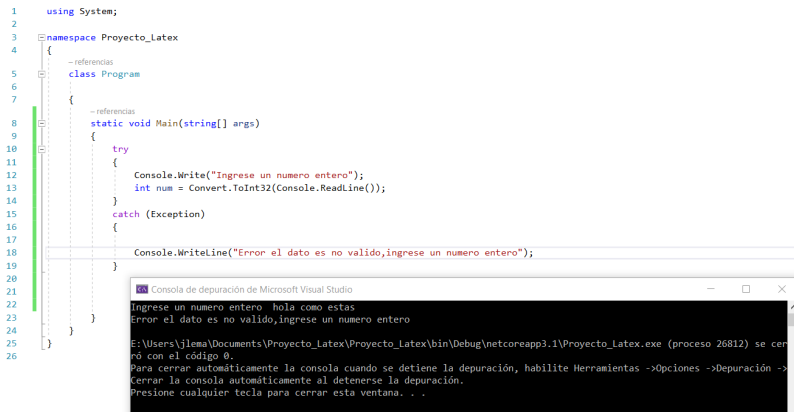


Figura 2.9: Comando Try-Catch.

términos que se relacionan con el funcionamiento de los Plcs y para que sea mas entendible se resumen varios términos como:

- Clases.
- Funciones.
- Métodos.
- Objetos.
- Constructores.
- Instancias.

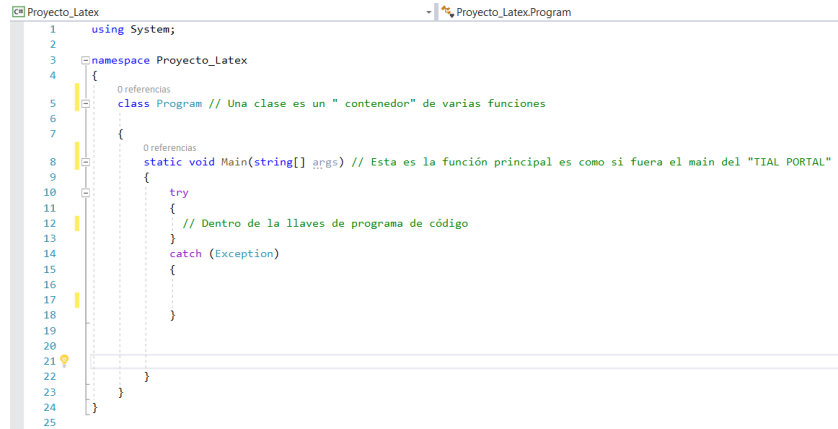
Al iniciar un proyecto el programa tendrá el aspecto como muestra en la figura 2.10.

Así como en tia portal se requiere variables para programar, en C Sharp es lo mismo se declara variables y se comienza por indicar el tipo de dato y un nombre. Hay varias formas de asignar un valor a la variable como se indica en la figura 2.11.

Como se hace en los PLCs en ocasiones es necesario convertir variables para que puedan ser compatibles con algún requerimiento como se muestra en la figura 2.12.

Leer datos por Consola

Con la instrucción `Console.ReadLine()` se puede leer datos de consola como se muestra en la figura 2.12.

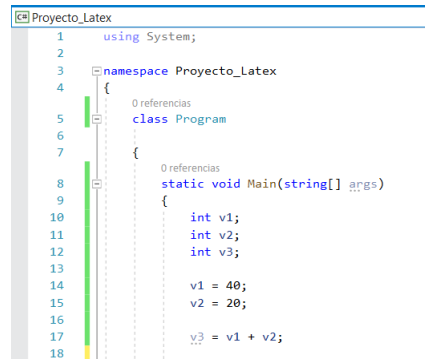


```

1  using System;
2
3  namespace Proyecto_Latex
4  {
5      0 referencias
6      class Program // Una clase es un " contenedor" de varias funciones
7      {
8          0 referencias
9          static void Main(string[] args) // Esta es la función principal es como si fuera el main del "TIAL PORTAL"
10         {
11             try
12             { // Dentro de la llaves de programa de código
13             }
14             catch (Exception)
15             {
16             }
17         }
18     }
19 }
20
21
22
23
24
25

```

Figura 2.10: Estructura del programa.



```

1  using System;
2
3  namespace Proyecto_Latex
4  {
5      0 referencias
6      class Program
7      {
8          0 referencias
9          static void Main(string[] args)
10         {
11             int v1;
12             int v2;
13             int v3;
14
15             v1 = 40;
16             v2 = 20;
17
18             v3 = v1 + v2;
19         }
20     }
21 }

```

Figura 2.11: Uso de variables Visual Studio.

Condicionales IF

El condicional IF es una instrucción que se usa en el momento del enlace con el PLC, es importante recordar su lógica y conocer la sintaxis que se maneja en C Sharp. La condición IF en este manuscrito servirá para saber si el PLC está o no conectado. El ejemplo de la condición IF se muestra en la figura 2.14.

Ciclo de repetición While

El ciclo de repetición While permite una rutina, mientras una condición sea verdadera. El uso en el presente proyecto es de enlazar al PLC lecturas

```

1 using System;
2
3 namespace Proyecto_Latex
4 {
5     class Program
6     {
7     {
8         static void Main(string[] args)
9         {
10            double n1 = Convert.ToDouble("13.5");
11            double n2 = 13.4;
12
13            Console.WriteLine("EL resultado es: {0}", n1+n2);
14        }
15    }
16 }
17
18

```

Selecionar Consola de depuración de Microsoft Visual Studio
 EL resultado es: 26.9

Figura 2.12: Convertir variables.

```

1 using System;
2
3 namespace Proyecto_Latex
4 {
5     class Program
6     {
7     {
8         static void Main(string[] args)
9         {
10            Console.WriteLine("Ingrese un numero entero");
11            int n1 = Convert.ToInt32(Console.ReadLine());
12
13            // Convert.ToInt32 = esta instrucción solo lee caracteres, requiere una conversión de datos
14            // (Console.ReadLine()) = este programa solicita la entrada de un numero entero
15        }
16    }
17 }
18
19

```

Figura 2.13: Leer datos por consola.

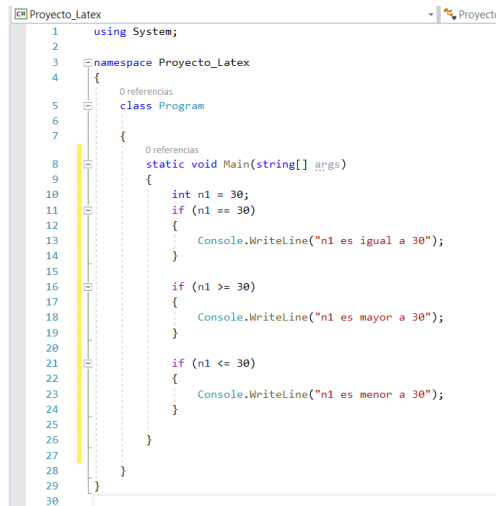
constantes de datos como se muestra en la figura 2.15.

Ciclo de repetición For

La instrucción For no se da casualmente en los programas que enlazan con el PLC, pero es necesario conocer su sintaxis, a continuación un ejemplo como muestra la figura 2.16.

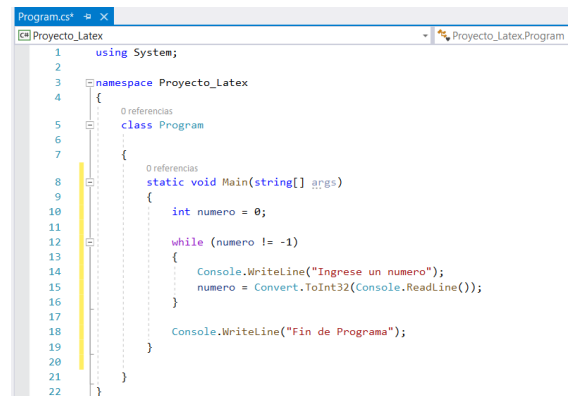
Matrices y arreglos

Los arreglos en C Sharp también conocidos como Arrays son indexados iniciando en cero(0). Esto permite decidir que tamaño va a tener según se necesite y se muestra en la figura 2.16.



```
1 using System;
2
3 namespace Proyecto_Latex
4 {
5     class Program
6     {
7
8         static void Main(string[] args)
9         {
10             int n1 = 30;
11             if (n1 == 30)
12             {
13                 Console.WriteLine("n1 es igual a 30");
14             }
15
16             if (n1 >= 30)
17             {
18                 Console.WriteLine("n1 es mayor a 30");
19             }
20
21             if (n1 <= 30)
22             {
23                 Console.WriteLine("n1 es menor a 30");
24             }
25         }
26     }
27 }
28
29
30
```

Figura 2.14: Condición IF.



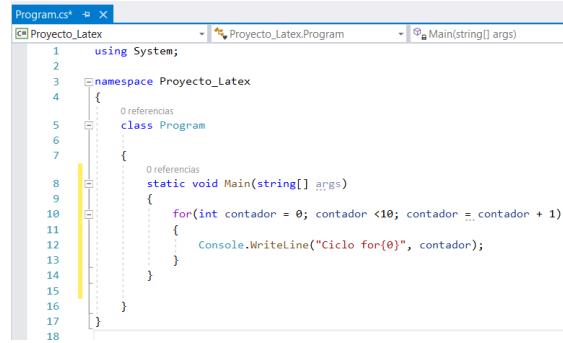
```
Program.cs* - X
Proyecto_Latex
1 using System;
2
3 namespace Proyecto_Latex
4 {
5     class Program
6     {
7
8         static void Main(string[] args)
9         {
10             int numero = 0;
11
12             while (numero != -1)
13             {
14                 Console.WriteLine("Ingrese un numero");
15                 numero = Convert.ToInt32(Console.ReadLine());
16             }
17
18             Console.WriteLine("Fin de Programa");
19         }
20     }
21 }
22
```

Figura 2.15: Ciclo de repetición While.

Variables globales

Las variables globales son declaradas dentro de la clase, no dentro del método o función.

Estas variables se crean para la comunicación con los PLCs y son requeridas en varias funciones, por ejemplo la dirección IP. Por lo tanto esta variable debe ser global como muestra en la figura 2.18.



```

1 using System;
2
3 namespace Proyecto_Latex
4 {
5     class Program
6     {
7     {
8         static void Main(string[] args)
9         {
10            for(int contador = 0; contador < 10; contador = contador + 1)
11            {
12                Console.WriteLine("Ciclo for{0}", contador);
13            }
14        }
15    }
16 }
17 }
18

```

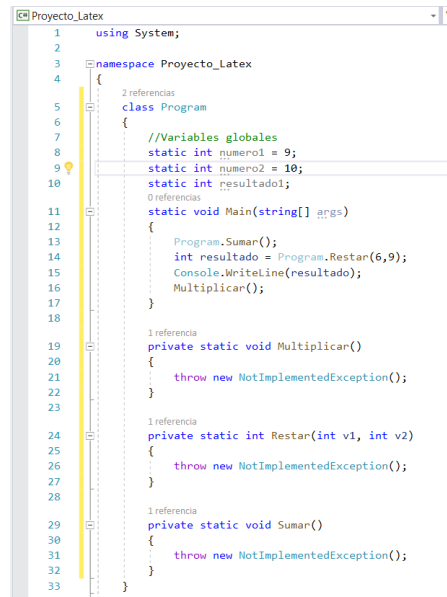
Figura 2.16: Ciclo de repetición For.

```

int[] valores; // valores sin inicializar
valores = new int[100]; // 100 elementos
valores = new int[20]; // contiene 20 elementos

```

Figura 2.17: Matrices o Arreglos.



```

1 using System;
2
3 namespace Proyecto_Latex
4 {
5     class Program
6     {
7         //Variables globales
8         static int numero1 = 9;
9         static int numero2 = 10;
10        static int resultado1;
11        static void Main(string[] args)
12        {
13            Program.Sumar();
14            int resultado = Program.Restar(6,9);
15            Console.WriteLine(resultado);
16            Multiplicar();
17        }
18
19        private static void Multiplicar()
20        {
21            throw new NotImplementedException();
22        }
23
24        private static int Restar(int v1, int v2)
25        {
26            throw new NotImplementedException();
27        }
28
29        private static void Sumar()
30        {
31            throw new NotImplementedException();
32        }
33    }
34 }

```

Figura 2.18: Variables globales.

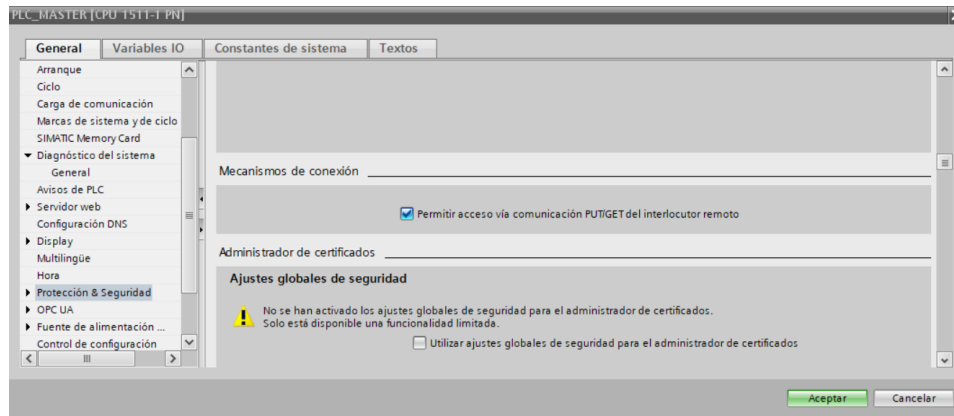


Figura 2.19: Permitir acceso Put/Get.

Preparación de los PLCs Simatic para la conexión con Visual Studio

En la habilitación de la comunicación de los PLCs S7-1200 y S7-1500 se requiere activar las funciones de comunicación Get/Put para enlazar ya sea a un Scada, Redes industriales o en este caso a un desarrollo HMI en Visual Studio, tal como se muestra en la siguiente figura [2.19](#).

Uso de las DBs

El tipo de las aplicaciones que se desarrolla en este proyecto están destinadas a enlazarse a PLCs ya operativos en donde se tendrán algunas variables. Sin embargo no todas las variables serán requeridas en el HMI.

En [Pérez Chueca, 2017] menciona que los DBs (Data Block) que corresponden con bases de datos secuenciales que pueden indexar diferentes tipos de datos en una lista en la que la posición viene indicada por un valor de offset por posicionamiento de memoria reservada en lista. Las DBs pueden contener UDTs, siendo las DBs concebidas como almacenamientos y las UDT como estructuraciones del mismo. Dentro de las DBs hay dos tipos, de datos o de instancia. Estas últimas siempre van asociadas a una FB (Function Block), que es un subprograma que emplea los datos remanentes contenidos en esa DB de instancia asociada. Estas últimas se suelen emplear cuando se repiten instrucciones referenciadas a grupos de datos similares. Las FCs (Function) son segmentos independientes de programa que fragmentan el mismo para hacerlo más manipulable y comprensible. Son cajas negras con

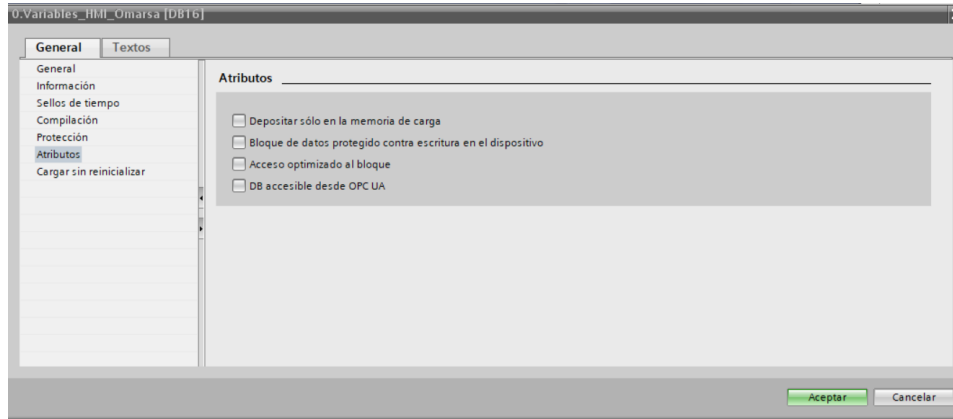


Figura 2.20: Acceso optimizado tia portal.

entradas, salida y valores internos (temporales y constantes) que ayudan a estructurar el programa y que son reutilizables en el caso de querer repetir operaciones iguales o similares (con distintas variables asociadas) a lo largo del programa. A diferencia de las FBs, las FCs no tienen DBs asociadas, por lo que en caso de necesitar de entradas, han de disponerse a forma de pines en el bloque que conforma los FC.

Habilitación de las DBs para la comunicación

De igual manera estos bloques deben habilitarse para el enlace, deshabilitando la opción Acceso optimizado al bloque.2.20.

El acceso optimizado es una característica exclusiva de Tia Portal y solo es para la serie S7-1200 y S7-1500.

Librerías de conexión S7- Visual Studio

La comunicación hacia los PLCs Simatic se logra con el uso de las librerías gratuitas prefabricadas, las cuales han sido desarrolladas por personas privadas ajenas a Siemens. Actualmente las librerías más conocidas y funcionales para Visual Studio son: SHARP7 figura 2.21, S7.NET figura 2.22.

Tipos de control

En el mundo de la automatización industrial existen dos grandes grupos de sistemas, los tipos on/off y los analógicos como se muestra en la figura 2.23. Para comenzar con los enlaces hacia el PLC desde aplicaciones desarrolladas

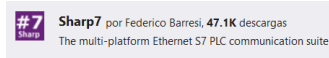


Figura 2.21: Enlace Sharp7.

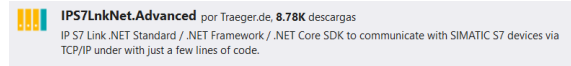


Figura 2.22: Enlace S7.NET.

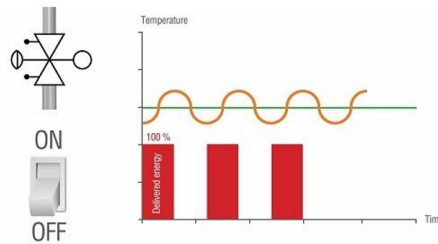


Figura 2.23: Control on-off/control analógico

con C Sharp en Visual Studio el tipo de control mas fácil es el analógico, ya que se contempla el uso de las variables tipo Int, Word, Dint, Dword, Real entre las mas utilizadas.

Administrar Paquetes NuGet

La librería Sharp 7 debe ser instalada por cada proyecto que se desarrolle y la forma de instalar es desde los paquetes NuGet en visual Studio.

NuGet es una extensión de Visual Studio que permite actualizar, agregar, eliminar librerías en proyectos de Visual Studio que utilizan .NET Framework.

Cuando se elimina una biblioteca, NuGet elimina y revierte los cambios que se hizo en el proyecto ejecutado.

Se puede realizar una verificación de obtención a fin de empezar a trabajar con las nuevas características.

Arquitectura de red

Para los enlaces de este proyecto se usa Ethernet arquitectura cliente

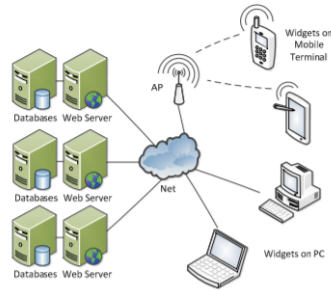


Figura 2.24: Arquitectura cliente servidor [565]

Tabla 2.3: Descripción S7 Client.

Función	Finalidad
ConnectTo	Conecta un cliente a un objeto PLC
SetConnectionType	Setea la conexión de tipo (PG/OP/S7Basic)
SetConnectionParams	Establece la dirección, el TSAP local y el remoto para la conexión
Connect	Conecta un objeto cliente a un PLC con parámetros implícitos
Disconnect	Desconecta al cliente

servidor en donde los clientes son las PCs o laptops que se comunican al PLC Máster que en este caso es el servidor. Los HMIs desarrollados se ubican en los Clientes para controlar los datos de los servidores. En las comunicaciones tradicionales desde una PC a una PLC de cualquier marca se usan los OPC para lograr el enlace, este proyecto se enlaza directamente sin OPCs o drivers o licencias de pago como lo muestra en la figura 2.24.

S7-Client

La primera clase de la librería Sharp7 que se emplea para la conexión es S7Client la cual permite crear un cliente que se comunice a un servidor PLC. Esta conexión se detalla en la tabla 2.3

Leer datos analógicos en Visual Studio

La mejor manera de iniciar la comunicación es con variables de tipo Int, Word, Dint, Dword, Real, y en la consola, posterior se realiza aplicaciones gráficas.

La clave para la comunicación es crear arreglos en Visual Studio como lo muestra en la figura 2.25 que sean mínimamente del mismo tamaño que sus

```

// Crear un arreglo
byte[] buffer = new byte[20];

```

Figura 2.25: Arreglo en Visual Studio.

1.GET_DATOS_AGUA			
	Nombre	Tipo de datos	Offset
1	Static		
2	RECIBE_DATOS_TQ...	Array[0..20] of Real	0.0
3	RECIBE_DATOS_TQ...	Real	0.0
4	RECIBE_DATOS_TQ...	Real	4.0
5	RECIBE_DATOS_TQ...	Real	8.0
6	RECIBE_DATOS_TQ...	Real	12.0
7	RECIBE_DATOS_TQ...	Real	16.0
8	RECIBE_DATOS_TQ...	Real	20.0
9	RECIBE_DATOS_TQ...	Real	24.0
10	RECIBE_DATOS_TQ...	Real	28.0
11	RECIBE_DATOS_TQ...	Real	32.0
12	RECIBE_DATOS_TQ...	Real	36.0
13	RECIBE_DATOS_TQ...	Real	40.0

Figura 2.26: Arreglo en Tia Portal.

```

// Lectura de la DB1 del plc
// plc.DBRead(1, 0, 10, buffer);

// Lectura de una memoria M:
//cpu.MBRead(200, 16, arreglo);

```

Figura 2.27: Funciones para leer datos.

correspondientes en Tia Portal como lo muestra en la figura 2.26.

Funciones para leer datos en Visual Studio

En la librería Sharp 7 se encuentran varias funciones para leer datos en una DB, y para leer datos en una memoria M, como lo muestra en la figura 2.27, los parámetros de las funciones son la clave para obtener una lectura correcta de los datos.

Interfaces gráficos (Windows Forms)

En C Sharp existen varias librerías de clase para implementar interfaces visuales. Se utiliza Windows Forms como plataforma.

Para generar una aplicación que utilice esta librería se debe crear un proyecto y los pasos son los siguientes:

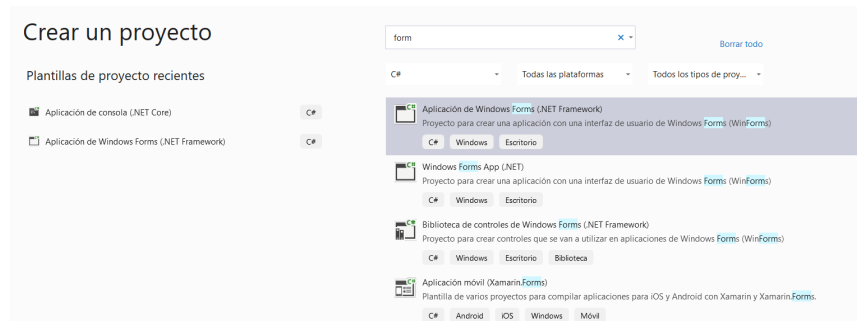


Figura 2.28: Crear proyecto (Window Forms).

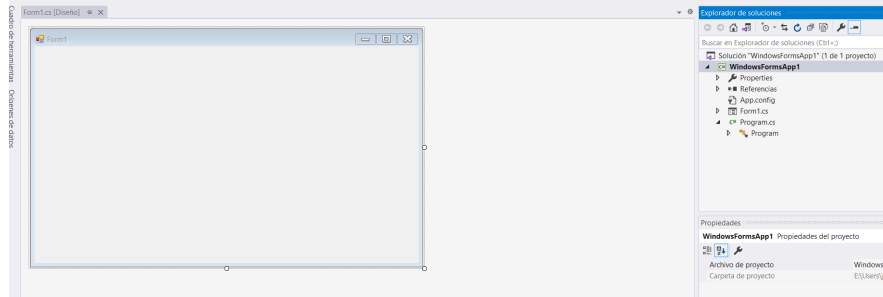


Figura 2.29: Explorador de soluciones.

- Desde el menú de aplicaciones de Visual Studio .Net se selecciona la opción: Archivo,Nuevo,Proyecto.
- Selecciona la plantilla Aplicación de Windows Forms (.NET Framework). La interfaz gráfica se visualiza como en la figura 2.28
- En el explorador de soluciones se crean tres archivos generados en forma automática:
Program.cs, Form1.cs y Form1.Designer.cs, como se muestra en la figura 2.29.

Dashboards en Visual Studio)

En Visual Studio hay una librería gratuita que permite la inserción de Dashboard y es LiveCharts.

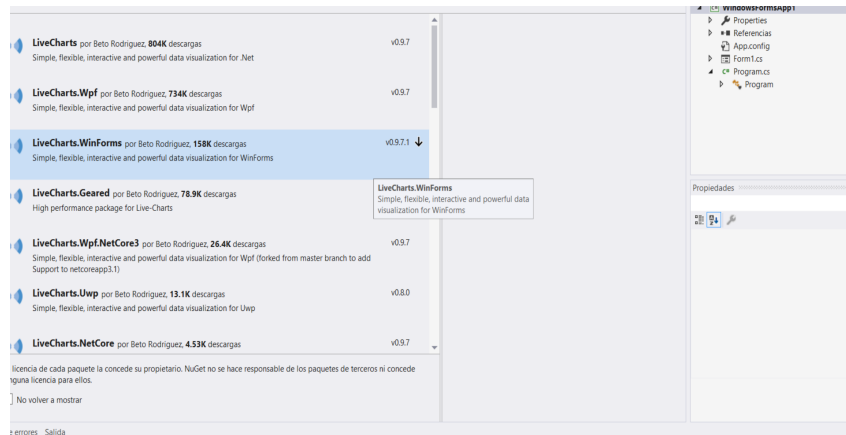


Figura 2.30: LiveCharts WinForms.

Esta librería se la puede instalar desde los paquetes NuGet y al buscar esta librería es necesario saber cual elegir, se debe dar clic en donde indica la figura 2.30.

LiveCharts

En el sitio oficial lvcharts.net, se encuentra gran cantidad de información sobre como insertar los diferentes visualizadores y como personalizarlos, como se ve en la figura 2.31.

Aplicaciones Runtime en Visual Studio

En los software de desarrollo industrial HMI, en forma general se tiene dos versiones, una que es para la programación y la otra que permite ejecutar aplicaciones finales, a esta versión se la conoce como el Runtime, y se muestra en la figura 2.32.

Proceso de Publicación

Al concluir un Proyecto en Visual Studio, el paso final es lograr una aplicación que funcione independiente de Visual Studio. La publicación consiste en hacer ejecutable (.exe) a la aplicación, entonces se puede decir que es el Runtime.

Paso1: En el explorador de soluciones, hacer clic derecho sobre el Proyecto y seleccionar la opción Publicar, como lo muestra en la figura 2.33.

Paso2: Se debe indicar la ubicación de la carpeta en donde se desea que se genere los archivos ejecutables, como lo muestra en la figura 2.34.

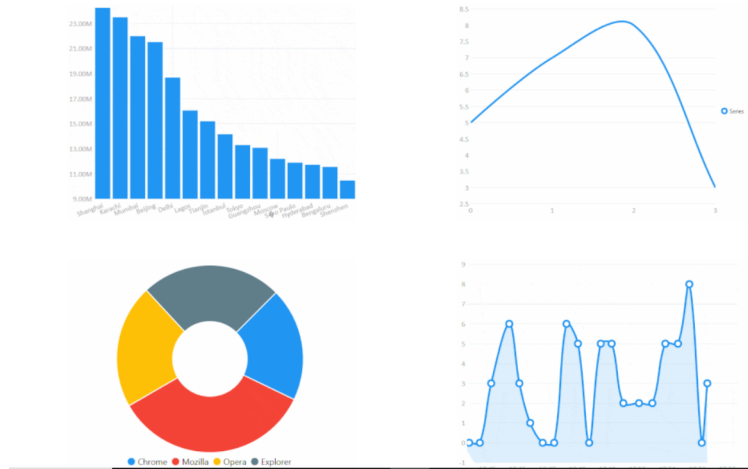


Figura 2.31: Plataforma LiveCharts.

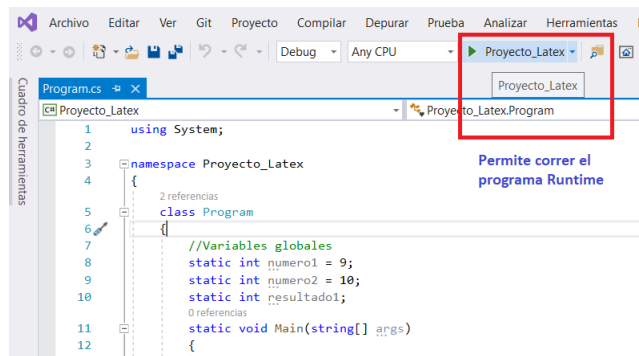


Figura 2.32: Entorno Runtime.

Paso3: Se debe indicar que la aplicación se instala "Desde un CD-ROM o un DVD-ROM", que es un instalador tradicional de Windows como lo muestra en la figura 2.35.

Paso4: Indicar que la aplicación no buscare actualizaciones. Este es el ultimo paso. Se presiona Finalizar y comienzan a generarse archivos de instalación como lo muestra en la figura 2.36.

Los archivos generados al terminar el asistente se generan dos archivos, uno con el nombre "Setupz el otro con mismo nombre de la aplicación;

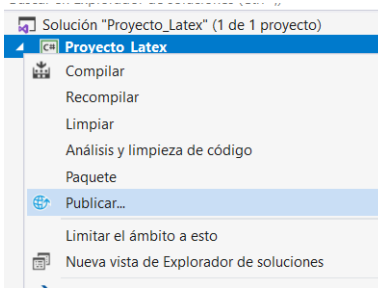


Figura 2.33: Publicar un proyecto-Paso1.

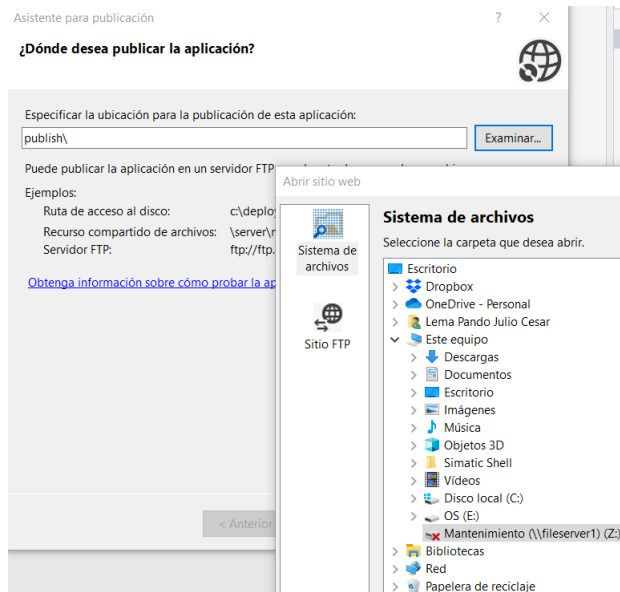


Figura 2.34: Publicar un proyecto-Paso2.

El archivo "Setup.es" el que debe ejecutarse, con lo cual debe aparecer la ventana de instalación y se muestra en la figura 2.37.

Posibles causas de error en la instalación

Suele ser común que al final del proceso no se genere el archivo ejecutable, la primera causa es que la solución se encuentre en una ruta muy extensa o que contenga caracteres especiales.

Este error sucede al momento de finalizar el asistente, en la parte baja de

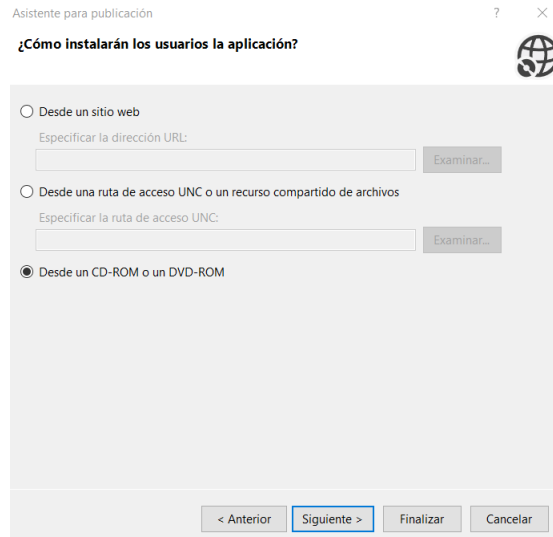


Figura 2.35: Publicar un proyecto-Paso3.

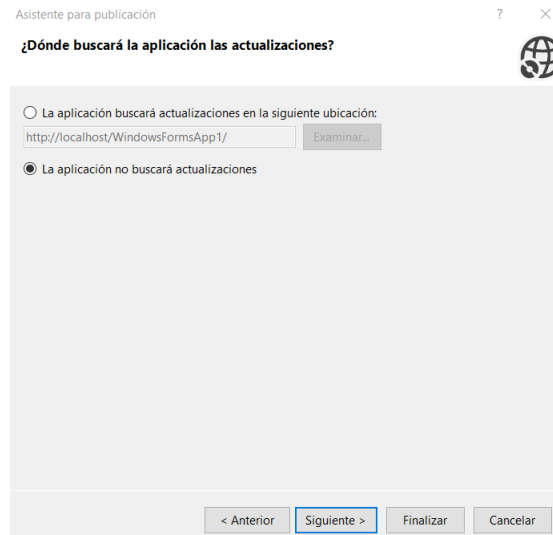


Figura 2.36: Publicar un proyecto-Paso4.

Visual Studio se verán una serie de iconos en forma de X, indicando que es un problema de compilación del programa, pero se debe a la ubicación El

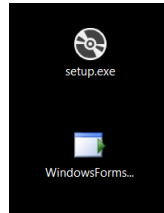


Figura 2.37: Icono Setup de instalación.

```

RUTAS DE UBICACION
C:\Users\jlema\Documents\Visual Studio 2019\DDS ---> Incorrecto
C:DDS ---> Correcto

```

Figura 2.38: Ruta de instalación.

error y la solución es copiar el proyecto lo más cercano de la raíz como se muestra en la figura 2.38.

Procedimiento general para crear un HMI

Ya que los diseños HMI son variables en dependencia del proyecto, el objetivo es crear una plantilla que pueda ser replicada a cualquier otro tipo de aplicación.

Para lograr esta plantilla se realiza una serie de pasos;

Paso 1: Insertar los controles básicos en el Formulario, Textos, botones, campos de textos, timer como se muestra en la figura 2.39.

A cada uno de estos controles es recomendable asignarle un nombre corto para optimizar la programación.

El timer es fundamental para lograr la lectura cíclica de las variables.

Paso 2: Iniciar con la programación en forma de variables globales y crear un objeto PLC, una variable para evaluar la conexión y los arreglos para lectura y escritura de bytes como se muestra en la figura 2.40.

Paso 3: En el timer se conecta al PLC y se evalúa su estado, esto se hace no solamente para conectar al PLC sino también para realizar una reconexión.

En ese mismo evento, cuando la conexión es exitosa se procede con la programación de las funciones de lectura como lo muestra la figura 2.41.

Paso 4: Validar desconexiones. En el mismo evento del Timer se realiza la programación para que en caso de que se desconecte el PLC, el HMI indique

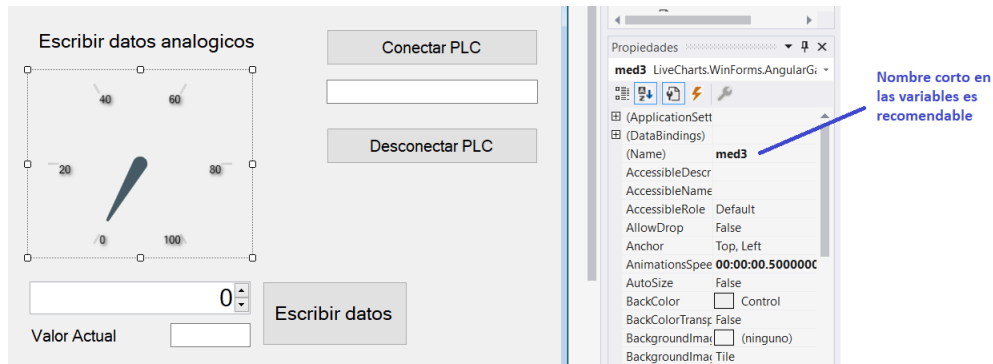


Figura 2.39: Pasos 1 para creación de un HMI.

```

namespace HMI_Planta_de_tratamiento
{
    2 referencias
    public partial class Form1 : Form
    {
        //Variables globales
        S7Client cpu = new S7Client();
        bool conectado = false;
        byte[] bufferRead = new byte[12];
        byte[] bufferWrite = new byte[12];

        1 referencia
        public Form1()
        {
            InitializeComponent();
        }
    }
}

```

Figura 2.40: Pasos 2 para creación de un HMI.

su desconexión como lo muestra la figura 2.42.

Paso 5: Aunque no es indispensable es recomendable colocar el componente gráfico.

Todos los paso se puede apreciar en la figura 2.43.

Función para leer variables booleanas

Las variables del PLC que se comunican con el HMI desarrollado en Visual Studio pueden estar en la tabla de las variables generales o e DBs. Al igual que como se hizo con las variables analógicas la recomendación es organizar

```

//Programacion en el temporizador
1referencia
private void timer1_Tick(object sender, EventArgs e)
{
    if (conectado == false)
    {
        int estado = cpu.ConnectTo("192.168.100.120",0,1);

        if (estado == 0)
        {
            conectado = true;
            campo7.Text = "PLC CONECTADO";
        }
        else
        {
            conectado = false;
            campo7.Text = "PLC DESCONECTADO";
        }
    }
    else
    {
        //Programacion lectura de datos del plc
        int estado = cpu.DBRead(1, 0, 12, bufferRead);
        if (estado == 0)
        {
            campo1.Text = S7.GetIntAt(bufferRead, 0).ToString();
            campo2.Text = S7.GetRealAt(bufferRead, 2).ToString();
        }
        else
        {
            conectado = false;
        }
    }
}

```

Figura 2.41: Pasos 3 para creación de un HMI.

```

//Programacion lectura de datos del plc
int estado = cpu.DBRead(1, 0, 12, bufferRead);
if (estado == 0)
{
    campo1.Text = S7.GetIntAt(bufferRead, 0).ToString();
    campo2.Text = S7.GetRealAt(bufferRead, 2).ToString();
}
else
{
    conectado = false;

    campo1.Text = "#####"; //Si el plc presenta una desconexion la variable
    campo2.Text = "#####"; // se pondra en asteriscos
}
}

```

Figura 2.42: Pasos 4 para creación de un HMI.

las variables tipo booleanas también en DBs como se muestra en la figura 2.44.

La librería Sharp7, también dispone de funciones de lectura para las variables de tipo booleana, la sintaxis de la función para leer datos tipo on/off es la siguiente.

```
bool bit1 = S7.GetBitAt (buffer,0,0);
```

Previamente a usar esta función los datos del buffer pueden haber sido cargados desde una DB o una tabla de variables tradicional.

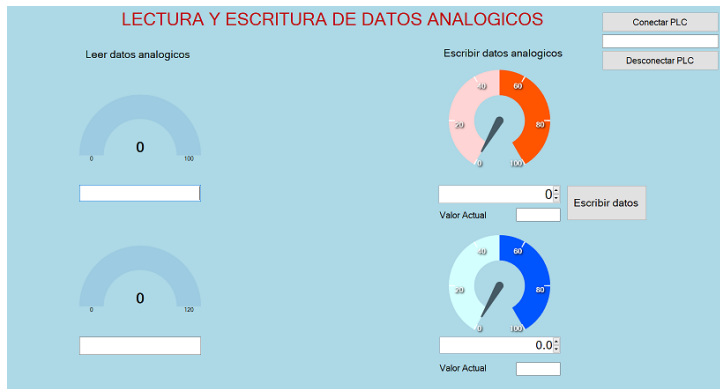


Figura 2.43: Pasos 5 para creación de un HMI.

PLC_T3_S6 ▶ PLC_1 [CPU 1512C-1 PN] ▶ Bloques de programa ▶ data1 [DB1]

Conservar valores actuales Instantánea Copiar instan...

data1					
	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...
1	Static				
2	Marcha	Bool	0.0	false	
3	Paro	Bool	0.1	false	
4	Bomba 1_in	Bool	0.2	false	
5	Bomba 2_Out	Bool	0.3	false	
6	Nivel_Tq	Int	2.0	0	
7	<Agregar>				
8	<Agregar>				
9	<Agregar>	Número del byte		Número del bit	
10	<Agregar>				

Figura 2.44: Variables booleanas Tia Portal.

Así mismo para escribir variables booleanas se mantiene la lógica , para leer se utiliza Get y para escribir se utiliza Set.

S7.SetBitAt (ref buffer,0,0, true);

En los parámetros de la función SetBit.At se encuentra como algo nuevo la palabra ref la cual indica que el argumento es trasladado en forma de referencia y no como un valor. La función esta prefabricada de la siguiente manera, se coloca el numero de byte y el numero de bit, así como lo hace la función de lectura de booleanos, sin embargo, aquí se aumenta un parámetro mas que corresponde al estado que se desea escribir en esa posición de la memoria.

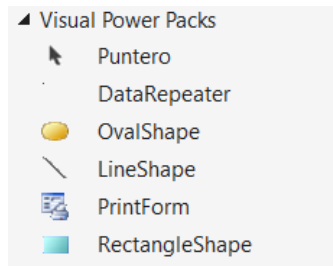


Figura 2.45: Librería Power Packs.

Insertar geometrías básicas en un Formulario

Colocar figuras geométricas básicas como círculos, líneas, rectángulos no resulta fácil en Visual Studio aunque esto parezca indispensable, la solución es insertar en el cuadro de herramientas la librería Visual Power Packs.

Esta librería es necesaria para representar luces piloto con círculos, también es de utilidad para dar una mejor presentación al diseño HMI para incluir líneas divisoras y secciones como los rectángulos como se puede apreciar en la figura 2.45.

La descarga de esta librería se puede hacer desde el sitio web oficial "**visual basic power packs download**"

Symbol Factory

Symbol Factory es una librería muy renombrada utilizada en varios Software de pago como Intouch y Wincc. Es básicamente un conjunto de componentes gráficos de equipamiento industrial como motores, transportadores, tanques, bombas etc como lo muestra en la figura 2.46.

Programación en Hilo

Un ciclo normal de un programa se ejecuta desde un inicio hasta un final, a esta ejecución se le conoce como Hilo Principal, esto se aprecia de mejor manera en una aplicación de consola que se puede observar en la figura 2.47.

Los hilos en C Sharp son el nombre que toman los procesos, para lograr comunicar varios PLCs al HMI y para crear varias pantallas en el mismo HMI se requiere que varios procesos funcionen a la vez, entonces cuando un programa tiene varios Hilos quiere decir que varios procesos se ejecutan en forma paralela y simultanea a la vez.

Así mismo a los Hilos secundarios se los puede mantener de manera cíclica en un laso infinito usando la instrucción While. Con esta técnica es posible

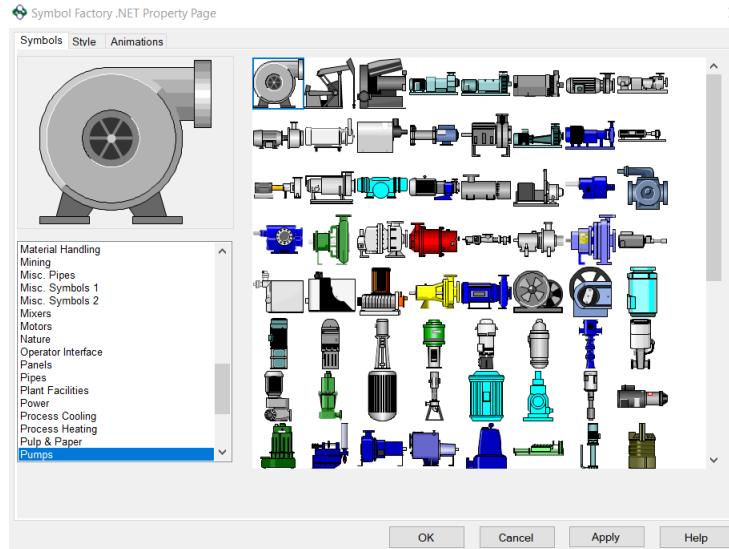


Figura 2.46: Plataforma Symbol Factory.

```

// Funcion para trabajo continuo y en forma paralela Hilo
1 referencia
public void iniciar()
{
    //Crear el hilo
    Thread th = new Thread(ciclo);
    proceso = true;

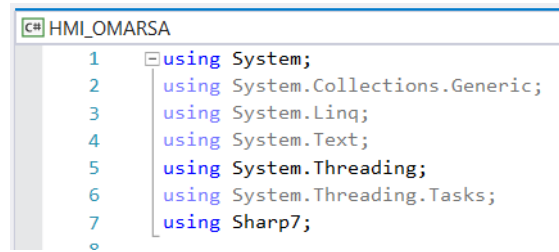
    // arrancar el hilo
    th.Start();
}

// Funcion para detener el hilo
1 referencia
public void detener()
{
    proceso = false;
}
}

```

Figura 2.47: Ejecución en Hilo.

tener varios PLCs comunicándose a la vez, cada PLC sera un hilo con una instrucción While.



```
C# HMI_OMARSA
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading;
6  using System.Threading.Tasks;
7  using Sharp7;
```

Figura 2.48: Librería Threading.

Para ejecutar una programación en hilo es necesario llamar a la librería `System.Threading` que es la que permite usar a la instrucción `Thread` la cual en si crea un hilo como se muestra en la figura 2.48.

2.2.4. IOT Industria 4.0

Los términos Internet de las Cosas, Internet of Things o IoT hacen referencia a una red industrial para poder digitalizarlo a través de puertas de enlaces o pasarelas.

Que se consigue con IOT en la Industria?

Implementando IOT en la industria se puede lograr;

- Control en la producción, ser más acertado ya que se tiene el conocimiento del estado de las variables en tiempo real.
- Mayor productividad y eficacia.
- Ahorro en costos, por predicciones de averías.
- Reducir los tiempos muertos incurridos.
- Mejorar la trazabilidad al tener una base de datos que permite visualizar una tendencia en la curva.

Evolución en el Mercado

Internet ha crecido rápidamente a lo largo del tiempo y esto ha hecho que IOT sea una realidad, no solo una visión del futuro.

La creación de imágenes con una buena combinación de colores y el manejo de datos a través de la nube hace muy vistoso este tipo de aplicaciones.

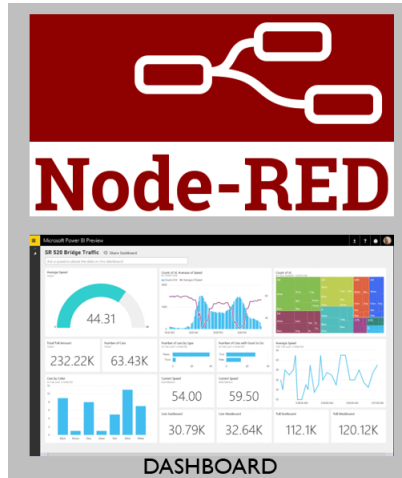


Figura 2.49: Icono Node-Red



Figura 2.50: Icono NodeJs.

Node-Red en la industria

Node-Red [2.48](#) es una plataforma de programación en diversos lenguajes que da la facilidad para interconectar dispositivos de hardware, API y servicios en línea de maneras nuevas y emocionantes, también se puede considerar como una herramienta de visualización para Internet de las cosas.

Node-Red [2.49](#) no quita la necesidad de programar por código nativo pero lo reduce. Node.js es una plataforma de tiempo de ejecución para plasmar un código en Javascript fuera de un navegador. En la figura [2.50](#) se ve su ícono.

Nodos

La comunidad de Node-Red facilita miles de nodos para usarlos en todo

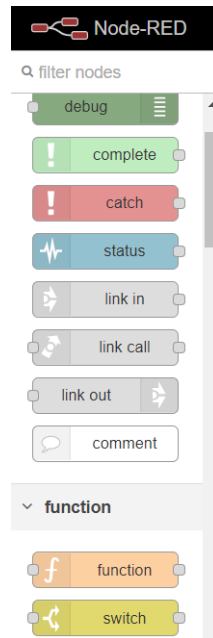


Figura 2.51: Nodos comunidad Node-Red

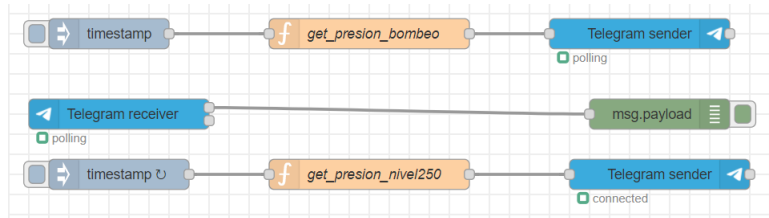


Figura 2.52: Nodos para envío de mensajes instantáneos.

tipo de proyectos o aplicaciones como lo muestra la figura 2.51.

Nodos de sistema

Estos nodos son ideales para el envío de mensajes, alertas, ubicación, etc como se muestra en la figura 2.52.

En este manuscrito se utiliza la plataforma telegram para el envío de alertas, basándose en otros proyectos como el de un Chatbot en Telegram para consultas de casos COVID-19 en el Ecuador, y que permite obtener



Figura 2.53: Aplicaciones en la nube.[826]

datos sobre el número de casos de contagio, recuperados, fallecidos y descartados, y que almacenen las consultas realizadas en una base de datos. Se crea nodos en Telegram para recibir y enviar los resultados y sacar una estadística de Covid-19 en en el país. [Prieto-Reinoso and Cabrera-Mejía, 2020].

Aplicaciones de Node-Red

Entre las principales aplicaciones esta el envío de datos a la nube como se nota en la figura 2.53 en el caso de este proyecto se envían datos a través de la plataforma Ubidots.

Cuenta también con nodos para realizar almacenamiento de información, en este proyecto se utiliza la base de datos phyMyAdmin que en [Carbonell, 2018] brinda mas información al respecto.

Capítulo 3

Marco Metodológico

En este capítulo se explica detalladamente diferentes metodologías que conlleva el automatismo de una planta industrial en este caso una planta dedicada al sector camaronero.

En una empacadora de camarón hay variables críticas como el agua y la temperatura que deben ser monitoreadas constantemente para evitar pérdidas en la producción.

La metodología empleada en este proyecto consta de 8 fases detallada a continuación;

- Fase 1: Migración de paneles eléctricos mas relevantes instalados en zonas conflictivas o de entornos hostiles a lugares secos, centralizados y de fácil acceso y cumpliendo con las normas de seguridad.
- Fase 2: Comunicaciones S7, configuración en la PC.
- Fase 3: Desarrollo de una plataforma HMI Web utilizando los servidores HTML embebidos en los PLCs Simatic.
- Fase 4: Desarrollo de un HMI industrial de escritorio utilizando Visual Studio que reemplaza a un Scada profesional pero que brinda ventajas superiores como cero costos de licencias, ilimitado numero de tags para monitoreo, ilimitados usuarios para usar el servicio, lo único que se requiere para ser monitoreado desde cualquier parte del mundo es una conexión a internet.
- Fase 5: Se Integra las variables deL PLC Master a una plataforma virtual como lo es Node-Red que su entorno visual permite visualizar las variables en Dashboards.

- Fase 6: Utilizando Node-Red se configura alarmas y alertas en mensajería instantánea multimedia a través de la plataforma Telegram a diferentes usuarios con el objetivo de facilitar el control de variables como presiones, niveles, temperaturas.
- Fase 7: Utilizando Node-Red se enlaza con una base de datos phpMyadmin y se registran valores como temperaturas, niveles para analizarlos en forma de indicadores de producción.
- Fase 8: Utilizando Node-Red se envía datos a la nube a través de la plataforma UBIDOTS, esta información puede ser vista en cualquier dispositivo móvil, celulares, tablets, únicamente transcribiendo un link en la web.

3.1. Desarrollo del Proyecto

3.1.1. Fase-1 Migración de Paneles Eléctricos.

Para el desarrollo del proyecto fue necesario realizar la migración de ciertos paneles antiguos que no cumplían con las normas y con la tecnología actual y de los cuales se va dar una breve descripción de su funcionamiento y de las variables que se controlan dentro de cada subproceso.

Planta de agua

Antes del desarrollo del proyecto se tenía varios inconvenientes con la distribución del agua en la planta, se implementó una nueva estación de bombeo pero uno de los inconvenientes es la distancia que queda entre la planta potabilizadora, la cisterna de almacenamiento y la estación de bombeo. Realizar una comunicación entre la potabilizadora y la estación de bombeo sería muy costoso al llevar señales cableadas ya que la distancia bordea los 300 metros.

Aprovechando la configuración de redes tipo árbol que tiene el departamento de infraestructura se solicita una VLAN exclusiva para el departamento de mantenimiento el segmento 172.25.144.XXX. La configuración de los dispositivos se muestra en la figura 3.1.

Para el automatismo de la planta potabilizadora se realiza conexiones Get-Put entre varios PLCs con el objetivo de controlar el encendido y apagado de las bombas según la marea, monitoreo de niveles, presiones, etc, en la figura 3.2 se muestra las pantallas del sistema de bombeo, en la figura 3.3 se detalla las pantallas de la planta potabilizadora, en la figura 3.4 se aprecia las pantallas de la planta de tratamientos de aguas residuales.

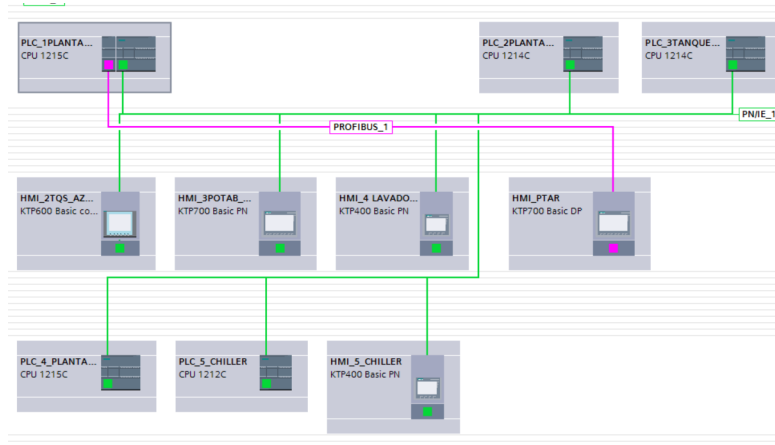


Figura 3.1: Configuración de dispositivos Planta Agua.

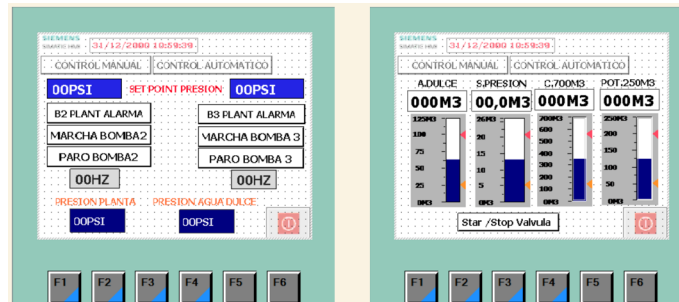


Figura 3.2: Pantallas sistema de bombeo.

Para realizar el automatismo en la planta de agua se implementó nuevos paneles eléctricos y se sustituyó otros que estaban obsoletos, en la figura 3.5 se muestra el panel eléctrico del sistema de bombeo, en la figura 3.6 se observa el reemplazo del panel eléctrico de la Ptar.

Tornado 1-2

Dentro del procesos de congelación se procede con la migración de los paneles de los equipos de congelación Tornado 1-2, se cambian de ubicación a una zona seca y centralizada y mediante profinet se envía señales booleanas y analógicas al PLC Máster. Esto minimiza considerablemente las fallas por cortocircuito, protege los elementos de control, facilita la maniobra al

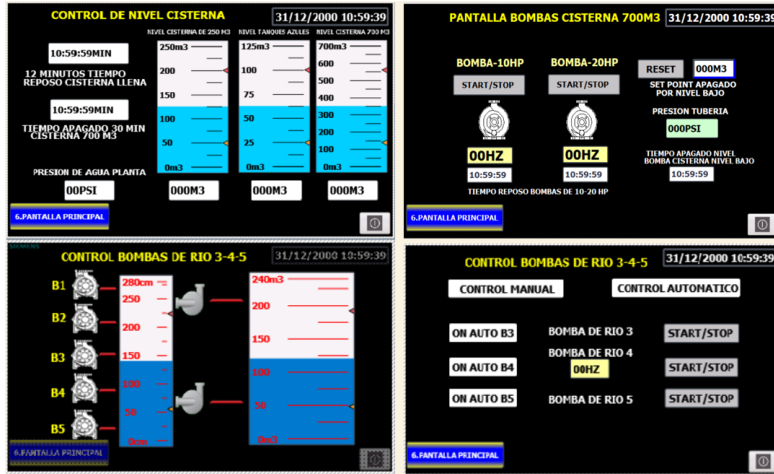


Figura 3.3: Pantallas planta de agua.

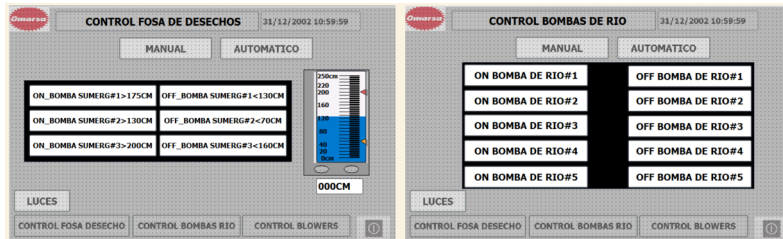


Figura 3.4: Pantallas Ptar.

operador de mandos manuales que tenían los paneles antiguos a llevar el control en un HMI, donde aparte cuenta con una ventana de alarmas que le facilita la detección de averías. La migración de los paneles eléctricos se detalla en la figura 3.7 y en la figura 3.8

La configuración de los dispositivos del proceso de congelación en Tia Portal se observa en la figura 3.9.

En el proceso de tornado 1-2 se realiza el montaje de una pantalla táctil que reemplaza a los botones, da un control de alarmas al operador y facilidad en el seteo de parámetros como se observa en la figura 3.10.

Palinox 2

Otro de los procesos críticos es el túnel de secado por salmuera y por



Figura 3.5: Pantallas sistema de bombeo.

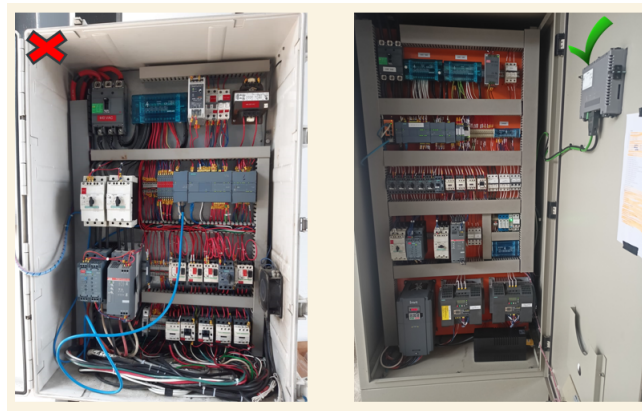


Figura 3.6: Pantallas sistema Ptar.

ventilación donde antes de la migración del panel eléctrico no se podía setear las velocidades de las bandas ni controlar de forma adecuada las temperaturas del proceso, ni controlar los tiempos de agitación de la

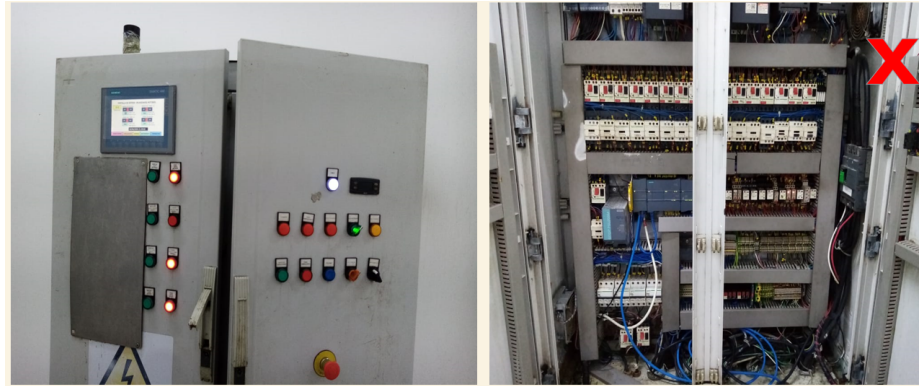


Figura 3.7: Paneles eléctricos obsoletos Tornado 1-2.

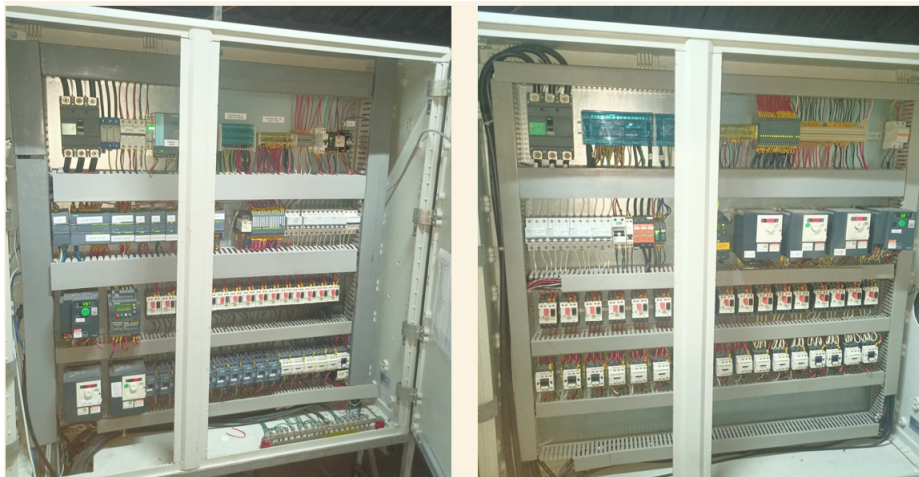


Figura 3.8: Paneles eléctricos actuales Tornado1-2.

salmuera.

Se procede a realizar la migración de todos los componentes eléctricos de control y de fuerza en un panel centralizado en una zona seca libre de humedad y de manipulación por parte de personal no autorizado.

Se procede a realizar una receta para reposo y trabajo en la agitación de la salmuera, se realiza un ahorro energético en la temperatura de la Palinox salmuera. Seteado en trabajo a -18 grados y en Reposo a -10grados centígrados.

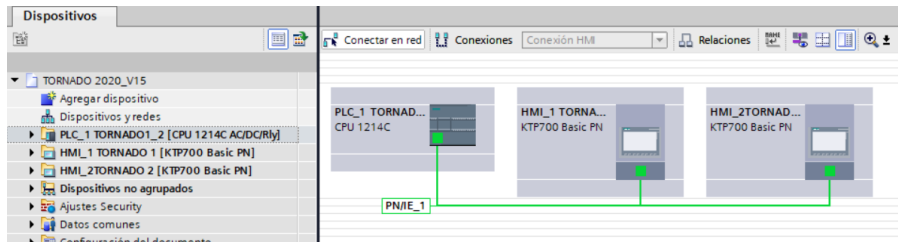


Figura 3.9: Configuración dispositivos tornados.

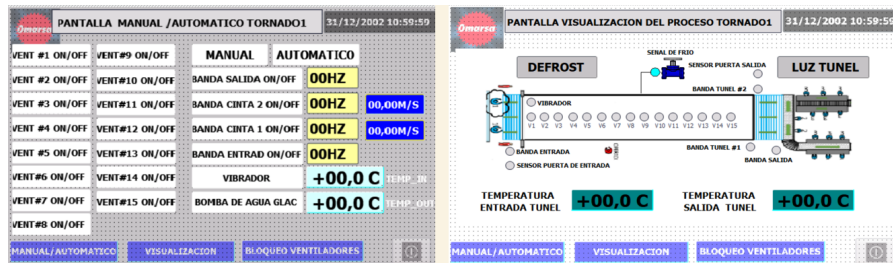


Figura 3.10: Pantallas Tornados.

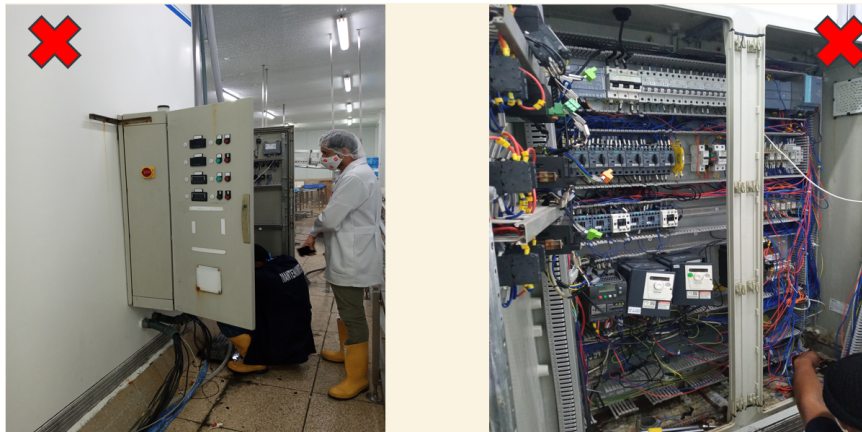


Figura 3.11: Panel eléctrico obsoleto Palinox-2.

En la figura 3.11 se observa el panel antiguo y en la figura 3.12 se muestra el panel eléctrico actual.



Figura 3.12: Paneles eléctricos actuales Palinox-2.

Sistema Dióxido de Cloro

Debido a la pandemia del Coronavirus desde el 2020 en Ecuador y pese a las restricciones por parte del gobierno Chino en las compras de camarón ecuatoriano y tras encontrar coronavirus en el embalaje, la Gerencia General de Omarsa solicita al departamento de mantenimiento montar un sistema de desinfección de dióxido de cloro al material de empaque y embalaje que sale al exterior.

Para ello se diseña un sistema de desinfección en línea que va desde el almacenamiento, la preparación, y la dosificación final al producto.

Dentro del diseño esta la sensorica empleada desde los controles de nivel hasta el on-off para la dosificación.

Esto como etapa inicial la implementación que se muestra en la figura [3.13](#).

Sistema centralizado Área Galpón 5

Una de las áreas fundamentales dentro de la compañía es Valor Agregado en el sector pelado Galpón 5, aquí hay varios subprocesos que conllevan una gran cantidad de variables a controlar, por lo que es otro sector en donde se integró a la red un panel eléctrico centralizado y que se controla procesos como máquinas de pelado, mesas de pelado y desvenado, cocederos, extractores.

Anterior al automatismo el sistema presentaba varios problemas como cortocircuitos en los equipos electrónicos debido a la humedad, falta de detección de fallos por parte de los eléctricos de planta, fallas inesperadas en la producción .

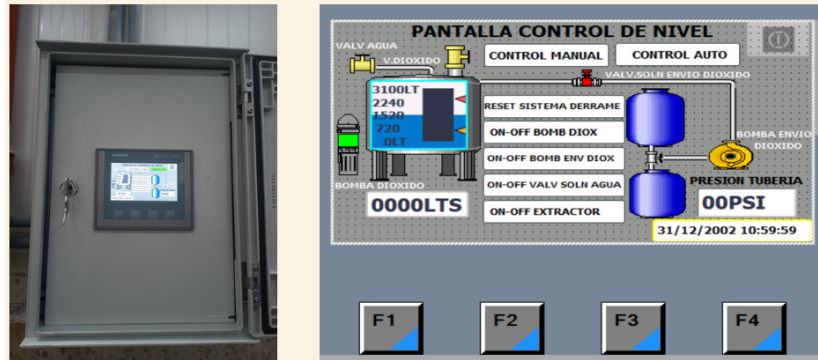


Figura 3.13: Panel Eléctrico Dióxido de Cloro.

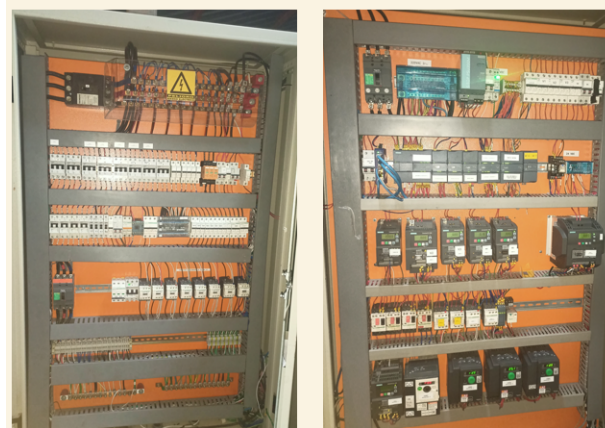


Figura 3.14: Paneles eléctricos Galpón 5.

Se instala un panel centralizado en una zona seca y de fácil acceso utilizando tecnología actual como muestra en la figura 3.14.

La configuración de dispositivos de este panel centralizado se muestra en la figura 3.15.

Una parte de las pantallas creadas en el área de Galpón 5 se muestra en la figura 3.16 donde se puede controlar y setear variables como temperatura, control de velocidad de bandas, control y monitoreo de alarmas etc.

Panel eléctrico Túneles de congelación, Sistema de Dióxido

El camarón que sale de las clasificadoras se empaquetan y se los traslada

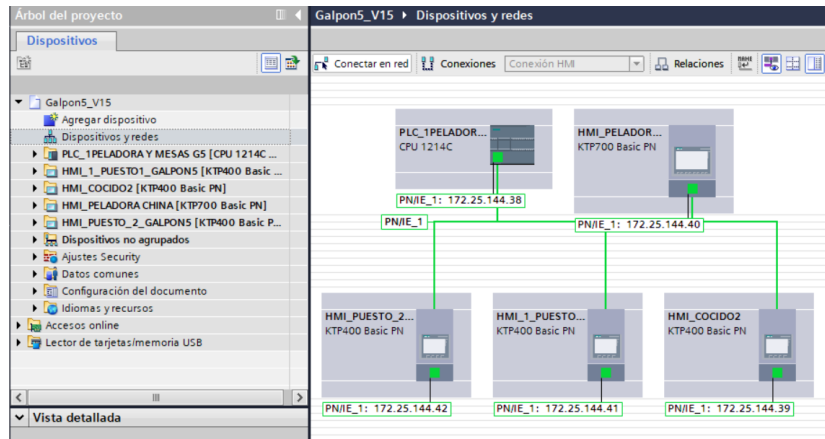


Figura 3.15: Configuración de dispositivos Pelado.

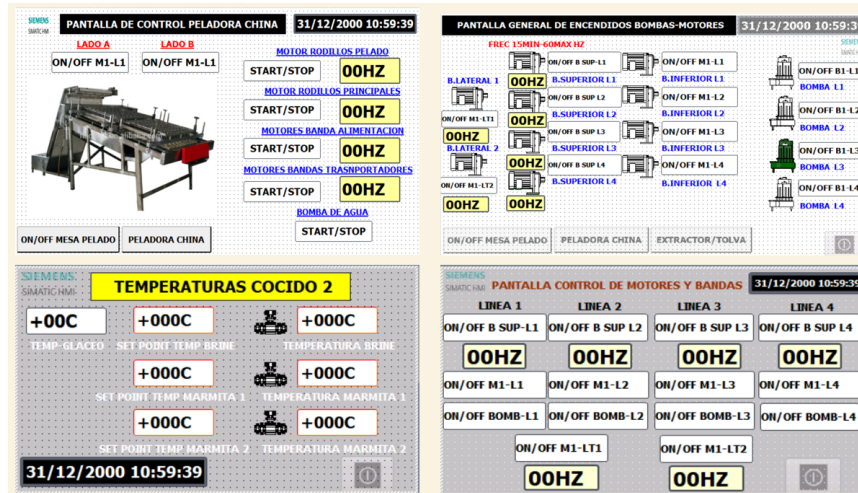


Figura 3.16: Pantallas Área Galpón 5.

en coches que tiene que pasar por la etapa de congelación en los túneles de frío , anteriormente el operador ingresaba cada cierto tiempo con un termómetro a medir la temperatura de las cajetas, viendo esta necesidad se procede a instalar sensores de temperatura en cada uno de los túneles y llevar estas señales a un PLC con su respectiva pantalla táctil para el control y monitoreo.

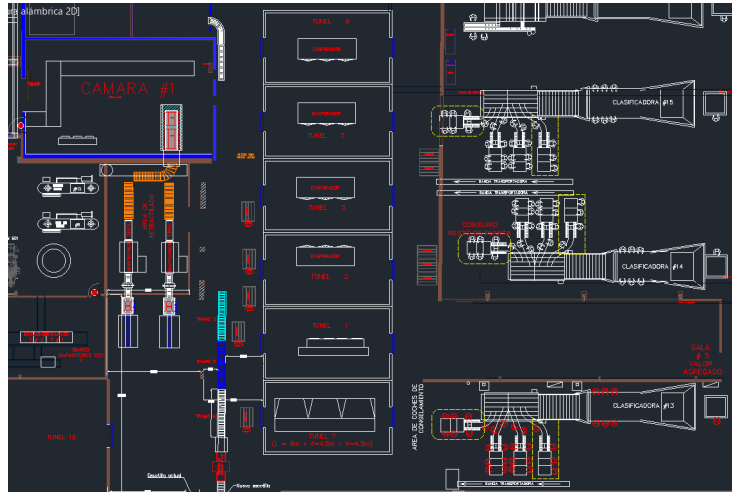


Figura 3.17: Plano de ubicación Túneles de congelación.



Figura 3.18: Tablero Túneles de congelación.

En la figura 3.17 se observa la ubicación de los túneles de congelación En la figura 3.18 se observa el Panel eléctrico de implementación.

En la figura 3.19 se detalla la configuración de los dispositivos integrados a la red de PLCs. Adicional se instala dispositivos de campo como sensores, válvulas para el control de desinfección de cajas en el área de planta general. En la figura 3.20 se muestra parte de las pantallas del proceso.

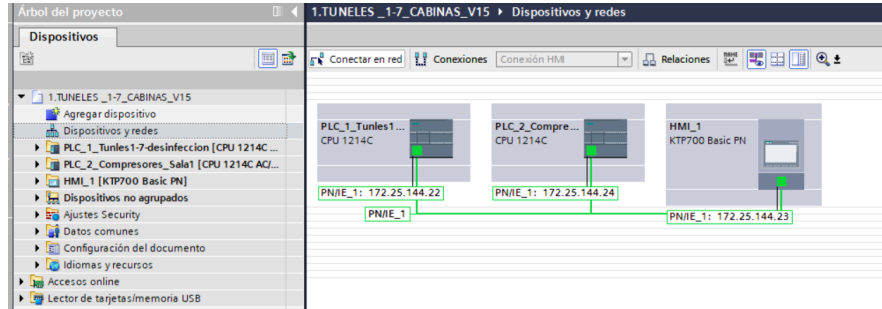


Figura 3.19: Configuración de dispositivos túneles y compresores.

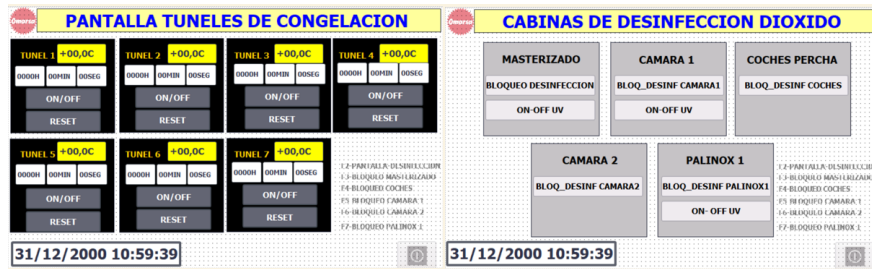


Figura 3.20: Pantallas túneles de congelación.

3.1.2. Fase-2 Fase 2: Comunicaciones S7, configuración en la PC.

El desarrollo de la comunicación S7 mediante el software de Siemens TIA PORTAL V15, se realizó enlaces PUT y GET, los cuales envían datos entre los Plcs esclavos y el Plc Master.

Para establecer comunicación lo primero es configurar la PC, como en la figura 3.21.

- Seleccionar Internet Protocolo Versión 4 (TCP/IPv4)
- Clic en Propiedades
- Configurar la Dirección IP y la Máscara subred
- Aceptar

Dentro del árbol del proyecto de Tia Portal se agrega una CPU 1511-1PN que dentro de sus características mas relevantes se destacan:

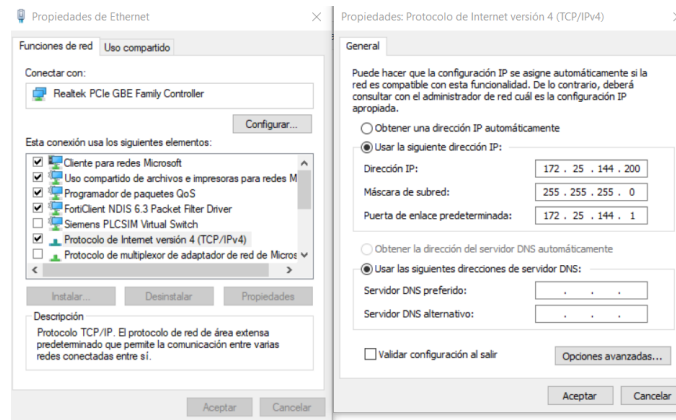


Figura 3.21: Configuración direcciones IP.

- Rack: CPU+31 Módulos
- Rack: 4 CM/CP(Profibus, profinet)
- Numero de elementos: total 2000 bloques(OB/FB/FC/DB) y UDT.
- Número de puertos: 2
- Número de conexiones S7 routing: 16
- Lenguajes de programación: KOP,FUP,AWL,SCL,GRAPH.

En este proyecto se utiliza la CPU 1511 como PLC maestro donde todas las señales de los PLCs esclavos se enlazan por medio del bloque PUT-GET a bloques FC creados dentro del árbol del PLC 1511, como se observa en la figura 3.22.

Es importante que todos los dispositivos se ejecuten en una mismo segmento de red como se muestra en la figura 3.23.

A continuación se muestran los pasos a seguir para una conexión S7-Get Put, entre PLCs de proyectos diferentes.

En este caso se hará la comunicación Get-Put entre el PLC Master S71500 y un PLC esclavo S7-1200 del Sistema de Bombeo, la interconexión con los demás PLCs es semejante.

Paso 1:

Permitir la comunicación Put/Get como se muestra en la figura 3.24.

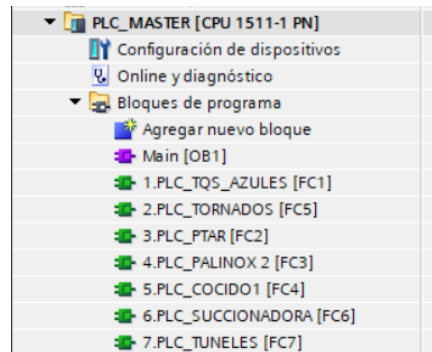


Figura 3.22: Subprocesos bloques FC.

- Seleccionar el PLC
- Propiedades
- General
- Mecanismos de conexión
- Habilitar la opción de Permitir el acceso vía comunicación PUT/GET del interlocutor remoto
- (Realizar este procedimiento a los dos dispositivos)
- NOTA: En versiones antiguas de firmware esta opción ya se encuentra activada

Paso 2:

Se activa las marcas de reloj como lo muestra en la figura [3.25](#).

- Seleccionar el dispositivo
- Propiedades
- General
- Marcas de sistema y de ciclo
- Activar
- (Realizar este procedimiento a los dos dispositivos)

DESCRIPCION	DIRECCION	MASCARA	SUBRED
PLC MAESTRO_1500	172.25.144.2	255.255.255.0	172.25.144.1
ESPIRAL 1PLC	172.25.144.20	255.255.255.0	172.25.144.1
ESPIRAL 1HMI	172.25.144.21	255.255.255.0	172.25.144.2
TUNELES 1-7PLC	172.25.144.22	255.255.255.0	172.25.144.3
TUNELES 1-7HMI	172.25.144.23	255.255.255.0	172.25.144.4
PLC COMPRESORES VILTER	172.25.144.24	255.255.255.0	172.25.144.5
PLC TORNADO 1-2	172.25.144.26	255.255.255.0	172.25.144.6
HMI TORNADO 1	172.25.144.27	255.255.255.0	172.25.144.7
HMI TORNADO 2	172.25.144.28	255.255.255.0	172.25.144.8
PLC_G6	172.25.144.12	255.255.255.0	172.25.144.1
PANTALLA DIOXIDO	172.25.144.13	255.255.255.0	172.25.144.1
PANTALLA PALINOX	172.25.144.14	255.255.255.0	172.25.144.1
PANTALLA CAMARA6	172.25.144.15	255.255.255.0	172.25.144.1
PLC RESIDUAL	172.25.144.16	255.255.255.0	172.25.144.1
HMI RESIDUAL	172.25.144.29	255.255.255.0	172.25.144.1
PLC PLANTA AGUA	172.25.144.30	255.255.255.0	172.25.144.1
HMI PLANTA DE AGUA	172.25.144.31	255.255.255.0	172.25.144.1
PLC TANQUES AZULES	172.25.144.34	255.255.255.0	172.25.144.1
HMI TANQUES AZULES	172.25.144.35	255.255.255.0	172.25.144.1
HMI LAVADORA GAVETAS	172.25.144.25	255.255.255.0	172.25.144.1
PLC_CHILLER	172.25.144.3	255.255.255.0	172.25.144.1
HMI_CHILLER	172.25.144.4	255.255.255.0	172.25.144.1
PLC_PLANTA2_POTABILIZADORA	172.25.144.74	255.255.255.0	172.25.144.1
HMI_PLANTA2_POTABILIZADORA	172.25.144.75	255.255.255.0	172.25.144.1
PLC SUCCIONADORA	172.25.144.32	255.255.255.0	172.25.144.1
HMI SUCCIONADORA	172.25.144.33	255.255.255.0	172.25.144.1
PLC CLASIF 1,2,3	172.25.144.36	255.255.255.0	172.25.144.1
PANTALLA DESCABEZADO	172.25.144.37	255.255.255.0	172.25.144.1
HMI_DISPENSADORES DE JABON	172.25.144.69	255.255.255.0	172.25.144.1
PLC_VALOR_AGREGADO	172.25.144.70	255.255.255.0	172.25.144.1
HMI_SUPERIOR_V.A	172.25.144.71	255.255.255.0	172.25.144.1
HMI_PUESTO1_V.A	172.25.144.72	255.255.255.0	172.25.144.1
HMI_PUESTO2_V.A	172.25.144.73	255.255.255.0	172.25.144.1
PLC GALPON 5	172.25.144.38	255.255.255.0	172.25.144.1
HMI COCIDO 2	172.25.144.39	255.255.255.0	172.25.144.1
HMI PELADORA CHINA	172.25.144.40	255.255.255.0	172.25.144.1
HMI PUESTO 1	172.25.144.41	255.255.255.0	172.25.144.1
HMI PUESTO 2	172.25.144.42	255.255.255.0	172.25.144.1
ESPIRAL 2 PLC	172.25.144.43	255.255.255.0	172.25.144.1
ESPIRAL 2 HMI	172.25.144.44	255.255.255.0	172.25.144.1
PLC PALINOX 1SALM	172.25.144.45	255.255.255.0	172.25.144.1
HMI PALINOX 1SALM	172.25.144.46	255.255.255.0	172.25.144.1
PLC SECADO PLNX1	172.25.144.47	255.255.255.0	172.25.144.1
HMI SECADO PLNX1	172.25.144.48	255.255.255.0	172.25.144.1
PLC TUNEL CHINO	172.25.144.49	255.255.255.0	172.25.144.1
HMI TUNEL CHINO	172.25.144.50	255.255.255.0	172.25.144.1
HMI TUMBLER	172.25.144.51	255.255.255.0	172.25.144.1
PLC COCIDO 1	172.25.144.52	255.255.255.0	172.25.144.1
HMI COCIDO 1	172.25.144.53	255.255.255.0	172.25.144.1
Nitrogeno t3 level	172.25.144.76	255.255.255.0	172.25.144.1
Nitrogeno t2 level	172.25.144.76	255.255.255.0	172.25.144.1
Logo siemens Web	172.25.144.81	255.255.255.0	172.25.144.1

Figura 3.23: Direcciones IP dispositivos Omarsa.

Paso 3:

Para establecer la conexión S7 como lo muestra la figura 3.26 se realiza el siguiente procedimiento.

- Seleccionar Conexiones
- El tipo es Conexión S7

Paso 4:

Es necesario agregar un nuevo bloque DB como lo muestra en la figura 3.27 en ambos PLCs, y crear un mapa de variables para enviar y recibir datos, adicional se debe quitar el acceso optimizado al bloque,este procedimiento se debe hacer en los dos PLCs.

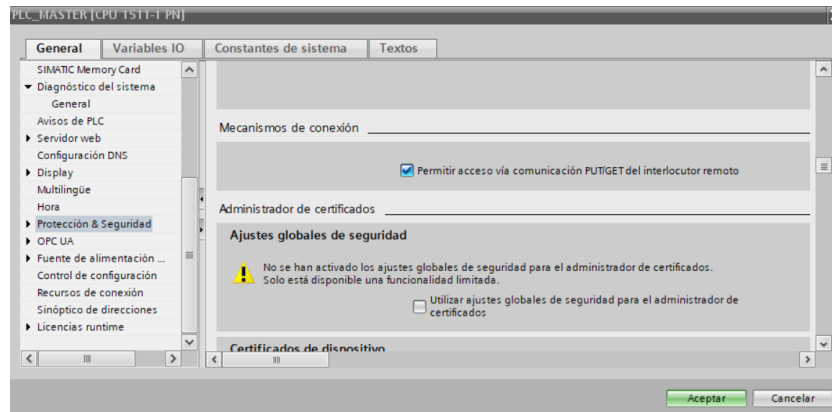


Figura 3.24: Permitir comunicación Put/Get.

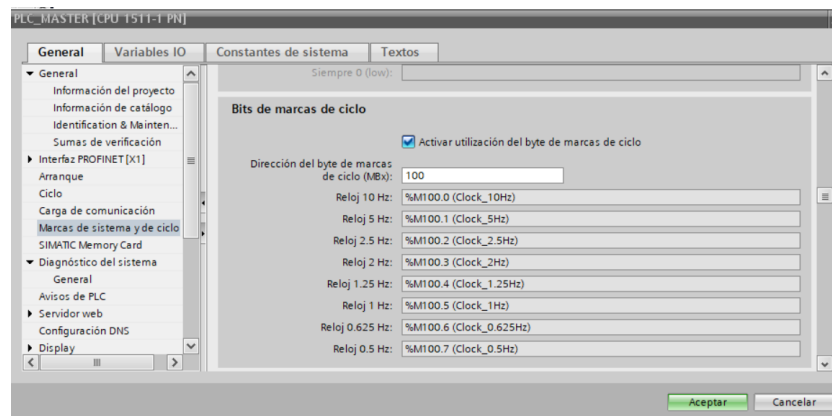


Figura 3.25: Activar marcas de ciclo.

En la configuración de bloques es importante saber la dirección del bloque para este caso el PLC del Sistema de Bombeo lleva la dirección DB51 y para el caso del PLC Máster la dirección es la DB1 como se indica en la figura 3.28

Paso 5:

Buscar en Comunicación S7 los bloques PUT y GET, para agregarlos al área programación.

Como se envía datos del PLC del Sistema de Bombeo se coloca un bloque Put y se lo configura como en la figura 3.29.



Figura 3.26: Configuraciones S7-Simatic.

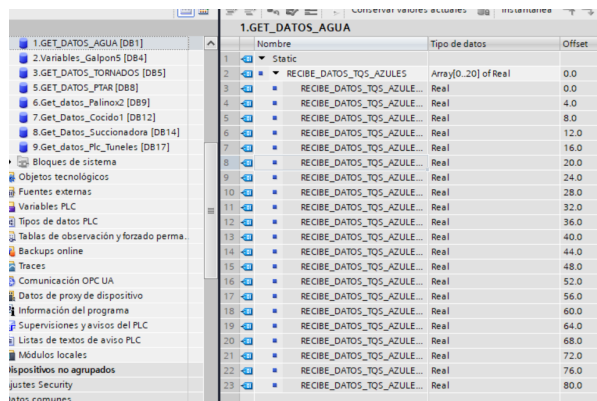


Figura 3.27: Variables Get PLC Máster.

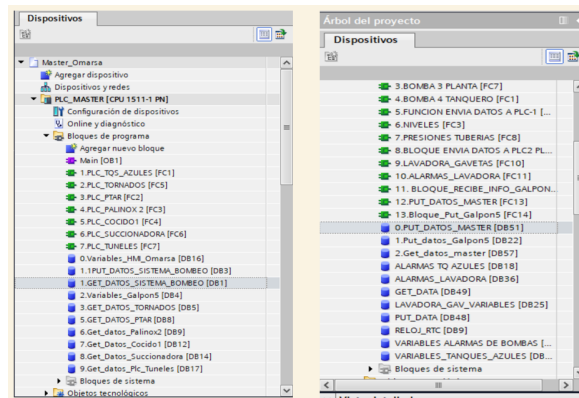


Figura 3.28: Configuración de bloques DB.

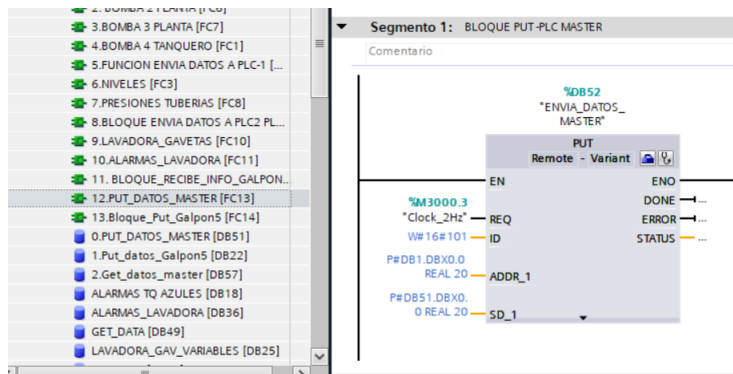


Figura 3.29: Bloque Put PLC Sistema de Bombeo.

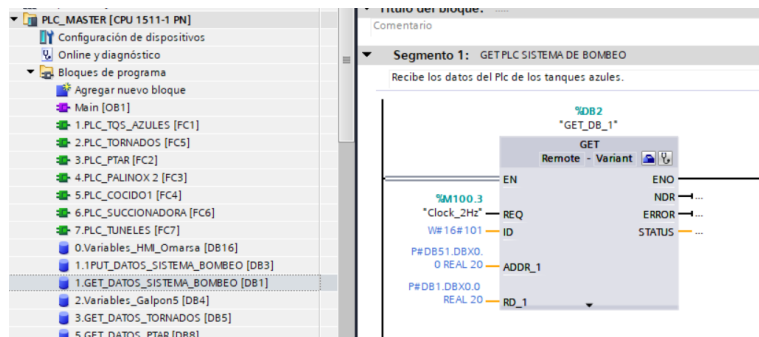


Figura 3.30: Bloque Get PLC Máster.

En el PLC Máster se configura el bloque GET como se muestra en la figura 3.30.

Paso 6:

Para configura el bloque Put es necesario seguir el siguiente procedimiento como lo muestra en la figura 3.31. Y se lo hace desde el PLC esclavo S71200 del Sistema de bombeo.

- REQ: Señal para la toma de información
- ID: Identificador del hardware
- ADDR1: Dirección local
- RD: Dirección remota

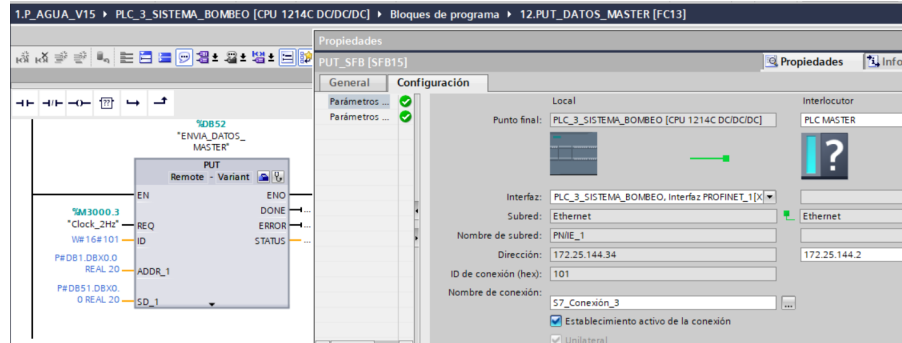


Figura 3.31: Parámetros conexión Put.

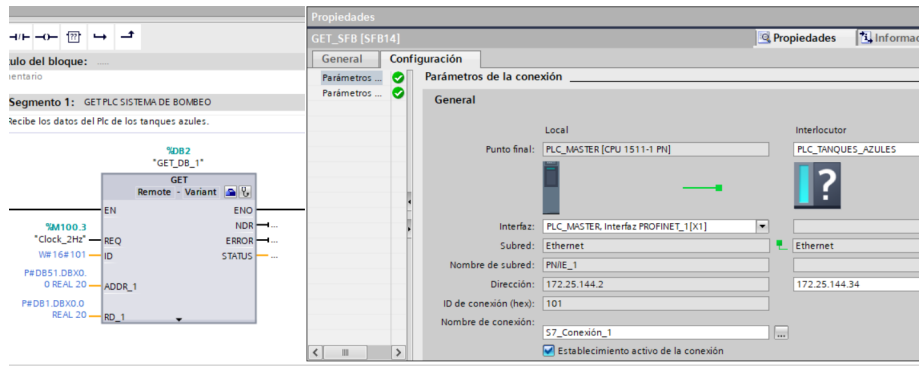


Figura 3.32: Parámetros conexión Get.

- Ir a Parámetros de la conexión
- Asignar una marca de Reloj en REQ
- Presionar el icono para seleccionar le dispositivo
- Seleccionar dispositivo
- Configurar la dirección local en donde se tiene el dato
- Configurar la dirección remota a donde se enviará el dato.

Para configura el bloque Get es necesario seguir el siguiente procedimiento como lo muestra en la figura 3.32. Y se lo hace desde el PLC S7-1500 Máster.



Figura 3.33: Traslado de datos entre PLCs.

- Señal para la toma de información.
- ID: Identificador del hardware.
- ADDR1: Dirección local.
- RD: Dirección remota.
- Ir a Parámetros de la conexión.
- Asignar una marca de Reloj en REQ.
- Presionar el icono para seleccionar le dispositivo.
- Seleccionar dispositivo.
- Configurar la dirección local en almacenará dato.
- Configurar la dirección remota de donde proviene el dato.

Paso 7:

Se agrega el programa en el PLC S71500 y S71200 y se realiza pruebas de conexión como lo muestra en la figura 3.33.

3.1.3. Fase-3 Desarrollo y aplicación HMI WEB

Otras de las técnicas realizadas en este proyecto es la creación de HMI Web, los lenguajes nativos para este desarrollo son HTML,CSS,JAVA SCRIPT.

La creación de códigos se puede realizar en: Sublime text o Visual Studio Code, en este proyecto se utilizó Sublime text.

En el diseño gráfico se utilizar: Smart Draw, Photoshop 6 (utilizado para quitar fondos).

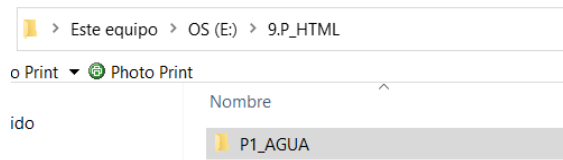


Figura 3.34: Carpetas de trabajo.

En Tia Portal se puede utilizar los siguientes lenguajes de programación: escalera, bloques, instrucciones, graficet, texto estructurado, AWP, para este proyecto se utilizó lenguaje de escalera.

En las carpetas de trabajo se recomienda utilizar nombres en minúscula, sin caracteres especiales, lo más cercano a la raíz como se muestra en la figura 3.34.

La familia de PICs Simatic que vienen embebidos el servidor WEB SERVER son los PLCs de la serie S7-1200, S71500 y deben tener un firmware igual o superior a la versión 4.0.

Es importante saber que para el desarrollo web se utilizó:

- HTML: para la estructura del proyecto
- CSS: para darle estilo al proyecto
- JAVASCRIPT el lenguaje de programación.

Los navegadores en los cuales se puede correr la aplicación son :

- Google Chrome
- Firefox
- Opera
- Explorer (No utilizar)

El procedimiento que se utilizó para subir una página web al PLC se detalla a continuación.

- Insertar el bloque WWW en Tia Portal como se muestra en la figura 3.37.
- Habilitación y configuración del servidor del PLC, como se muestra en la figura 3.36.

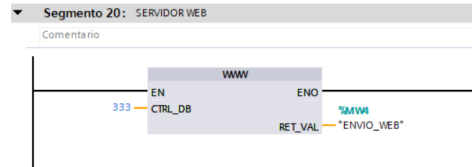


Figura 3.35: Bloque de envío de datos Web Tia Portal.

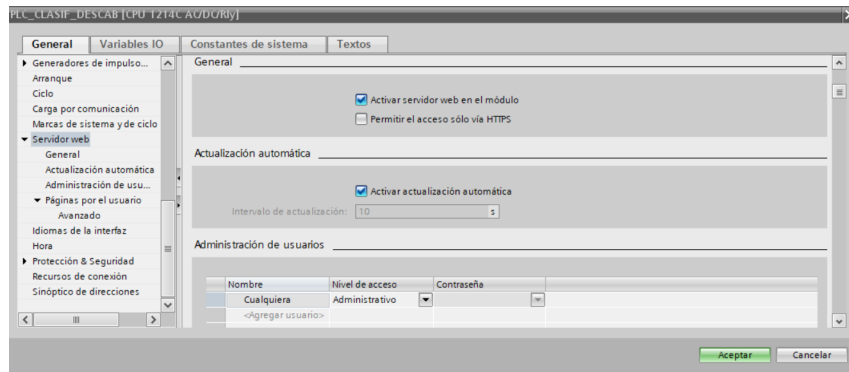


Figura 3.36: Páginas de Usuarios HTML.

- Crear los usuarios (Administrador) como se muestra en la figura ??.
- Indicar la ruta de la carpeta.

Dentro del proceso de empacado se realizó un proyecto de ahorro de agua en los equipos de laboratorio kjeldahl, los cuales antes de la mejora consumían agua potable de tanquero sin renovación, y el agua era desperdiciada en una cantidad considerable.

El promedio de consumo anterior de estos equipos era de 50 m³ diarios ya que necesitaban una toma constante de agua que se utiliza como enfriamiento.

Como mejora se instaló un sistema de recirculación de agua en circuito cerrado y pasando la tubería por un túnel de congelación para que el agua reduzca su temperatura en un promedio de 16 a 20 grados centígrados como se muestra en la figura 3.38.

Se realiza un cuadro comparativo de retorno de inversión de los materiales utilizados para el automatismo y se llega a la conclusión que el retorno se lo recupera en 2.35 meses como se muestra en la figura 3.39.

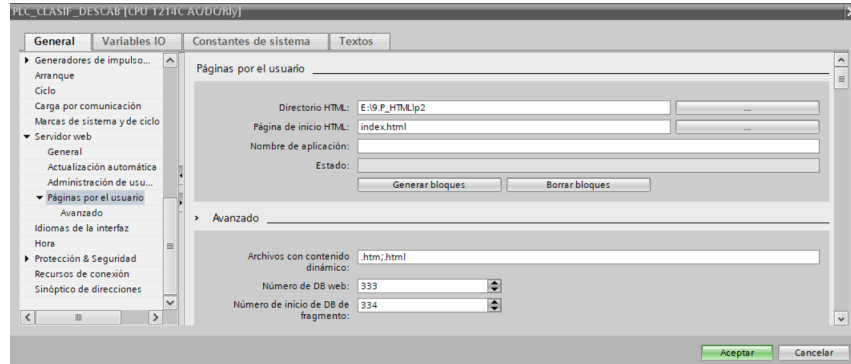


Figura 3.37: Servidor Web administrar usuarios.

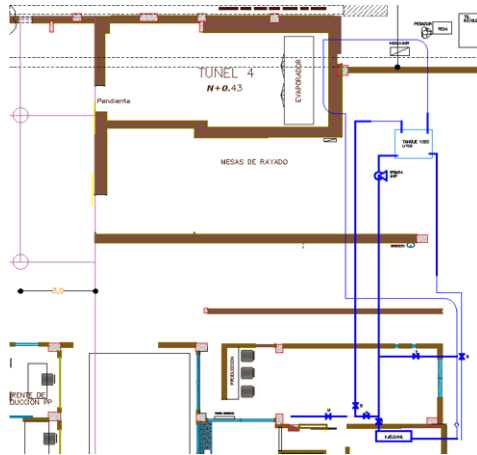


Figura 3.38: Diseño recirculación de agua.

En la figura 3.40 se muestra la implementación en sitio.

El sistema de recirculación está ubicado en un lugar alejado del laboratorio de calidad para ello se genera una pagina web indispensable para el monitoreo del sistema. En donde el personal del área va a tener una herramienta para verificación del nivel y temperatura del tanque. Los pasos a seguir a seguir son los siguientes.

- Se crea una pagina Web en HTML utilizando el editor de códigos Sublime Text como se muestra en las figuras 3.41, 3.42, 3.43 y se carga el código creado en HTML al PLC.

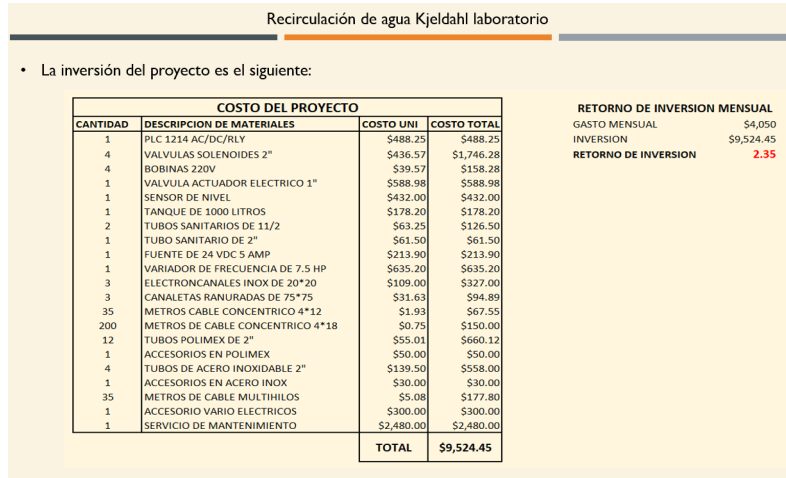


Figura 3.39: Análisis costo beneficio.

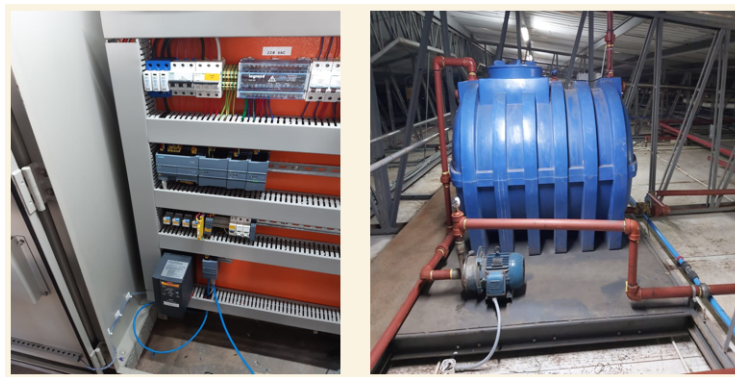


Figura 3.40: Implementación del sistema de recirculación kjeldahl

- En la barra de navegación de internet se coloca la dirección IP del PLC 172.25.144.36, como se muestra en la figura 3.44, y se da clic en páginas de usuario
- Finalmente se tiene el entorno web esperado como se muestra en la figura 3.45.

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Kjendall</title>
5 <link rel="stylesheet" type="text/css" href="main.css">
6 </head>
7 <body class="contenido">
8 <h1>TEMPERATURAS TORNADOS </h1>
9 <br>
10 <br>
11 <center>
12 <table border="1" width="40%">
13
14 <div class="valor1"><h2><label id="TempInT1">0 </label></h2>
15 </div>
16
17 <div class="valor2"><h2><label id="TempOutT1">0 </label></h2>
18 </div>
19 </table>
20 </center>
21 <script src="jquery-2.0.2.min.js"></script>
22
23 <script type="text/javascript">
24 $(document).ready(function(){
25 $.ajaxSetup({ cache: false });
26 setInterval(function() {
27 $.get("es_index.html", function(result){
28 $('#TempInT1').text(result.trim());
29 });
30 },1000);
31
32 setInterval(function() {
33 $.get("es_index.html", function(result){
34 $('#TempOutT1').text(result.trim());
35 });
36 },1000);
37
38
39 });
40 </script>

```

Figura 3.41: Códigos del programa paso 1.

3.1.4. Fase 4: Desarrollo HMI industrial Visual Studio

Una vez realizado los enlaces Put-Get entre los PLCs esclavos y el el PLC Maestro es recomendable elaborar un bloque DB destinado a las variables, para el caso de este manuscrito se trabajó en la DB16 del PLC Máster como se indica en la figura y se controla un aproximado de 357 variables [3.46](#).

El detalle paso a paso del desarrollo de un HMI industrial en Visual Studio se lo puede encontrar en el capítulo 2 en este capítulo se concreta a los resultados finales y la metodología empleada.

Para el desarrollo de un HMI industrial es necesario seguir una secuencia que se detalla a continuación.

Paso 1:

Crear un proyecto en Windows Forms para este caso se llamará HMI OMARSA.

Se crea varias clases o pantallas para los diferentes procesos como se muestra en la figura [3.47](#) y en la figura [3.48](#).

- Nombre del proyecto

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <!-- AWP_In_Variable Name="TEMPERATURA" -->
4  <!-- AWP_In_Variable Name="NIVEL" -->
5  <!-- AWP_In_Variable Name="RESULTADO" -->
6  <!-- AWP_In_Variable Name="FRECUENCIA" -->
7  <!-- AWP_In_Variable Name="VELOCIDAD" -->
8  <!-- AWP_In_Variable Name="SALIDA_ESCALIZADA" -->
9
10
11 <!-------ENCABEZADO----->
12 <head>
13   <meta charset="utf-8">
14   <meta http-equiv="refresh" content = "1">
15   <title>Recirculacion de agua </title>
16
17   <link rel="stylesheet" type="text/css" href="main.css"> <!--
18   Define css -->
19 </head>
20
21 <body class="contenido">
22
23 <center><h1 class="titulo">SISTEMA DE RECIRCULACION</h1></center>
24 <hr>
25
26
27 <hr> <!--define cambio de tematica con una linea transversal-->
28 <br> <!--define salto de linea-->
29
30 <div class="logo">
31   
32 </div>
33
34 <div class="tanque">
35   
36 </div>
37

```

Figura 3.42: Códigos del programa paso 2.

- Clic derecho, Agregar Formulario(Windows Forms)
- Se crea un nuevo formulario con extensión cs que quiere decir que es una clase.

Para este proyecto se ha creado varias clases o pantallas como se muestra el detalle en la figura 3.49.

Paso 2:

Se crea una clase llamada PLCController, esta pantalla es la encargada de controlar el PLC del proyecto en este caso el PLC S7-1500 Máster.

Antes de comenzar con la programación es necesario instalar las librerías de comunicación Sharp7 en la plataforma Administrar paquetes NuGet.

Posterior se crea un objeto PLC, y una variable tipo String para la dirección IP del PLC y se genera una variable para saber si el PLC esta o no conectado como se muestra en la figura 3.50.


```

39 <div class="kendall">
40   
41 </div>
42
43 <center>
44   <table border="1" width= 40%>
45     <!--Define una fila en una tabla-->
46     <tr> <!--INGRESO DE TABLA-->
47       <td> <b>DETALLE DE VARIABLES</b> <!--define texto en negrita
48         -->
49       </td>
50
51       <td>
52         <b>VALOR ACTUAL</b>
53       </td>
54     </tr>
55     <!--Define una fila en una tabla-->
56     <tr>
57       <td>
58         <form> <!--Define un formulario en html-->
59           TEMPERARURA DEL AGUA=
60         </form>
61       <td>:= "TEMPERATURA":</td>
62     </tr>
63     <tr>
64     <td>
65       <form> <!--Define un formulario en html-->
66         NIVEL TANQUE 1000 LITROS=
67       </form>
68     <td>:= "NIVEL":</td>
69     </tr>
70   </tr>
71 </table>
72 </center>
73
74

```

Figura 3.43: Códigos del programa paso 3.

Paso 3:

En la clase Variables se crea todas las variables y deben de ser de tipo publicas para que puedan ser llamados de otros formularios.

Como recomendación es necesario que las variables creadas en Visual Studio tengan el mismo nombre y el tipo que las variables creadas en Tia Portal como se muestra en la figura 3.51.

Paso 4:

Se regresa a la clase PLCController y se realiza la conexión para la,lectura y escritura de las variables.

Y se crea una función para establecer la comunicación con el PLC como se muestra en la figura 3.52.

Paso 5:

El siguiente paso es crear una función para leer las variables del PLC ya sean booleanas, o reales 3.53. El procedimiento es el mismo para la conexión

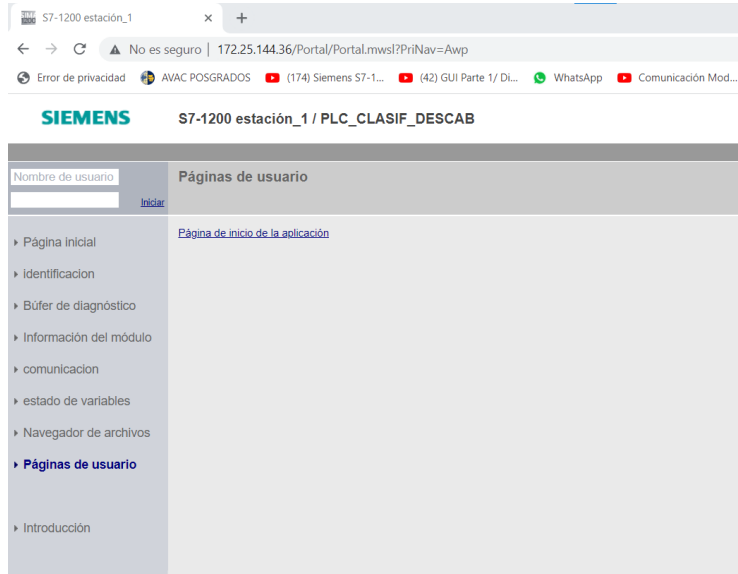


Figura 3.44: Página de usuario estación Siemens



Figura 3.45: HMI Web equipos kjendal.

de todos los procesos, la recomendación es que se trabaje en una DB en este caso la DB16.

Paso 6:

Se crea una función clave para el funcionamiento de uno o varios PLCs o de varias pantallas.

0.Variables_HM_Omarsa			0.Variables_HM_Omarsa		
Nombre	Tipo de datos	Offset	Nombre	Tipo de datos	Offset
1	Static		3	Termico_cinta_2_T1	Bool
2	Presion_bombeo_planta	Real	4	Termico_cinta_1_T1	Bool
3	Presion_agua_dulce_planta	Real	5	Termico_cinta_entrada_T1	Bool
4	Nivel_cisterna_700m3	Real	6	Paro_emergencia_Tornado1	Bool
5	Nivel_tanques_azules	Real	7	Termico_bomba_glacero_T1	Bool
6	Frecuencia_bomba_agua_dulce	Real	8	Sensor_puerta_entrada_T1	Bool
7	Frecuencia_bomba2_planta	Real	9	Sensor_puerta_salida_T1	Bool
8	Frecuencia_bomba3_planta	Real	10	Flotador_nivel_agua_glacero_T1	Bool
9	Termico_bomba_agua_dulce	Bool	11	Termico_cinta_salida_T2	Bool
10	Termico_bomba2_planta	Bool	12	Termico_cinta_2_T2	Bool
11	Termico_bomba3_planta	Bool	13	Termico_cinta_1_T2	Bool
12	Termico_bomba_tanquero	Bool	14	Termico_cinta_entrada_T2	Bool
13	Piloto_B1_Agua_dulce	Bool	15	Paro_emergencia_Tornado2	Bool
14	Piloto_B2_Bombeo	Bool	16	Termico_bomba_glacero_T2	Bool
15	Piloto_B3_Bombeo	Bool	17	Sensor_puerta_entrada_T2	Bool
16	Piloto_B4_Tanquero	Bool	18	Sensor_puerta_salida_T2	Bool
17	Piloto_Valvula_sistema_presion	Bool	19	Flotador_nivel_agua_glacero_T2	Bool
18	Piloto_valvula_floculante	Bool	20	Piloto_cinta_entrada_Tornado1	Bool
19	Alarma_falla_compresor_Potb	Bool	21	Piloto_cinta_entrada_Tornado2	Bool
20	Marca_Manual	Bool	22	Piloto_cinta_metalica_1_Tornado1	Bool
21	Marca_Auto	Bool	23	Piloto_cinta_metalica_1_Tornado2	Bool
22	Nivel_tq_25m3	Real	24	Piloto_cinta_metalica_2_Tornado1	Bool
23	Nivel_fosa_Ptar	Real	25	Piloto_cinta_metalica_2_Tornado2	Bool
24	Nivel_cisterna_250m3	Real	26	Piloto_cinta_salida_Tornado1	Bool
25	Frecuencia_bomba20hp_700m3	Real	27	Piloto_cinta_salida_Tornado2	Bool
26	Frecuencia_bomba10hp_700m3	Real	28	Piloto_bomba_glacero_tornado1	Bool
27	Termico_B1_Potb	Bool	29	Piloto_bomba_glacero_tornado2	Bool
28	Termico_B2_Potb	Bool			
29	Termico_Bomb_Sumrg1	Bool	357	Piloto_senal_frio_tornado1	Bool

Figura 3.46: Variables DB16 PLC Máster.

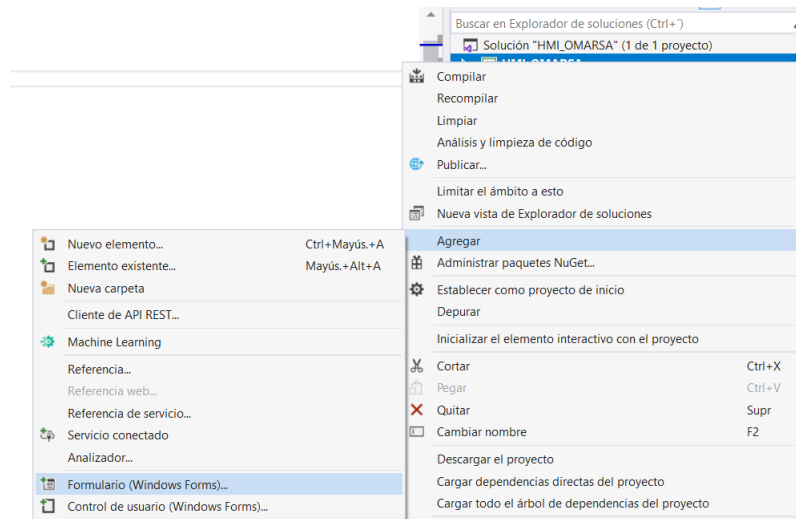


Figura 3.47: Crear una clase paso 1.

Esta función contiene el hilo que permite controlar procesos en forma simultanea y paralela como se indica en la figura 3.54.

Paso 7:

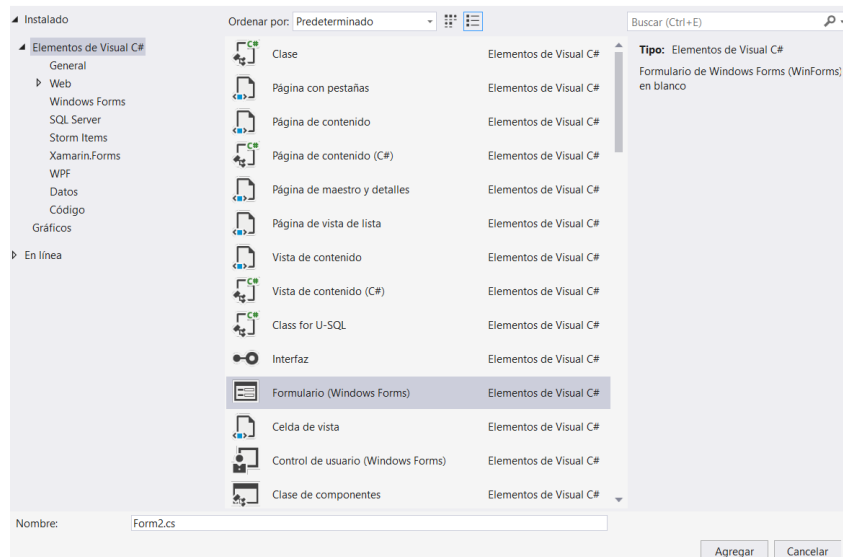


Figura 3.48: Crear una clase paso 2.



Figura 3.49: Pantallas del proceso Omarsa.

```

ladora.cs [Diseño] | PLCController.cs # X | PlantasDeAguas.cs
_OMARSA | HMI_OMARSA
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading;
6 using System.Threading.Tasks;
7 using Sharp7;
8
9 namespace HMI_OMARSA
0 {
11     18 referencias
12     public class PLCController
13     {
14         S7Client cpu1;
15         string IP1;
16         int estado;
17         bool proceso = false;
18         public bool conectado = false;
19         public Variables db16 = new Variables();
20
21         // Constructor
22         1 referencia
23         public PLCController (string param_ip)
24         {
25             IP1 = param_ip;
26             cpu1 = new S7Client();
27             estado = cpu1.ConnectTo(IP1, 0, 1);
28         }
29
30     }

```

Figura 3.50: Crear el objeto PLC.

Se recomienda utilizar la clase form1.cs como la pantalla Maestro de un proyecto en Visual Studio y no reemplazar su nombre original, en la ventana form.cs se desarrolla la programación para el llamado al control del PLC como se nota en la figura 3.55 y para la navegación entre pantallas como se aprecia en la figura 3.56.

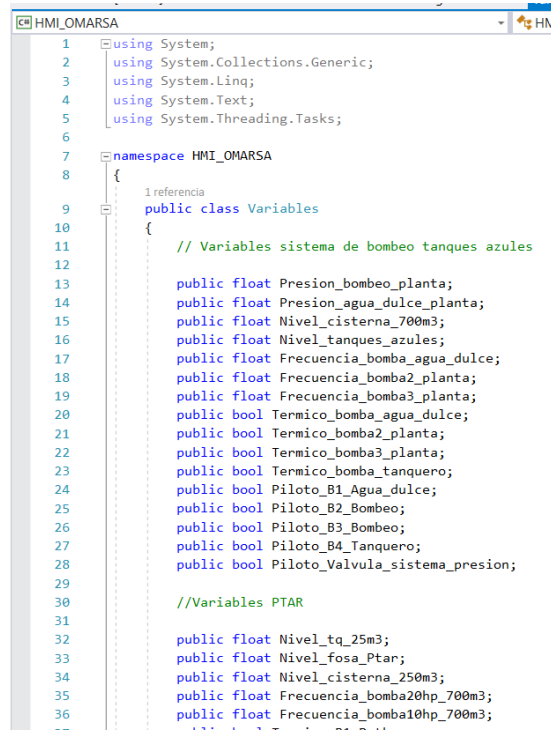
Paso 8:

Se inserta los controles gráficos en el formulario de todos los procesos y se navega entre pantallas.

Dentro del formulario HMI Omarsa se muestra las siguientes pantallas:

Pantalla Menú, figura 3.57 se crea una pantalla con botones de accesos a los diferentes procesos y poder navegar entre ellos, se instala un botón OFF para salir de la aplicación.

Pantalla del Sistema de Bombeo, figura 3.58, en esta pantalla permite a los usuarios revisar los controles de nivel de los tanques de agua dulce 25 m3, de la cisterna de 700 m3 y del tanque de almacenamiento de 25 m3 presión constante. Permite monitorear el encendido de las bombas del sistema de bombeo, así como saber el estado de las frecuencias y la presión



```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace HMI_OMARSA
8  {
9      1 referencia
10     public class Variables
11     {
12         // Variables sistema de bombeo tanques azules
13
14         public float Presion_bombeo_planta;
15         public float Presion_agua_dulce_planta;
16         public float Nivel_cisterna_700m3;
17         public float Nivel_tanques_azules;
18         public float Frecuencia_bomba_agua_dulce;
19         public float Frecuencia_bomba2_planta;
20         public float Frecuencia_bomba3_planta;
21         public bool Termico_bomba_agua_dulce;
22         public bool Termico_bomba2_planta;
23         public bool Termico_bomba3_planta;
24         public bool Termico_bomba_tanquero;
25         public bool Piloto_B1_Agua_dulce;
26         public bool Piloto_B2_Bombeo;
27         public bool Piloto_B3_Bombeo;
28         public bool Piloto_B4_Tanquero;
29         public bool Piloto_Valvula_sistema_presion;
30
31         //Variables PTAR
32
33         public float Nivel_tq_25m3;
34         public float Nivel_fosa_Ptar;
35         public float Nivel_cisterna_250m3;
36         public float Frecuencia_bomba20hp_700m3;
37         public float Frecuencia_bomba10hp_700m3;
38     }
39 }

```

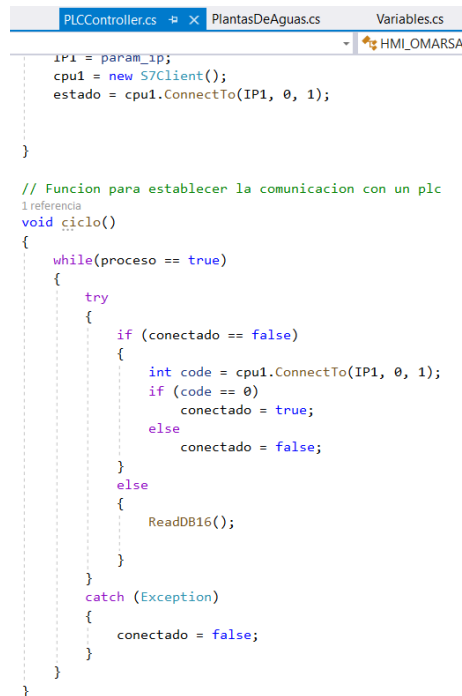
Figura 3.51: Clase Variables.

en PSI de la tubería. Adicional el monitoreo de las fallas térmicas.

Pantalla Ptar-Potabilizadora, figura 3.59, en este proceso se monitorea el nivel de la del pozo primario que receipta el agua gris que viene de planta, a su vez es bombeado por las bombas sumergibles bomba 1,2,3 que envían a los tanques de aireación donde dos blowers se alternan cada cinco horas y mantienen vivas unas bacterias que purifican el agua y previo a un análisis son devueltas al río.

En el proceso de la potabilizadora se monitorea el encendido de las cinco bombas que captan agua del río y que pasan a una cisterna de varias etapas que son dosificadas por dos químicos que es el MasterFloc Y Flasmaster, esto hace que las partículas de lodo se asienten y el agua al llegar a la etapa de clorificación se apta para el proceso. Dentro del Hmi se controla que la válvula de dosificación este siempre activa y se verifica el nivel del tanque de clorificacion de 280 cm3.

Se verifica el encendido de las bombas de filtro 1-2,3-4 que bombean a los



```

PLCController.cs  X  PlantasDeAguas.cs  Variables.cs
HML_OMARSA

IP1 = param_ip;
cpu1 = new S7Client();
estado = cpu1.ConnectTo(IP1, 0, 1);
}

// Funcion para establecer la comunicacion con un plc
1referencia
void ciclo()
{
    while(proceso == true)
    {
        try
        {
            if (conectado == false)
            {
                int code = cpu1.ConnectTo(IP1, 0, 1);
                if (code == 0)
                    conectado = true;
                else
                    conectado = false;
            }
            else
            {
                ReadDB16();
            }
        }
        catch (Exception)
        {
            conectado = false;
        }
    }
}

```

Figura 3.52: Función para establecer conexión al PLC.

tanques de carbón activado hasta la etapa de recepción en la cisterna de 250 m³.

Posterior al almacenamiento en la cisterna de 250 m³ hace un tras-base a otra cisterna ubicada a 300 metros de distancia de mas capacidad ubicada en la zona de bombeo, Estas bombas pueden ser monitoreadas tanto el estado on-off como la presión que levantan al llenar la cisterna de 700 m³.

Dentro de la solución se ha creado una pantalla para el control de alarmas o alertas.

Pantalla Succionadora, figura 3.60, en este proceso se controla el vacío que genera la tubería de succión tanto del sistema de cáscaras de camarón como el de cabezas de camarón.

Se monitorea la secuencia de los cilindros neumáticos tipo guillotina y encendido y apagado de las bombas.

Dentro de la solución se instala una pantalla de control de alarmas y alertas.

Pantalla Sistema de Desinfección, figura 3.61, debido a la pandemia

```

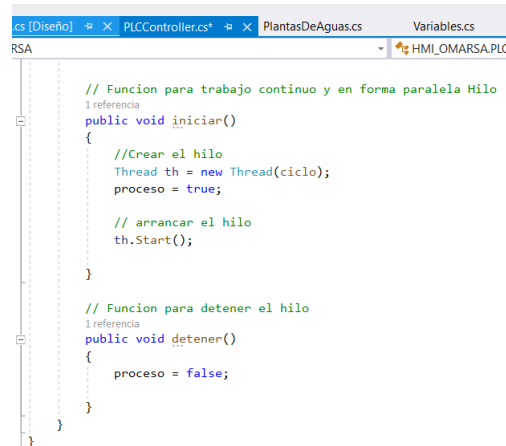
// Funcion para leer los datos del PLC1
1 referencia
void ReadDB16()
{
    byte[] buffer = new byte[263]; // # de los bytes de la DB
    int code = cpu1.DBRead(16, 0, 263, buffer);

    if (code != 0)
    {
        conectado = false;
    }
    else
    {
        // Pantalla Sistema de bombeo tanques Azules

        db16.Presion_bombeo_planta = (float)Math.Round(57.GetRealAt(buffer, 0),1); // Se coloca db16 por que en esa DB se trabaja en el PLC Master
        db16.Presion_agua_dulce_planta = (float)Math.Round(57.GetRealAt(buffer, 4),1); // La funcion (float)Math.Round se coloca para el # de decimales
        db16.Nivel_cisterna_700m3 = (float) Math.Round(57.GetRealAt(buffer, 8),1);
        db16.Nivel_tanques_azules = (float)Math.Round(57.GetRealAt(buffer, 12),1);
        db16.Frecuencia_bomba_agua_dulce = (float)Math.Round(57.GetRealAt(buffer, 16),1);
        db16.Frecuencia_bomba2_planta = (float)Math.Round(57.GetRealAt(buffer, 20),1);
        db16.Frecuencia_bomba3_planta = (float)Math.Round(57.GetRealAt(buffer, 24),1);
        db16.Termico_bomba_agua_dulce = 57.GetBitAt(buffer, 28, 1);
        db16.Termico_bomba2_planta = 57.GetBitAt(buffer, 28, 1);
        db16.Termico_bomba3_planta = 57.GetBitAt(buffer, 28, 1);
        db16.Termico_bomba_tanquero = 57.GetBitAt(buffer, 28, 3);
        db16.Piloto_B1_Agua_dulce = 57.GetBitAt(buffer, 28, 4);
        db16.Piloto_B2_Bombeo = 57.GetBitAt(buffer, 28, 5);
        db16.Piloto_B3_Bombeo = 57.GetBitAt(buffer, 28, 6);
        db16.Piloto_B4_Tanquero = 57.GetBitAt(buffer, 28, 7);
        db16.Piloto_Valvula_sistema_presion = 57.GetBitAt(buffer, 29, 0);
    }
}

```

Figura 3.53: Función de lectura de variables.



```

cs [Diseño]  PLCController.cs*  PlantasDeAguas.cs  Variables.cs
RSA  HMI_OMARSA.PLC

// Funcion para trabajo continuo y en forma paralela Hilo
1 referencia
public void iniciar()
{
    //Crear el hilo
    Thread th = new Thread(ciclo);
    proceso = true;

    // arrancar el hilo
    th.Start();
}

// Funcion para detener el hilo
1 referencia
public void detener()
{
    proceso = false;
}
}

```

Figura 3.54: Función del Hilo.

mundial este proceso es crítico ya que todas las cajas de camarón que salen al mercado Asiático deben ir desinfectadas con una solución de dióxido de cloro.

Para ello se creó una pantalla para el monitoreo de todos los puntos de desinfección ya sea por dióxido de cloro o lámparas UV.

Es de gran ayuda para el eléctrico de turno la visualización en la pantalla


```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace HMI_OMARSA
{
    10 referencias
    public partial class Form1 : Form
    {
        PlantasDeAguas ptaguas;
        Tratamiento_Aguas presidual;
        Succionadora succ;
        Sistema_Dioxido sistdiox;
        Palinox2 plnx2;
        Tornadol_2 torn1_2;
        Palinox1 plnx1;
        Coccido1 csd1;

        // Llamado al control del plc
        PLCController plc1 = new PLCController("172.25.144.2");

        9 referencias
        public Form1()
        {
            InitializeComponent();
            plc1.iniciar();
            this.ClientSize = new Size(1920, 1080);
            this.StartPosition = FormStartPosition.CenterScreen;
        }
    }
}

```

Figura 3.55: Llamado al control PLC.

para saber el estado de los sensores.

El control del nivel del tanque de preparación se lo puede visualizar como el de la presión de la red, son datos valiosos para mantener el estado del proceso en línea.

Dentro de la solución se instala una pantalla de control de alarmas y alertas.

Pantalla Palinox-2 figura 3.62, este equipo de alta producción es necesario mantenerlo controlado es por eso que se realizó la instalación de sensores de temperatura para ver el estado de la temperatura tanto en la salmuera como en secado.

Se verifica el estado de los agitadores y de los ventiladores, así como la velocidad de las bandas.

Dentro de la solución se instala una pantalla de control de alarmas y alertas, muy importante dentro del proceso ya que trabajar menos un ventilador en la zona de secado disminuye notablemente la velocidad de la línea de producción.

Pantalla Tornado 1-2 figura 3.63, en este túnel de congelación se

```

1 referencia
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    plc1.detener(); // Detiene el proceso del hilo para que el programa se cierre correctamente
}

1 referencia
private void btn2_Aguas_Click(object sender, EventArgs e)
{
    if (ptaguas == null || ptaguas.IsDisposed) ptaguas = new PlantasDeAguas(plc1);
    ptaguas.Show();
}

1 referencia
private void btn_salir_Click(object sender, EventArgs e)
{
    Application.Exit();
}

1 referencia
private void btn3_residuales_Click(object sender, EventArgs e)
{
    if (presidual == null || presidual.IsDisposed) presidual = new Tratamiento_Aguas(plc1);
    presidual.Show();
}

1 referencia
private void btn4_Succionadora_Click(object sender, EventArgs e)
{
    if (succ == null || succ.IsDisposed) succ = new Succionadora(plc1);
    succ.Show();
}

```

Figura 3.56: Función para navegar entre pantallas.



Figura 3.57: Pantalla Menús del Proceso.

monitorea tanto la temperatura de entrada como la de salida , la apertura y cierre de la válvula de amoniaco, el estado de los quince ventiladores, el estado y la velocidad de las bandas de transmisión y alarmas importantes como señales térmicas, aperturas de puertas, activación de paros de emergencia.

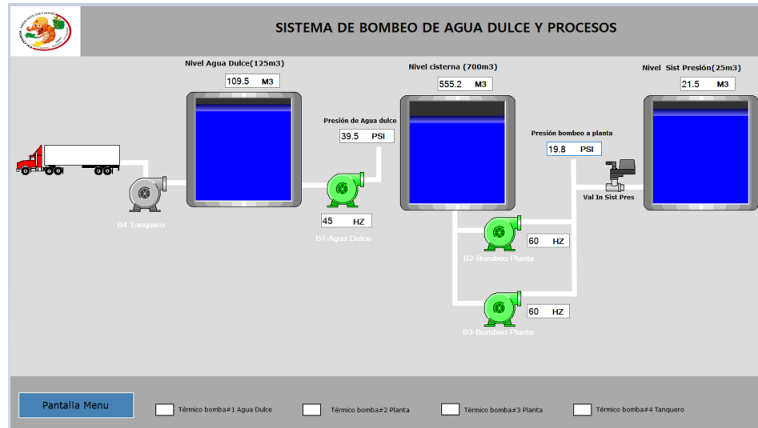


Figura 3.58: Pantalla Sistema de Bombeo.

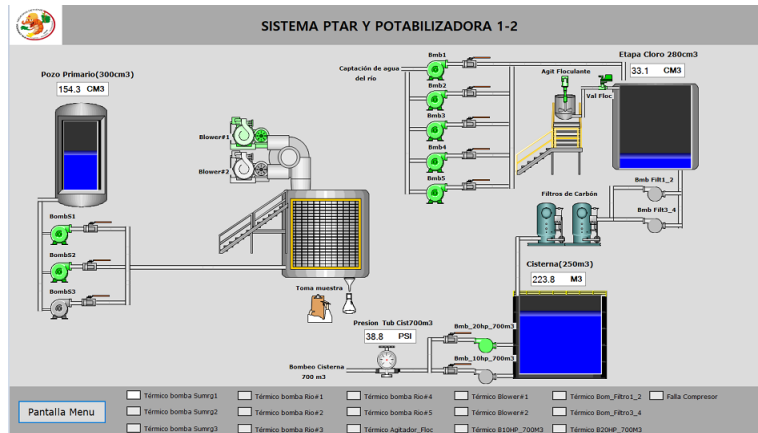


Figura 3.59: Pantalla Ptar-Potabilizadora 1-2.

3.1.5. Fase-5 Integración Node-Red y datos a la Nube

En esta fase se desarrolla otra técnica para el monitoreo y control de variables, es netamente programación por nodos, que facilita al programador enlazar las variables a la nube, sistema de bases de datos, mensajería de textos etc.

Para utilizar Node-Red como plataforma de monitoreo es indispensables realizar los siguientes pasos.

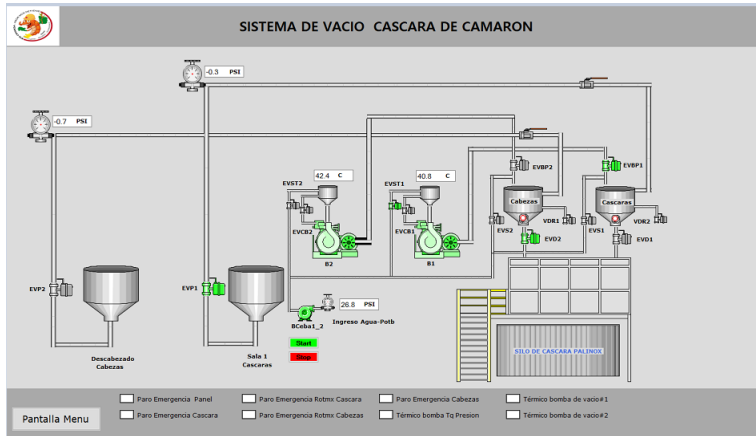


Figura 3.60: Pantalla Succionadora de Cascaras.

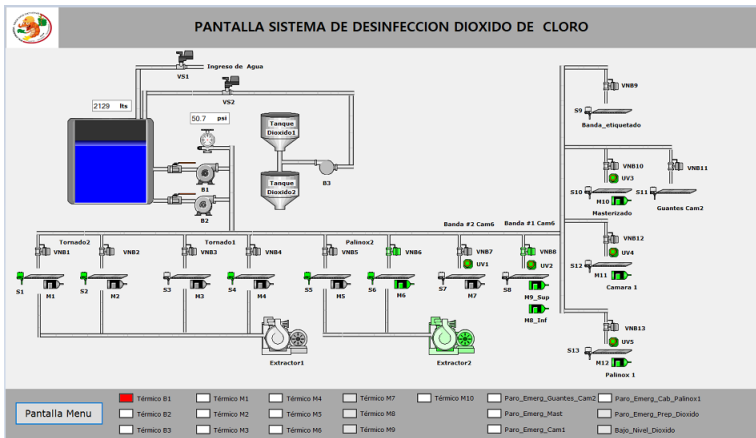


Figura 3.61: Pantalla Sistema Dióxido de cloro.

Paso 1 Ventana de Instalación Nodejs:

Si se trabaja desde una PC, o un servidor, y no se cuenta con un dispositivo como el IOT-2040 es importante instalar una dependencias como lo es Nodejs como lo muestra en la figura 3.64 que corre en JavaScript.

Pero como recomendación es indispensable desarrollar el proyecto en una PC, debido a la limitante de memoria Ram que conlleva el dispositivo, esto agiliza el desarrollo del proyecto y al final Node-Red permite exportar al

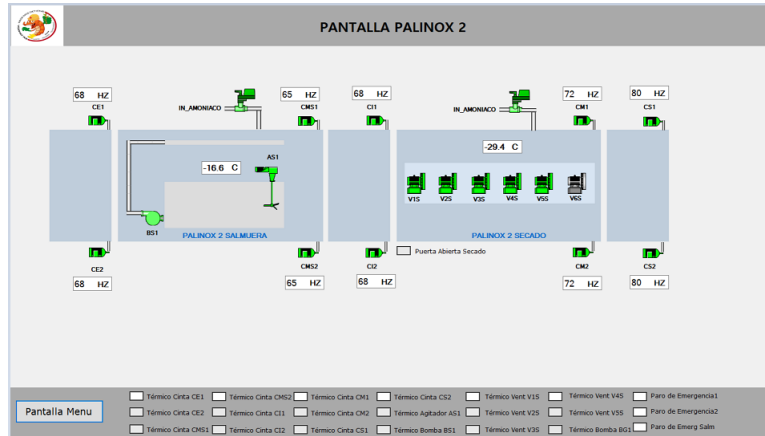


Figura 3.62: Pantalla Palinox 2.

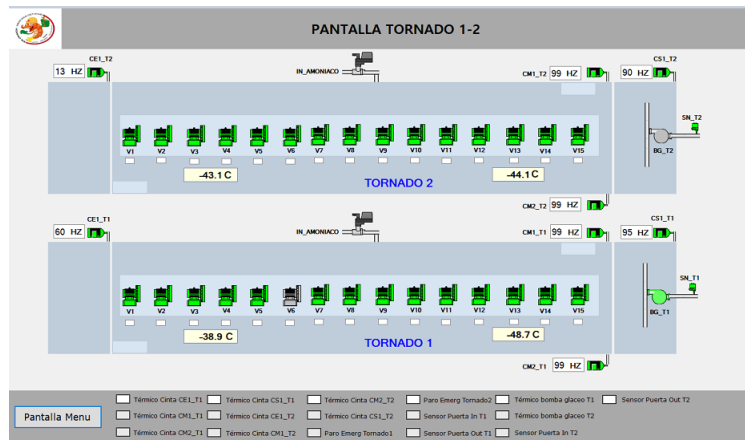


Figura 3.63: Pantalla Tornado 1-2.

dispositivo IOT 2040 el proyecto en formato Jason.

Paso 2 Instalación de Nodejs:

Asegurarse que Nodejs este instalado en la PC, servidor. Se debe ejecutar en la ventana CMD el comando `node -v`, como lo muestra en la figura 3.65

Nodejs trabaja con un gestor de librerías en la nube que se llama npm que está a la espera de la petición de instalación de un cliente.

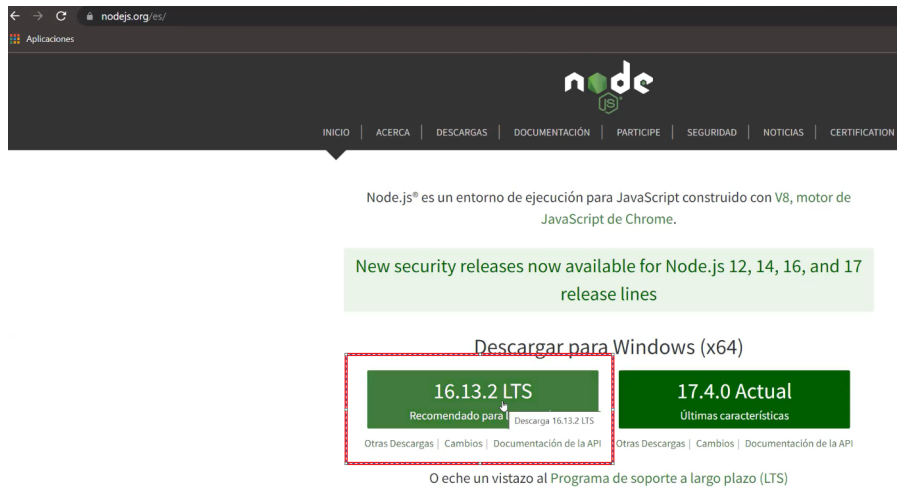


Figura 3.64: Descargar Nodejs versión LTS.

```

C:\> Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.1466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jlema>node --version
v16.13.2

C:\Users\jlema>

```

Figura 3.65: Versión Nodejs.

Este gestor de paquetes nos ayuda a la comunicación con Ubidots, base de datos, telegram etc. Para ello se verifica la correcta instalación del gestor de paquetes escribiendo `npm --version` como lo indica en la figura.3.66

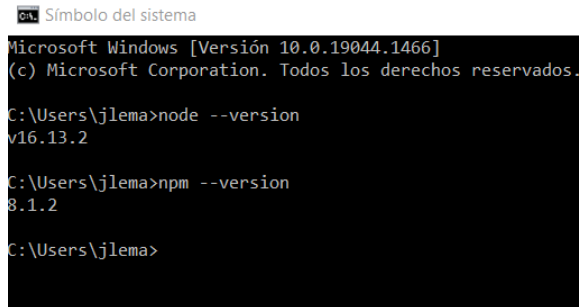
Paso 3 Instalación de Node-Red:

Instalar Node-Red desde el gestor de paquetes npm en la nube.

Ejecutar el comando `>npm install -g --unsafe-perm node-red`, como lo muestra en la figura 3.67 y que indica que en la plataforma npm se va instalar de manera global sin problemas node-red.

Paso 4 Ejecución cmd Node-Red:

Se ejecuta en cmd Node-Red y se verifica que la instalación de Node-Red



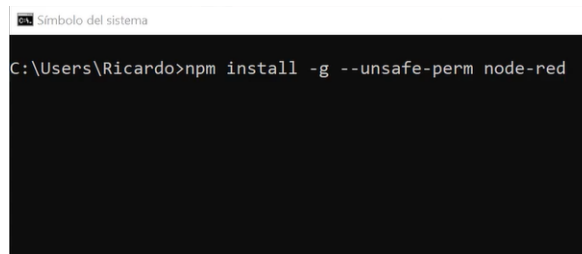
```
ca Símbolo del sistema
Microsoft Windows [Versión 10.0.19044.1466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jlema>node --version
v16.13.2

C:\Users\jlema>npm --version
8.1.2

C:\Users\jlema>
```

Figura 3.66: Versión npm.



```
ca Símbolo del sistema

C:\Users\Ricardo>npm install -g --unsafe-perm node-red
```

Figura 3.67: Instalación Node-Red.

es satisfactoria, como se aprecia en la figura 3.68.

Es importante cada vez que se quiera desarrollar en Node-Red o que se enlace con las variables S7-Simatic se debe ejecutar este comando.

Paso 5 Ventana de instalación de Librerías:

En la ventana Manage Palette se encuentra toda la librería de los nodos que se necesita para instalar como por ejemplo la librería Dashboard que se muestra en la figura 3.69, adicional se descarga las librerías que contienen los nodos para comunicarse con un PLC Simatic, llevar los datos a una base de datos, mensajería de texto etc, como lo muestra en la figura 3.70.

Paso 6 Creación de Fichas y Grupos:

Se procede con el desarrollo del proyecto en Node-Red para ello es debe crear los tab en la opción dashboard y crear los subgrupos de trabajo, eso da facilidad en el momento de asignar los nombres y la ubicación a los diferentes nodos como se indica en la figura 3.71.

```

node-red
Microsoft Windows [Versi3n 10.0.19044.1466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\jlema\node-red
12 Feb 10:54:26 - [info]

Welcome to Node-RED
=====
12 Feb 10:54:26 - [info] Node-RED version: v2.1.6
12 Feb 10:54:26 - [info] Node.js version: v16.13.2
12 Feb 10:54:26 - [info] Windows_NT 10.0.19044 x64 LE
12 Feb 10:54:28 - [info] Loading palette nodes
12 Feb 10:54:30 - [info] Dashboard version 3.1.5 started at /ui
12 Feb 10:54:31 - [info] Settings file : C:\Users\jlema\node-red\settings.js
12 Feb 10:54:31 - [info] Context store : 'default' [module=memory]
12 Feb 10:54:31 - [info] User directory : \Users\jlema\node-red
12 Feb 10:54:31 - [warn] Projects disabled : editorTheme.projects.enabled=false
12 Feb 10:54:31 - [info] Flows file : \Users\jlema\node-red\flows.json
12 Feb 10:54:31 - [info] Server now running at http://127.0.0.1:1880/
12 Feb 10:54:31 - [warn]

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

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

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----
12 Feb 10:54:31 - [info] Starting flows
12 Feb 10:54:31 - [warn] [S7 endpoint:PLC_S7_1500] No variables configured, skipping cyclic reading
12 Feb 10:54:31 - [warn] [S7 endpoint:PLC_S7_1500] No variables configured, skipping cyclic reading
12 Feb 10:54:31 - [info] Started flows
12 Feb 10:54:31 - [error] [MySQLdatabase:MySQL] Error: connect ECONNREFUSED 127.0.0.1:3306
ubidots Publisher connected

```

Figura 3.68: Ejecuci3n Node-Red.

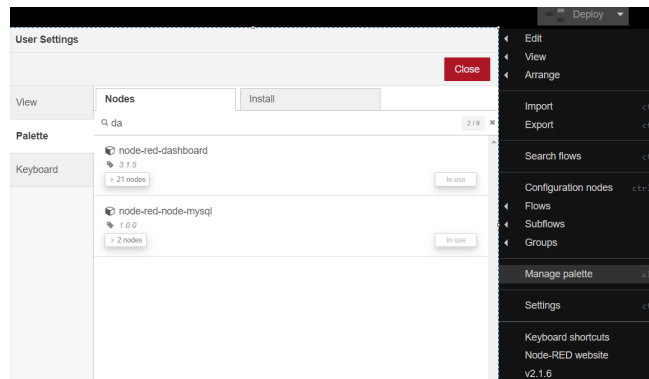


Figura 3.69: Ventana Manage Palette.

Paso 7 Crear conexi3n S7 Simatic- Node-RED:

Es necesario crear una conexi3n S7 con Node red, para el caso de este proyecto se trabaj3 con el PLC M3ster S7-1500 como se aprecia en la figura [3.72](#).



Figura 3.70: Gestor de paquetes librería Node-Red.

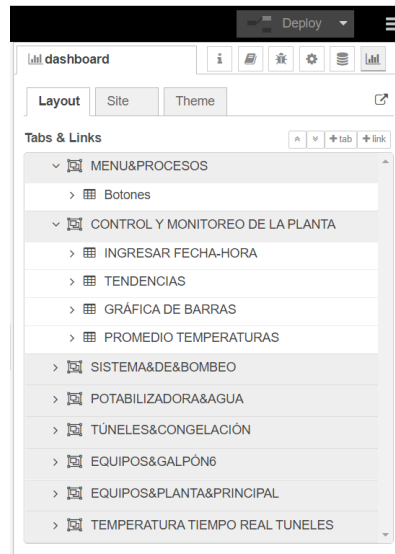


Figura 3.71: Creación de grupos y subgrupos en Node-Red.

Paso 8 Creación de variables globales:

Se crea una tabla de variables, semejante a las creadas en la DB16 del

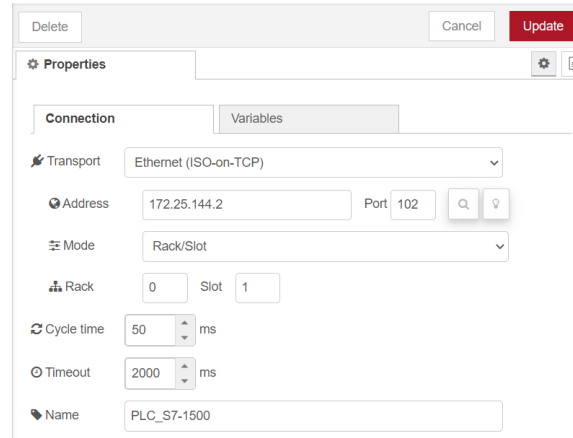


Figura 3.72: Crear conexión Simatic con Node-Red.

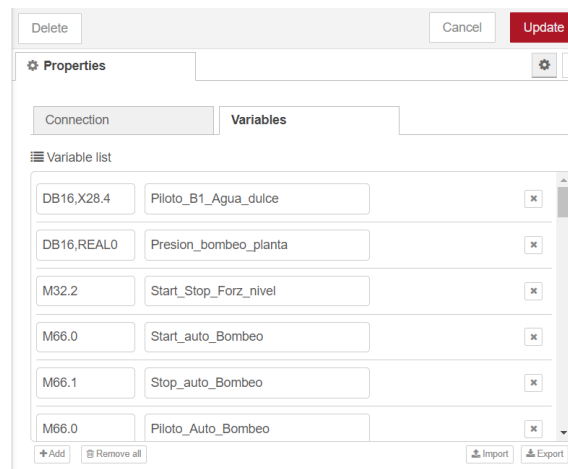


Figura 3.73: Creación de variables en Node-Red.

PLC que se va a enlazar, es recomendable utilizar los mismo nombres que se asignó en Tia Portal como se muestra en la figura 3.73.

Paso 9 Creación de plantillas:

Como en todo software de programación es importante la creación de plantillas y luego replicar a los demás proyectos. En este paso se procede a crear plantillas para señales booleanas, reales, de tiempo etc.

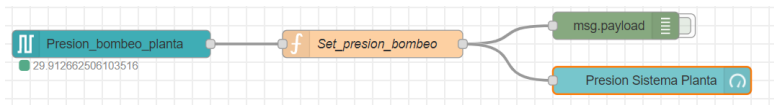


Figura 3.74: Nodos lectura de datos Reales.

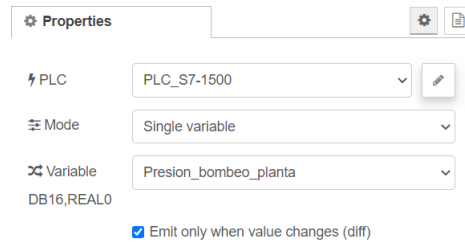


Figura 3.75: Configuración Nodo S7 in.

Para realizar la lectura de variables Reales se utilizan tres nodos detallados a continuación y visto en la figura 3.74.

- Se configura el nodo S7-in, se asigna la variable a medir y el PLC del que se toma el dato como lo indica en la figura 3.75.
- Se configura el nodo función en programación JavaScript y por el comando `toFixed` se trunca el número de decimales a observar en el dashboard como se muestra en la figura 3.76.
- Se elige un nodo Gauge para la visualización en el dashboard y se configura como se muestra en la figura 3.77.

Para realizar la escritura de variables Reales se utilizan dos nodos detallados a continuación y visto en la figura 3.78.

- Se utiliza el nodo text in para realizar la escritura de datos reales y se configura como lo detallado en la figura 3.79.
- Se utiliza el nodo S7 out para escribir en el PLC el dato escrito en Node-Red y se configura como en la figura 3.80.

Para las soluciones de escritura de datos booleanos se ha creado la siguiente plantilla 3.81.

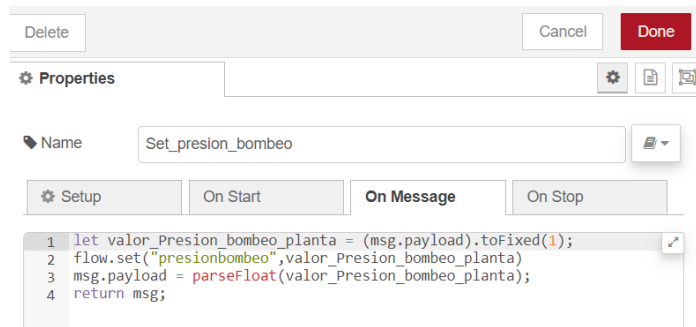


Figura 3.76: Comandos para lectura de variables.

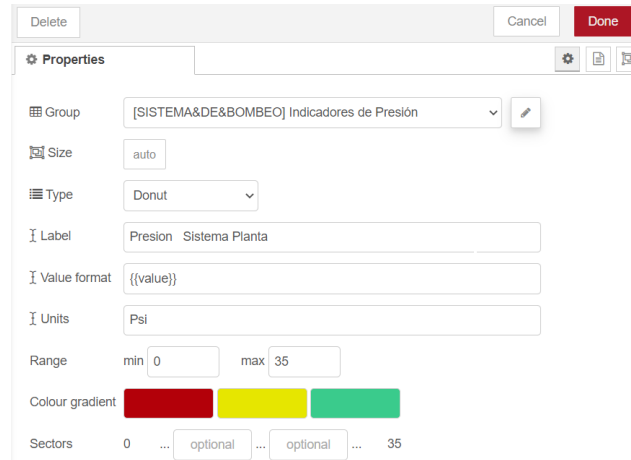


Figura 3.77: Configuración Nodo Gauge.



Figura 3.78: Configuración Nodo text in.

Para el nodo función se ejecuta los comandos vistos en la figura 3.82, tanto para el valor falso como para el valor verdadero

Para leer variables booleanas desde el PLC hacia Node-Red se integran los siguientes nodos como se nota en la figura 3.83.

La configuración de un led para visualizar el encendido y apagado se

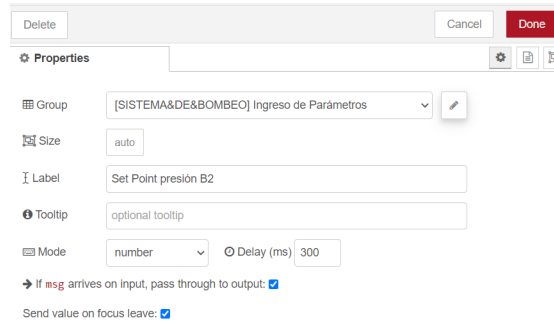


Figura 3.79: Configuración nodo text variables.

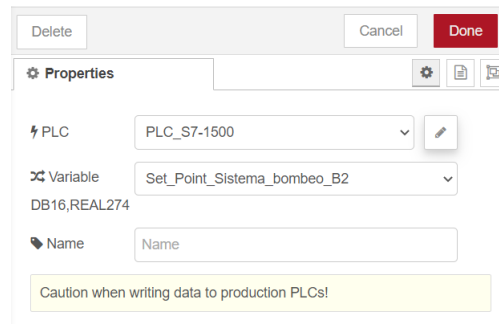


Figura 3.80: Configuración Nodo S7 out.

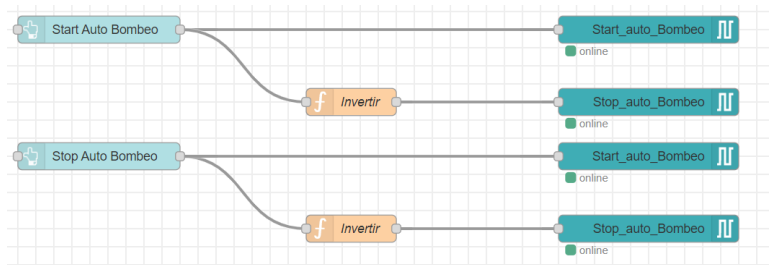


Figura 3.81: Configuración Nodos de escritura booleanos.

muestra en la figura 3.84.

Para leer variables tipo time off day se integran los siguientes nodos como se nota en la figura 3.85. La programación en JavaScript en el nodo función

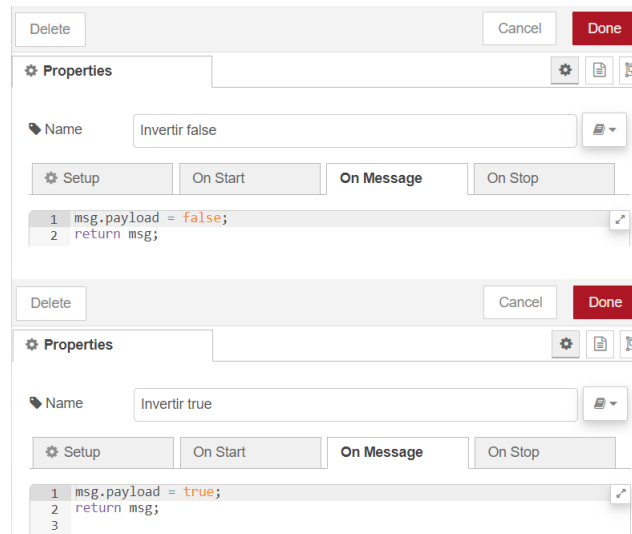


Figura 3.82: Comandos booleanos.



Figura 3.83: Configuración Nodos de lectura Booleanos.

para leer los datos de tiempo en horas, minutos, segundos se aprecia en la figura 3.86.

Paso 10 Envió de datos por correo electrónico:

Para el envío de datos por correo electrónico es necesario integrar los nodos que se visualizan en la figura 3.87.

Para realizar el envío de datos por correo electrónico es necesario seguir los siguientes pasos.

- En el nodo email se debe configurar el correo de la persona que emite como la que recibe la información y se muestra en la figura 3.88.
- En el correo electrónico que se emite el mensaje se debe activar en la cuenta de google la pestaña permitir el acceso de aplicaciones poco seguras, como en la figura 3.89.

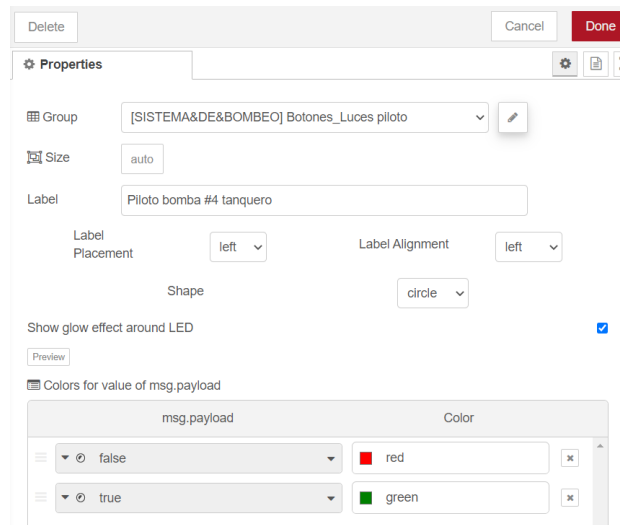


Figura 3.84: Configuración led booleano.

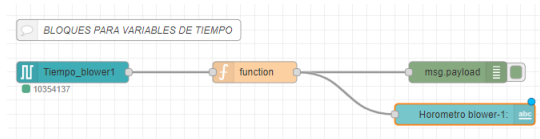


Figura 3.85: Configuración led booleano.

- En el nodo Función se realiza la programación en JavaScript para que se visualice el texto a enviar, se la realiza con una variable tipo flow Get del sistema de presión como se aprecia en la figura 3.90.
- Finalmente en un nodo tipo inyect se programa el intervalo de tiempo que se requiere para la recepción del correo electrónico al destinatario se nota en la figura 3.91. y el mensaje se aprecia en la figura 3.92.

Paso 11 Envié de mensajes instantáneos por Telegram:

En la pestaña Manage Palette descargar la librería que se muestra en la figura 3.93.

Posterior se ingresa en Telegram Web, y se sigue los siguientes pasos:

- Ingresar en Telegram web y dar clic en BotFaher-Start como se muestra en la figura 3.94.

```

1 let t = new Date(msg.payload);
2
3 let h = t.getHours() - 19;
4 let m = t.getMinutes();
5 let s = t.getSeconds();
6
7 if(h<10){
8   h = "0"+h;
9 }
10 if(m<10){
11   m = "0"+m;
12 }
13 if(s<10){
14   s = "0"+s;
15 }
16
17 msg.payload = h+"-"+m+"-"+s;
18
19 return msg;
    
```

Figura 3.86: Comandos de tiempo.

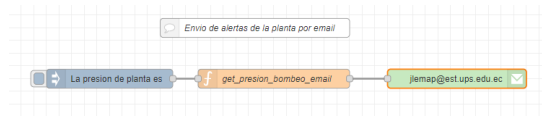


Figura 3.87: Integración de Nodos para envío de correos.

Properties

To

Server

Port Use secure connection.

Userid

Password

TLS option Check server certificate is valid

Name

Figura 3.88: Configuración correo electrónico



Figura 3.89: Acceso aplicaciones google.

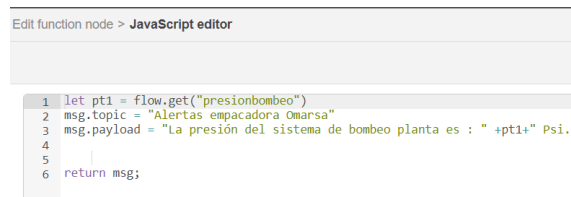


Figura 3.90: Programación Java envió de texto

- Dar clic en New Bot como en la figura 3.95 y se crea un nuevo Bot.
- Se crea un nuevo bot para el caso de este manuscrito se ha creado el enlace jlemabot, y un número de token que se debe ingresar en el nodo de telegram como lo muestra en la figura 3.96.
- La programación de los nodos es la siguiente se copia el número de token en el nodo telegram sender como se muestra en la figura 3.97
- Se inserta un script en un nodo función para realizar el envió de mensajes, cabe mencionar que este procedimiento se debe hacer a cada dispositivo que se requiera enviar este tipo de alertas como se muestra en la figura 3.98.
- Al final se configura los nodos para el envió de mensajes por telegram y se estructura como en la figura 3.99.

Paso 12 Base de datos:

Para crear una base de datos y poder administrar un panel visual se descarga WampServer que es un paquete informático que contiene Apache que es un

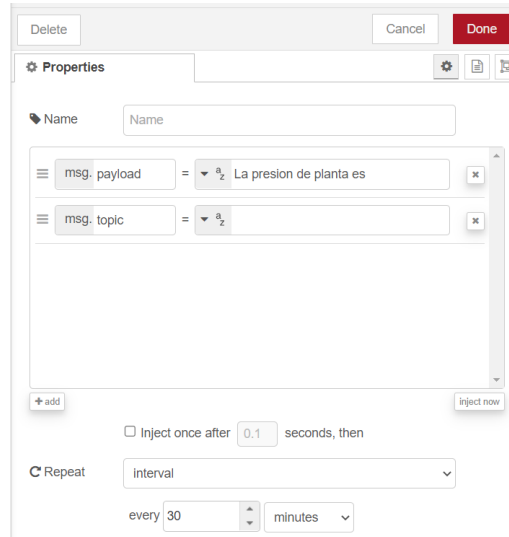


Figura 3.91: Intervalo de tiempo correo electrónico.



Figura 3.92: Alertas desde Node-Red al correo electrónico

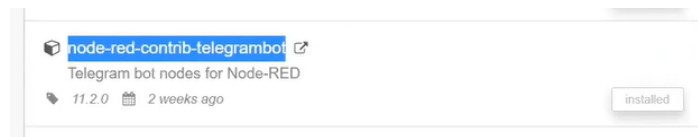


Figura 3.93: Alertas desde Node-Red.

servidor web HTTP, contiene a My SQL que es un gestor de base datos, y contiene a phpMyAdmin que es una aplicación Web para administrar base de datos, este paquete se muestra en la figura 3.100.

Una vez completada la instalación del WampServer, se puede navegar a

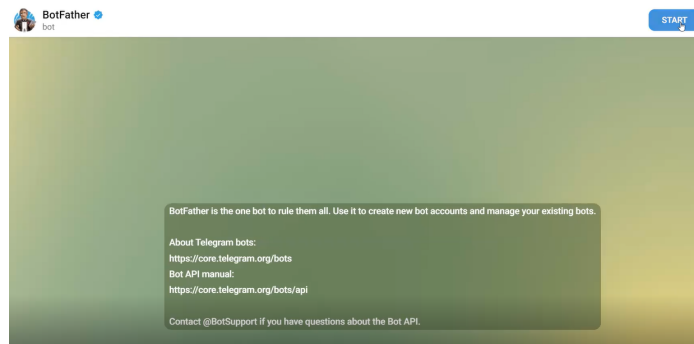


Figura 3.94: Paquete TelegramBot.

I can help you create and manage Telegram bots. If you're new to the Bot API, please [see the manual](#).

You can control me by sending these commands:

[/newbot](#) - create a new bot
[/mybots](#) - edit your bots [beta]

Edit Bots

[/setname](#) - change a bot's name
[/setdescription](#) - change bot description
[/setabouttext](#) - change bot about info
[/setuserpic](#) - change bot profile photo
[/setcommands](#) - change the list of commands
[/deletebot](#) - delete a bot

Figura 3.95: Crear un nuevo bot.

los diferentes paquetes, es como una consola con menús, se aprecia en la figura 3.101.

Para estructurar la base de datos es necesario seguir los siguientes paso que se desarrolló en el presente proyecto:

- Se crea un usuario y contraseña para ingresar en phpMyAdmin como se visualiza en la figura 3.102.
- Se crea dos nuevas carpetas dentro de la base de datos la primera es para la temperatura de los túneles de congelación y la segunda es para los niveles de agua de la planta Omarsa y se visualiza en la figura 3.103.
- Se procedió a crear la estructura y la creación de las variables como se

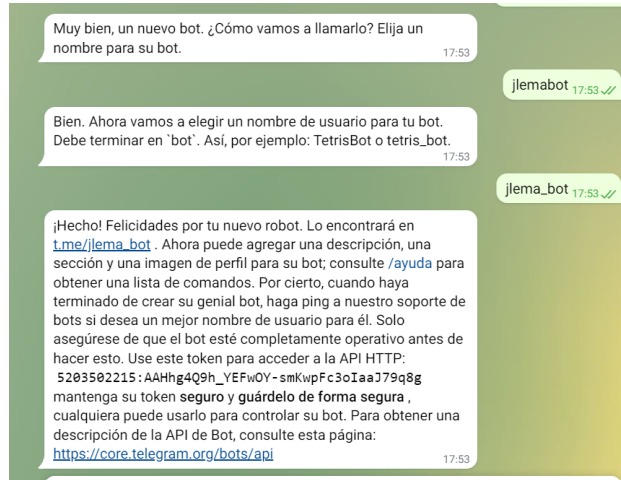


Figura 3.96: Crear token para enlace.

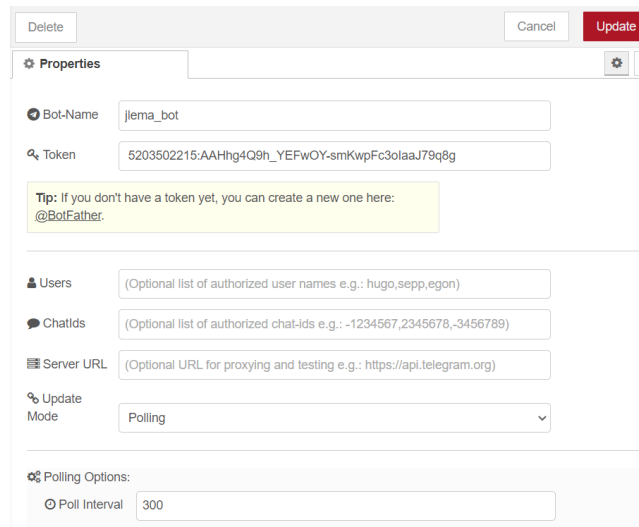


Figura 3.97: Insertar token en nodo telegram.

muestra en la figura 3.104.

- Es importante en una base de datos como estándar que se tenga opciones para crear, insertar, actualizar, y borrar, o mas conocido como

```

Delete Cancel Done
Properties
Name get_presion_bombeco
Setup On Start On Message On Stop
1 let pt1 = flow.get("presionbombeco")
2
3- msg.payload={
4   chatId:5249475236,
5   type:"message",
6   content: "La presión del sistema de bombeo planta es : " +pt1+" Psi.
7- }
8 return msg;
    
```

Figura 3.98: Función envío de mensajes Telegram.

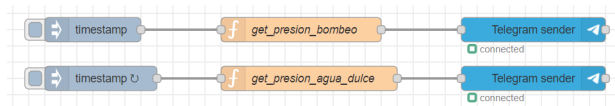


Figura 3.99: Estructura envío de mensajes Telegram.

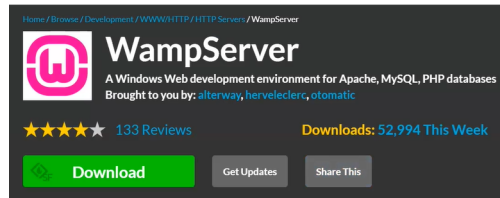


Figura 3.100: Instalación paquete WampServer.

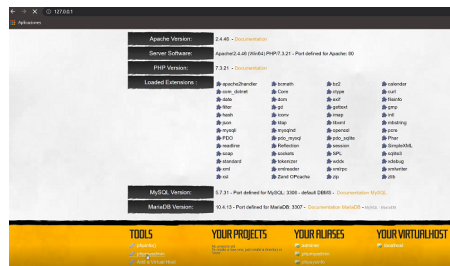


Figura 3.101: Consola WampServer.

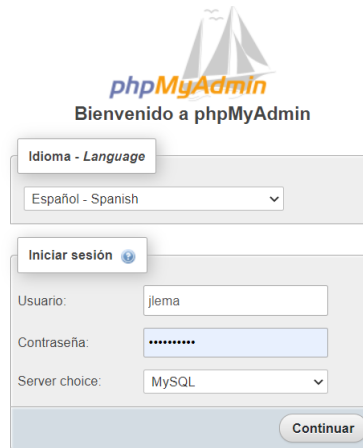


Figura 3.102: Ingresar en phpMyAdmin.

CRUD visto en la figura 3.105, en este proyecto se utiliza nodos para insertar y borrar los datos.

- Para inserta variables se realizó la siguiente programación en el nodo función en JavaScript como lo muestra la figura 3.106, y se puede observar en la figura 3.107, los datos extraídos del PLC Master cuyas variables pertenecen a la temperatura de los túneles.
- Esta base de datos también permite verificar los datos en forma de curva, únicamente se debe setear el numero de las filas a ser vistas como en la figura 3.108.
- Permite exportar los datos en formato excel para realizar comparativos. Como se indica en la figura 3.109.
- Para encerrar los valores de la base de datos se ha insertado un nodo función con el siguiente comando en JavaScript que permite borrar los datos sin necesidad de acceder a la tabla y se muestra en la figura 3.110.
- Los mismos paso se utilizaron para la tabla de control de niveles.
- Finalmente una vista en la figura 3.111 los nodos utilizados para realizar la base de datos.

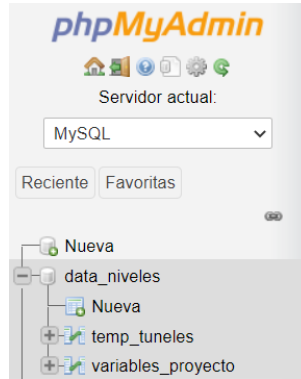


Figura 3.103: Creación base de datos Omarsa.

The screenshot shows the table structure for 'temp_tuneles'. The table has 10 columns. The first column is 'id_temp' (int(10), AUTO_INCREMENT). The second is 'fecha_hora' (timestamp(6), CURRENT_TIMESTAMP(6)). The next seven columns are 'temp_tunel1' through 'temp_tunel7', all of type float. The last column is 'temp_tunel15' (float, NULL). Each column has 'Cambiar' and 'Eliminar' icons.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id_temp	int(10)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	fecha_hora	timestamp(6)			No	CURRENT_TIMESTAMP(6)			Cambiar Eliminar Más
3	temp_tunel1	float			No	Ninguna			Cambiar Eliminar Más
4	temp_tunel2	float			No	Ninguna			Cambiar Eliminar Más
5	temp_tunel3	float			No	Ninguna			Cambiar Eliminar Más
6	temp_tunel4	float			No	Ninguna			Cambiar Eliminar Más
7	temp_tunel5	float			No	Ninguna			Cambiar Eliminar Más
8	temp_tunel6	float			No	Ninguna			Cambiar Eliminar Más
9	temp_tunel7	float			No	Ninguna			Cambiar Eliminar Más
10	temp_tunel15	float			Sí	NULL			Cambiar Eliminar Más

Figura 3.104: Base de datos temperaturas Túneles.

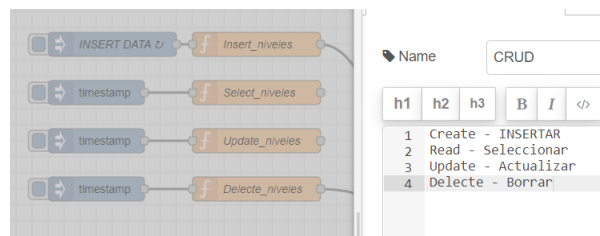


Figura 3.105: Terminología CRUD.

```

1 const temp5 = parseFloat(flow.get("temp_tune11"));
2 const temp4 = parseFloat(flow.get("temp_tune12"));
3 const temp3 = parseFloat(flow.get("temp_tune13"));
4 const temp2 = parseFloat(flow.get("temp_tune14"));
5 const temp1 = parseFloat(flow.get("temp_tune15"));
6 const temp6 = parseFloat(flow.get("temp_tune16"));
7 const temp7 = parseFloat(flow.get("temp_tune17"));
8 const temp8 = parseFloat(flow.get("temp_tune18"));
9
10 mgc.topic = "tune18" INTO temp_tune16,temp_tune17,temp_tune18,temp_tune19,temp_tune20,temp_tune21,temp_tune22,temp_tune23,temp_tune24,temp_tune25 VALUES ("temp1","temp2","temp3",
11 "temp4","temp5","temp6","temp7","temp8");
12
13 return mgc;
    
```

Figura 3.106: Inserta variables a la base de datos.

id_temp	fecha_hora	temp_tune1	temp_tune2	temp_tune3	temp_tune4	temp_tune5	temp_tune6	temp_tune7	temp_tune8	temp_tune9	temp_tune10	temp_tune11	temp_tune12	temp_tune13	temp_tune14	temp_tune15
45	2022-02-13 20:24:37	130296	-33.5	-39.1	-31.3	5.9	42.9	-50.7	-28.4	17.4						
46	2022-02-13 20:44:37	137775	-34	-38.7	-31.4	5.6	42.5	-31.4	-28.8	17.2						
47	2022-02-13 21:04:37	143561	-35.2	-40	-32.6	5.6	39.2	-29.4	-29.5	17.4						
48	2022-02-13 21:24:37	160130	-35.8	-38.8	-32.8	7	8.7	-6	-29.8	17.2						
49	2022-02-13 21:44:37	160651	-35.8	-38.9	-33.1	7.5	0.2	-24.2	-30.1	17.4						
50	2022-02-13 22:04:37	166448	-37.6	-39.3	-32	6.9	2.8	-28.6	-32.2	17.4						
51	2022-02-13 22:24:37	188766	-37.7	-39.4	-32.3	6.7	4.1	-30.1	-28.2	17.4						
52	2022-02-13 22:44:37	196898	-38.2	-40.3	-33.2	6.5	4.7	-31.3	3.3	17.4						
53	2022-02-13 23:04:37	208939	-36.5	-37.8	-32.5	6.3	5	-31.9	-27.5	17.2						
54	2022-02-13 23:24:37	211630	-38.3	-40.5	-33.6	6.3	4.6	-32.6	-31.6	17.2						
55	2022-02-13 23:44:37	212483	-38.5	-41	-34.2	6.6	6.2	-36.1	-32.7	17.2						
56	2022-02-14 19:19:59	460847	-24.2	-21.7	-30.3	-27.5	-4.7	-28.3	15.5	16.4						
57	2022-02-14 19:39:59	458674	-23.2	-20.6	-25.4	-26.6	-25.5	-27.8	-15	16.2						
58	2022-02-14 19:59:59	474598	-24	-21.2	-25.9	-27.4	-27.4	-28	-26.1	16.2						
59	2022-02-14 20:19:59	472796	-25.4	-22.4	-13.6	-28.6	-28.9	-26.8	-29	16.2						
60	2022-02-14 20:39:59	477973	-21.2	-23.3	7.1	-29.8	-29.8	-27.8	-29.7	16.2						
61	2022-02-14 23:11:56	300964	-29.5	-29.3	-34.4	-35.6	-34.1	-0.4	-22.3	16.7						
62	2022-02-14 23:31:56	310652	-29.5	-30.8	-35.5	-36.5	-37.1	-0.9	-28.7	16.7						
63	2022-02-15 09:11:51	542481	5.3	2.5	2.8	-8.6	-20.6	-22.4	-40.4	18.6						
64	2022-02-15 09:31:51	537184	-12.2	9.8	4.7	-9.1	-22.6	-21.6	-39.8	17.2						
65	2022-02-15 09:51:51	555827	-31.1	10.4	-24.1	-0.6	-22.5	-22.4	-16.7	16.9						

Figura 3.107: Extracción de datos temperaturas túneles.

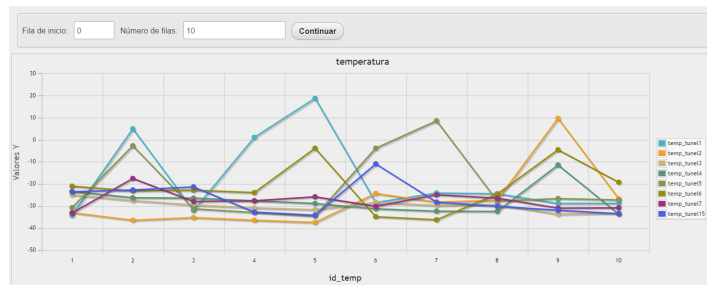


Figura 3.108: Curva de temperaturas túneles.

Paso 13 Ubidots:

Para el envío de datos a la nube se ha realizado los siguientes pasos que se detallan a continuación:

- En credenciales API se copia el número token y se pega en el nodo

A	B	C	D	E	F	G	H	I	J
id_temp	fecha_hora	temp_tunel1	temp_tunel2	temp_tunel3	temp_tunel4	temp_tunel5	temp_tunel6	temp_tunel7	temp_tunel15
1	03/02/2022 15:36	-34.3	-33.1	-25	-23.5	-30.8	-21.1	-33.2	-23.6
2	03/02/2022 16:06	4.9	-36.5	-27.6	-26.3	-2.8	-23.2	-17.6	-22.8
3	03/02/2022 16:36	-32.1	-35.3	-29.6	-26.5	-31.3	-22.9	-28	-21.4
4	03/02/2022 17:06	1	-36.5	-30.9	-27.7	-33	-24	-27.7	-32.8
5	03/02/2022 17:36	18.7	-37.5	-31.7	-28.9	-34.6	-3.9	-25.9	-34.2
6	04/02/2022 09:42	-28.5	-24.4	-28.4	-31.3	-3.9	-34.9	-30.2	-11
7	04/02/2022 10:12	-24.2	-28.3	-29.8	-32.4	8.5	-36.3	-25	-28.3
8	04/02/2022 10:42	-24.5	-27.5	-29.4	-32.5	-27.8	-24.6	-26.5	-30.2
9	04/02/2022 11:12	-28.9	9.6	-33.8	-11.5	-26.7	-4.6	-31	-32
10	04/02/2022 11:42	-28.9	-26.8	-33.3	-33.7	-27.3	-19.3	-30.9	-33.5
11	04/02/2022 12:12	-29.9	-28.4	-34.6	-34.8	-28.2	-20	-32.1	-34.2
12	04/02/2022 12:42	-27.2	-29.7	-35.8	-35.8	-26.9	-19.3	-32.2	-34.7
13	04/02/2022 13:12	-28.3	-30.3	-34.7	-0.4	-27.8	-20.4	-33.6	-34.7
14	04/02/2022 13:42	-28.7	-29.5	-33.9	-15.9	-28.1	-21.2	-33.6	-33
15	04/02/2022 14:12	-30.8	-31.5	16.3	-19	-29.4	-22.9	-36	-37.4
16	04/02/2022 14:42	-32.9	-32.3	-22	-20	-11.7	-22.8	-36.8	-38.7

Figura 3.109: Exportar datos a excel.

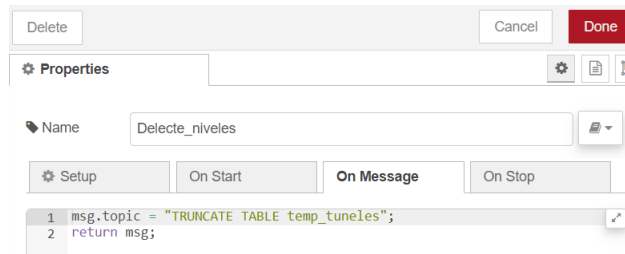


Figura 3.110: Eliminar historial base de datos.

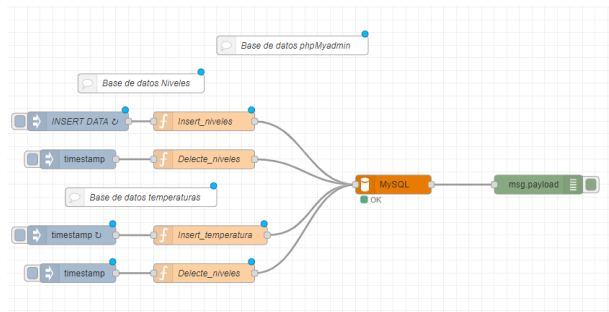


Figura 3.111: Nodos base de datos.

Ubidots de Node-Red como lo muestra la figura 3.112.

- Se copiar el token de ubidots se da clic en un nodo out Ubidots y se copia como lo muestra la figura 3.113.
- Para crear nuevos dispositivos es importante que el nombre sea en

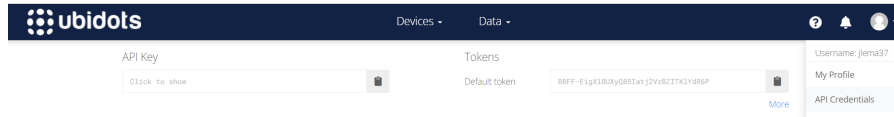


Figura 3.112: Credenciales Api Ubidots.

minúsculas, sin caracteres y sin espacios como se muestra en la figura 3.114.

- Se crea las variables dentro de Ubidots hay dos opciones la cruda que es para revisión de datos en tiempo real y la sintética que es para hacer análisis de datos, se da clic en la cruda y se generan los cuadros de variables como se muestra en la figura 3.115.
- El script en el nodo función para realizar el enlace con Ubidots se muestra en la figura 3.116, es necesario crear variables tipo flow Get.
- Para crear un medidor en Ubidots es necesario crear un nuevo Widget insertar el nombre de la variable, el método de agregación, el período de tiempo, el nombre del dashboard, el numero de decimales y los rango de lectura, se visualiza en la figura 3.117.
- Ubidots permite cambiar el color al fondo, a la letra, a los bordes, etc, y tener dashboards personalizados como se muestra en la figura 3.118, todo eso generado en códigos hexadecimales, en internet existe una herramienta colores CSS que puede ser de ayuda en el momento de elegir los tonos.
- Finalmente se generan los dashboards que pueden ser leídos desde cualquier dispositivo electrónico en tiempo real como lo muestra la figura 3.119.

Paso 14 Creación de Dashboard Node-Red:

La estructurada de la parte de programación de los nodos se detalla en una vista del Menú generado dentro de Nodo-RED como lo muestra en la figura 3.120.

- Se crea un dashboard para monitorear y controlar las variables mas importantes dentro del sistema de bombeo, como controles de niveles de los tanques, seteos de presiones para el bombeo, visualización del

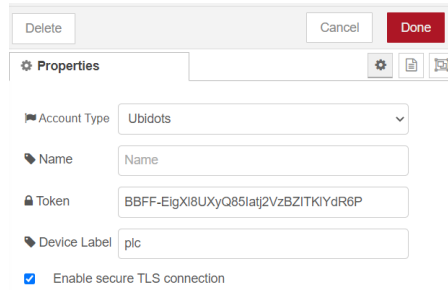


Figura 3.113: Copiar token en nodo Ubidots.



Figura 3.114: Crear dispositivos en Ubidots.

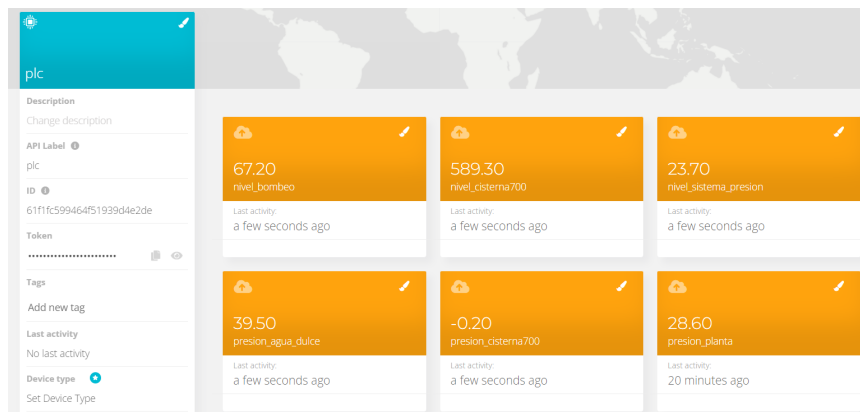


Figura 3.115: Variables Ubidots.

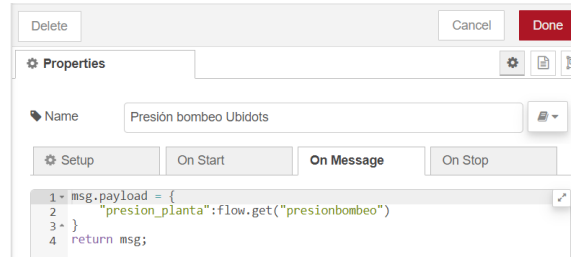


Figura 3.116: Programación nodo función Ubidots.



Figura 3.117: Agregar Widget.

estado de los motores, monitoreo de la frecuencia de las bombas, todo esto visto en la siguiente figura 3.121.

- Se crea un dashboard para monitorear y controlar las variables mas importantes de la planta potabilizadora como los controles de nivel, control de presiones, estado de los motores, variables de tiempo de



Figura 3.118: Editor de estilos.

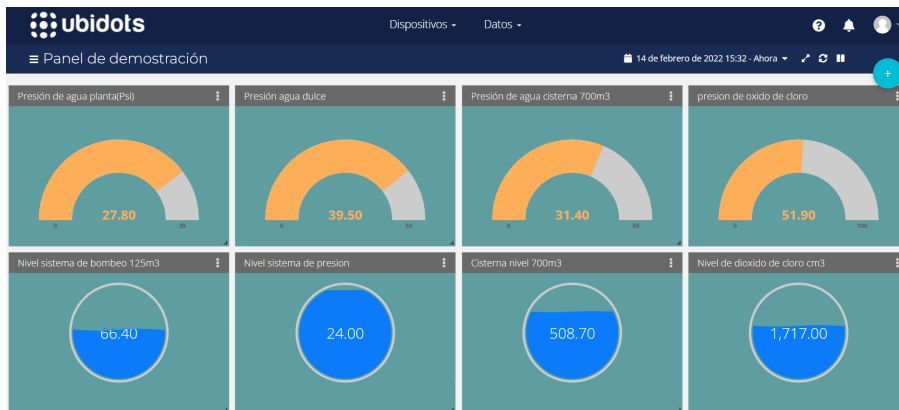


Figura 3.119: Dashboards planta Omarsa.

encendido de los blowers, vistos en la figura 3.122.

- Se crea un dashboard para monitorear la temperatura de los túneles de congelación y verificar su curva en el tiempo como se muestra en la figura 3.123 y 3.124.

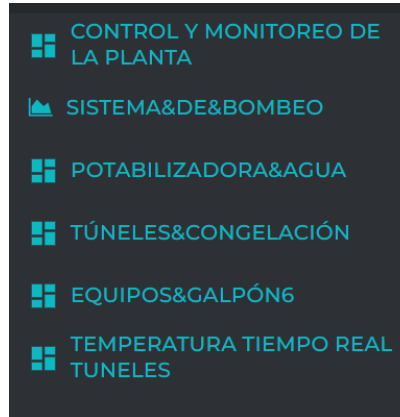


Figura 3.120: Menú Node Red.

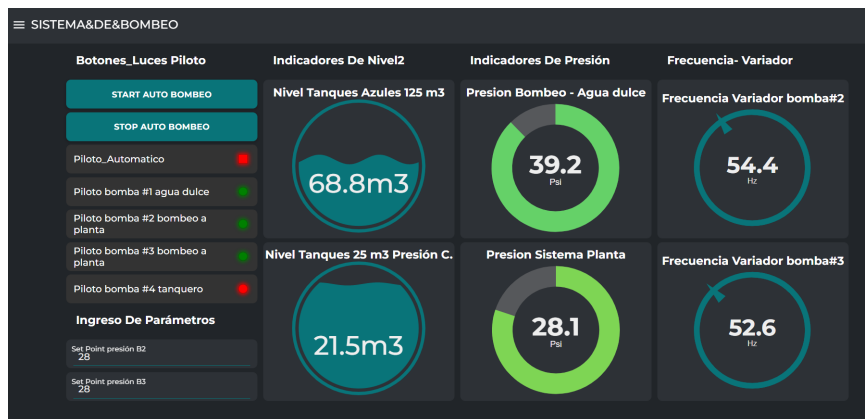


Figura 3.121: Dashboard Sistema de Bombeo.

- Se crea un dashboard para monitorear las variables que se encuentran dentro del Galpón 6 de la planta Omarsa se visualiza la temperatura de los equipos de congelación Tornado 1-2, Palinox 2, revisión de nivel y presión del tanque de preparación de dióxido de cloro y esto se muestra en la figura 3.125.

Paso 15 Instalación Pasarella IOT2040 Siemens:

Los pasos que se realizó en la instalación del sistema operativo IOT2040 son:

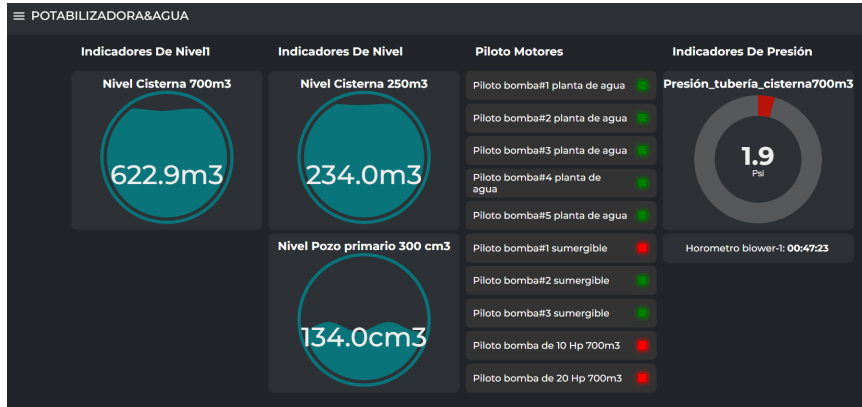


Figura 3.122: Dashboard Planta Potabilizadora.

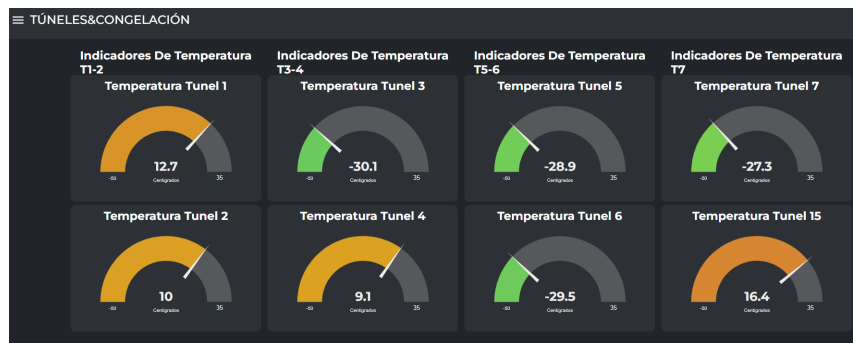


Figura 3.123: Dashboard Temperatura de los túneles.

- Se instala el disco duro
- Se instala el sistema operativo
- Se instala Node-Red
- Se instala el software Putty
- Se instala Linux administrador root como se muestra en la figura 3.126.
- Se instala iot2000 setup.
- Se cambia la dirección IP para ponerlo en red con el PLC Máster como se muestra en la figura 3.127.

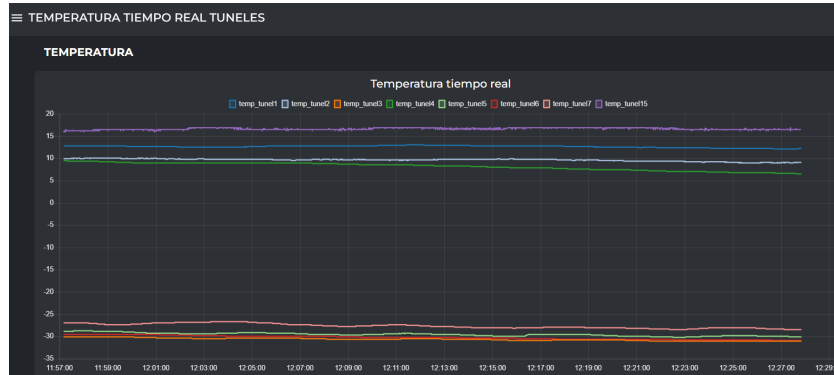


Figura 3.124: Dashboard Temperaturas tiempo real.

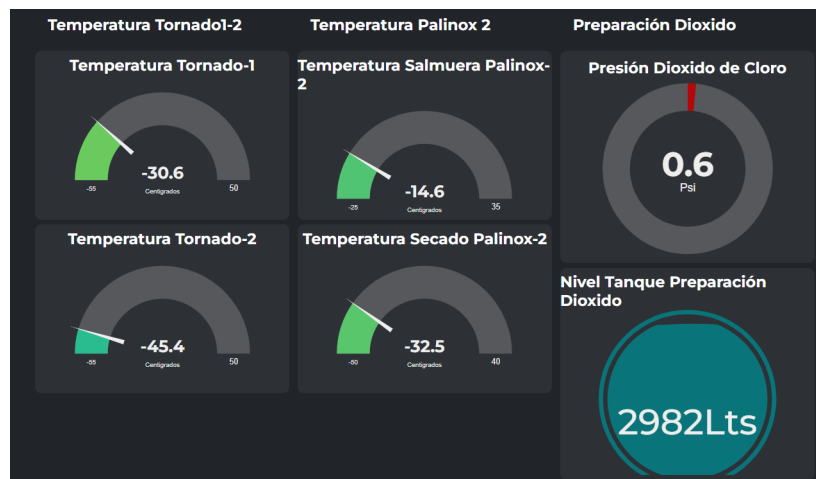


Figura 3.125: Dashboard Variables Galpón 6.

- Se reinicia el equipo IOT2040
- Se instala la tarjeta SD mínimo debe ser de 4 GB maximo 8GB.
- Se carga la imagen del sistema operativo en la tarjeta usando PC.SO.Yocto Linux.
- La tarjeta se formatea a Linux y no puede acceder con Windows.
- Se activa Node-Red

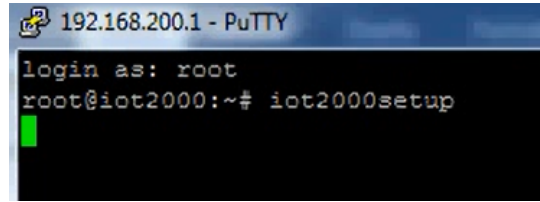


Figura 3.126: Administrador root.

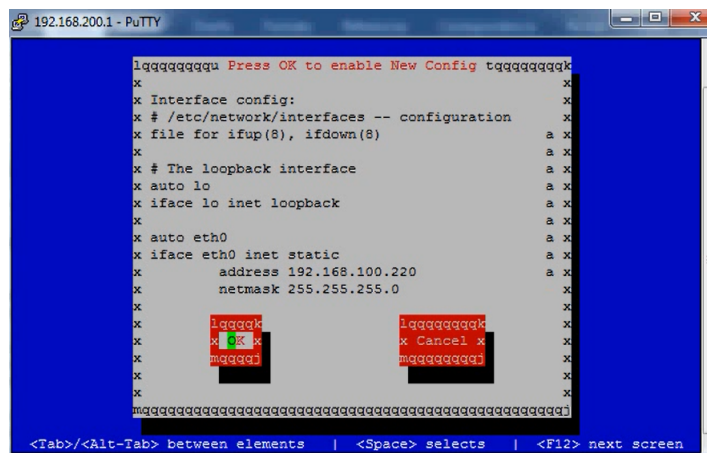


Figura 3.127: Configuración IP Putty.

- Se ingresa con Putty al IOT como se ve en la figura 3.128.
- Finalmente se instala los nodos por consola ingresando a la carpeta de nodos como se aprecia en la figura 3.129.
- Físicamente en la figura 3.130 se muestra la instalación del IOT2040 junto al PLC Máster S71500.

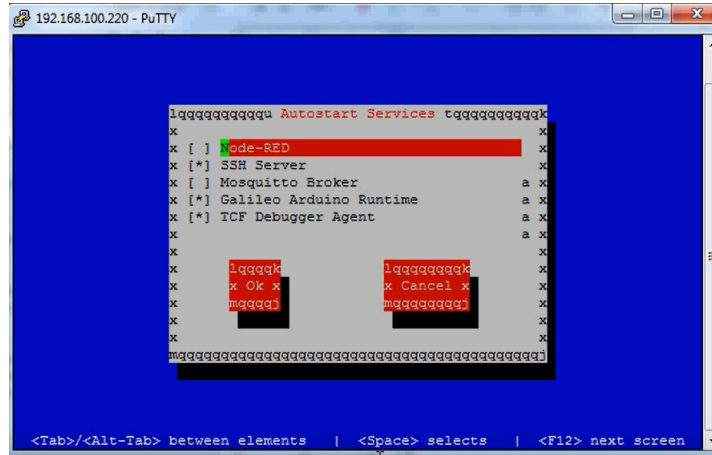


Figura 3.128: Activación librerías Putty.

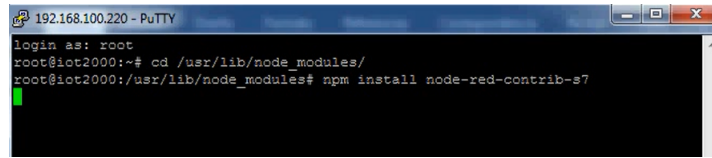


Figura 3.129: Instalación de nodos por consola.



Figura 3.130: Instalación IOT-2040 Siemens.

Capítulo 4

Resultados

En este capítulo se describe los resultados de la metodología empleada para el análisis de las variables dentro del proceso de empaquetado de camarón.

Gracias a los diferentes métodos utilizados en este proyecto y utilizando tecnología de punta e innovadora como es la Industria 4.0, se ha logrado sacar resultados óptimos en diferentes procesos con la finalidad de bajar los costos de producción o de repotenciar algún equipo o maquinaria.

Node-Red y sus paquetes de integración informática permite al usuario llevar históricos del comportamiento de ciertas variables y analizarlas en el tiempo con el fin de tomar acciones predictivas y optimizar los recursos ya sea en producción o en mantenimiento.

Mensajes instantáneos multimedia

En la parte de mensajes multimedia se ha facilitado el monitoreo de variables críticas a diferentes usuarios y únicamente creando el acceso desde un teléfono celular por la aplicación telegram.

Antes los operadores debían tomar datos desde diferentes sitios de la planta ,variables como niveles, presiones, temperaturas,etc.

En el actualidad se configura a un cierto grupos de usuarios las alarmas vía telegram para el monitoreo por mensajes de textos como se muestra en la figura 4.1.

Envío de alertas por correo electrónico

Para el personal mantenimiento en el área de planificación, jefaturas, etc, se ha diseñado un método realizar mantenimiento preventivo, en este caso como prueba piloto el mantenimiento de motores críticos o te potencias elevadas.

Toda planta industrial en general no tiene aún definido el método para

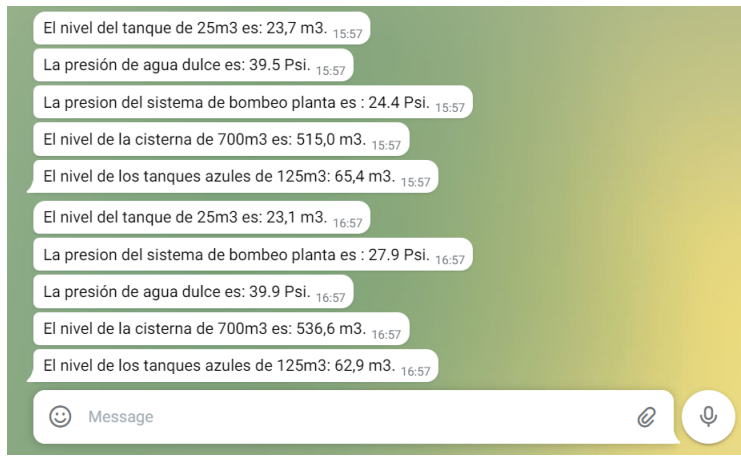


Figura 4.1: Alertas Telegram.

el mantenimiento preventivo de motores, ciertas empresas trabajan bajo frecuencias periódicas ya sean semestrales o anuales, este método no es confiable ya que dependen de otros factores como el de la demanda de producción, si es alta el motor trabajará a toda su capacidad, si la producción es baja el motor va a estar en espera.

Normalmente los tipos de software de mantenimiento llevan la frecuencia en el tiempo y no distinguen si el motor estuvo trabajando o no, es ahí cuando los costos de mantenimiento se elevan.

Como solución a este problema y con el afán de optimizar los recursos de un planta industrial se ha definido realizar un control de mantenimiento preventivo por horas de trabajo a los motores blowers de 50 HP del sistema de aireación de la Ptar.

Para ello fue necesario realizar los siguientes pasos que pueden ser replicados para diferentes equipos.

- Se realizó una lógica de programación en el Tia Portal en lenguaje SCL para el control de horas de trabajo de cada motor como se muestra en la figura 4.3 y en la figura 4.4.
- Se programó unas líneas de comandos en lenguaje de escalera para que una vez cumplido las 8000 horas de trabajo el blower 1 y 2, envíe un Bit al correo electrónico como se muestra en la figura 4.5 y sea una alerta para programar el respectivo mantenimiento. Este aviso se repetirá

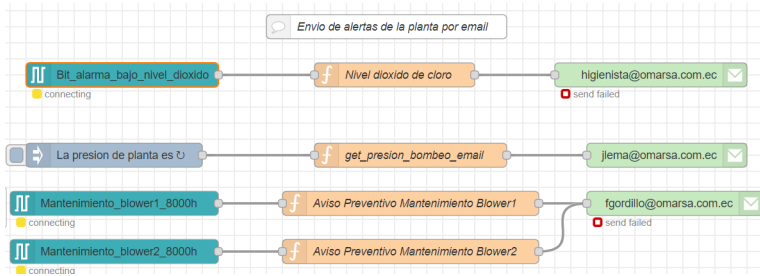


Figura 4.2: Nodos envío de correos electrónicos.

Nombre	Tipo de datos	Valor predet.	Remanencia
entradat2	Bool	false	Remanente
reseteot2	Bool	false	Remanente
pulsot2	Bool	false	Remanente
segundot2	Dint	0	Remanente
minutost2	Dint	0	Remanente
horast2	Dint	0	Remanente

```

1 IF #entradat2 THEN
2   IF #pulsot2 THEN
3     #segundot2 := #segundot2 + 1;
4     IF #segundot2 > 60 THEN
5       #segundot2 := 0;
6       #minutost2 := #minutost2 + 1;
7       IF #minutost2 >= 60 THEN
8         #minutost2 := 0;
9         #horast2 := #horast2 + 1;
10        END_IF;
11      END_IF;
12    END_IF;
13  END_IF;
14 IF #reseteot2 THEN
15   #segundot2 := 0;
16   #minutost2 := 0;
17   #horast2 := 0;
18 END_IF;
19

```

Figura 4.3: Lógica de programación conteo SCL.

cada 24 horas hasta que el mantenimiento del motor sea ejecutado, y se pueda encerrar.

- La configuración de los nodos se muestra en la figura 4.2 y el mensaje de alerta se muestra en la figura 4.6, cabe recalcar que si se quiere enviar a diferentes usuarios se debe crear un nodo semejante al creado.

Análisis de las temperaturas en los túneles de congelación

Siguiendo con los resultados obtenidos se realizó un análisis del comportamiento de las temperaturas en los túneles de congelación y se busca estructurar la eficiencias de los mismos.

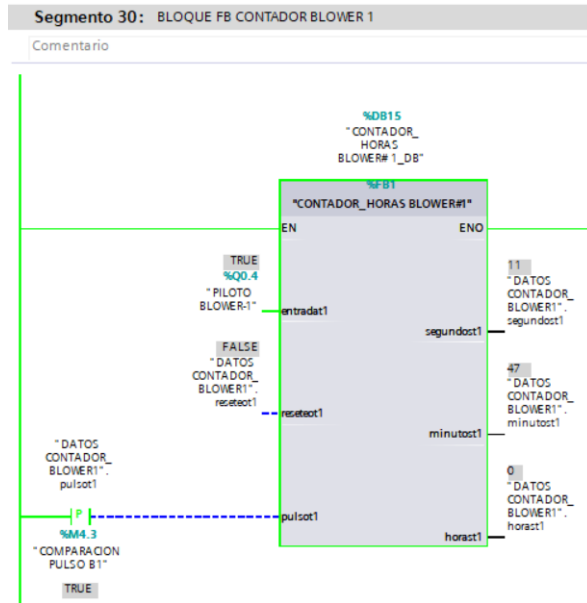


Figura 4.4: Bloque S7 Contador.

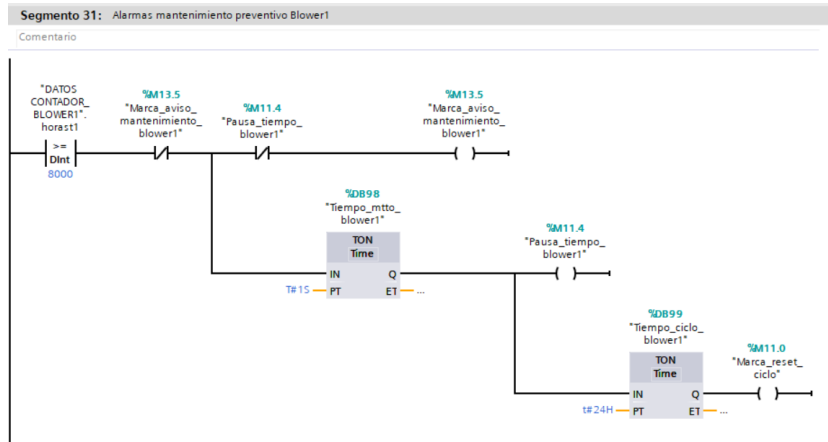


Figura 4.5: Instrucciones tia portal ciclos de envío alertas.

Todo el producto que sale de las máquinas clasificadoras de camarón son colocadas en cajas con una temperaturas promedio entre 7 a 10 grados centígrados como se muestra en la figura 4.8, estas cajas son almacenadas

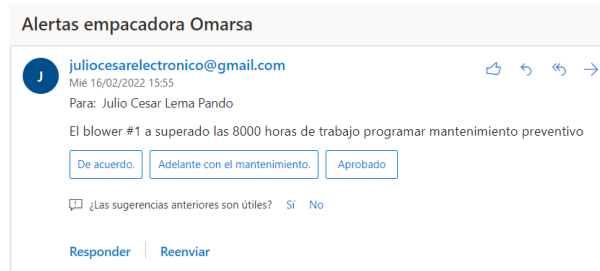


Figura 4.6: Avisos de alertas por correo electrónico.

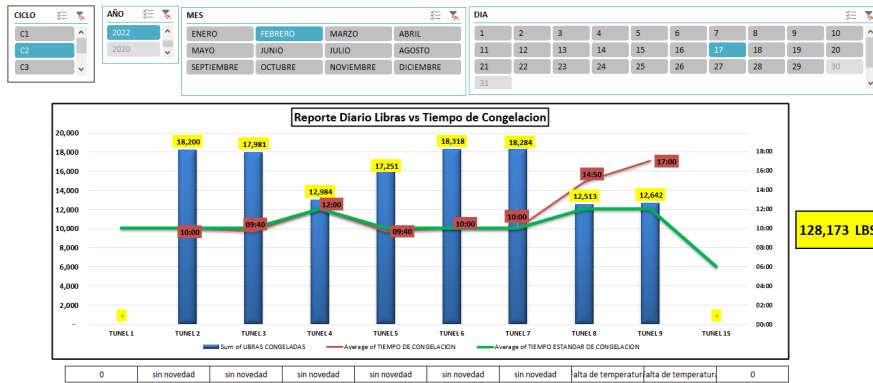


Figura 4.7: Resumen libras congeladas.

en coches y son trasladadas a los túneles de congelación, como se indica en la figura 4.9.

El análisis que hace producción en los tiempos de congelación vs libras congeladas no refleja el comportamiento de las temperaturas en el interior de los túneles y se nota en la figura 4.7.

Se diseña una base de datos para monitorear el comportamiento de la temperatura de cada túneles de congelación, con la ayuda de Nodo-Red se elabora un dashboard dinámico como se indica en la figura 4.10, donde el usuario puede setear la fecha que requiera revisar. El dashboard permite también revisar la curva de temperatura por túnel de congelación. En la tabla con los datos de producción se hace un comparativo de la curva de las temperaturas en el tiempo para verificar la eficacia de cada túnel.

Entre las ventajas de la creación de esta base de datos también brinda una oportunidad de mejora en la parte de mantenimiento para revisar la eficiencia



Figura 4.8: Temperatura de las cajas.



Figura 4.9: Coches ingreso a túneles de congelación.

de las unidades condensadoras, si un túnel de congelación no mantiene su curva de temperatura o tarda demasiado tiempo en sacar libras congeladas, se debe programar su revisión o en su lugar migrar por una unidad evaporativa con mas toneladas de refrigeración por hora.

En el análisis de congelación que se realizó los días del 16-18 de febrero del 2022 para los cuales se ha pedido información complementaria a producción

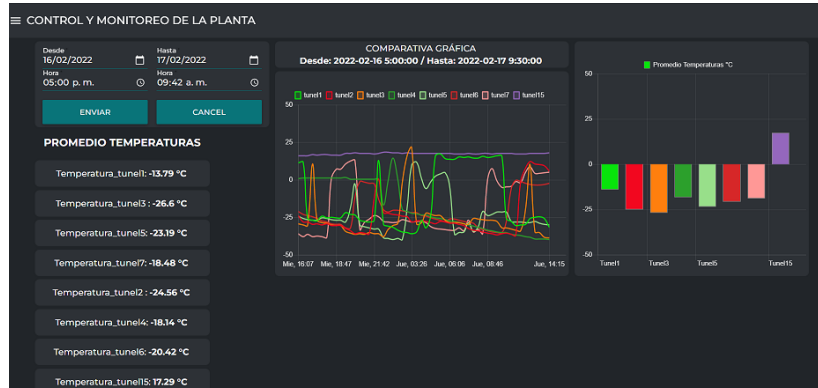


Figura 4.10: Curva de temperatura Túneles.



Figura 4.11: Curva de temperatura Túnel-1.

como el número de coches que ingresan, el número de libras congeladas, y las horas de congelación, el análisis que les falta es el comportamiento de la curva de temperaturas en el tiempo, dato que se genera en la base de datos implementada.

Se revisa las curvas de temperaturas en el túnel 1 figura 4.11 , en el túnel 2 figura 4.12, en el túnel 3 figura 4.13, el túnel 4 figura 4.14, en el túnel 5 figura 4.15, en el túnel 6 figura 4.16, en el túnel 7 figura 4.17.

El resultado del análisis se puede apreciar en la figura 4.18.

Para llevar a acabo este análisis se configuró en Node-red los nodos que se muestran en la figura 4.19

En el editor de JavaScript en el segmento de fecha y hora se configura los

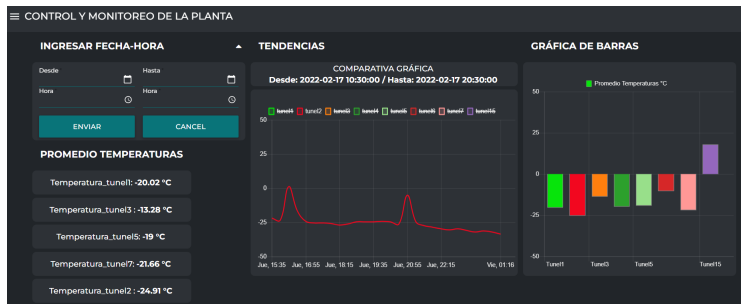


Figura 4.12: Curva de temperatura Túnel-2.

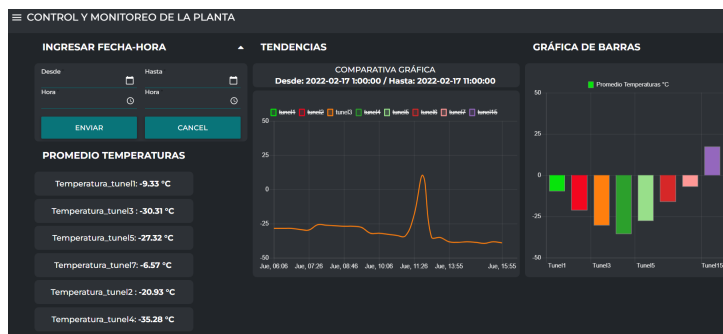


Figura 4.13: Curva de temperatura Túnel-3.



Figura 4.14: Curva de temperatura Túnel-4.

siguientes comandos como se muestra en la figura 4.20 y 4.21.

En el editor de JavaScript en la función Chart Prep se configura los

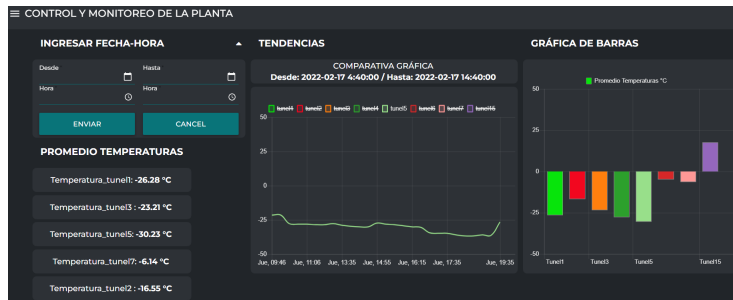


Figura 4.15: Curva de temperatura Túnel-5.



Figura 4.16: Curva de temperatura Túnel-6.

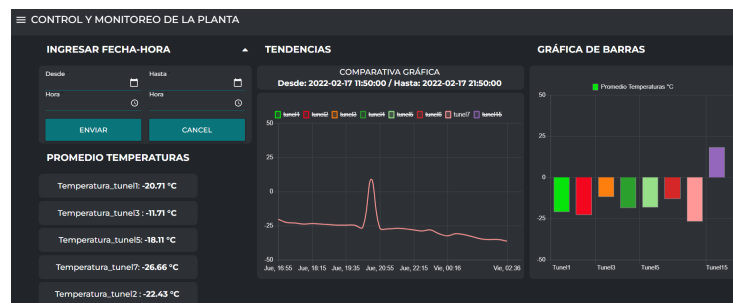


Figura 4.17: Curva de temperatura Túnel-7.

siguientes comandos como se muestra en las figuras 4.22,4.23,4.24, estos comandos se configuran para preparar las tendencias en el dashboard de

CUADRO DE EFICIENCIAS TUNELES DE CONGELACION												
NOMBRE TUNEL	DIA-1	DIA-2	MES	AÑO	CAPACIDAD COCHES	#COCHES INGRESADOS	HORA INICIO	HORA FINAL	TOTAL HORAS	LIBRAS CONGELADAS	PROM TEMP	PROM TEMP
TUNEL-1	16	2	2022		30	22	13:50	23:00	09:10	18239	-26.87	27.77
TUNEL-2	16	17	2	2022	30	30	19:00	06:00	11:00	17547	-29.97	-27.44
TUNEL-3	16		2	2022	30		09:40	21:00	11:20	17712	-30.53	-30.42
TUNEL-4												-26.11
TUNEL-5	16		2	2022	30	30	11:10	22:00	10:50	18540	-29.19	-29.71
TUNEL-6	16	17	2	2022	30	30	17:30	04:30	11:00	18904	-26.24	-23.07
TUNEL-7	16	17	2	2022	30	30	16:40	02:00	09:20	17871	-29.24	-25.45

CUADRO DE EFICIENCIAS TUNELES DE CONGELACION												
NOMBRE TUNEL	DIA-1	DIA-2	MES	AÑO	CAPACIDAD COCHES	#COCHES INGRESADOS	HORA INICIO	HORA FINAL	TOTAL HORAS	LIBRAS CONGELADAS	PROM TEMP	PROM TEMP
TUNEL-1	17		2	2022	30	22	07:10	16:40	09:30	14324	-28.67	-28.67
TUNEL-2	17		2	2022	20	30	10:30	20:30	10:00	18200	-24.91	-24.91
TUNEL-3	17		2	2022	30	30	01:00	11:00	10:00	18176	-30.31	-30.31
TUNEL-4	17	18	2	2022	21	21	13:50	01:50	12:00	12984	-26.11	-26.11
TUNEL-5	17		2	2022	30	30	04:40	14:40	10:00	18611	-30.23	-30.23
TUNEL-6	17	18	2	2022	30	30	14:50	00:50	10:00	18318	-19.90	-19.90
TUNEL-7	17		2	2022	30	30	11:50	21:50	10:00	18284	-21.66	-21.66

Figura 4.18: Cuadro de eficiencias túneles.

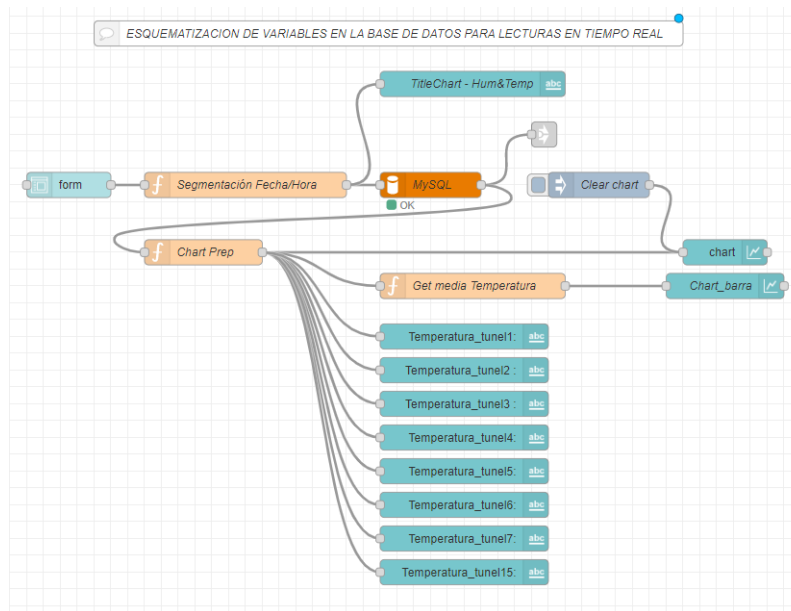


Figura 4.19: Nodos base de datos temperatura.

control y monitoreo de planta y para ello toca configurar la ventana chart barra para que muestre los valores entre -50 a 50 grados centígrados como se muestra en la figura 4.25.

Como resultado final se pudo apreciar que el túnel 3 es el mas eficiente y túnel 6 es la mas deficiente, por lo que se debe realizar la acción correctiva para revisar si tiene algún problema en la unidad evaporativa o las puertas no esta sellando bien.

```

Edit function node > JavaScript editor

1 //Preparación de fecha inicial
2 msg.fecha_from = msg.payload.fecha_from.substring(0,10);
3 msg.h_from = (msg.payload.hora_from.substring(11,13)-5).toString();
4
5 //Si el usuario elige horas en P.M
6 let hora_from = Number(msg.h_from);
7 if (hora_from < 0) {
8     hora_from += 24;
9     msg.h_from = (hora_from).toString();
10 }
11 msg.min_from = msg.payload.hora_from.substring(14,16);
12 msg.sec_from = msg.payload.hora_from.substring(17,19);
13
14 msg.hora_from = msg.h_from+":"+msg.min_from+":"+msg.sec_from;
15
16 let tiempo_from = msg.fecha_from+" "+msg.hora_from;
17 msg.tiempo_from = tiempo_from;
18 flow.set('startData',msg.tiempo_from);
19
20 //Preparación de fecha final
21 msg.fecha_to = msg.payload.fecha_to.substring(0,10);
22 msg.h_to = (msg.payload.hora_to.substring(11,13)-5).toString();
    
```

Figura 4.20: Editor fecha y hora parte 1.

```

24 //Si el usuario elige horas en P.M
25 let hora_to = Number(msg.h_to);
26 if (hora_to < 0) {
27     hora_to += 24;
28     msg.h_to = (hora_to).toString();
29 }
30
31 msg.min_to = msg.payload.hora_to.substring(14,16);
32 msg.sec_to = msg.payload.hora_to.substring(17,19);
33
34 msg.hora_to = msg.h_to+":"+msg.min_to+":"+msg.sec_to;
35
36 let tiempo_to = msg.fecha_to+" "+msg.hora_to;
37 msg.tiempo_to = tiempo_to;
38 flow.set('endData', msg.tiempo_to);
39
40 //Fin de flujo de datos
41 let state_sum = msg.payload.sum;
42 let state_avg = msg.payload.avg;
43 let state_max = msg.payload.max;
44 let state_min = msg.payload.min;
45
46 msg.payload = { 'resultado': {
47     msg.payload: "SELECT * FROM temp_tendencias WHERE fecha_hora >='"+ tiempo_from +"'" AND fecha_hora <='"+ tiempo_to +"'"// ORDER BY sensor_id DESC LIMIT '"+ longitud;
48 }
49 }
50 let text_title = `Resumen $[msg.tiempo_from] / hasta $[msg.tiempo_to]`;
51 msg.setTitle(text_title);
52 return msg;
    
```

Figura 4.21: Editor fecha y hora parte 2.

```

Edit function node > JavaScript editor

1 //var tzoffset = (new Date()).getTimezoneOffset() * 60000; //Offset in milliseconds
2 let cantidad = msg.payload.length;
3 let m = {};
4 const dias = [
5     'Dom',
6     'Lun',
7     'Mar',
8     'Mie',
9     'Jue',
10    'Vie',
11    'Sab',
12 ];
13 let media_tune1 = 0;
14 let media_tune2 = 0;
15 let media_tune3 = 0;
16 let media_tune4 = 0;
17 let media_tune5 = 0;
18 let media_tune6 = 0;
19 let media_tune7 = 0;
20 let media_tune15 = 0;
21
22 m.labels = []; //Etiquetas en el eje 'X'
23
24 m.series = ['tune1','tune2','tune3','tune4','tune5','tune6','tune7','tune15'];
25
    
```

Figura 4.22: Función preparar tendencias parte 1.

```

26 let linea_1 = [];//tunel1
27 let linea_2 = [];//tunel2
28 let linea_3 = [];//tunel3
29 let linea_4 = [];//tunel4
30 let linea_5 = [];//tunel5
31 let linea_6 = [];//tunel6
32 let linea_7 = [];//tunel7
33 let linea_15 = [];//tunel15
34
35
36 let arrayLength = msg.payload.length;
37
38 for (let i = 0; i < arrayLength; i++) {
39   let fechahora_1 = msg.payload[i].fecha_hora;
40   let cadenaFechaHora = fechahora_1.toISOString().slice(0, -5).replace('T', ' ');
41
42   const numeroDia = new Date(cadenaFechaHora).getDay();
43   const nombreDia = dias[numeroDia];
44
45   let fecha_hora = fechahora_1.toISOString().slice(10, -8).replace('T', ' ');
46
47   m.labels.push(nombreDia + " " + fecha_hora);
48   linea_1.push((msg.payload[i].temp_tunel1));
49   linea_2.push((msg.payload[i].temp_tunel2));
50   linea_3.push((msg.payload[i].temp_tunel3));
51   linea_4.push((msg.payload[i].temp_tunel4));
52   linea_5.push((msg.payload[i].temp_tunel5));
53   linea_6.push((msg.payload[i].temp_tunel6));
54   linea_7.push((msg.payload[i].temp_tunel7));
55   linea_15.push((msg.payload[i].temp_tunel15));

```

Figura 4.23: Función preparar tendencias parte 2.

```

56   media_tunel1 = media_tunel1+msg.payload[i].temp_tunel1;
57   media_tunel2 = media_tunel2+msg.payload[i].temp_tunel2;
58   media_tunel3 = media_tunel3+msg.payload[i].temp_tunel3;
59   media_tunel4 = media_tunel4+msg.payload[i].temp_tunel4;
60   media_tunel5 = media_tunel5+msg.payload[i].temp_tunel5;
61   media_tunel6 = media_tunel6+msg.payload[i].temp_tunel6;
62   media_tunel7 = media_tunel7+msg.payload[i].temp_tunel7;
63   media_tunel15 = media_tunel15+msg.payload[i].temp_tunel15;
64
65 }
66 }
67
68 media_tunel1 = parseFloat((media_tunel1/arrayLength).toFixed(2));
69 media_tunel2 = parseFloat((media_tunel2/arrayLength).toFixed(2));
70 media_tunel3 = parseFloat((media_tunel3/arrayLength).toFixed(2));
71 media_tunel4 = parseFloat((media_tunel4/arrayLength).toFixed(2));
72 media_tunel5 = parseFloat((media_tunel5/arrayLength).toFixed(2));
73 media_tunel6 = parseFloat((media_tunel6/arrayLength).toFixed(2));
74 media_tunel7 = parseFloat((media_tunel7/arrayLength).toFixed(2));
75 media_tunel15 = parseFloat((media_tunel15/arrayLength).toFixed(2));
76
77 let array_media = [media_tunel1,media_tunel2,media_tunel3,media_tunel4,media_tunel5,media_tunel6,media_tunel7,media_tunel15];
78
79
80 m.data = [linea_1, linea_2, linea_3, linea_4, linea_5, linea_6, linea_7, linea_15];
81
82 return (payload[m],topic:msg.topic,media:array_media);
83

```

Figura 4.24: Función preparar tendencias parte 3.

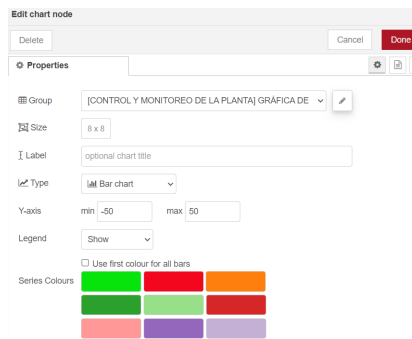


Figura 4.25: Edición de seteos barra de temperatura.

Capítulo 5

Conclusiones

Se analizó el estado del arte y el desarrollo de HMIs Industriales con diferentes metodologías y lenguajes de programación.

Se diseñó una plataforma Web con lenguaje de programación nativo en JavaScript y utilizando el servidor embebido en los PLCs Simatic S7.

Se diseñó una plataforma de HMIs industriales en Visual Studio sin OPCS, ni licencias y con las mismas virtudes incluso superiores al de un Scada Profesional, ya que permite integrar infinidad de tags a cero costo, y se consigue únicamente con horas de ingeniería. Otra de las ventajas del desarrollo de esta solución es que permite instalar la aplicación a un número ilimitado de usuarios y no es necesario instalar Visual Studio en el computador ya que la instalación es muy similar a la partición de un disco.

Se diseñó Dashboards dinámicos en Node-Red para el monitoreo de variables ya que está optimizado para poder tratar múltiples conexiones de una manera óptima.

Se desarrolló el monitoreo por mensajes instantáneos multimedia Telegram donde se enlaza las variables S7 del PLC Master con los nodos Telegram posterior se crea un Bot y se programó usando Node-RED.

Se desarrolló una base de datos utilizando la plataforma compacta WampServer que es un paquete informático que contiene Apache que es un servidor web HTTP, contiene a My SQL que es un gestor de base datos, y contiene a phpMyAdmin que es una aplicación Web para administrar base de datos, esto se lo integro con Nodo-Red y se pudo hacer un análisis de eficiencia a los túneles de congelación mediante dashboard dinámicos.

PhpMyAdmin facilitó a los usuarios de revisar estadística de curvas, como se indica en la figura 5.1, donde se realizó un análisis del comportamiento de los niveles de las cisternas de agua en la planta Omarsa.

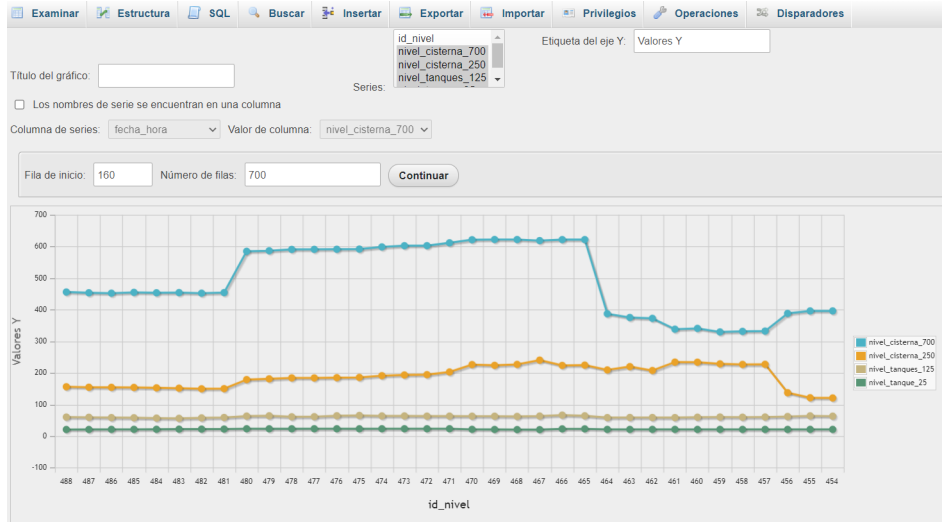


Figura 5.1: Curva de Niveles Cisternas.

PhpMyAdmin es un software gratuito y con conocimientos previos de lenguajes de programación es JavaScript su plataforma y entorno es muy amigable.

Capítulo 6

Recomendaciones

Como recomendación utilizar la metodología Web-HTML para proyectos pequeños o de variables críticas en el proceso que puedan ser monitoreadas desde un móvil o tablet.

Para proyectos donde impliquen mas variables utilizar el entorno de Visual Studio la versatilidad de trabajar con plantillas hace mas fácil al programador integrar soluciones y su sistema que no requiere licencias ni OPCs lo hace muy atractivo frente a un Scada profesional que tienen características semejantes pero a costos inaccesibles para ciertas industrias.

Para trabajar con variables que requieran indicadores, base de datos, mensajería de correo electrónico y multimedia lo recomendable es trabajar con Node-Red que gracias a por su plataforma multifunción permite interconectar y enlazar nodos de bases de datos, mensajería instantánea, etc, y sacar indicadores de producción,

Es recomendable que para para el desarrollo de las variables a monitorear y la integración de los indicadores de producción se trabaje en una PC, hasta afinar detalles, cuando ya se tenga culminado el proyecto se debe cargar a la pasarela IOT Siemens en el caso de este proyecto el IOT2040.

Capítulo 7

Glosario

AWS Amazon Web Server.

TCP Tráfico de baja prioridad.

RT Tráfico real time.

IDE Desarrollo integrado intuitivo.

IoT Internet Of Everything (Internet de las cosas).

IP Protocolo de Internet

PLC Controlador lógico programable.

HMI Interfase hombre máquina.

UDT Estructuras de datos simples.

OPC Protocolo abierto de comunicación.

API Interfaz de programación de aplicaciones.

JSON Formato de texto sencillo para el intercambio de datos.

KPI Key Performance Indicator (Indicador clave de rendimiento).

TAGS Etiquetas con palabras claves.

SCADA Control supervisor y adquisición de datos.

MySQL My Structured Query Language (Lenguaje de Consulta Estructurado).

HTTP Lenguaje que media entre las peticiones del cliente y las respuestas del servidor en Internet.

Modbus Protocolo de comunicaciones, basado en la arquitectura maestro/esclavo o cliente/servidor.

Token Componente léxico es una cadena de caracteres.

VLAN Red de área local virtual.

Bibliografía

Algoritmo de supervisión del protocolo smtp del sistema de correo electrónico activo. In *2009 2nd IEEE International Conference on Broadband Network Multimedia Technology*. doi: 10.1109/ICBNMT.2009.5348490.

Una nueva arquitectura de aplicaciones web: la arquitectura widget/servidor. In *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*. doi: 10.1109/ICNIDC.2010.5657919.

Detección de clones en lenguajes de programación microsoft .net. In *19.ª Conferencia de trabajo sobre ingeniería inversa de 2012*. doi: 10.1109/WCRE.2012.50.

Exploración de la migración de aplicaciones al entorno de la nube. In *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*. doi: 10.1109/CONFLUENCIA.2016.7508097.

Internet de las cosas usando node-red y alexa. In *17º Simposio Internacional sobre Comunicaciones y Tecnologías de la Información (ISCIT) de 2017*. doi: 10.1109/ISCIT.2017.8261194.

Monitoreo de consumo de agua de iot amp; sistema de alerta. In *Conferencia Internacional de Ingeniería Eléctrica e Informática (ICELTIC) de 2018*. doi: 10.1109/ICELTICS.2018.8548930.

Un panel de control basado en nodered para implementar canalizaciones sobre la infraestructura de iot. In *Conferencia internacional IEEE de 2020 sobre informática inteligente (SMARTCOMP)*. doi: 10.1109/SMARTCOMP50058.2020.00036.

Diseño de una red industrial ethernet con autómatas simatic s7-1200.

M. M. Carbonell. Phpmyadmin. *Recuperado de <http://personales.upv.es/moimacar/download/servidores/phpmyadmin.pdf>, el, 1, 2018.*

- J. Coronel, F. Blanes, P. Pérez, G. Benet, and J. Simó. Arquitectura de control distribuida usando nodos empotrados con rt-linux sobre le protocolo de comunicaciones scocan. *XXV Jornadas de Automática. Alicante (España)*, 2004.
- M. Corral González et al. Diseño y desarrollo de un sistema hmi para una aplicación de industria 4.0. 2021.
- S. N. Z. Naqvi, S. Yfantidou, and E. Zimányi. Time series databases and influxdb. *Studienarbeit, Université Libre de Bruxelles*, 12, 2017.
- E. Ollora Zaballa et al. Implementación de una plataforma de análisis y visualización de datos iot multi-protocolo en tiempo real sobre node-red. 2021.
- B. Oviedo, L. Suarez Litardo, E. Zhuma Mera, A. Puris, and R. Hernández. Visualizador de tráfico de red de comunicación basadas en la arquitectura tcp/ip. *Revista Universidad y Sociedad*, 11(2):193–202, 2019.
- P. Pérez Chueca. *Automatización de una planta de tratamientos de aguas residuales*. PhD thesis, Universitat Politècnica de València, 2017.
- E. S. Prieto-Reinoso and J. B. Cabrera-Mejía. Chatbot en telegram para consultas de casos covid-19 en el ecuador. *Dominio de las Ciencias*, 6(4): 987–1000, 2020.
- J. A. C. PULIDO. *METODOLOGÍA DE DESARROLLO DE APLICACIONES SOBRE LA PLATAFORMA MICROSOFT .NET FRAMEWORK SDK*. PhD thesis, UNIVERSIDAD AUSTRAL DE CHILE, 2003.
- F. Rivalta. *Studio ed implementazione di un posizionatore interfacciato su rete ProfiNET*. PhD thesis.
- J. Rodríguez De La Rosa et al. Desarrollo de sistema scada para la automatización de células de fabricación. 2021.
- V. Tapia. Industria 4.0–internet de las cosas. *UTCienciaCiencia y Tecnología al servicio del pueblo"*, 1(1):51–60, 2017.