

Universidad Católica De Santa María

Facultad De Ciencias e Ingenierías Físicas Y Formales

Escuela Profesional De Ingeniería Electrónica



**“AUTOMATIZACIÓN DE PLANTA PILOTO PARA LA REUTILIZACIÓN
DE AGUA POR FILTRADO Y CONDENSACIÓN; UTILIZANDO
PROGRAMACIÓN PARALELA Y SECUENCIAL, MEDIANTE CONTROL
FUZZY”**

Tesis presentada por el Bachiller:

Allasi Gonzales, Jonathan Richard

Para Optar el Título Profesional de:

Ingeniero Electrónico

Asesor de Tesis:

Ing. Rivera Chávez, Víctor Hugo

AREQUIPA – PERÚ

2017

DEDICATORIA

A Dios, a la Virgencita de Chapi por recibir cada día su bendición y sobre todo darme la fuerza para seguir adelante.

A mis padres Jesús Allasi y Juana Gonzales por acompañarme en todos los momentos de mi vida, buenos y malos.

A los Ingenieros de la carrera Profesional de Ingeniería Electrónica, por sus enseñanzas que impartieron en las aulas de mi alma Mater la Universidad Católica de Santa Maria.

A mis amigos Juan Pablo Paredes, Cristian Bustamante, Ryan Casapia, Janet Uribarri, Yuri Ruiz, Grivas Girón, Moisés Vilca, Ricardo Fajardo, Ronald Torres, Edson Cartagena, Isaac Sánchez, Marco Quispe, Robert Chuco, Percy Pari, por sus enseñanzas y consejos que me sirvieron para la vida.

INTRODUCCIÓN

Hoy en día el desecho de las aguas al sistema de drenaje es un problema, debido a las grandes cantidades que desperdiciamos cada día, la falta de información y de sistemas que permitan aprovechar al máximo el agua hace que cada día la demanda de este recurso sea mayor.

La reutilización del agua deberá ser parte de la solución al problema del recurso hídrico, ya que si la aprovechamos podemos generar grandes ahorros tanto para los hogares, industrias y como para el abastecimiento de nuestras represas. Pero para ello si pensamos en la reutilización no solo basta con almacenar en tanques y después utilizarlas, ya que siendo así estamos siendo parte del aumento de la contaminación del suelo, debido a que el agua no ha tenido un tratamiento.

Es por esto que necesitamos plantas que puedan ser de gran ayuda para el tratamiento del agua y así reutilizarla, por lo tanto la reutilización del agua lo podemos emplear desde nuestras casas, utilizando filtros con materia prima y con la ayuda de la energía solar para su purificación. Y de la mano con la tecnología que hoy en día se hace más accesible, permitiendo así que los sistemas sean confiables.

En el mercado se cuenta con purificadores de alto costo, nosotros planteamos una alternativa para el tratamiento del agua y así poder ser reutilizada, yendo de la mano con la tecnología y así ser partícipes de la solución al problema del agua.

RESUMEN

El presente proyecto es una Planta Piloto para la Reutilización de aguas, utilizando un filtro natural a base de materia prima y un filtro a base de energía solar, en la que aprovecharemos el recurso del sol para la evaporación del agua. El control automatizado se desarrolló a base de un Arduino Mega 2560 y un PLC Siemens S7-1200, en la que ambos dispositivos mantendrán una comunicación constante a base del protocolo Modbus TCP/IP.

Al utilizar ambos dispositivos realizamos dos tipos de programación una Secuencial dada por el Arduino y otra Paralela dada por el PLC. Respecto a la programación del Arduino nos enfocamos para controlar al seguidor solar para un mejor aprovechamiento de la energía solar y también para su medición de la Temperatura en el Filtro y del Sistema, como también para la medición del caudal que pasara a la Etapa de Evaporación y Condensación de la Planta.

Utilizamos un Control Fuzzy para nuestro Filtro por evaporación, en la que nos basaremos en el lenguaje de programación STL. Los datos que será enviado por el Arduino, lo utilizaremos para nuestra programación, de tal manera demostraremos como dos controladores se pueden comunicar de manera eficiente y en tiempo real.

Se desarrolló su SCADA aprovechando el Web Server del PLC S7-1200, de tal manera utilizando el código de HTML, donde se desarrolló la visualización de los parámetros de la planta para poder ver a través del internet. El fin fue que al ser una plataforma libre, nos evitamos de las licencias para un SCADA.

El proyecto logro las expectativas de la Institución de CETEMIN, los que invirtieron en el proyecto con el fin de que sea aprovechado para sus instalaciones y a la vez que sea didáctico para la enseñanza de sus alumnos.

Palabras Claves: Control Fuzzy.

ABSTRACT

The present project is a Pilot Plant for the Reuse of water, using a natural filter based on raw material and a filter based on solar energy, in which we will take advantage of the sun's resource for the evaporation of water. The automated control was developed based on an Arduino Mega 2560 and a Siemens PLC S7-1200, in which both devices will maintain a constant communication based on the protocol Modbus TCP / IP.

When using both devices we perform two types of programming a Sequential given by the Arduino and another Parallel given by the PLC. Regarding the programming of the Arduino we focus on controlling the solar tracker for a better use of solar energy and also for its measurement of the Temperature in the Filter and the System, as well as for the measurement of the flow that will pass to the Evaporation Stage and Condensation of the Plant.

We use a Fuzzy Control for our Evaporative Filter, in which we will be based on the STL programming language. The data to be sent by the Arduino, we will use it for our programming, so we will demonstrate how two controllers can communicate efficiently and in real time.

Its SCADA was developed taking advantage of the Web Server of the S7-1200 PLC, in such a way using the code of HTML, where it was developed the visualization of the parameters of the plant to be able to see through Internet. The end was that being a free platform, we avoided the licenses for a SCADA.

The project met the expectations of the CETEMIN Institution, which invested in the project in order to be used for its facilities and at the same time it was didactic for the teaching of its students.

Keywords: Fuzzy control.

INDICE GENERAL

INTRODUCCIÓN	ii
RESUMEN	iii
ABSTRACT	iv
INDICE GENERAL.....	v
INDICE DE FIGURAS	viii
INDICE DE TABLAS	x
CAPÍTULO 1	1
PLANTEAMIENTO METODOLOGICO	1
1.1. DESCRIPCIÓN DEL PROBLEMA	1
1.2. JUSTIFICACIÓN.....	3
1.3. OBJETIVOS.....	3
1.3.1. OBJETIVO PRINCIPAL	3
1.3.2. OBJETIVOS ESPECÍFICOS.....	3
1.4. HIPÓTESIS.....	3
1.5. ALCANCE DEL PROYECTO	4
1.6. VARIABLE	4
1.6.1. VARIABLES INDEPENDIENTES.....	4
1.6.2. VARIABLES DEPENDIENTES	4
CAPÍTULO 2	5
MARCO TEÓRICO	5
2.1. ENERGIA SOLAR EN EL PERU ¹	5
2.1.1. ENERGÍA CALORIFÍCA ³	9
2.1.2. TRANSMISIÓN DE LA ENERGÍA TÉRMICA ³	10
2.2. FILTROS DE AGUA DE MATERIA PRIMA ⁴	11
2.3. PROGRAMACIÓN SECUENCIAL Y PARALELA	13
2.3.1. PROGRAMACIÓN SECUENCIAL	13
2.3.2. PROGRAMACIÓN EN PARALELO	13
2.4. ARDUINO	14

2.4.1.	ARDUINO MEGA	14
2.4.2.	MODULO SHIELD ETHERNET ⁷	17
2.5.	SENSOR DE FLUJO ⁸	19
2.6.	SENSOR LDR (Light Dependent Resistor)	20
2.7.	SENSOR DE TEMPERATURA LM35 ⁹	21
2.8.	VALVULAS SELENOIDES ¹⁰	21
2.9.	PLC S7-1200 ¹¹	22
2.9.1.	COMUNICACIÓN MODBUS TCP/IP ¹²	24
2.10.	LÓGICA DIFUSA ¹³	26
2.10.1.	CONTROL FUZZY ¹³	29
2.10.2.	FUZIFICACIÓN	31
2.10.3.	INFERENCIA – MANDAMI ¹³	31
2.10.4.	DEFUZIFICACIÓN – MÉTODO MANDAMI ¹³	33
2.10.5.	MÉTODO DE TAKAGI – SUGENO ¹⁴	34
2.11.	SCADA POR WEB SERVER	36
2.11.1.	WEB SERVER DEL S7-1200	36
2.11.2.	HTML ¹⁵	38
2.11.3.	JAVASCRIPT ¹⁶	39
	CAPÍTULO 3	41
	DESARROLLO DE INGENIERIA DE LA PLANTA PILOTO	41
3.1.	DIAGRAMA DEL PROCESO	41
3.2.	CONTROL DE NIVEL DE TANQUES	42
3.2.1.	CONTROL DE NIVEL DEL FILTRO 01	42
3.2.2.	CONTROL DE NIVEL DEL TANQUE N° 2, TANQUE N°3	45
3.2.3.	CONTROL DE NIVEL DEL FILTRO 02	46
3.3.	INTERCAMBIADOR DE CALOR	48
3.4.	SEGUIDOR SOLAR	48
3.4.1.	SENSOR DE LUZ O LDR	49
3.4.2.	MOTORES PARA EL MOVIMIENTO DEL SEGUIDOR SOLAR	51
3.4.2.1.	CIRCUITO PARA EL ACCIONAMIENTO DE LOS MOTORES	51
3.4.2.2.	CIRCUITO DE LA ETAPA DE CONTROL PARA LOS MOTORES	53

3.4.3.	SENSOR DE TEMPERATURA.....	54
3.4.4.	SENSORES FINES DE CARRERA	55
3.4.5.	SENSOR DE FLUJO.....	56
3.5.	IMPLEMENTACIÓN DEL PROGRAMA DE LA PLANTA PILOTO.....	56
3.6.	PROGRAMA EN EL ARDUINO MEGA 2560.....	57
3.6.1.	CONFIGURACIÓN DE LAS LIBRERIAS PARA LA COMUNICACIÓN MODBUS	59
3.6.2.	DECLARACIÓN DE LAS VARIABLES.....	62
3.6.3.	PROGRAMA DEL SEGUIDOR SOLAR	63
3.6.4.	PROGRAMA PARA MEDICIÓN DE TEMPERATURA.....	65
3.6.5.	PROGRAMA DE LECTURA DEL FLUJOMETRO	66
3.7.	DESARROLLO DEL PROGRAMA EN EL PLC SIEMENS S7-1200.....	67
3.7.1.	COMUNICACIÓN MODBUS DEL S7-1200 CON EL ARDUINO.....	69
3.7.2.	DECLARACIÓN DE VARIABLES EN BLOQUE DE DATOS PARA EL SCADA.....	72
3.7.3.	PRIMERA ETAPA: CONTROL DEL PRIMER FILTRADO DE LA PLANTA.....	74
3.7.4.	SEGUNDA ETAPA: CONTROL FUZZY CON LENGUAJE STL PARA EL FILTRADO POR CONDENSACIÓN	77
3.7.5.	SEGUNDA ETAPA: CONTROL CON GRAFCET Y LADDER PARA EL FILTRADO POR CONDENSACIÓN	94
3.8.	IMPLEMETACIÓN DEL SCADA EN HTML	104
3.9.	SISTEMA COMPLETO EN FUNCIONAMIENTO.....	115
	CONCLUSIONES	116
	RECOMENDACIONES	117
	BIBLIOGRAFIA	118
	ANEXOS	121

INDICE DE FIGURAS

Fig. 2. 1 Potencial de Energía Solar en el Perú	8
Fig. 2. 2 Sistema Termo Solar.....	9
Fig. 2. 3 Transmisión de Energía Térmica	11
Fig. 2. 4 Filtro de Arena y Grava	13
Fig. 2. 5 Tarjeta de Arduino Mega.....	17
Fig. 2. 6 Modulo de Shiefl Ethernet.....	18
Fig. 2. 7 Modulo de Shiefl Ethernet.....	20
Fig. 2. 8 Sensor LDR.....	20
Fig. 2. 9 Valvula Selenoide	22
Fig. 2. 10 PLC S7-1200.....	23
Fig. 2. 11 Tipos de Conexión de Comunicación.....	26
Fig. 2. 12 Conjuntos Difusos	27
Fig. 2. 13 Función de Membresía.....	27
Fig. 2. 14 Formula para el Cálculo de Membresías del Triangulo.....	28
Fig. 2. 15 Formula para el Cálculo de Membresías del Triangulo.....	28
Fig. 2. 16 Formula para el Cálculo de Membresías del Trapezoide	29
Fig. 2. 17 Estructura de un Controlador Difuso	30
Fig. 2. 18 Método de Inferencia Máximo – Mínimo (Mandami)	33
Fig. 2. 19 Método de Centroide (Mandami).....	34
Fig. 2. 20 Funcionamiento del Sistema Difuso Sugeno.....	35
Fig. 2. 21 Visualización de un SCADA a través del Web Server	37
Fig. 3. 1 Diagrama de Bloques del Sistema	41
Fig. 3. 2 Diagrama del Diseño a Desarrollar	41
Fig. 3. 3 Los Tanques 01, Filtro 01 y Tanque 02.....	42
Fig. 3. 4 a) Capa de Cascajo b) Capa de Gasa y Algodón	43
Fig. 3. 5 Soporte para el Filtro 01.....	43
Fig. 3. 6 Montaje del Sensor de Nivel The Probe.....	44
Fig. 3. 7 Acondicionamiento para el Sensor Ultrasónico PROBE	45
Fig. 3. 8 Conexión para la medición del nivel.....	45
Fig. 3. 9 Electrodo para la Medición del Nivel	46
Fig. 3. 10 Filtro 02 para la Evaporación del Agua	47
Fig. 3. 11 Intercambiador de Calor.....	48
Fig. 3. 12 Estructura para Seguidor Solar	49
Fig. 3. 13 Sensores de Luz para Seguidor Solar.....	50
Fig. 3. 14 Circuito para el Sensor LDR.....	50
Fig. 3. 15 a) Motor de Traslación. b) Motor de Rotación.	51
Fig. 3. 16 Driver Dual L298N.....	52
Fig. 3. 17 Regulador de Tensión con Salida de 5V	52

Fig. 3. 18 Circuito Regulador de Tensión Variable y Corrientes Elevadas	53
Fig. 3. 19 Circuito con Optoacoplador 4N35	54
Fig. 3. 20 Sensor de Temperatura LM35	54
Fig. 3. 21 Conexión con el Arduino	55
Fig. 3. 22 Circuito para el Fin de Carrera	55
Fig. 3. 23 Conexión con el Arduino	56
Fig. 3. 24 Configuración de la Tarjeta que se utilizara	58
Fig. 3. 25 Configuración del Procesador del Arduino Mega 2560	58
Fig. 3. 26 Comentar la instrucción TCP_KEEP_ALIVE	59
Fig. 3. 27 Comentar la instrucción USE_HOLDING_REGISTERS_ONLY	60
Fig. 3. 28 Instrucción Modbus TCP	69
Fig. 3. 29 Bloque de Función para la Comunicación con el Arduino	70
Fig. 3. 30 Creación de Bloque de Datos para el Arduino	70
Fig. 3. 31 Iniciación de Configuración Datos Arduino	71
Fig. 3. 32 Configuración de Datos Arduino	71
Fig. 3. 33 Bloque de Funciones en el Main	72
Fig. 3. 34 Bloque de Datos para el almacenamiento del dato para lectura del HTML	73
Fig. 3. 35 Instrucción de Move para el almacenamiento del dato	74
Fig. 3. 36 . Lógica en Grafcet para la etapa del Filtrado 01	75
Fig. 3. 37 Entrada del Sistema - Rangos de Temperatura	78
Fig. 3. 38 Entrada del Sistema - Rangos de Nivel	79
Fig. 3. 39 Salida del Sistema - Rangos de la Frecuencia	79
Fig. 3. 40 Entrada de Temperatura en fuzzyLogicDesigner	81
Fig. 3. 41 Entrada de Nivel Filtro 02 en fuzzyLogicDesigner	82
Fig. 3. 42 Salida del Variador en fuzzyLogicDesigner	82
Fig. 3. 43 Reglas de Control en fuzzyLogicDesigner	83
Fig. 3. 44 Resultado a una Frec 46 en fuzzyLogicDesigner	83
Fig. 3. 45 Resultado a una Frec 50 en fuzzyLogicDesigner	84
Fig. 3. 46 Resultado a una Frec 54 en fuzzyLogicDesigner	84
Fig. 3. 47 Resultado a una Frec 58 en fuzzyLogicDesigner	85
Fig. 3. 48 Método de Sugeno	93
Fig. 3. 49 Lógica en Grafcet para la Etapa de Condensación	94
Fig. 3. 50 Activación del Servidor Web del PLC S7-1200	104
Fig. 3. 51 Generación de Carpetas, HTML	111
Fig. 3. 52 Ingreso de la Ruta para el SCADA	113
Fig. 3. 53 Visualización del SCADA	114
Fig. 3. 54 Visualización del SCADA en Funcionamiento	114
Fig. 3. 55 Ensamblaje de la Planta Piloto	115
Fig. 3. 56 Funcionamiento de la Planta Piloto	115

INDICE DE TABLAS

Tabla 2. 1 Características del Arduino Mega.....	15
Tabla 2. 2 Módulo de Entrada y Salida del PLC S7-1200	24
Tabla 3. 1 Distancias de los Electrodo de Nivel	46
Tabla 3. 2 Distancias de los Electrodo del Filtro 02.....	47
Tabla 3. 3 Salidas Arduino	57
Tabla 3. 4 Entradas Arduino.....	57
Tabla 3. 5 Entradas Analógicas Arduino	57
Tabla 3. 6 Entradas Digitales PLC S7-1200.....	68
Tabla 3. 7 Salidas Digitales PLC S7-1200	68
Tabla 3. 8 Entrada Analógica PLC S7-1200	68
Tabla 3. 9 Salida Analógica PLC S7-1200	68
Tabla 3. 10 Reglas para el Control Fuzzy	80



CAPÍTULO 1

PLANTEAMIENTO METODOLOGICO

1.1. DESCRIPCIÓN DEL PROBLEMA

Debido al crecimiento de la población a nivel mundial, ha aumentado el consumo del recurso hídrico. En la actualidad se observa la problemática que cada país está afrontando respecto a la gestión del agua, y también de la poca sensibilización de parte de los gobiernos respecto al uso racional del agua.

Una persona por día debería estar consumiendo 100 litros de agua para cubrir sus necesidades (Fuente: Organización Mundial de la Salud OMS), el agua que desperdiciamos es mayor a la que consumimos. Otro aspecto que hace aún más crítico es el problema del calentamiento global, donde el derretimiento de los glaciales hace que el agua dulce se descongele y este se vaya al mar.

Hemos identificado también que en una familia hoy en día ya cuenta con lavadoras domésticas, ya que al ser utilizada observamos como el agua con detergente es desperdiciada, y así se utiliza aproximadamente 150 litros de agua por cada carga y si sumamos que una lavadora realiza 3 cargas por cada cantidad de ropa, entonces en un solo lavado tenemos 450 litros de agua que se desperdicia, esto también depende de cuánto es la capacidad de la lavadora. Esta cantidad de agua se dirige al sistema de desagüe de la ciudad, donde un gran porcentaje su destino final será el mar, ocasionado así el aumento de la contaminación. Este recurso puede ser reutilizado y/o hasta tratada para su utilización en riego de jardines, lavado de automóviles,

hasta la utilización de los servicios higiénicos y a lo que deseamos es poder ser reutilizada para el consumo humano.

Para la implementación de una planta de tratamiento de agua, las grandes empresas usan equipamientos y dispositivos de costo elevado, pero las industrias medianas y pequeñas, le es difícil acceder a tecnologías por el costo. Es por ello que debido al crecimiento de la industria, se hace necesario la automatización a un coste e infraestructura adecuada, para que las industrias que desechen aguas con pequeñas partículas sólidas o aguas con sustancias jabonosas puedan implementar una planta en que pueda reutilizar sus aguas.

Por otra parte, la creciente información, hace que integremos las tecnologías existentes, para poder mejorar los procesos de automatización a toda escala.

Los fabricantes ponen a disposición de las nuevas tecnologías, el uso de nuevos recursos como programación secuencial C++, además del lenguaje html, web2.0 que se viene aplicado a los SCADAs, con la finalidad de integrar la programación de tipo secuencial en sus controladores de programación paralela.

Lo que se desea es que en la planta de tratamiento de agua, es utilizar las herramientas de la electrónica, implementando un control automático para el sistema y el aprovechamiento de la energía solar para una condensación del agua.

1.2. JUSTIFICACIÓN

Es la utilización de los conocimientos de la electrónica para una automatización y control de una planta piloto para el tratamiento de las aguas con pequeñas partículas sólidas, aplicando la tecnología que hoy en día tenemos y poder ser parte de la solución para la contaminación del agua.

1.3. OBJETIVOS

1.3.1. OBJETIVO PRINCIPAL

Diseño e Implementación de una planta piloto para la reutilización de agua por filtrado y condensación; utilizando programación paralela y secuencia, mediante Control Fuzzy.

1.3.2. OBJETIVOS ESPECÍFICOS

- Estudio, diseño y elaboración del sistema de filtrado utilizando materia prima y del sistema por condensación del agua.
- Desarrollo de la programación en Arduino para el control del seguidor solar y censado de temperatura.
- Integración de PLC y Arduino.
- Estudio e implementación de Lógica Difusa en lenguaje STL (C++)
- Estudio de HTML, para el diseño en SCADA.
- Selección de la instrumentación adecuada para el sistema.
- Evaluar del Funcionamiento del Sistema de reutilización de agua.

1.4. HIPÓTESIS

Es posible realizar una integración de plataformas de programación paralela y secuencial a través de software abierto.

1.5. ALCANCE DEL PROYECTO

El proyecto está enfocado en el tratamiento del agua, mediante un filtro natural y un sistema por condensación, mediante un seguidor solar para dicho sistema.

De estos proyectos lo que se pretende es la unión de las etapas de filtrado y de condensación del agua por un seguidor solar, con la finalidad de mejorar y actualizar la tecnología y ver la comunicación entre un Arduino y un PLC para controlar las dos etapas mediante un SCADA por Web Server y así obtener mejor resultados en el tratamiento del agua. Es por ello que se realizara esta planta piloto para ver el funcionamiento de los dos sistemas interactuando los dos controladores.

1.6. VARIABLE

1.6.1. VARIABLES INDEPENDIENTES

- El agua, ya que es un recurso que se va a tratar.
- Control de la Planta Piloto.

1.6.2. VARIABLES DEPENDIENTES

- Energía Térmica del Sol para la generación de altas temperaturas, para la condensación del agua.
- Nivel del agua en las diferentes etapas del proceso y temperatura del agua.
- Posición del sol para realizar el seguimiento.

CAPÍTULO 2

MARCO TEÓRICO

2.1. ENERGIA SOLAR EN EL PERU ¹

La energía solar es una de las opciones que se están desarrollando como alternativas a las energías provenientes de la quema de combustibles fósiles. A diferencia de los países nórdicos, el territorio peruano, por estar mucho más próximo al Ecuador, cuenta con sol durante la mayor parte del año. Según el Atlas Solar del Perú elaborado por el Ministerio de Energía y Minas, el Perú tiene una elevada radiación solar anual siendo en la sierra de aproximadamente 5.5 a 6.5 kWh/m²; 5.0 a 6.0 kWh/m² en la Costa y en la Selva de aproximadamente 4.5 a 5.0 kWh/m².

En el Perú hay tres ámbitos donde se ha desarrollado el uso de energía solar en el Perú. El primer ámbito (y más tradicional) es el uso como fuente térmica a través de termas de agua en zonas del sur peruano, principalmente Arequipa y Puno, departamentos en los que existe cerca de 30 empresas dedicadas a la fabricación y mantenimiento de estos aparatos. No obstante, aún es amplio el camino a recorrer para masificar el uso de paneles solares tanto para áreas urbanas como rurales destinados al uso térmico el cual implicaría menor consumo de la red eléctrica en los hogares (una terma eléctrica es uno de los principales consumidores de energía eléctrica en un hogar). Asimismo su uso no se limitaría a lo domestico sino también podría incluirse en usos productivos como secadores de granos para la agricultura (en la zona sur la producción de granos andinos como kiwicha, quinua, kañihua es alta) así como para como la potabilización de agua en aquellas zonas que lo requieran.

Otro ámbito donde existen avances es en la provisión de electricidad a las zonas rurales. Según datos del 2011, el 16% población peruana no tiene electricidad en sus casas, cifra que se eleva a 22% en las zonas rurales. Según la Dirección General de Electrificación Rural aún existen cerca de 500 000 hogares ubicados en zonas rurales que quedarían sin ser atendidos por los programas públicos de electrificación. El Plan de Electrificación Nacional de Electrificación Rural cerca de 345 823 hogares deberán ser cubiertos con módulos fotovoltaicos en espacios rurales.

Entre los proyectos existentes está el financiado por el Banco Mundial, el Global Environment Facility – GEF y el MEM que ya ha subvencionado la provisión de electricidad a 2 216 hogares que con sistemas fotovoltaicos pilotos. Asimismo, dentro de este esquema existiría en cartera otro subproyectos para llegar a 7 000 hogares más. Otro programa es Euro Solar, que provee 130 pequeñas centrales de energía híbrida (eólico-solar) destinadas a abastecer de energía a postas, colegios y locales comunales rurales. Asimismo, el programa Luz para Todos del Gobierno Central contempla que cerca de 11 640 nueva localidades con servicio eléctrico serán atendidas con fuentes renovables siendo una buena parte de ellas a través de sistemas fotovoltaicos. Entre las opciones para la electrificación rural están los sistemas fotovoltaico domiciliario (SFD). La empresa estatal ADINELSA, encargada de la promoción de la electrificación rural en áreas no concesionadas, ya posee más 1500 SFDs operativos en el sur del país.

El tercer ámbito de desarrollo, y el más promisorio, es el que ha surgido con la concesión de las 4 centrales solares que se enlazarán al Sistema Eléctrico Nacional (SEIN) luego de la primera subasta de suministro de electricidad de Recursos Energéticos Renovables (RER) llevada a cabo por el Ministerio de Energía y Minas. Las compañías españolas T-Solar Global y Solarpark Corporacion Tecnológica son las que construirán estas cuatro centrales

fotovoltaicas, con una potencia conjunta de 80 megavatios (Mw). Estas empresas han firmado contratos con el Gobierno Peruano que les permite asegurar la venta de electricidad producida de fuentes solares durante un lapso de 20 años. Según Juan Laso, Director General de T Solar, esta adjudicación le permitirá “incrementar la cartera de proyectos en fase de desarrollo de T-Solar, que suman una potencia superior a los 650 MW”.

Como vemos, el sector de la energía solar va desde pequeñas instalaciones familiares hasta grandes proyectos de centrales solares. Es interesante que los avances, en este último caso, generen el desarrollo tecnológico y la difusión de esta fuente de energía renovable en el país. Una característica primordial de la energía solar es su capacidad para adecuarse a proyectos de mediana y pequeña envergadura para usuarios individuales. Por ejemplo, en ámbitos urbanos se pueden desarrollar instalaciones fotovoltaicas que se integren a grandes superficies expuestas como estacionamientos, edificios, marquesinas. De hecho, la T –Solar ya desarrolla proyectos de este tipo en España. Este tipo de innovaciones permite acercar la producción de electricidad al punto de consumo evitando pérdidas durante el transporte y además de reemplazar el consumo de energía eléctrica de la red nacional y ahorrar costos a quienes la implementan.

No cabe duda de que las opciones de uso de la energía solar son grandes. Lamentablemente, aún existe desconocimiento de aquellos sectores que pueden aprovecharlo más intensamente. Desarrollar este subsector energético sería crucial ya que es una de las mejores opciones para cambiar la actual matriz energética mundial intensiva en gases de efecto invernadero.

¹ (Rural, 2017)



Fig. 2. 1 Potencial de Energía Solar en el Perú
(Fuente: ² (Zeballos Paz, 2017, pág. 32))

2.1.1. ENERGÍA CALORÍFICA ³

La energía térmica (también energía calórica o energía calorífica) es la manifestación de la energía en forma de calor. En todos los materiales los átomos que forman sus moléculas están en continuo movimiento ya sea trasladándose o vibrando. Este movimiento implica que los átomos tengan una determinada energía cinética a la que nosotros llamamos calor, energía térmica o energía calorífica.

Si se aumenta temperatura a un elemento aumenta su energía térmica; pero no siempre que se aumenta la energía térmica de un cuerpo aumenta su temperatura ya que en los cambios de fase (de líquido a gas, por ejemplo) la temperatura se mantiene. Por ejemplo, al calentar un cazo de agua, poco a poco le vamos dando energía térmica y va aumentando su temperatura, pero cuando llega a los 100°C (temperatura de ebullición) la energía térmica que le suministramos a partir de este momento se utiliza para cambiar de fase (de líquido a gas, es decir, a vapor de agua) pero no para aumentar su temperatura. ³ (Energía Nuclear, 2016)

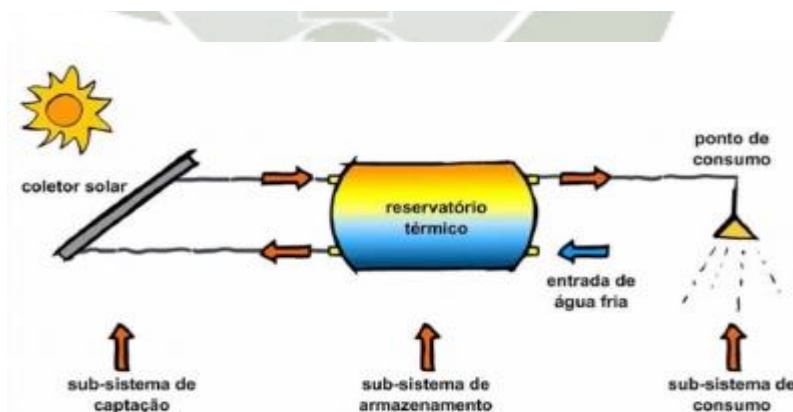


Fig. 2. 2 Sistema Termo Solar
(Fuente: www.dasolabrava.org.br)

2.1.2. TRANSMISIÓN DE LA ENERGÍA TÉRMICA ³

La energía térmica puede transmitirse de un cuerpo a otro siguiendo las leyes de la termodinámica de tres modos distintos:

- Transmisión de calor por radiación.
- Transmisión de energía térmica por conducción.
- Transmisión de energía calorífica por convección.

La energía térmica por radiación se transmite a través de ondas electromagnéticas. Es el modo con el que nos llega la energía térmica proveniente del Sol. El principal ejemplo de este caso lo encontramos en las instalaciones de energía solar térmica, que aprovechan la radiación solar para calentar agua. El agua caliente que se obtiene de estas instalaciones solares habitualmente se utiliza para calefacción y para usarla como agua caliente sanitaria.

La transmisión de la energía térmica por conducción se experimenta cuando un cuerpo caliente está en contacto físico con otro cuerpo más frío. La energía se transmite siempre del cuerpo caliente al cuerpo frío. Si ambos cuerpos están a la misma temperatura no hay transferencia energética. Cuando tocamos un trozo de hielo con la mano parte de la energía térmica de nuestra mano se transfiere al hielo, por eso tenemos sensación de frío.

La transmisión de la energía térmica por convección se produce cuando se trasladan las moléculas calientes de un lado a otro. Sería el caso del viento, capaz de mover moléculas con cierta energía calorífica de un lado a otro. ³ (Energía Nuclear, 2016)

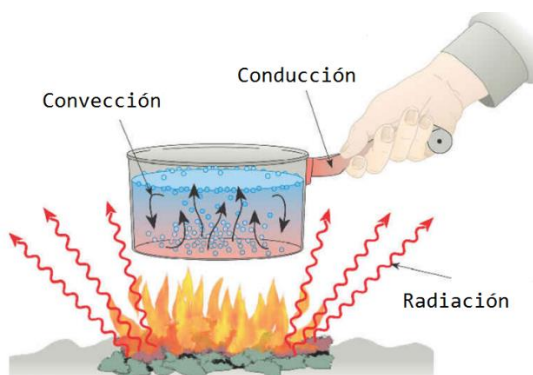


Fig. 2. 3 Transmisión de Energía Térmica
(Fuente: www.esss.com.br)

2.2. **FILTROS DE AGUA DE MATERIA PRIMA** ⁴

El filtro de arena es un método de tratamiento de los efluentes ecológicos, relativamente sencillo y poco costoso. Su principio consiste en hacer percolar el agua a través de un bloque de arena.

Esquemáticamente, los granos de arena forman una capa atravesada por el agua y que detiene por simple efecto de tamizado las partículas de tamaño superior al de los espacios existentes entre dichos granos. Si a lo largo de su avance tocan un grano, las partículas más pequeñas también quedarán retenidas sobre la superficie de estos por el efecto pared. La capacidad de frenado del filtro será tanto mayor cuanto menor sea el diámetro de sus granos y más largo sea el tiempo de permanencia de las partículas.

Hay tres tipos de filtración por arena:

- Los filtros de arena rápidos. Deben limpiarse con frecuencia debido a su alisado, que invierte la dirección del agua.

- Los filtros de arena semirrápidos.
- Los filtros de arena lentos.

Los dos primeros requieren del uso de bombas y productos químicos (principio de floculación). Se utiliza un floculante que, por un principio químico, atrapa los materiales en suspensión y las partículas, formando grandes copos que se depositarán por sedimentación (lo cual significa que las partículas en suspensión detienen su movimiento y se depositan).

A diferencia de otros métodos de filtración por arena, los filtros de arena lentos emplean procesos biológicos para limpiar el agua y son sistemas no presurizados. Pueden tratar el agua y reducir la presencia de microorganismos (bacterias, virus, microbios, etc.) sin necesidad de productos químicos. No precisan electricidad para funcionar. ⁴ (Wiki Water, s.f.)

2.2.1. EL FILTRO DE ARENA DE PRETRATAMIENTO ⁴

La instalación de este filtro resulta muy sencilla. Sin embargo, por sí mismo solo permite realizar un tratamiento somero del agua.

Consiste en llenar un recipiente, que suele ser una sencilla cubeta de hormigón provista de un grifo u otro medio de vaciado, con una capa de grava bastante fina y con otra capa de arena, mucho más gruesa que la anterior, sobre la que se vierte el agua a tratar, que solo hay que recoger por la parte inferior del recipiente.

Es realmente eficaz cuando se utiliza como pretratamiento para otros métodos de desinfección (por radiación solar, cloración o ebullición) en aquellas ocasiones en las que el agua no tratada de la que se dispone está algo turbia; a pesar de ello, no basta por sí mismo. ⁴ (Wiki Water, s.f.)



Fig. 2. 4 Filtro de Arena y Grava
(Fuente: www.imoralesm.wordpress.com)

2.3. PROGRAMACIÓN SECUENCIAL Y PARALELA

2.3.1. PROGRAMACIÓN SECUENCIAL

Tradicionalmente, los programas informáticos se han escrito para el cómputo en serie. Para resolver un problema, se construye un algoritmo y se implementa como un flujo en serie de instrucciones. Estas instrucciones se ejecutan en una unidad central de procesamiento en un ordenador. Sólo puede ejecutarse una instrucción a la vez y un tiempo después de que la instrucción ha terminado, se ejecuta la siguiente. ⁵ (Barney, 2007)

2.3.2. PROGRAMACIÓN EN PARALELO

La computación en paralelo, por el contrario, utiliza simultáneamente múltiples elementos de procesamiento para resolver un problema. Esto se

logra mediante la división del problema en partes independientes de modo que cada elemento de procesamiento pueda ejecutar su parte del algoritmo de manera simultánea con los otros. Los elementos de procesamiento son diversos e incluyen recursos tales como una computadora con múltiples procesadores, varios ordenadores en red, hardware especializado, o cualquier combinación de los anteriores. ⁵ (Barney, 2007)

2.4. ARDUINO

2.4.1. ARDUINO MEGA

El Arduino Mega 2560 es una placa microcontroladora basada en el ATmega1280. Cuenta con 54 pines digitales de entrada / salida (de los cuales 14 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UART (puertos serie de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador; simplemente conéctelo a un ordenador con un cable USB o conéctelo con un adaptador AC-DC o una batería para empezar. Respecto a sus características son las siguientes:

Microcontrolador	ATmega1280
Tensión de funcionamiento	5V
Tensión de entrada (recomendado)	7-12V
Tensión de entrada (límites)	6-20V
Digital I / O Pins	54 (de los cuales 15 proporcionan salida de PWM)
Analog Input Pins	16
Corriente CC por pin de E / S	40 mA

Corriente de CC para clavija de 3.3V	50 mA
Memoria flash	128 KB de los cuales 4 KB utilizados por bootloader
SRAM	8 KB
EEPROM	4 KB
Velocidad de reloj	16 MHz

Tabla 2. 1 Características del Arduino Mega
(Fuente: www.arduino.cc)

El ATmega1280 tiene 128 KB de memoria flash para almacenar código (de los cuales 4 KB se utiliza para el gestor de arranque), 8 KB de SRAM y 4 KB de EEPROM

Respecto a las entradas y salidas cada uno de los 54 pines digitales del Mega se puede utilizar como entrada o salida, utilizando las funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Funcionan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia pull-up interna (desconectada por defecto) de 20-50 kOhms. Además, algunos pines tienen funciones especializadas:

- ❖ Serie: 0 (RX) y 1 (TX); Serie 1: 19 (RX) y 18 (TX); Serie 2: 17 (RX) y 16 (TX); Serie 3: 15 (RX) y 14 (TX). Se utiliza para recibir (RX) y transmitir (TX) datos en serie TTL. Los pines 0 y 1 también están conectados a las clavijas correspondientes del chip serie FTDI USB-to-TTL. ⁶ (Arduino, 2017)
- ❖ Interrupciones externas: 2 (interrupción 0), 3 (interrupción 1), 18 (interrupción 5), 19 (interrupción 4), 20 (interrupción 3) y 21 (interrupción 2). Estos pines se pueden configurar para activar una interrupción en un

valor bajo, un flanco ascendente o descendente o un cambio de valor. Vea la función `attachInterrupt()` para más detalles. ⁶ (Arduino, 2017)

- ❖ PWM: 2 a 13 y 44 a 46. Proporcione una salida PWM de 8 bits con la función `analogWrite()`. ⁶ (Arduino, 2017)
- ❖ SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). Estos pines admiten comunicación SPI, que, aunque provistos por el hardware subyacente, no está incluido actualmente en el lenguaje Arduino. Los pines SPI también se dividen en la cabecera ICSP, que es físicamente compatible con Duemilanove y Diecimila. ⁶ (Arduino, 2017)
- ❖ LED: 13. Hay un LED incorporado conectado al pin digital 13. Cuando el pin es valor ALTO, el LED está encendido, cuando el pin es LOW, está apagado.
- ❖ I²C: 20 (SDA) y 21 (SCL). Apoye la comunicación de I²C (TWI) usando la biblioteca del alambre (documentación en el Web site del cableado). Tenga en cuenta que estos pines no están en el mismo lugar que los pins I²C en el Duemilanove o Diecimila.
- ❖ El Mega tiene 16 entradas analógicas, cada una de las cuales proporciona 10 bits de resolución (es decir, 1024 valores diferentes). Por defecto miden desde tierra hasta 5 voltios, aunque es posible cambiar el extremo superior de su rango usando la función de alfiler `AREF` y `analogReference()`.
- ❖ Hay un par de otros alfileres en el tablero:
- ❖ `AREF`. Tensión de referencia para las entradas analógicas. Se utiliza con `analogReference()`.
- ❖ Reiniciar. Lleve esta línea LOW para reiniciar el microcontrolador. Normalmente se utiliza para agregar un botón de reinicio a los escudos que bloquean el de la placa.



Fig. 2. 5 Tarjeta de Arduino Mega
(Fuente: www.arduino.cc)

2.4.2. MODULO SHIELD ETHERNET ⁷

El módulo Arduino Shield Ethernet tiene una conexión RJ-45 estándar, con un transformador de línea integrado. El chip Wiznet W5100 ofrece una red (IP) capaz de usar TCP y UDP.

Tiene una ranura para tarjetas micro-SD, que se puede utilizar para almacenar archivos para servir a través de la red. Es compatible con las placas Arduino/Genuino Uno y Mega (utilizando la Biblioteca Ethernet). El lector de tarjetas microSD es accesible a través de la Biblioteca SD.

El escudo también incluye un controlador de reajuste, para asegurar que el módulo Ethernet W5100 se restablece correctamente en el encendido. Las revisiones anteriores del escudo no eran compatibles con el Mega y la necesidad de restablecer manualmente después del encendido.

Arduino se comunica tanto con el chip W5100 como con la tarjeta SD usando el bus SPI (a través de la cabecera ICSP). Para ello utiliza los pines digitales 10, 11, 12, y 13 en las placas Arduino/Genuino Uno y los pines 50, 51 y 52 en la Mega. En ambas placas, el pin 10 se utiliza para seleccionar el chip W5100 y el pin 4 para la tarjeta SD. Estos pines no se pueden utilizar para otras entradas o salidas.

Tenga en cuenta que debido a que la cuota del bus SPI del chip W5100 y la tarjeta SD, sólo uno puede estar activo en cada momento. Si está utilizando ambos periféricos en su programa, esto debe ser atendido por las bibliotecas correspondientes. Si usted no está utilizando uno de los periféricos en su programa, tendrá que anular la selección explícitamente. Para hacer esto con la tarjeta SD, ajuste el pin 4 como salida y de un valor alto a la misma. Para el chip W5100, establezca pin digital 10 como una salida alta.

El botón de reinicio en el escudo restablece tanto el chip W5100 como la placa Arduino/Genuino. ⁷ (Web - Robotica.com, 2017)

El módulo Arduino Shield Ethernet contiene una serie de LEDs informativos:

- PWR: indica que la placa y el módulo están encendidos.
- LINK: indica la presencia de un enlace de red y parpadea cuando el módulo transmite o recibe datos.
- FULLD: indica que la conexión de red es full duplex
- 100M: indica la presencia de un conexión de red a Mb/s 100 (en contraposición a 10 Mb/s)
- RX: Parpadea cuando el escudo recibe datos
- TX: parpadea cuando el escudo envía datos
- COLL: parpadea cuando se detectan colisiones de red



Fig. 2. 6 Modulo de Shieft Ethernet
(Fuente: www.web-robotica.com)

2.5. **SENSOR DE FLUJO** ⁸

Los sensores de flujo son muy usados en varias aplicaciones, puesto que el flujo o caudal es parámetro necesario en varios procesos, a nivel doméstico lo podemos usar para medir el consumo de agua.

El sensor internamente tiene un rotor cuyas paletas tiene un imán, la cámara en donde se encuentra el rotor es totalmente aislado evitando fugas de agua, externamente a la cámara tiene un sensor de efecto hall que detecta el campo magnético del imán de las paletas y con esto el movimiento del rotor, el sensor de efecto hall envía los pulsos por uno de los cables del sensor, los pulsos deberán ser convertidos posteriormente a flujo pero esto ya es tarea del Arduino o controlador que se desee usar.

Todos los modelos tienen tres cables para su conexión, rojo y negro para la alimentación y amarillo para la salida de los pulsos.

La salida de pulsos es una onda cuadrada cuya frecuencia es proporcional al caudal. El factor de conversión de frecuencia (Hz) a caudal (L/min) varía entre modelos y depende de la presión, densidad e incluso del mismo caudal.

Para el caso del sensor de ½" el factor de conversión promedio proporcionado por el fabricante es:

$$f(\text{Hz}) = 7.5 \times Q(\text{L/min})$$

Llamaremos nosotros K al factor de conversión, siendo K=7.5 para el sensor de ½", K=5.5 para el sensor de ¾" y 3.5 para el sensor de 1", trabajar con dichos valores no nos garantiza precisión, pero nos pueden servir para aplicaciones simples, si necesitamos mayor exactitud necesitamos calibrar y calcular dicho factor. ⁸ (NAYLAMP MECHATRONICS, 2017).



Fig. 2. 7 Modulo de Shielf Ethernet
(Fuente: www.naylampmechatronics.com)

2.6. ***SENSOR LDR (Light Dependent Resistor)***

Una LDR es una resistencia que varía su valor en función de la luz que incide sobre ella. Así de fácil. También se llama fotocélula o resistencia LDR.

Cuando hay luz sobre la LDR su resistencia disminuye, cuanto menos luz tenga la LDR mayor será su resistencia. Los valores de las resistencias con la luz dependen del tipo del LDR.

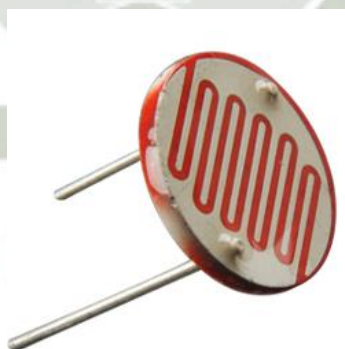


Fig. 2. 8 Sensor LDR
(Fuente: www.prometec.net)

2.7. SENSOR DE TEMPERATURA LM35 ⁹

El LM35 es un sensor de temperatura digital. A diferencia de otros dispositivos como los termistores en los que la medición de temperatura se obtiene de la medición de su resistencia eléctrica, el LM35 es un integrado con su propio circuito de control, que proporciona una salida de voltaje proporcional a la temperatura.

La salida del LM35 es lineal con la temperatura, incrementando el valor a razón de 10mV por cada grado centígrado. El rango de medición es de -55°C (-550mV) a 150°C (1500 mV). Su precisión a temperatura ambiente es de 0,5°C.

Los sensores LM35 son relativamente habituales en el mundo de los aficionados a la electrónica por su bajo precio, y su sencillez de uso. ⁹ (Llamas, 2015)

2.8. VALVULAS SELENOIDES ¹⁰

Controlar el flujo de agua en una tubería es ahora más sencillo con la ayuda de esta válvula solenoide. Las válvulas solenoides son un tipo de electroválvula todo/nada o abierto/cerrado. Tienen dos partes: el solenoide y el cuerpo de plástico.

El solenoide es un electroimán que al ser energizado se desplaza junto con el diafragma de la válvula y permite el paso del fluido. La válvula se mantiene abierta mientras el solenoide está energizado. Cuando no está alimentado un resorte se encarga de regresar la válvula a su posición de reposo, que en este caso es del tipo normalmente cerrada (NC).

El cuerpo de la válvula está fabricado en plástico con roscados machos a ambos lados de 1/2" para conexión estándar NPS (recta).

Para su control se puede utilizar un microcontrolador como Arduino o un ESP8266 y utilizar un protocolo de comunicación para el control a distancia, como: Wifi, RF, RS-485, Z-wave, etc. Es necesario utilizar un driver de potencia entre el microcontrolador y la válvula, en este caso como el actuador de la válvula es un solenoide es suficiente con utilizar transistor o relay. Nuestra recomendación son los transistores TIP121 o el Módulo Mosfet, si se desea utilizar un relay se puede usar el Módulo Relay de 2 canales. ¹⁰ (NAYLAMP MECHATRONICS, 2017)

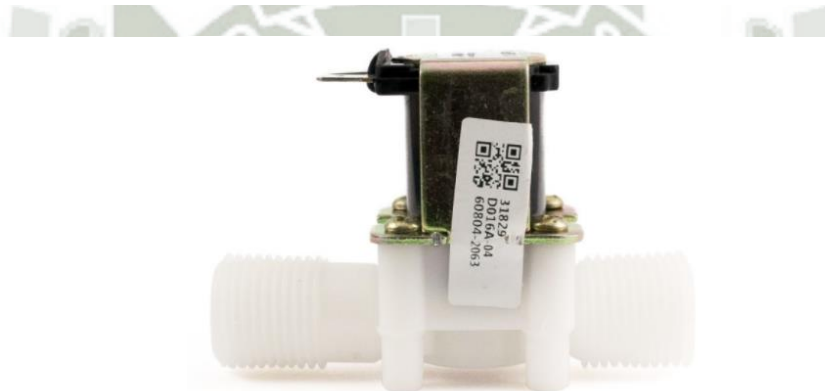


Fig. 2. 9 Válvula Selenoide
(Fuente: <http://www.naylampmechatronics.com>)

2.9. PLC S7-1200 ¹¹

El controlador lógico programable (PLC) S7-1200 ofrece la flexibilidad y capacidad de controlar una gran variedad de dispositivos para las distintas tareas de automatización. Gracias a su diseño compacto, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones.

La CPU incorpora un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada y salida en una carcasa compacta, conformando así un potente PLC. Una vez cargado el programa en la CPU, ésta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que puede incluir lógica booleana, instrucciones de conteo y temporización, funciones matemáticas complejas, así como comunicación con otros dispositivos inteligentes.

La CPU incorpora un puerto PROFINET para la comunicación en una red PROFINET. Los módulos de comunicación están disponibles para la comunicación en redes RS485 o RS232. ¹¹ (Siemens, 2009, pág. 12)

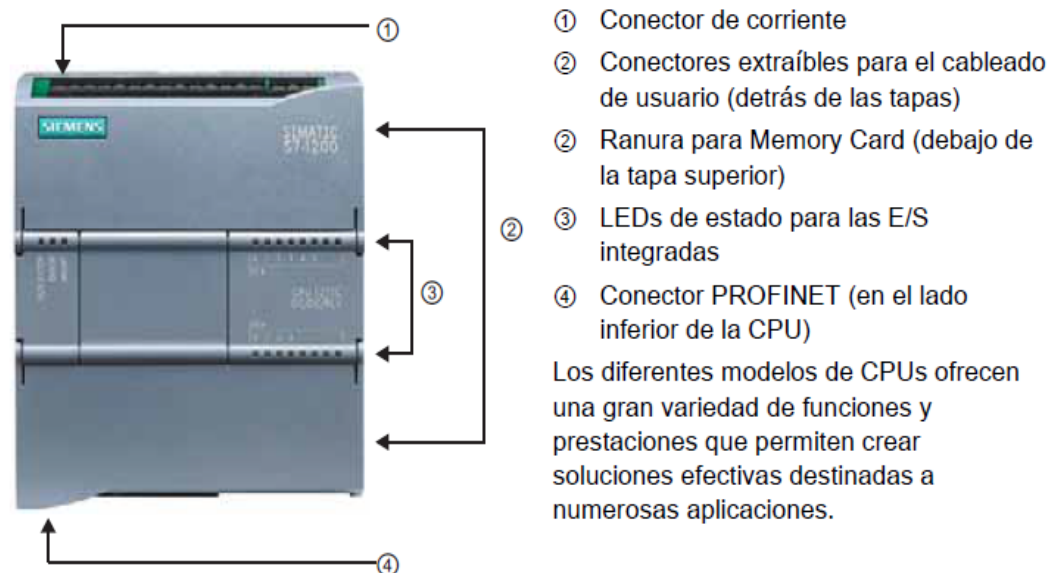


Fig. 2. 10 PLC S7-1200
(Fuente: Manual de PLC S7-1200)

La gama S7-1200 ofrece una gran variedad de módulos de señales y Signal Boards que permiten ampliar las prestaciones de la CPU. También es posible instalar módulos de comunicación adicionales para soportar otros protocolos de comunicación. ¹¹ (Siemens, 2009, pág. 13)

Función	CPU 1211C	CPU 1212C	CPU 1214C
Dimensiones físicas (mm)	90 x 100 x 75		110 x 100 x 75
Memoria de usuario	• 25 KB • 1 MB • 2 KB		• 50 KB • 2 MB • 2 KB
E/S integradas locales	• 6 entradas/4 salidas • 2 entradas		• 8 entradas/6 salidas • 2 entradas
Tamaño de la memoria imagen de proceso	1024 bytes para entradas (I) y 1024 bytes para salidas (Q)		
Área de marcas (M)	4096 bytes		8192 bytes
Ampliación con módulos de señales	Ninguna	2	8
Signal Board	1		

Tabla 2. 2 Módulo de Entrada y Salida del PLC S7-1200
(Fuente: Manual de PLC S7-1200)

2.9.1. COMUNICACIÓN MODBUS TCP/IP ¹²

Modbus es un protocolo de comunicaciones situado en el nivel 7 del Modelo OSI, basado en la arquitectura maestro/esclavo (RTU) o cliente/servidor (TCP/IP), diseñado en 1979 por Modicon para su gama de controladores lógicos programables (PLCs). Convertido en un protocolo de comunicaciones estándar de facto en la industria, es el que goza de mayor disponibilidad para la conexión de dispositivos electrónicos industriales.

Las principales razones por las cuales el uso de Modbus en el entorno industrial se ha impuesto a otros protocolos de comunicaciones son:

- Se diseñó teniendo en cuenta su uso para aplicaciones industriales
- Es público y gratuito
- Es fácil de implementar y requiere poco desarrollo
- Maneja bloques de datos sin suponer restricciones

Modbus permite el control de una red de dispositivos, por ejemplo un sistema de medida de temperatura y humedad, y comunicar los resultados a un ordenador. Modbus también se usa para la conexión de un ordenador de

supervisión con una unidad remota (RTU) en sistemas de supervisión adquisición de datos (SCADA). Existen versiones del protocolo Modbus para puerto serie y Ethernet (Modbus/TCP).

Cada dispositivo de la red Modbus posee una dirección única. Cualquier dispositivo puede enviar órdenes Modbus, aunque lo habitual es permitirlo sólo a un dispositivo maestro. Cada comando Modbus contiene la dirección del dispositivo destinatario de la orden. Todos los dispositivos reciben la trama pero sólo el destinatario la ejecuta (salvo un modo especial denominado "Broadcast"). Cada uno de los mensajes incluye información redundante que asegura su integridad en la recepción. Los comandos básicos Modbus permiten controlar un dispositivo RTU para modificar el valor de alguno de sus registros o bien solicitar el contenido de dichos registros.

Existe gran cantidad de modems que aceptan el protocolo Modbus. Algunos están específicamente diseñados para funcionar con este protocolo. Existen implementaciones para conexión por cable, wireless, SMS o GPRS. La mayoría de problemas presentados hacen referencia a la latencia y a la sincronización. ¹² (Wikipedia - La enciclopedia libre, 2017)



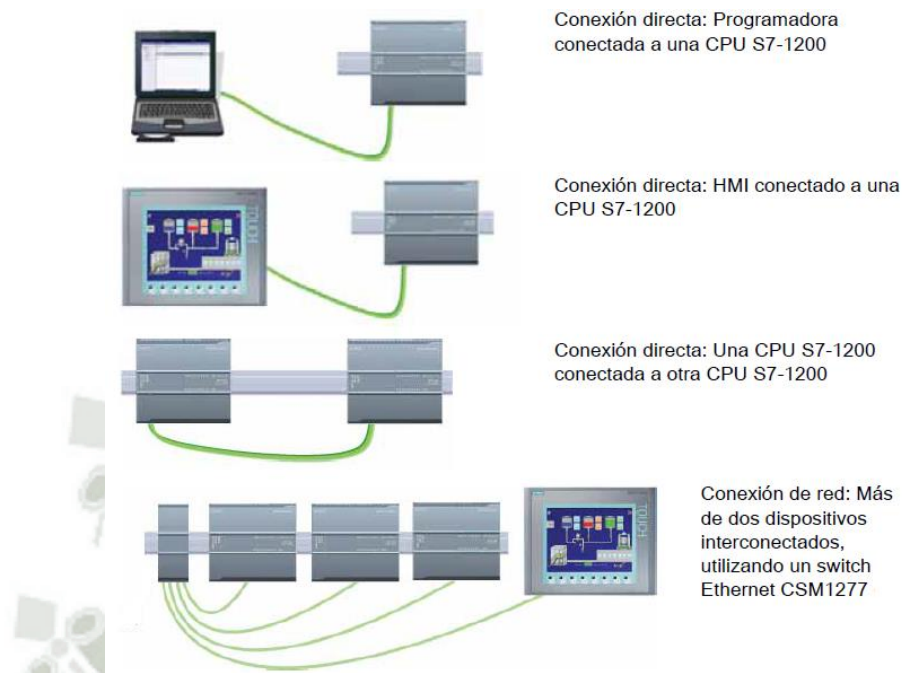


Fig. 2. 11 Tipos de Conexión de Comunicación
(Fuente: Manual de PLC S7-1200)

2.10. LÓGICA DIFUSA ¹³

La lógica difusa permite a los sistemas tratar con información que no es exacta; es decir, dicha información contiene un alto grado de imprecisión, contrario a la lógica tradicional que trabaja con información definida y precisa. Como ejemplo de información que maneja la lógica difusa tenemos: estatura media, temperatura alta, etc., que en términos difusos son realmente imprecisos. La teoría de conjuntos difusos parte de la similitud con los conjuntos clásicos en las cuales se tienen una función de pertenencia de 0 ó 1. En los conjuntos difusos se tiene la característica de que la función de pertenencia pueda adquirir valores en el rango de 0 a 1. Es así como se introduce el concepto de conjunto difuso, el cual se encuentra asociado con un determinado valor lingüístico que está definido por una etiqueta, palabra o adjetivo. (Catarina UDLAP, pág. 37)

Así se define una función de pertenencia de una variable para cada conjunto difuso, en la que se indica el grado en que se encuentra incluida en el concepto representado por la etiqueta. De esta manera los conjuntos difusos pueden agrupar objetos por el valor de una cierta magnitud; como por ejemplo, las personas pueden ser agrupadas por su altura. (Catarina UDLAP, pág. 37)

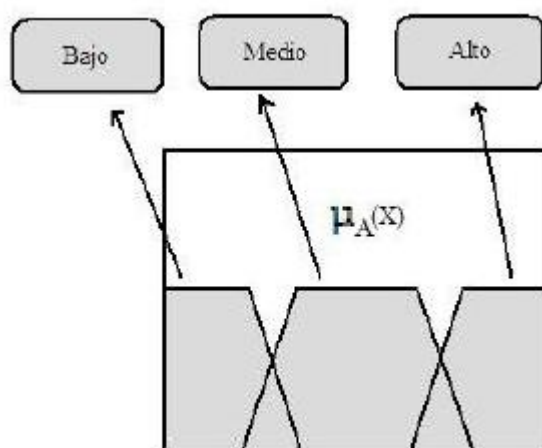


Fig. 2. 12 Conjuntos Difusos
(Fuente: www.catarina.udlap.mx)

Además puede observarse claramente que la transición del valor del conjunto entre el cero y el uno es gradual y no cambia instantáneamente entre ellos como el caso de la lógica clásica. El concepto de función de membresía en un conjunto difuso se ilustra en la siguiente figura. (Catarina UDLAP, pág. 38)

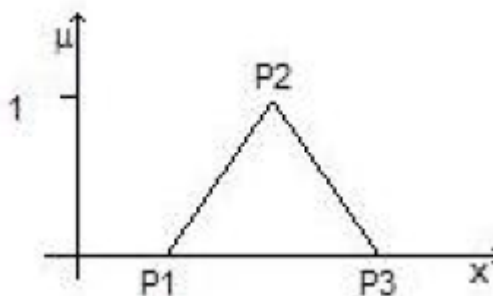


Fig. 2. 13 Función de Membresía
(Fuente: www.catarina.udlap.mx)

Donde la función de membresía o pertenencia de un elemento X en el conjunto difuso triangular está dado por: (Catarina UDLAP, pág. 38)

$$\mu_A(x) = \begin{cases} \frac{x - p_1}{p_2 - p_1} & p_1 < x < p_2 \\ \frac{p_3 - x}{p_3 - p_2} & p_2 < x < p_3 \\ 0 & \text{en otro lado} \end{cases}$$

Fig. 2. 14 Formula para el Cálculo de Membresías del Triangulo
(Fuente: www.catarina.udlap.mx)

De esta forma, existen otras formas de conjuntos difusos, como son trapezoidales, tipo campana de Gauss, tipo S, etc. Para todos ellos el concepto de función de membresía es el mismo; es decir, el tener un grado de pertenencia entre 0 y 1. Así, a grandes rasgos, este es el concepto básico de la lógica, sus conjuntos difusos. (Catarina UDLAP, pág. 39)

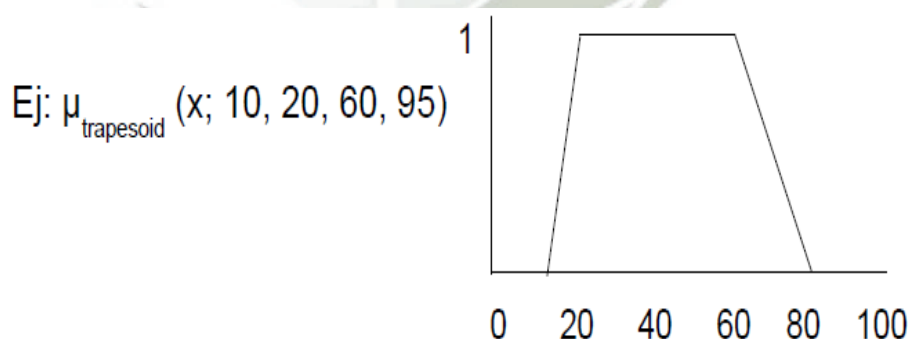


Fig. 2. 15 Formula para el Cálculo de Membresías del Triangulo
(Fuente: www.catarina.udlap.mx)

$$\mu_{trapezoida}(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$$

Fig. 2. 16 Formula para el Cálculo de Membresías del Trapezoide
(Fuente: www.catarina.udlap.mx)

2.10.1. CONTROL FUZZY¹³

La principal aplicación actual de la lógica difusa son los sistemas de control difuso, que utilizan las expresiones difusas para formular las reglas que controlaran dichos sistemas. Como la lógica difusa sugiere un cierto grado de pertenencia para un dato que se presente dentro de los conjuntos difusos, permite a un controlador difuso tomar diferentes grados de acción en un sistema. En los sistemas de control debe tomarse en cuenta el conocimiento experto de una o varias personas para la realización de la base de conocimientos sobre el cual se basaran la toma de decisiones.

El control Difuso puede aplicarse en innumerables sistemas, tanto sencillos, como brazos articulados y vehículos autónomo, en los cuales los modelos matemáticos son muy complejos; así, empleando técnicas de razonamiento aproximado es posible controlar sistemas superiores cuando el entorno no se conocen de forma precisa. Dicha característica permite mayor flexibilidad que el control clásico en la que para la realización de un controlador se requiere de un alto grado de cálculo matemático. Así, al desarrollar un

controlador difuso es posible prescindir de la rigidez matemática y transmitir el raciocinio humano hacia un sistema. (Catarina UDLAP, págs. 39-40)

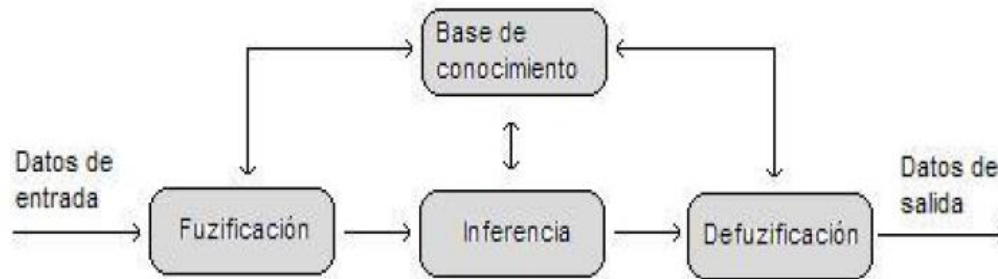


Fig. 2. 17 Estructura de un Controlador Difuso
(Fuente: www.catarina.udlap.mx)

En la que se puede ver en el primero bloque llamado fuzificador en ellos datos de entrada son procesados para calcular el grado de membresía que tendrán dentro del controlador. Posteriormente se tiene el dispositivo de inferencia que junto con la base de conocimientos realizan la toma de decisiones que dictaran la forma en que actuara el sistema. El método de inferencia se basa en el grado de pertenencia de los datos de entrada en los conjuntos difusos de los espacios correspondientes, siempre que estas se den, para tomar una decisión en el espacio de salida. El conjunto de reglas, que son la base de conocimientos, es el tipo antecedente – consecuente; es decir *si...entonces...*, y existen distintos métodos de llevar acabo (mínimo - máximo, máximo, producto, etc.). La última etapa que se tiene dentro del controlador es el defuzificador, que es quien realiza el procesado final con el fin de adecuar los valore difusos obtenidos de la inferencia en valores no difusos útiles para el proceso que se ha de controlar, como ejemplos tenemos el método del centro de área, de la máxima pertenencia, etc. De esta manera obtenemos un panorama general de cómo se estructura un controlador difuso. (Catarina UDLAP, págs. 40-41)

2.10.2. FUZIFICACIÓN

El proceso de fuzificación consiste en convertir una variable real en un grado de pertenencia que cuantifica el grado de posesión hacia su correspondiente variable lingüística. Las variables lingüísticas son representativas de situaciones como: Positivo, Muy Positivo, Bajo, Alto, etc.

El primer paso consiste en tomar las entradas y determinar el grado al que ellos pertenecen a cada uno de los conjuntos difusos apropiados. La entrada siempre es un valor numérico limitado al universo del discurso de la variable de entrada.

2.10.3. INFERENCIA – MANDAMI ¹³

Conocida también como las reglas del controlador o base de conocimiento, es el punto de partida para la generación del conjunto de reglas sobre las que se rige la inferencia. Se describe los términos usados y se muestra la base en forma de matriz de asociación difusa.

Para el desarrollo, es necesario realizar una investigación de campo, esta nos permite confirmar los conocimientos empíricos para lograr tener un respaldo adecuado y que el controlador se basara en la mejor información posible. (Catarina UDLAP, pág. 47)

El método de inferencia mínimo – máximo conocido como Mandami, una de sus características es apropiado para realizar un controlador compilado. Este método se basa en lo siguiente:

- **Valores de Entrada:** El resultado de la fuzificación arroja ciertos valores de membresía para distintos clusters en cada espacio de entrada. Ejemplo:

$\mu_{A1}(x) = 0.7$; Valor de membresía de la variable X en la entrada A.

$\mu_{A2}(x) = 0.3$; Valor de membresía de la variable X en la entrada A.

0 para los demás clusters del espacio de entrada A

$\mu_{B1}(y) = 0.4$; Valor de membresía de la variable Y en la entrada B.

$\mu_{B2}(y) = 0.6$; Valor de membresía de la variable Y en la entrada B.

0 para los demás clusters del espacio de entrada B

- **Aplicación de Reglas:** Las pertenencias que existen se aplican sobre la base de reglas para saber en qué cluster del espacio de salida se produce el consecuente de cierta combinación de antecedentes descritos en dichas reglas. Ejemplo.

R1: Si A1 y B2 entonces C3

R2: Si A3 y B1 entonces C2, etc.

- **Membresía de Salida:** El valor de membresía que se hereda al cluster de salida en cada regla cumplida, es el valor mínimo de las membresías de los clusters de los espacios de entrada que se involucran en la regla. Ejemplo:

$\mu_{C3}(z) = 0.6$; valor de membresía de la variable Z en la salida C.

$\mu_{C5}(z) = 0.3$; valor de membresía de la variable Z en la salida C.

- **Formación del Polígono:** Después de aplicar todas las reglas que se cumplen, podemos tener varios valores de membresía para un mismo cluster del espacio de salida, si varias reglas heredaron en él. Adicionalmente es necesario formar el polígono de salida. Para obtener el producto final del proceso de inferencia, se debe entonces hacer un barrido por cada cluster, tomado siempre el máximo valor de membresía que presente en cualquier regla. Entre clusters, se debe tomar también

como el valor de membresía, la mayor de cada uno. (Catarina UDLAP, págs. 49-50-51)

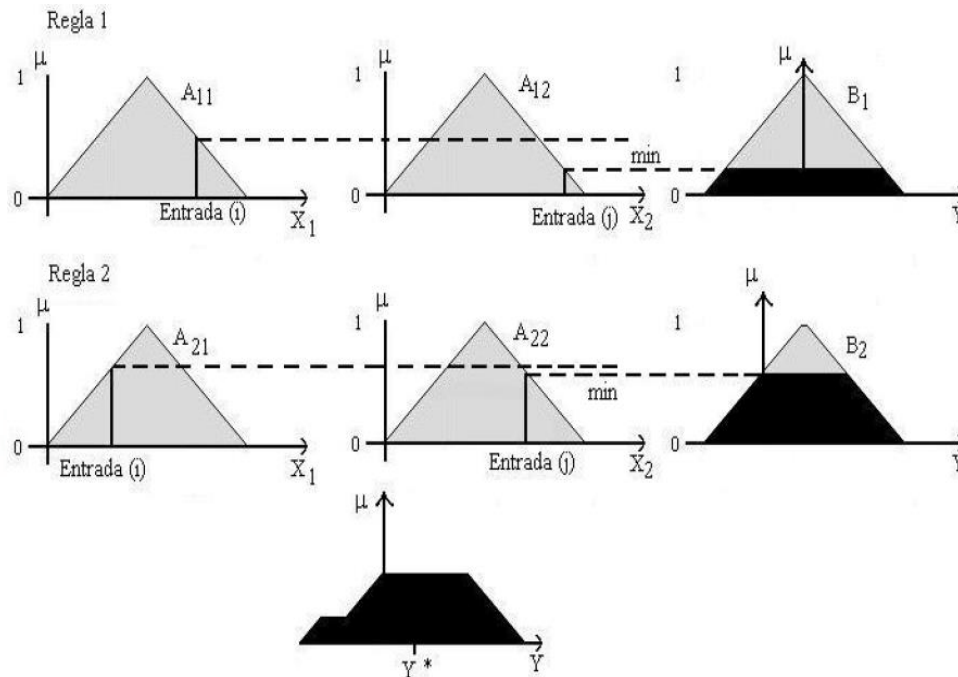


Fig. 2. 18 Método de Inferencia Máximo – Mínimo (Mamdani)
(Fuente: www.catarina.udlap.mx)

2.10.4. DEFUZIFICACIÓN – MÉTODO MANDAMI ¹³

Como última parte de los procesos del controlador difuso, se lleva a cabo la defuzificación. El método utilizado hay varios pero explicaremos el del Centroide, su selección se debe a la misma razón que método de inferencia elegido; es decir practicidad para ser implementado. Es importante mencionar, que no por ello, se compromete de ninguna manera su eficiencia, ya que estos métodos son también muy confiables.

El proceso de defuzificación por centroide es simple, dado el polígono de salida por el proceso de inferencia, simplemente se calcula el centro de gravedad de la figura mediante la fórmula. (Catarina UDLAP, pág. 52)

$$Z' = \frac{\int \mu c(z) z \, dz}{\int \mu c(z) \, dz}$$

Donde Z' es el centroide.

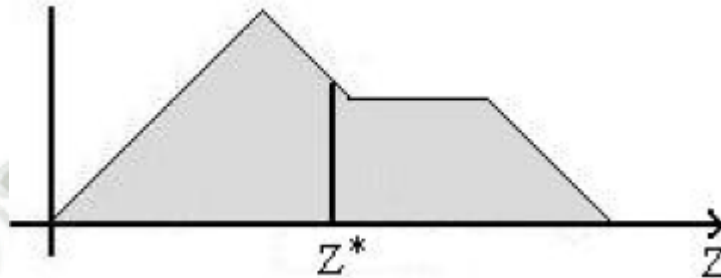


Fig. 2. 19 Método de Centroide (Mamdani)
(Fuente: www.catarina.udlap.mx)

2.10.5. MÉTODO DE TAKAGI – SUGENO ¹⁴

En los sistemas difusos Sugeno se distinguen las siguientes partes:
Fuzzificador: Realiza la misma función que en los sistemas Mamdani explicados anteriormente.

Mecanismo de inferencia difusa: Realiza la misma función que en los sistemas Mamdani explicados anteriormente.

Base de Reglas Difusas: Las reglas de la base de conocimiento de un sistema Sugeno es diferente a las de los sistemas Mamdani pues el consecuente de estas reglas ya no es una etiqueta lingüística sino que es una función de la entrada que tenga el sistema en un momento dado, esto se ilustra a continuación:



En el sistema difuso tipo Sugeno, los valores que arrojan los consecuentes de las diferentes reglas que se han activado en un momento determinado ya son valores numéricos por lo que no se necesita una etapa de defuzificación.

Para calcular la salida del sistema difuso se ponderan los diferentes consecuentes teniendo en cuenta el valor que se activó el antecedente de cada una de las reglas, para un sistema con dos reglas la salida del sistema difuso sería:

$$y = \frac{w_1 y_1 + w_2 y_2}{w_1 + w_2}$$

En este caso:

$$\begin{aligned} y_1 &= f_1(x) \\ y_2 &= f_2(x) \end{aligned}$$

Serían las funciones que permiten calcular el consecuente de cada una de las dos reglas implicadas. En el siguiente gráfico se muestra de manera más detallada como es el funcionamiento de un sistema difuso Sugeno. ¹⁴ (Takagi - Sugeno, 2017)

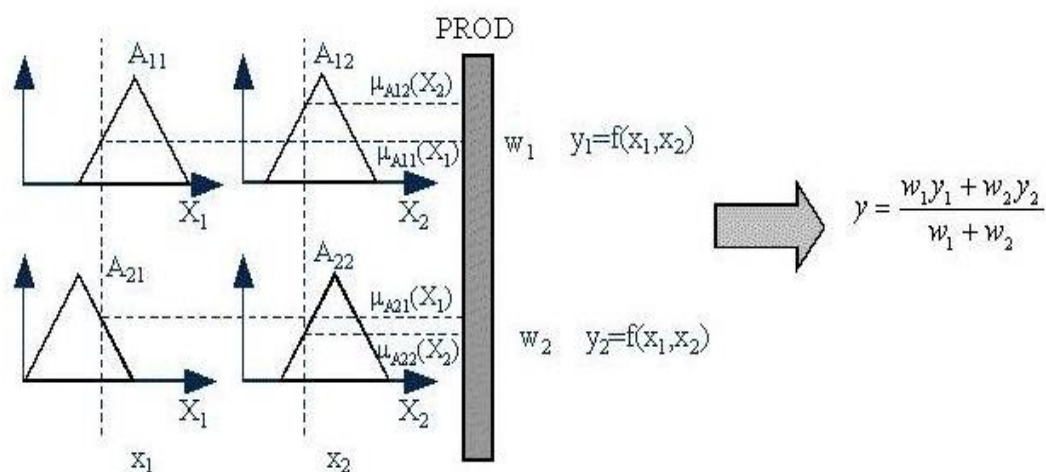


Fig. 2. 20 Funcionamiento del Sistema Difuso Sugeno

(Fuente: [www. takagi-sugeno.wikispaces.com](http://www.takagi-sugeno.wikispaces.com))

2.11. SCADA POR WEB SERVER

2.11.1. WEB SERVER DEL S7-1200

Las técnicas de automatización modernas integran cada vez más tecnologías de Internet que permiten accesos directos a la instalación a través de Intranet/Internet utilizando una comunicación transparente basada en Ethernet. El operario de puesta en servicio quiere, durante la fase de prueba y puesta en servicio, acceder de manera flexible a la CPU, así como visualizar datos individuales durante el funcionamiento para tareas de diagnóstico. Lo más lógico es utilizar estándares ya existentes para el acceso a través de Intranet o Internet, como por ejemplo tecnología HTTP, buscadores web estándar o "lenguajes" muy extendidos como HTML o JavaScript.

Las CPUs SIMATIC con interfaz PROFINET, como las del S7-1200, ofrecen la posibilidad de acceder a variables de la CPU a través de páginas web disponibles en el sistema. A través de un buscador web estándar puede acceder al servidor web integrado en la CPU S7. En este ejemplo se guardan las variables a transferir dentro de una cadena. Y esta cadena se transfiere a continuación. Todo esto reduce el tiempo de actualización. Adicionalmente a los mecanismos estándar de la página web como la identificación, el búfer de diagnóstico, el estado de los módulos, la comunicación, el estado de las variables y el archivo de datos, el usuario puede crear páginas web individuales para sus aplicaciones especiales. Con ayuda de esta aplicación mostramos cómo puede crear páginas web propias para el S7-1200 de forma rápida y sencilla. No es obligatorio disponer de conocimientos previos sobre programación en HTML.

Para la creación individual de su página web (página web definida por el usuario) puede utilizar cualquier herramienta, como por ejemplo Microsoft Frontpage, Notepad, etc. Para el diseño de la página web puede utilizar todas las posibilidades que ofrecen las tecnologías HTML, CSS y JavaScript. Además se dispone de instrucciones especiales (instrucciones AWP) para la comunicación directa con la CPU S7.



Fig. 2. 21 Visualización de un SCADA a través del Web Server
(Fuente: support.industry.siemens.com)

Los beneficios son los siguientes:

- Control y supervisión paralelo del operador basado en la web (MS Internet Explorer)
- Configuración integrada y administración de usuarios / autorizaciones de acceso
- Funcionalidades específicas del usuario, por ejemplo, idioma, planta (unidad) ver

- Tasas de actualización rápida (por ejemplo, después del cambio de valor, cambio de imagen)
- Costes mínimos de mantenimiento gracias a la administración centralizada de software
- Altos estándares de seguridad de Internet
- Supervisión remota mediante la aplicación segura de WinCC Web Viewer
- Soluciones flexibles de cliente ligero para una variedad de plataformas (PC, panel in situ, PDA móvil)
- Diagnóstico remoto eficiente con el servidor / cliente WebDiagnostics
- Mayor seguridad y disponibilidad gracias a la separación de WinCC y del servidor Web
- Máxima disponibilidad y rendimiento gracias a las granjas de servidores Web (equilibrio de carga)

2.11.2. **HTML** ¹⁵

HTML, que significa Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript).

HTML le da "valor añadido" a un texto estándar en español. Hiper Texto se refiere a enlaces que conectan una página Web con otra, ya sea dentro de una página web o entre diferentes sitios web. Los vínculos son un aspecto

fundamental de la Web. Al subir contenido a Internet y vincularlo a páginas de otras personas, te haces participante activo de esta Red Mundial.

HTML usa "markup" o marcado para anotar textos, imágenes, y otros contenidos que se muestran en el Navegador Web. El lenguaje de marcado HTML incluye "elementos" especiales tales como <head>, <title>, <body>, , <div> y muchos otros más. ¹⁵ (Barakat, 2017)

2.11.3. JAVASCRIPT ¹⁶

El uso más común de JavaScript es añadir el comportamiento del lado del cliente a las páginas HTML, también conocido como HTML dinámico (DHTML). Los scripts se incorporan o se incluyen desde páginas HTML e interactúan con el DOM (Document Object Model) de la página. Algunos ejemplos simples de este uso son:

- Cargar contenido de página nueva o enviar datos al servidor a través de Ajaxsin recargar la página (por ejemplo, una red social puede permitir al usuario publicar actualizaciones de estado sin salir de la página).
- Animación de elementos de página, desvanecimiento de ellos, cambio de tamaño, desplazamiento, etc.
- Contenido interactivo, por ejemplo juegos, y reproducción de audio y video.
- Validar valores de entrada de un formulario Web para asegurarse de que son aceptables antes de ser enviados al servidor.
- Transmitir información sobre los hábitos de lectura del usuario y las actividades de navegación a varios sitios web. Las páginas Web suelen hacer esto para análisis Web, seguimiento de anuncios, personalización u otros propósitos.

Dado que el código JavaScript puede ejecutarse localmente en el navegador de un usuario (en lugar de hacerlo en un servidor remoto), el navegador puede responder rápidamente a las acciones del usuario, lo que hace que una aplicación sea más sensible. Además, el código JavaScript puede detectar acciones del usuario que el HTML por sí solo no puede, como las pulsaciones de teclas individuales. Las aplicaciones como Gmail se aprovechan de esto: gran parte de la lógica de la interfaz de usuario está escrita en JavaScript y JavaScript envía las solicitudes de información (como el contenido de un mensaje de correo electrónico) al servidor. La tendencia más amplia de la programación de Ajax explota de manera similar esta fuerza.

Un navegador Web es, con mucho, el entorno de host más común para JavaScript. Los navegadores Web normalmente crean "objetos de host" para representar el DOM en JavaScript. El servidor Web es otro entorno de host común. Un servidor Web JavaScript típicamente expone objetos host que representan HTTP petición y objetos de respuesta, que un programa JavaScript podría entonces interrogar y manipular para generar dinámicamente páginas Web. ¹⁶ (Wikipedia - La enciclopedia libre, 2017)

CAPÍTULO 3

DESARROLLO DE INGENIERIA DE LA PLANTA PILOTO

3.1. DIAGRAMA DEL PROCESO

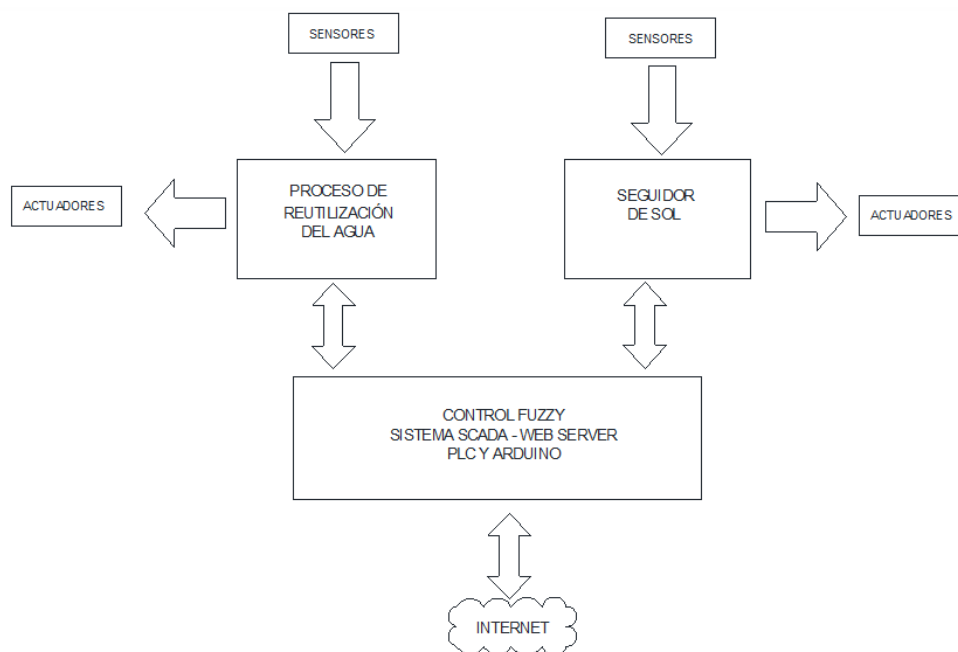


Fig. 3. 1 *Diagrama de Bloques del Sistema*
(Fuente: Elaboración Propia)

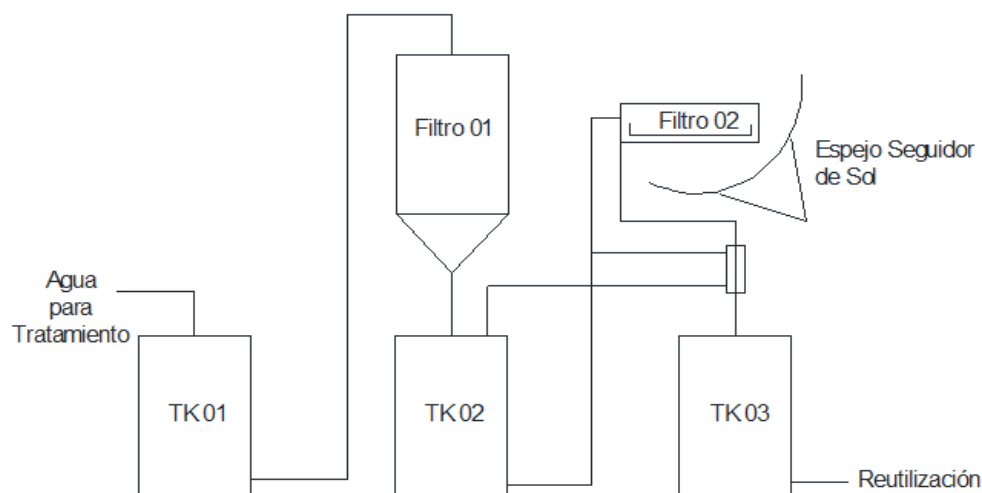


Fig. 3. 2 *Diagrama del Diseño a Desarrollar*
(Fuente: Elaboración Propia)

3.2. CONTROL DE NIVEL DE TANQUES

El Tanque n°1 será el almacenamiento de nuestra agua a tratar, el agua pasara por medio de una bomba 01 a nuestro Filtro 01, donde el encendido y apagado de la bomba 01 dependerá de lo niveles alto y bajo del Tanque 02 y del sensor de nivel ultrasónico.



Fig. 3. 3 Los Tanques 01, Filtro 01 y Tanque 02
(Fuente: Elaboración Propia)

Respecto a la capacidad que almacena el Tanque 01 es de 200 Litros de igual manera el Tanque 02. Para el Filtro 01 su capacidad es de 90 Litros.

3.2.1. CONTROL DE NIVEL DEL FILTRO 01

Para la primera etapa del proceso, en el Filtro 01 en su interior se utilizó materia prima como cascajo, piedra de diferentes dimensiones. En cada intermedio de material, se le añadió una gasa con algodón y gasa, con el fin de mejorar el filtrado.



a)



b)

Fig. 3. 4 a) Capa de Cascajo b) Capa de Gasa y Algodón

(Fuente: Elaboración Propia)

Se desarrolló el diseño del soporte de acuerdo al diagrama esquemático mostrado en la figura 3.1. La caída del agua será por gravedad que se dirigirá al Tanque N°2. Donde, esta agua al ya estar filtrada ya se puede reutilizar, mas no consumir.



Filtro 01

Tanque 02

Fig. 3. 5 Soporte para el Filtro 01
(Fuente: Elaboración Propia)

Para la medición del nivel en Filtro 01, realizamos por medio de un sensor analógico con una salida de 4 - 20mA. El sensor que se utilizó es el THE PROBE y por medio de la activación de la Bomba 01 el nivel. Su señal se dirige al PLC para su procesamiento.



Fig. 3. 6 Montaje del Sensor de Nivel The Probe
(Fuente: Elaboración Propia)

Como el PLC lee lecturas analógicas en tensión de 0 a +10V, realizamos un reacondicionamiento de la señal, de la siguiente manera:

- Como la tensión máxima que recibe el PLC es de +10V y la corriente máxima que da el sensor es de 20mA. Por la ley de Ohm se obtiene:

$$V_{plc} = I_{sensor} * R \quad (3.1)$$

$$R = \frac{10}{20mA}$$

$$R = 500 \, \Omega$$

Como no es comercial la resistencia de 500Ω , ponemos dos resistencias de $1k\Omega$ en paralelo.

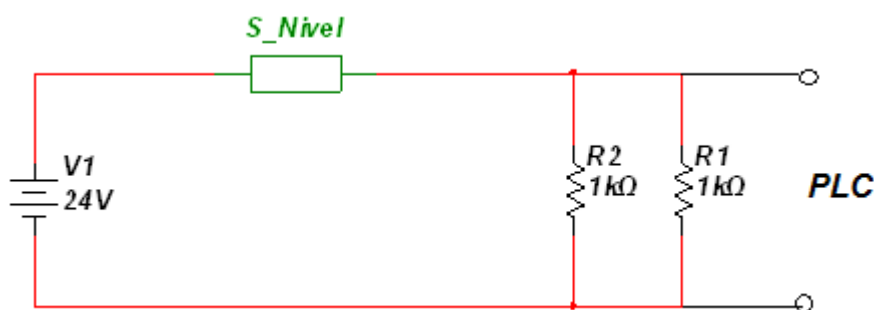


Fig. 3. 7 Acondicionamiento para el Sensor Ultrasónico PROBE
(Fuente: Elaboración Propia)

3.2.2. CONTROL DE NIVEL DEL TANQUE N° 2, TANQUE N°3

El tanque N° 2 y 3 se almacenara el agua ya tratada por el Filtro 01 y Filtro 02 respectivamente, su control será por medio de unos electrodos, donde aprovecharemos la conductividad del agua, para que cuando el nivel del agua llegue a donde se encuentre los electrodos, este cierre el circuito y podrá enviar una señal al PLC.

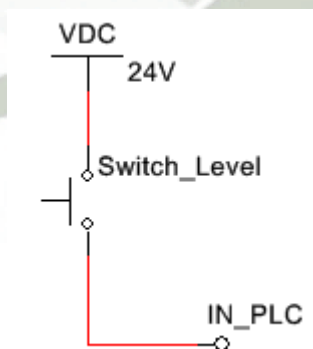


Fig. 3. 8 Conexión para la medición del nivel
(Fuente: Elaboración Propia)

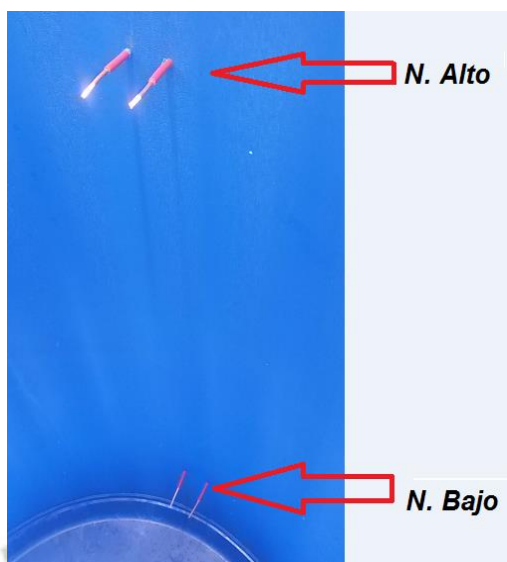


Fig. 3. 9 Electrodo para la Medición del Nivel
(Fuente: Elaboración Propia)

Las distancias que se colocaron los electrodos son los siguientes:

<i>TANQUE 02</i>	<i>DISTANCIA (cm)</i>	<i>TANQUE 03</i>	<i>DISTANCIA (cm)</i>
NIVEL BAJO	18	NIVEL BAJO	12
NIVEL ALTO	78	NIVEL ALTO	33

Tabla 3. 1 Distancias de los Electrodos de Nivel
(Fuente: Elaboración Propia)

3.2.3. CONTROL DE NIVEL DEL FILTRO 02

Para el Filtro 02 se diseñó un depósito totalmente hermético, en la cual se realizó así para obtener la mayor eficiencia del calor para que cuando sea calentado por el sol el vapor no pueda fugar por ningún lado.

De igual modo para el control del nivel se utilizó la misma lógica empleando electrodos, pero para este caso tendremos 4 niveles: Nivel Muy Bajo, Nivel Bajo, Nivel Alto y Nivel Muy Alto. La capacidad del Filtro es de 4 Litros. Las

distancias que se colocaron fueron las siguientes, tomando como referencia la base del Filtro 02.

FILTRO 02	DISTANCIA (cm)
N. Muy Bajo	3
N. Bajo	7
N. Alto	11
N. Muy Alto	15

Tabla 3. 2 Distancias de los Electrodo del Filtro 02
(Fuente: Elaboración Propia)

En el diseño de nuestro filtro 02 se consideró una entrada que será para el ingreso del agua y una salida que será para el vapor de agua. Donde este se dirigirá a un intercambiador de calor, que consistirá en enfriar el agua caliente y que pueda fluir hacia el Tanque 03.



Fig. 3. 10 Filtro 02 para la Evaporación del Agua
(Fuente: Elaboración Propia)

3.3. **INTERCAMBIADOR DE CALOR**

Se realizó un intercambiador de calor donde consiste en que la manguera atraviese el tubo, donde este circulara agua proveniente del Tanque 02 y el que dará paso será la válvula de circulación. Este sistema consta en que el agua de estar en estado gaseosos pase a estado líquido.



Fig. 3. 11 Intercambiador de Calor
(Fuente: Elaboración Propia)

3.4. **SEGUIDOR SOLAR**

Para la construcción del seguidor solar se diseñó para que tenga dos grados de libertad y una parte fija; donde ahí se colocara el Filtro 02 para el calentamiento del agua.

La superficie donde capta los rayos del sol es cóncava y de un material refractante donde su concentración máxima será en el foco, donde ahí estará colocado el Filtro 2. Esta superficie rotará de acuerdo a como se va trasladando el sol.

La otra parte móvil del seguidor solar girara de acuerdo a las estaciones ya que por cada tiempo el sol tiene un eje específico de salida de acuerdo al transcurrir el año.



Fig. 3. 12 Estructura para Seguidor Solar
(Fuente: Elaboración Propia)

Los sensores que forman parte para el control del seguidor solar son los siguientes:

3.4.1. SENSOR DE LUZ O LDR

Para el control de luz se instalaron dos LDR (light dependent resistor) o fotoresistencia, para el seguimiento del sol, donde su función principal es la de estar perpendicularmente con el sol para una mejor captación de los rayos solares.

La fotorresistencia consta de una resistencia donde su valor aumenta a medida que no le incide la luz y su valor de la resistencia es muy bajo al incidirle la luz, su rango va desde los 10 Ohm hasta los Megaohmios.

Para que nuestro controlador (Arduino) pueda leer la señal que envía el LDR, hemos acondicionado un circuito que se hará una comparación de voltaje, en la que me enviara una tensión de acuerdo a la incidencia de la luz en el LDR. Hemos utilizado dos LDR donde uno será Horizontal y el otro Vertical.



Fig. 3. 13 Sensores de Luz para Seguidor Solar
(Fuente: Elaboración Propia)

El circuito que se implementó para que el Arduino pueda procesar la señal mediante el LDR es el siguiente:

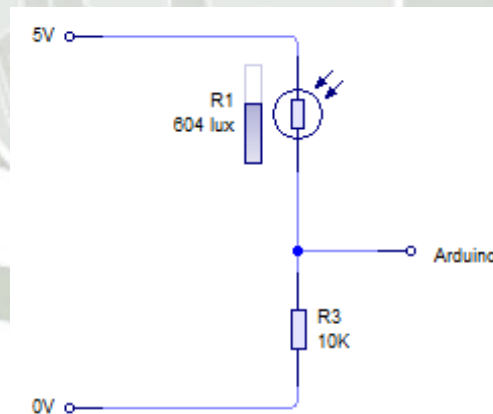


Fig. 3. 14 Circuito para el Sensor LDR
(Fuente: Elaboración Propia)

3.4.2. MOTORES PARA EL MOVIMIENTO DEL SEGUIDOR SOLAR

Para el movimiento de la estructura móvil del seguidor solar se instaló dos motores de 12 Vdc de 600mA, con su caja reductora, con el fin de darle el movimiento deseado con un par de 15kg/cm. De acuerdo a su manera de conexión me permiten que los motores giren en ambos sentidos.



a)



b)

Fig. 3. 15 a) Motor de Traslación. b) Motor de Rotación.
(Fuente: Elaboración Propia)

3.4.2.1. CIRCUITO PARA EL ACCIONAMIENTO DE LOS MOTORES

Para el accionamiento de la parte de potencia de los motores utilizamos el driver dual L298N, que me permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 A.

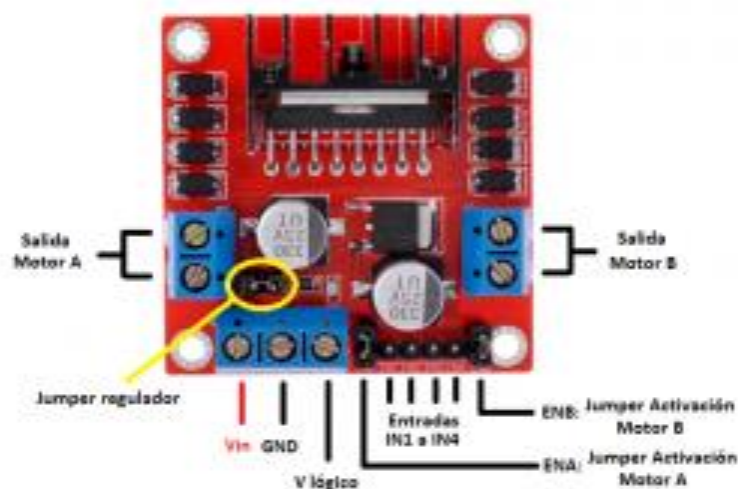


Fig. 3. 16 Driver Dual L298N
(Fuente: www.prometec.net)

Deseamos que los motores realice un movimiento suave por lo que hacemos trabajar a los motores a una tensión de 6V, por lo que su corriente llega a consumir hasta los 900mA, lo que nos lleva a implementar un circuito que maneje corriente elevada y regular la tensión.

Para la etapa de potencia para nuestro seguidor solar utilizamos una fuente de 24V y para poder regular a una tensión a partir de los 5V, utilizamos un regulador de tensión el LM7405, implementamos el siguiente circuito.

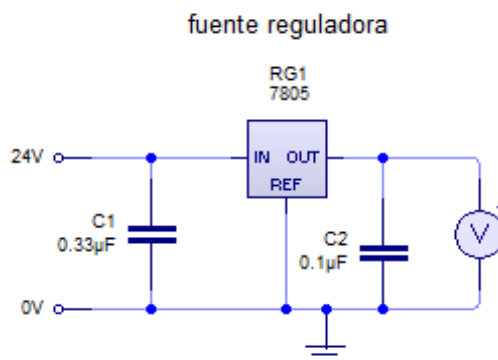


Fig. 3. 17 Regulador de Tensión con Salida de 5V
(Fuente: Elaboración Propia)

Luego desarrollamos el circuito para regular la tensión y a la vez para que soporte la corriente de consumo de los motores.

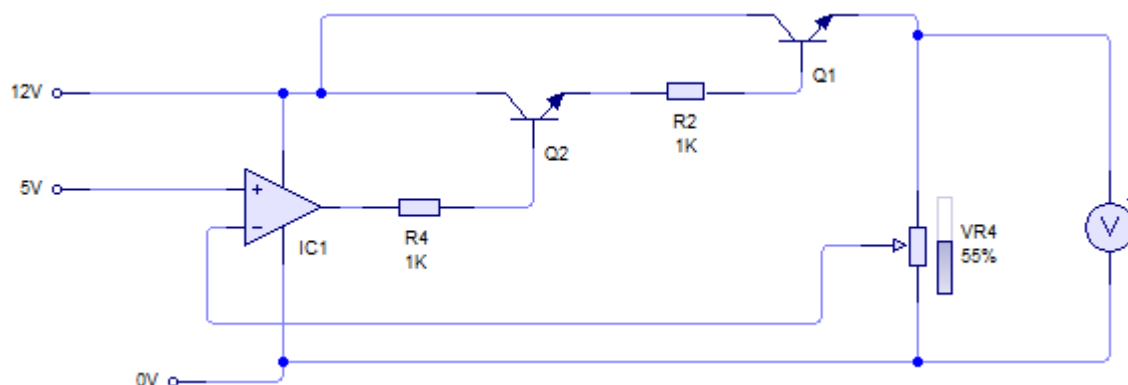


Fig. 3. 18 Circuito Regulador de Tensión Variable y Corrientes Elevadas
(Fuente: Elaboración Propia)

Donde:

IC1 = Lm358

Q1= 2N3055

Q2= BC548

Potenciómetro VR4 = 100K Ω

3.4.2.2. CIRCUITO DE LA ETAPA DE CONTROL PARA LOS MOTORES

Para esta etapa utilizamos opto-acopladores 4N35, por lo que para cada motor son dos para activar en ambos sentidos. Con el fin de aislar las tensiones y así separar la etapa de control con la etapa de potencia.

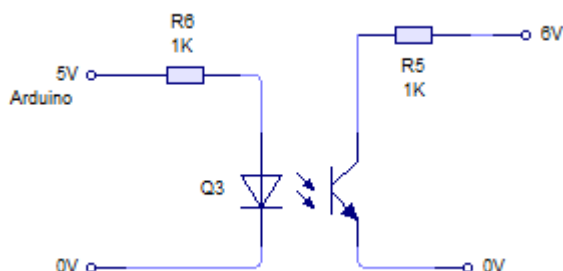


Fig. 3. 19 Circuito con Optoacoplador 4N35
(Fuente: Elaboración Propia)

3.4.3. SENSOR DE TEMPERATURA

Para la medición de la temperatura utilizamos el integrado LM35. Tiene un rango de medición de -55°C hasta 150°C , tiene una impedancia baja y una salida lineal, por lo que la tensión que entrega es proporcional a la temperatura.

Su montaje es simple ya que al ser un encapsulado tipo transistor solo tiene dos entradas de alimentación y una salida.



Fig. 3. 20 Sensor de Temperatura LM35
(Fuente: www.luisllamas.es)

Su conexión con el microcontrolador Arduino es de la siguiente manera:

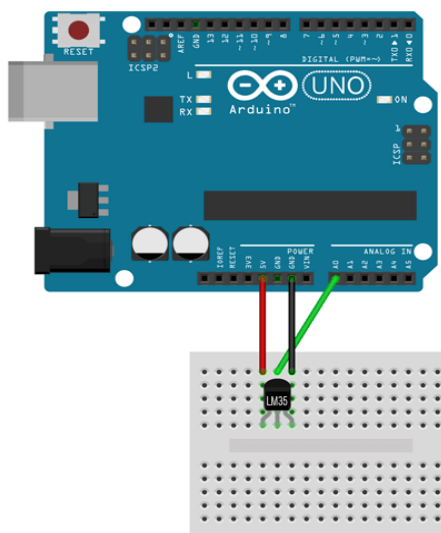


Fig. 3. 21 Conexión con el Arduino
(Fuente: www.luisllamas.es)

3.4.4. *SENSORES FINES DE CARRERA*

Para los sensores fines de carrera, que serán los que me pondrán el límite de que mi seguidor solar se detenga, estos sensores son importantes también para la lógica que tendrán los motores. Al ser un contacto su circuito es de la siguiente manera.

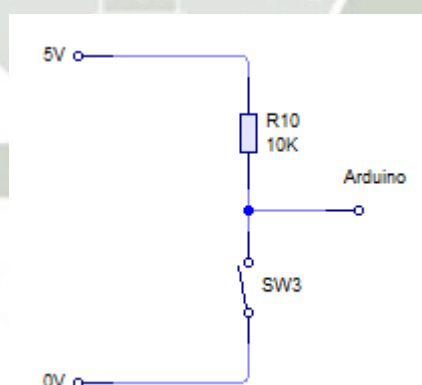


Fig. 3. 22 Circuito para el Fin de Carrera
(Fuente: Elaboración Propia)

3.4.5. SENSOR DE FLUJO

Para la conexión del flujometro con el Arduino lo conectamos al pin 2 del PWM, esto debido a que este pin permite las interrupciones, con el fin de que en la programación podamos leer los datos que enviara el sensor. El sensor de flujo realiza medición de 1 – 30 L/min.



Fig. 3. 23 Conexión con el Arduino
(Fuente: www.naylampmechatronics.com)

3.5. IMPLEMENTACIÓN DEL PROGRAMA DE LA PLANTA PILOTO

Para la implementación del programa de control del seguidor solar, se desarrolló por medio de un Arduino Mega 2560, donde realizara el censado de temperatura en el filtro 02, la temperatura en el sistema, y la medición del caudal de flujo. El microcontrolador se conectó con una tarjeta de Shield Ethernet W5100.

Para el control de los Tanques 02 y 03, el Filtro 02, válvulas y bombas, se desarrollara mediante el PLC S7-1200. La comunicación se realizara por medio del protocolo de comunicación ModBus TCP/IP, que se comunicara el PLC, Arduino y la PC.

3.6. PROGRAMA EN EL ARDUINO MEGA 2560

Para iniciar la programación primero realizamos la designación de las entradas y salidas para el microcontrolador Arduino, que es de la siguiente manera:

Salidas		
Dirección	Descripción	TAG
Pin 3 PWM	Motor Traslación Horario	103-SE-001
Pin 4 PWM	Motor Rotación Horario	103-SE-002
Pin 5 PWM	Motor Traslación Antihorario	103-SE-003
Pin 6 PWM	Motor Rotación Antiorario	103-SE-004

Tabla 3. 3 Salidas Arduino
(Fuente: Elaboración Propia)

Entradas		
Dirección	Descripción	TAG
Pin 2 PWM	Flujometro	102-FT-001
Pin 7 PWM	Final de carrera Este	103-ZS-001
Pin 8 PWM	Final de carrera Oeste	103-ZS-002

Tabla 3. 4 Entradas Arduino
(Fuente: Elaboración Propia)

Entrada Analógicas		
Dirección	Descripción	TAG
A0	LDR Vertical	103-RS-001
A1	LDR Horizontal	103-RS-002
A2	Temperatura del Filto 2	103-TT-001
A3	Temperatura en el plato Solar	103-TT-002

Tabla 3. 5 Entradas Analógicas Arduino
(Fuente: Elaboración Propia)

Para iniciar la programación en el Arduino procedemos abrir el programa y empezamos a la configuración.

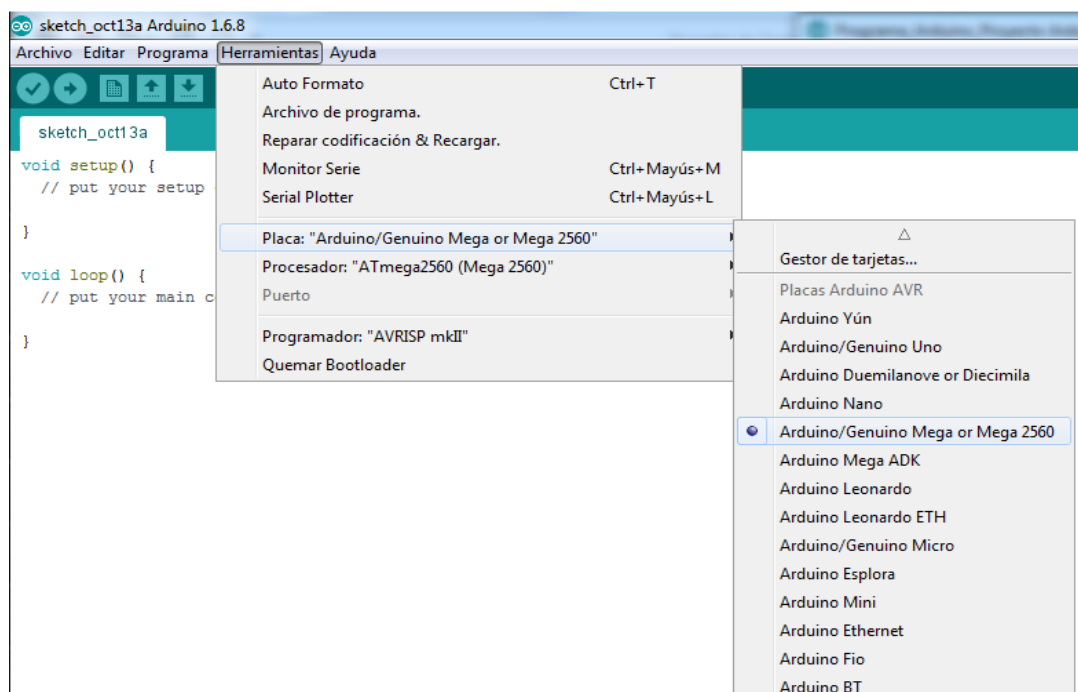


Fig. 3. 24 Configuración de la Tarjeta que se utilizara
(Fuente: Arduino - Home)

Luego configuramos el procesador del Arduino Mega

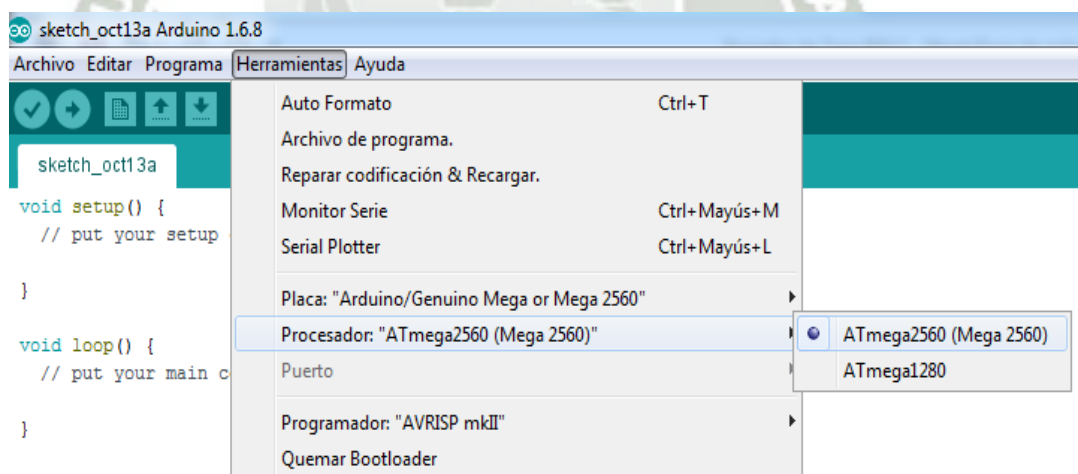


Fig. 3. 25 Configuración del Procesador del Arduino Mega 2560
(Fuente: Arduino - Home)

3.6.1. CONFIGURACIÓN DE LAS LIBRERÍAS PARA LA COMUNICACIÓN

MODBUS

Para la configuración descargamos las librerías Modbus.cpp, Modbus.h, ModbusIP.cpp y ModbusIP.h. Estas librerías me permiten comunicarme a través del protocolo Modbus. El Modbus es un protocolo maestro- esclavo utilizado en la automatización industrial y permite utilizarse en otras áreas.

Estas bibliotecas permiten que el Arduino funcione como esclavo, admitiendo Modbus Serial y Modbus IP, cuando usamos el Modbus, el protocolo de transporte es TCP (puerto 502).

Como en nuestra transmisión de datos no necesitamos una comunicación TCP keep-alive, ya que envía telegramas cíclicamente para determinar si la estación del otro lado del enlace esta todavía disponible, en caso de que no esté disponible durante un tiempo se desactiva el enlace, esto es importante solo para aplicaciones se servidores, es por ello que en la librería ModbusIP.h, se comenta el TCP_KEEP_ALIVE.

Programa_Arduino_Proyecto	Modbus.cpp	Modbus.h	ModbusIP.cpp	ModbusIP.h
---------------------------	------------	----------	--------------	------------

```

/*
  ModbusIP.h - Header for Modbus IP Library
  Copyright (C) 2015 André Sarmiento Barbosa
*/
#include <Arduino.h>
#include "Modbus.h"
#include <SPI.h>
#include <Ethernet.h>

#ifndef MODBUSIP_H
#define MODBUSIP_H

#define MODBUSIP_PORT 502
#define MODBUSIP_MAXFRAME 200

// #define TCP_KEEP_ALIVE

```

Fig. 3. 26 Comentar la instrucción TCP_KEEP_ALIVE
(Fuente: www.github.com/modbusarduino)

En la biblioteca de Modbus.h hay una opción para limitar la operación a las funciones de Holding Register, para ello se comenta esa instrucción.



```

/*
  Modbus.h - Header for Modbus Base Library
  Copyright (C) 2014 André Sarmiento Barbosa
*/
#include "Arduino.h"

#ifndef MODBUS_H
#define MODBUS_H

#define MAX_REGS    32
#define MAX_FRAME   128
// #define USE_HOLDING_REGISTERS_ONLY

```

Fig. 3. 27 Comentar la instrucción USE_HOLDING_REGISTERS_ONLY
(Fuente: www.github.com/modbusarduino)

Al optar por Modbus IP, debe incluirse en el encabezado las instrucciones:

```

#include <SPI.h>
#include <Ethernet.h>
#include "Modbus.h"
#include "ModbusIP.h"

```

Para establecer el registro de Modbus para representar el interruptor. Este valor es el desplazamiento basado en 0 que se colocara en su software de supervisión o prueba. Tener en cuenta que si su software utiliza compensaciones en 1, el valor establecido debe ser 100. Para nuestro programa declaramos los siguientes registros:

```
const int TEMP_FILTRO02 = 100;
const int TEMP_PLATOSOLAR = 101;
const int FLUJOMETRO = 102;
const int POSICION_PLATO = 103;
const int LDR_VERTICAL = 104;
const int LDR_HORIZONTAL = 105;
```

Luego se declara la instrucción de:

```
ModbusIP mb;
```

Donde se crea la instancia de mb (ModbusIP) que se utilizara.

Ahora declaramos la dirección MAC y la IP que pasa por el método de config().

```
void setup() {
    Serial.begin(9600);

    // The media access control (ethernet hardware) address for the
    shield

    byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

    // The IP address for the shield
    byte ip[] = { 192, 168, 0, 12 };

    //Config Modbus IP

    mb.config(mac, ip);
```

Después proseguimos con agregar el tipo de registro de estado de entrada analógica, solo lectura, que es responsable de detectar si un interruptor está en estado activo o desactivado.

```
mb.addlreg(TEMP_FILTRO02);  
mb.addlreg(TEMP_PLATOSOLAR);  
mb.addlreg(FLUJOMETRO);  
mb.addlreg(POSICION_PLATO);  
mb.addlreg(LDR_VERTICAL);  
mb.addlreg(LDR_HORIZONTAL);
```

La instrucción “mb.task()” responde las solicitudes y cambia los registros si es necesario, debe ser llamado una sola vez, al inicio del ciclo.

Y finalmente, los valores de los registros cambian de acuerdo a los datos que se vaya enviado de acuerdo a nuestro programa o valor que deseamos que se almacene.

```
mb.lreg(LDR_VERTICAL, ldr_vertical);  
mb.lreg(LDR_HORIZONTAL, ldr_horizontal);  
mb.lreg(TEMP_FILTRO02, celsius_filtro);  
mb.lreg(TEMP_PLATOSOLAR, celsius_plato);  
mb.lreg(FLUJOMETRO, caudal_L_m);
```

3.6.2. DECLARACIÓN DE LAS VARIABLES

Para esta etapa procedemos a la declaración de todas nuestras entradas y salidas y variables que utilizaremos para realizar las ecuaciones para el

calculo respectivo de cada sensor, de acuerdo al tipo de entrada ya sea int, const, float.

//Declaracion de Variables

// Variables para LDR

```
const int LDR_Vertical= A0;
const int LDR_Horizontal= A1;
int ldr_vertical_adc=0;
int ldr_horizontal_adc=0;
float ldr_vertical=0;
float ldr_horizontal=0;
const int PosPlato = A4;
```

//Variables para temperatura

```
const int TempFiltro2 = A2;
const int TempPlato = A3;
//Variables de Motor
int Motor_T_Ho=3; // Motor de traslación para giro HORARIO
int Motor_T_An timer=5; //Motor de traslación para giro ANTIHORARIO
int Motor_R_Ho=4; //Motor de rotación para giro HORARIO
int Motor_R_An timer=6; //Motor de rotación para giro ANTIHORARIO
```

//Variables de Finales de Carrera

```
int Este= 7;
int Oeste=8;
int valor_Este=0;
int valor_Oeste=0;
```

//Variable del Flujometro

```
volatile int NumPulsos; //variable para la cantidad de pulsos recibidos
int Flujometro=2;
float fc=7.5; //Factor de conversión
```

3.6.3. PROGRAMA DEL SEGUIDOR SOLAR

A continuación la descripción del programa del seguidor solar.

```
pinMode(Motor_T_Ho,OUTPUT); //Activación para el motor de
Traslación(Giro_Horario)
pinMode(Motor_T_An timer,OUTPUT); //Activación para el motor de
Traslación(Giro_Antihorario)
pinMode(Motor_R_Ho,OUTPUT); //Activación para el motor de
Rotación(Giro_Horario)
pinMode(Motor_R_An timer,OUTPUT); //Activación para el motor de
Rotación(Giro_Antihorario)
pinMode(Este,INPUT); //Entrada fin de carrera lado Este
pinMode(Oeste,INPUT); //Entrada fin de carrera lado Oeste

void loop() {
//Call once inside loop() - all magic here
mb.task();
//Read each two seconds
if (millis() > ts + 350) {
ts = millis();

//Logica de LDR
ldr_vertical_adc=analogRead(LDR_Vertical);
ldr_vertical=ldr_vertical_adc*(5.0/1023.0); //Ecuación para saber el
nivel de tensión en el LDR Vertical
ldr_horizontal_adc=analogRead(LDR_Horizontal);
ldr_horizontal=ldr_horizontal_adc*(5.0/1023.0); //Ecuación para
saber el nivel de tensión en el LDR Horizontal
mb.lreg(LDR_VERTICAL, ldr_vertical);
mb.lreg(LDR_HORIZONTAL, ldr_horizontal);
Serial.print("Voltaje_V: ");
Serial.print(ldr_vertical);
Serial.print(" Voltaje_H: ");
Serial.print(ldr_horizontal);
```

//Logica de los finales de carrera

```
valor_Este=digitalRead(Este); //Lectura del sensor de fin de carrera
lado Este
valor_Oeste=digitalRead(Oeste); //Lectura del sensor de fin de
carrera lado Oeste
```

//Logica para los motores con los finales de carrera

```
if(valor_Este==0)
digitalWrite(Motor_T_Ho,LOW);
if((4.70<=ldr_vertical)&&(valor_Este==0))
digitalWrite(Motor_T_Anh,HIGH);

if((3.95>ldr_vertical)&&(ldr_vertical>=2.5)&&(valor_Este==1)&&(valor
_Oeste==1))
digitalWrite(Motor_T_Anh,HIGH);
if((ldr_vertical>=4.10)&&(valor_Este==1))
digitalWrite(Motor_T_Anh,LOW);
if(valor_Oeste==0)
digitalWrite(Motor_T_Anh,LOW);
if((ldr_vertical<=1.5)&&(valor_Este==1))
digitalWrite(Motor_T_Ho,HIGH);
```

3.6.4. PROGRAMA PARA MEDICIÓN DE TEMPERATURA

Ahora desarrollamos la lógica para la lectura de los sensores de temperatura tanto para el filtro 02 y el sistema del seguidor solar

//Logica de lectura de la temperatura

//sensor de temperatura filtro 02

```
int value_filtro = analogRead(TempFiltro2);
float millivolts_filtro = (value_filtro / 1024.0) * 5000;
float celsius_filtro = millivolts_filtro / 10;
Serial.print(" TEMPERATURA FILTRO 02: ");
Serial.print(celsius_filtro);
mb.lreg(TEMP_FILTRO02, celsius_filtro);
```

//sensor de temperatura Plato

```
int value_plato = analogRead(TempPlato);
float millivolts_plato = (value_plato / 1024.0) * 5000;
float celsius_plato = millivolts_plato / 10;
Serial.print(" TEMPERATURA PLATO: ");
Serial.print(celsius_plato);
//Serial.println();
mb.lreg(TEMP_PLATOSOLAR, celsius_plato);
```

3.6.5. PROGRAMA DE LECTURA DEL FLUJOMETRO

Para este caso se tuvo que desarrollar un void ContarPulsos(), debido a que la lectura es por medio de PWM, por lo cual se tuvo que desarrollar un conteo de pulsos y habilitación de interrupción y por lo cual obtenemos una frecuencia.

```
//Función que se ejecuta en interrupción
void ContarPulsos()
{
    NumPulsos++; //incrementamos la variable de los pulsos
}

//Función para obtener frecuencia de los pulsos
int ObtenerFrecuencia()
{
    int frecuencia;
    NumPulsos = 0; //Ponemos a 0 el número de pulsos
    interrupts(); //Habilitamos las interrupciones
    delay(1000); //muestra de 1 segundo
    noInterrupts(); //Desabilitamos las interrupciones
    frecuencia=NumPulsos; //Hz(pulsos por segundo)
    return frecuencia;
}

//Logica de Lectura de Flujometro
    float frecuencia=ObtenerFrecuencia(); //obtenemos la Frecuencia
    de los pulsos en Hz
    float caudal_L_m=frecuencia/fc; //calculamos el caudal en L/m
    float caudal_L_h=caudal_L_m*60; //calculamos el caudal en L/h
    Serial.print( "Caudal: ");
    Serial.println(caudal_L_m);
    mb.lreg(FLUJOMETRO, caudal_L_m);
```

3.7. **DESARROLLO DEL PROGRAMA EN EL PLC SIEMENS S7-1200**

Para la implementación del programa en el PLC, primeramente se tuvo que identificar todas las entradas y salidas que se tendrá, para ello se desarrolló el siguiente cuadro:

Entradas Digitales		
Dirección	Descripción	TAG
I0:0	Level_Low_Tk-02	101-LSL-021
I0:1	Level_High_Tk-02	101-LSH-022
I0:2	Level_Very_Low_FI-02	102-LSLL-020
I0:3	Level_Low_FI-02	102-LSL-021
I0:4	Level_High_FI-02	102-LSH-022
I0:5	Level_Very_High_FI-02	102-LSHH-023
I0:6	Level_Low_Tk-03	102-LSL-024
I0:7	Level_High_Tk-03	102-LSH-025

Tabla 3. 6 Entradas Digitales PLC S7-1200
(Fuente: Elaboración Propia)

Salidas Digitales		
Dirección	Descripción	TAG
Q0:0	PU_01	101-PU-001
Q0:1	PU_02	102-PU-002
Q0:2	Valve_FI_2	102-XS-001
Q0:3	Valve_Circulación	102-XS-002

Tabla 3. 7 Salidas Digitales PLC S7-1200
(Fuente: Elaboración Propia)

Entrada Analógica		
Dirección	Descripción	TAG
Iw64	Level_FI-01	101-LIT-020

Tabla 3. 8 Entrada Analógica PLC S7-1200
(Fuente: Elaboración Propia)

Salida Analógica		
Dirección	Descripción	TAG
Qw80	VFD_Pum_01_y_02	101-VFD-001

Tabla 3. 9 Salida Analógica PLC S7-1200
(Fuente: Elaboración Propia)

3.7.1. COMUNICACIÓN MODBUS DEL S7-1200 CON EL ARDUINO

Primero lo que realizamos es configurar el IP del PLC que para este proyecto lo configuramos con el IP: 192.168.0.1. Luego creamos un Bloque de Función con el nombre “ModbusArduino”, luego procedemos a añadir la instrucción de Comunicación Modbus TCP/ MB_Client. Se le configura como un DB multi-instancia.

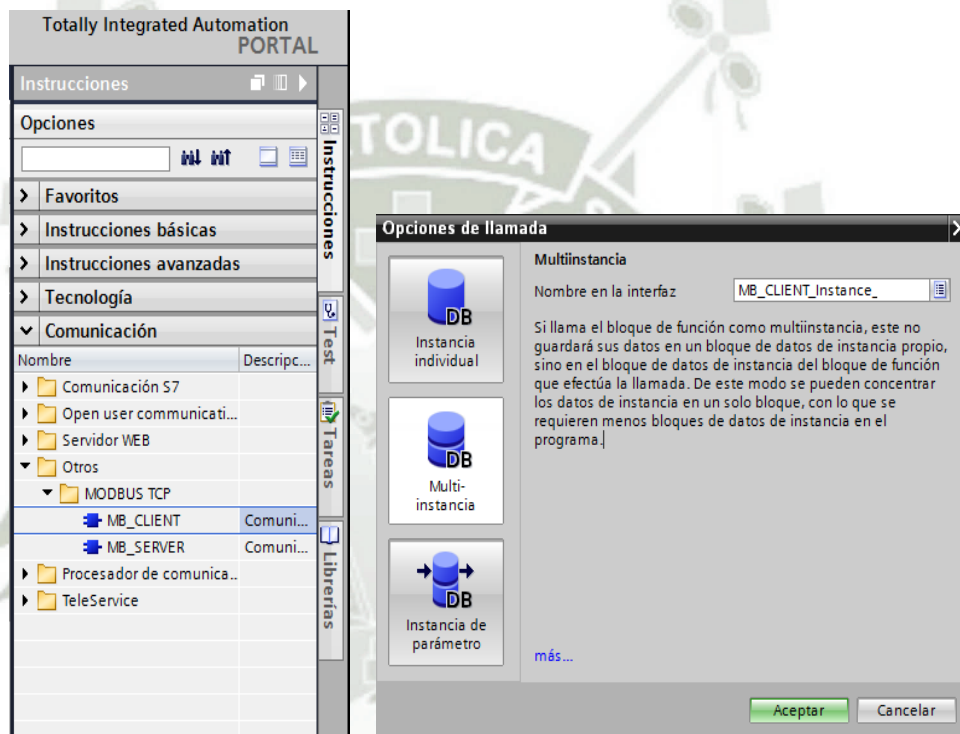


Fig. 3. 28 Instrucción Modbus TCP
(Fuente: PLC Siemens S7-1200)

Luego procedemos a la configuración del MB_CLIENTE, introduciendo todos los parámetros.

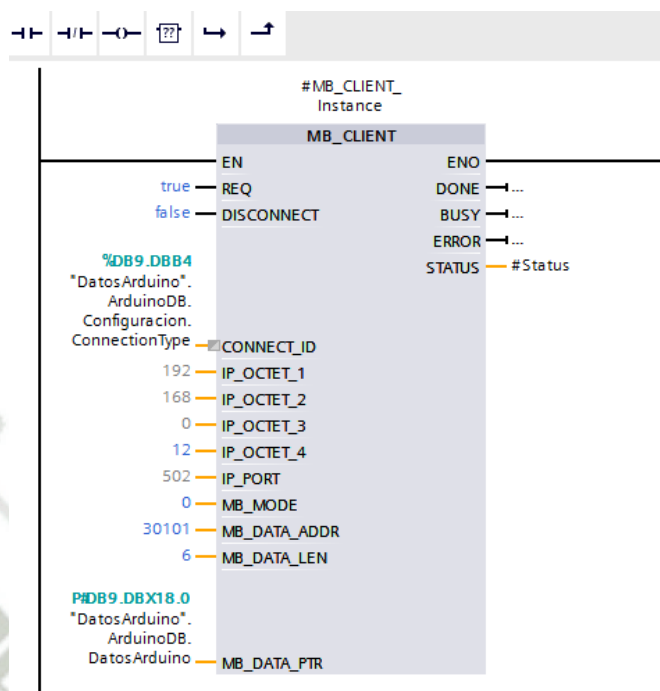


Fig. 3. 29 Bloque de Función para la Comunicación con el Arduino
(Fuente: PLC Siemens S7-1200)

Respecto a “CONNECT_ID” y “MB_DATA_PTR” creamos un bloque de datos.

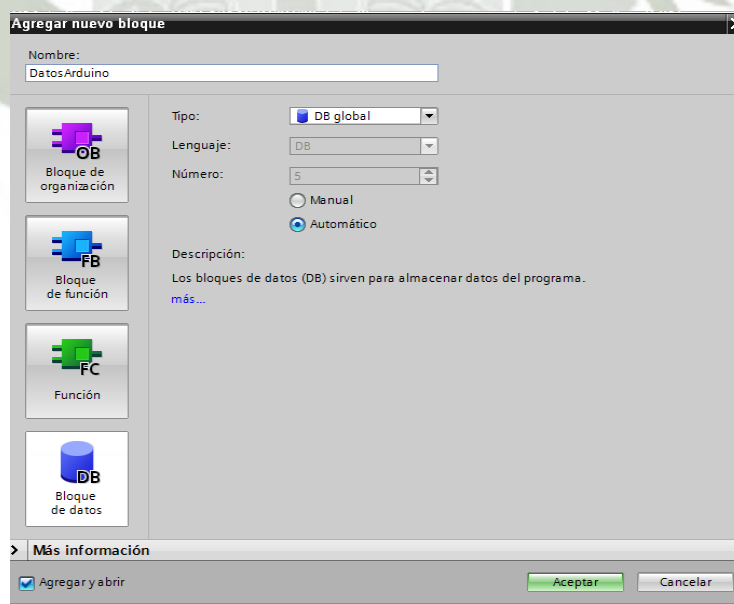
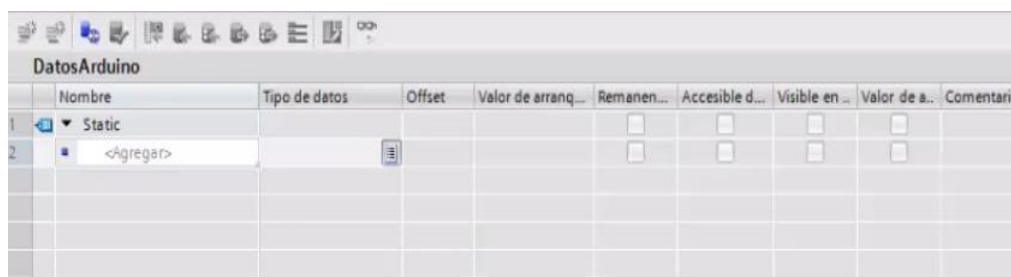


Fig. 3. 30 Creación de Bloque de Datos para el Arduino
(Fuente: PLC Siemens S7-1200)

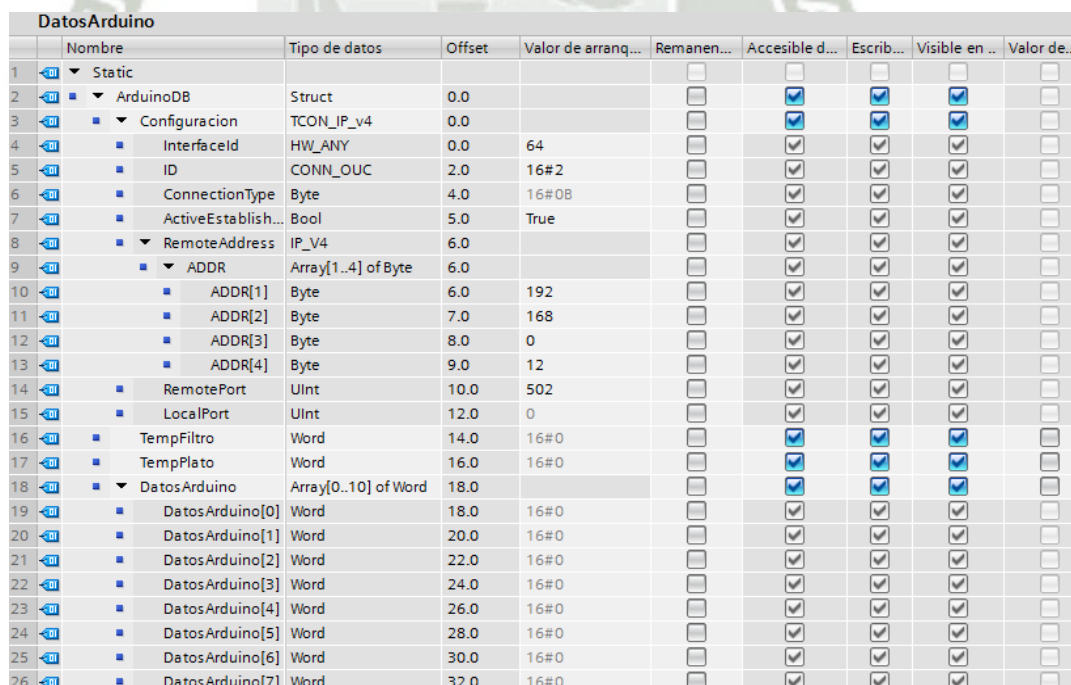
Luego nos vamos a Propiedades / Atributos y desactivamos el check de Acceso optimizado al bloque, con el fin de ver los valor de los bits que se están marcando. Luego nos aparecerá la siguiente pantalla:



	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Visible en ..	Valor de a...	Comentario
1	Static				☐	☐	☐	☐	
2	<Agregar>				☐	☐	☐	☐	

Fig. 3. 31 Iniciación de Configuración Datos Arduino
(Fuente: PLC Siemens S7-1200)

- Empezamos con la configuración colocando el nombre “ArduinoDB” con tipo de dato “Struct”.
- Luego escribimos “Configuración” y su tipo de dato será un “TCON_IP_v4, se desplegara una serie de valores que configuramos como se muestra a continuación:



	Nombre	Tipo de datos	Offset	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ..	Valor de...
1	Static				☐	☐	☐	☐	
2	▼ ArduinoDB	Struct	0.0		☐	☒	☒	☒	
3	▼ Configuracion	TCON_IP_v4	0.0		☐	☒	☒	☒	
4	InterfaceId	HW_ANY	0.0	64	☐	☒	☒	☒	
5	ID	CONN_OUC	2.0	16#2	☐	☒	☒	☒	
6	ConnectionType	Byte	4.0	16#08	☐	☒	☒	☒	
7	ActiveEstablish...	Bool	5.0	True	☐	☒	☒	☒	
8	▼ RemoteAddress	IP_V4	6.0		☐	☒	☒	☒	
9	▼ ADDR	Array[1..4] of Byte	6.0		☐	☒	☒	☒	
10	ADDR[1]	Byte	6.0	192	☐	☒	☒	☒	
11	ADDR[2]	Byte	7.0	168	☐	☒	☒	☒	
12	ADDR[3]	Byte	8.0	0	☐	☒	☒	☒	
13	ADDR[4]	Byte	9.0	12	☐	☒	☒	☒	
14	RemotePort	UInt	10.0	502	☐	☒	☒	☒	
15	LocalPort	UInt	12.0	0	☐	☒	☒	☒	
16	TempFiltro	Word	14.0	16#0	☐	☒	☒	☒	
17	TempPlato	Word	16.0	16#0	☐	☒	☒	☒	
18	▼ DatosArduino	Array[0..10] of Word	18.0		☐	☒	☒	☒	
19	DatosArduino[0]	Word	18.0	16#0	☐	☒	☒	☒	
20	DatosArduino[1]	Word	20.0	16#0	☐	☒	☒	☒	
21	DatosArduino[2]	Word	22.0	16#0	☐	☒	☒	☒	
22	DatosArduino[3]	Word	24.0	16#0	☐	☒	☒	☒	
23	DatosArduino[4]	Word	26.0	16#0	☐	☒	☒	☒	
24	DatosArduino[5]	Word	28.0	16#0	☐	☒	☒	☒	
25	DatosArduino[6]	Word	30.0	16#0	☐	☒	☒	☒	
26	DatosArduino[7]	Word	32.0	16#0	☐	☒	☒	☒	

Fig. 3. 32 Configuración de Datos Arduino

(Fuente: PLC Siemens S7-1200)

Luego ya en el Bloque de Funciones vamos y configuramos en MB_DATA_PTR, con "DatosArduino".ArduinoDB.DatosArduino. Es ahí donde se escribirán los datos que se envían del Arduino.

Y en CONNECT se configura con "DatosArduino".ArduinoDB.Configuracion.ConnectionType y en STATUS definimos la variable #Status.

En la que podemos observar que para cada dato que recibe el PLC es de tipo Word y para este programa solo utilizaremos los 5 primeros, desde el 18 al 26, ya que se creó un Array 10.

Y por último en el Bloque de Main agregamos el Bloque de Funciones del Arduino.

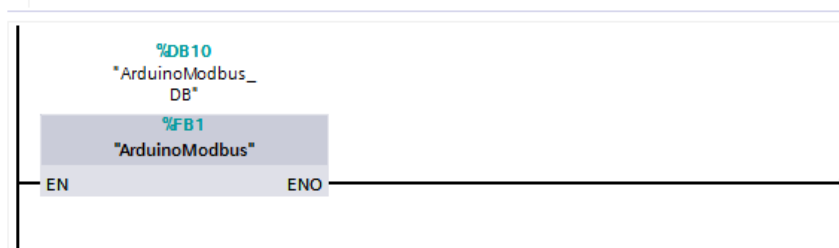
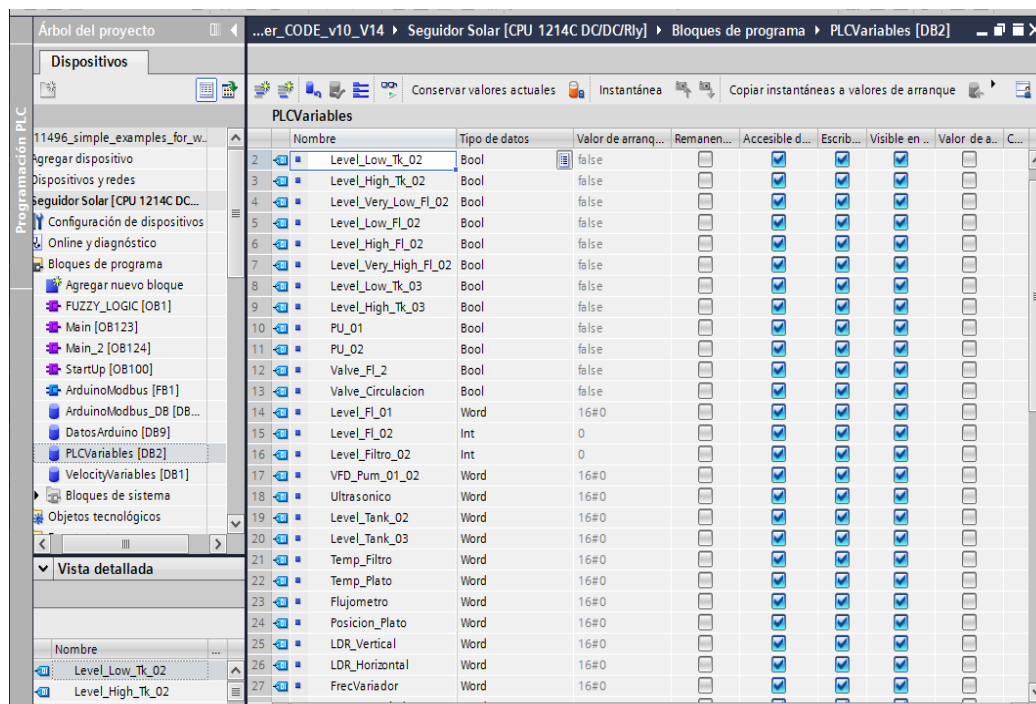


Fig. 3. 33 Bloque de Funciones en el Main
(Fuente: PLC Siemens S7-1200)

3.7.2. DECLARACIÓN DE VARIABLES EN BLOQUE DE DATOS PARA EL SCADA

Ahora sabiendo que datos vamos a utilizar, necesitamos crear un bloque de datos donde almacenaremos los datos en memorias donde servirán para la lectura de datos para el SCADA desarrollada por HTML.



Nombre	Tipo de datos	Valor de arranq...	Remanen...	Accesible d...	Escrib...	Visible en ..	Valor de a...	C...
2	Level_Low_Tk_02	Bool	false		☒	☒	☒	
3	Level_High_Tk_02	Bool	false		☒	☒	☒	
4	Level_Very_Low_Fl_02	Bool	false		☒	☒	☒	
5	Level_Low_Fl_02	Bool	false		☒	☒	☒	
6	Level_High_Fl_02	Bool	false		☒	☒	☒	
7	Level_Very_High_Fl_02	Bool	false		☒	☒	☒	
8	Level_Low_Tk_03	Bool	false		☒	☒	☒	
9	Level_High_Tk_03	Bool	false		☒	☒	☒	
10	PU_01	Bool	false		☒	☒	☒	
11	PU_02	Bool	false		☒	☒	☒	
12	Valve_Fl_2	Bool	false		☒	☒	☒	
13	Valve_Circulacion	Bool	false		☒	☒	☒	
14	Level_Fl_01	Word	16#0		☒	☒	☒	
15	Level_Fl_02	Int	0		☒	☒	☒	
16	Level_Filtro_02	Int	0		☒	☒	☒	
17	VFD_Pum_01_02	Word	16#0		☒	☒	☒	
18	Ultrasonico	Word	16#0		☒	☒	☒	
19	Level_Tank_02	Word	16#0		☒	☒	☒	
20	Level_Tank_03	Word	16#0		☒	☒	☒	
21	Temp_Filtro	Word	16#0		☒	☒	☒	
22	Temp_Plato	Word	16#0		☒	☒	☒	
23	Flujometro	Word	16#0		☒	☒	☒	
24	Posicion_Plato	Word	16#0		☒	☒	☒	
25	LDR_Vertical	Word	16#0		☒	☒	☒	
26	LDR_Horizontal	Word	16#0		☒	☒	☒	
27	FrecVariador	Word	16#0		☒	☒	☒	

Fig. 3. 34 Bloque de Datos para el almacenamiento del dato para lectura del HTML
(Fuente: PLC Siemens S7-1200)

Una vez declarado todas las variables a utilizar, ahora en el Ladder realizamos el movimiento del dato del que envia el arduino para que se almacene en la base de datos creada “PLCVariables”.

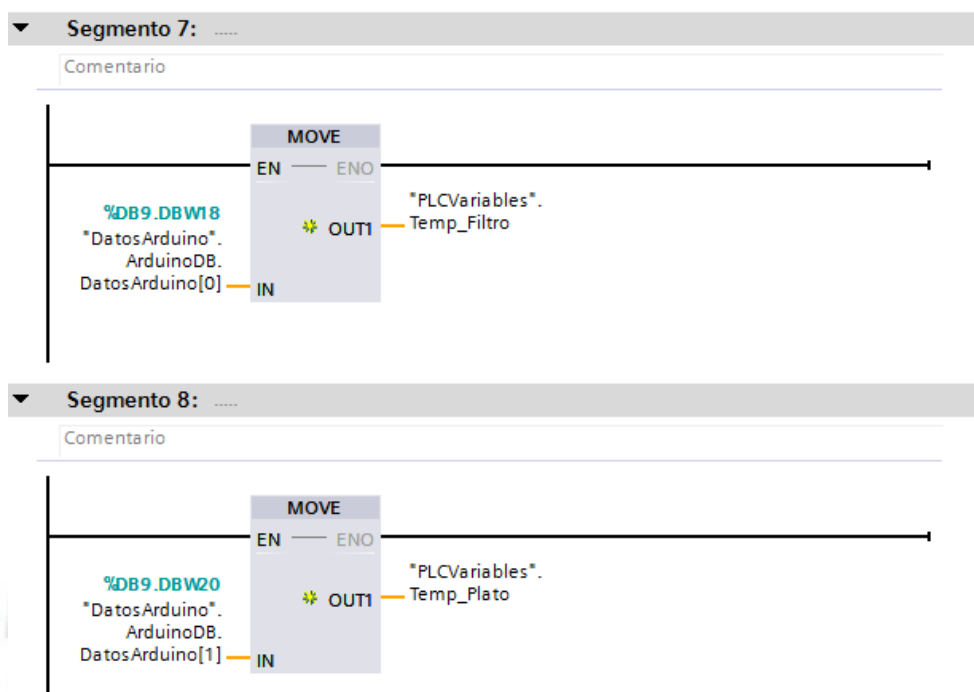


Fig. 3. 35 Instrucción de Move para el almacenamiento del dato
(Fuente: PLC Siemens S7-1200)

Así desarrollamos con todas las variables que deseamos leer del Arduino para que sea almacenada en el PLC, y así poder utilizarla para la programación, en este caso la que utilizaremos será:

- Temperatura del Filtro 02.
- Temperatura del Sistema.
- Flujometro

3.7.3. PRIMERA ETAPA: CONTROL DEL PRIMER FILTRADO DE LA PLANTA

Para la primera etapa la programación lo realizamos aplicando Grafcet para luego convertirlo a Ladder.

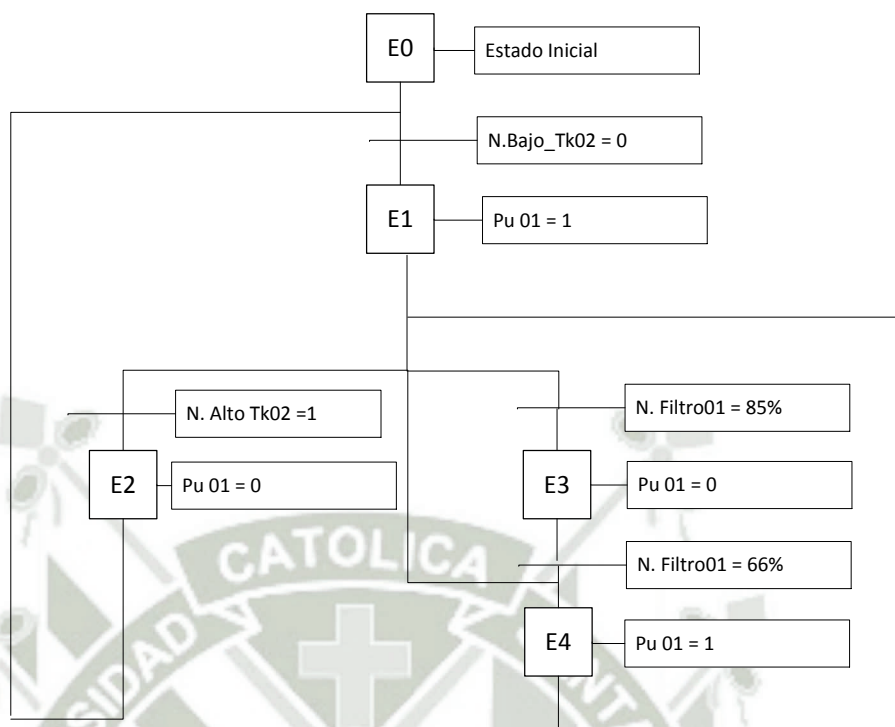
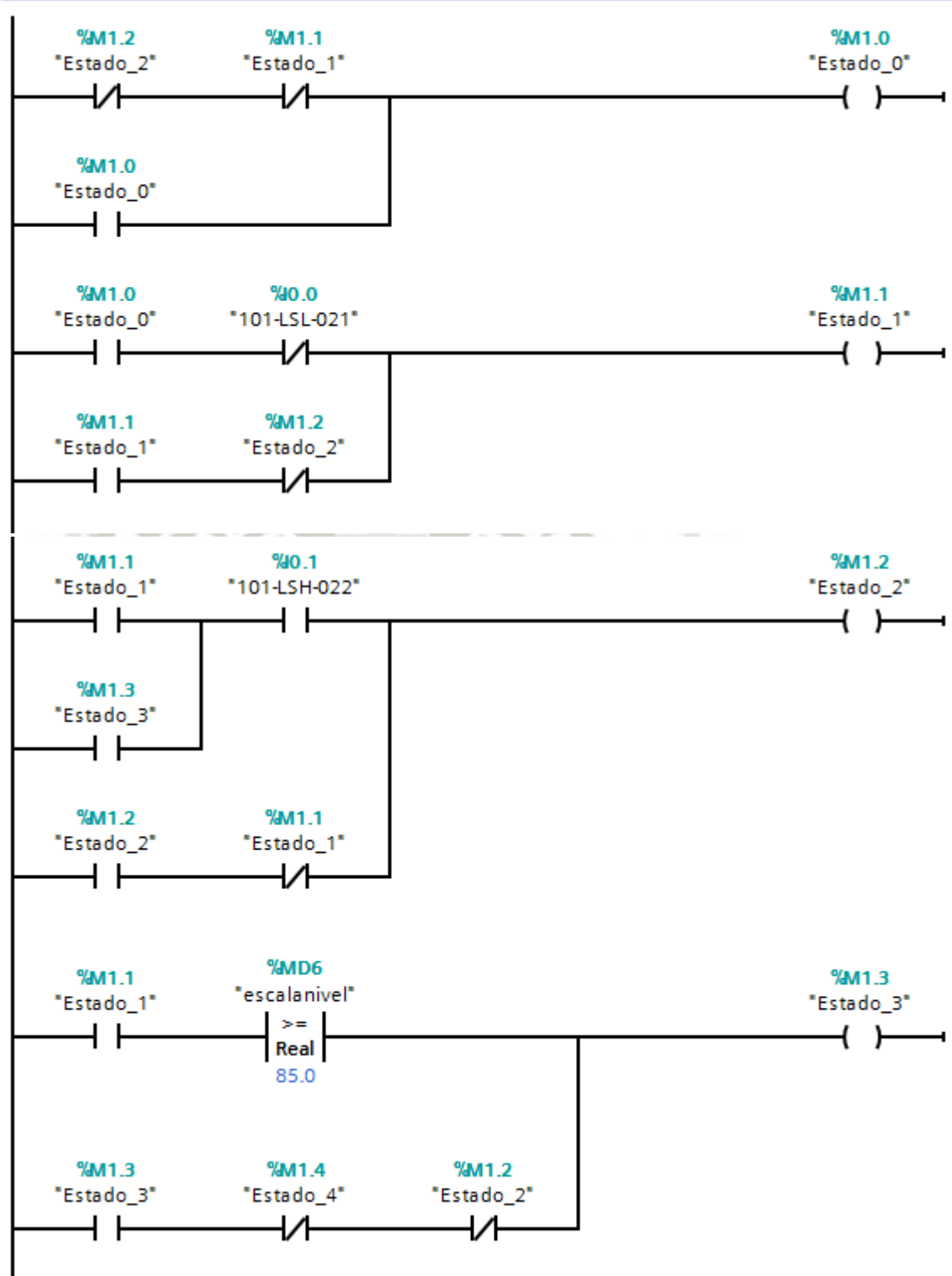
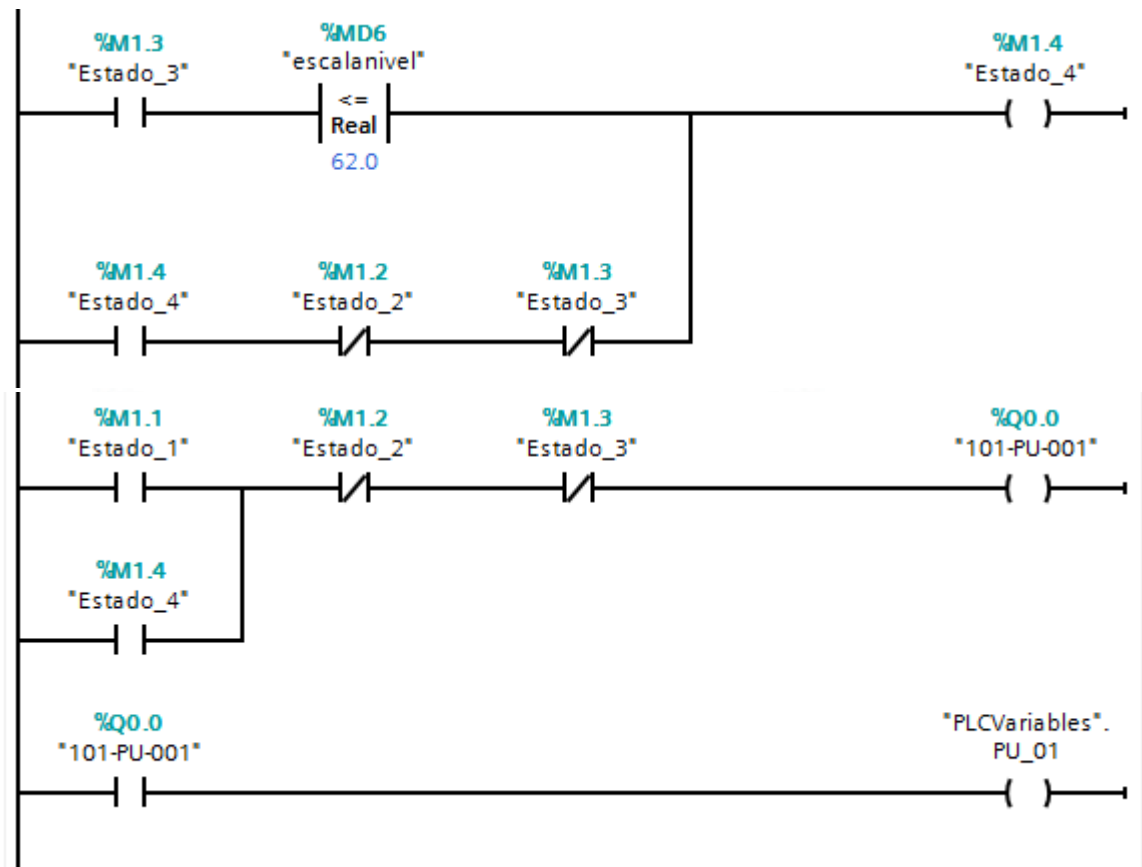


Fig. 3. 36 . Lógica en Grafcet para la etapa del Filtrado 01
(Fuente: Elaboración Propia)

Pasamos del Grafcet a Ladder de la siguiente manera:





3.7.4. SEGUNDA ETAPA: CONTROL FUZZY CON LENGUAJE STL PARA EL FILTRADO POR CONDENSACIÓN

Respecto al Control Fuzzy se implementó para la etapa del Filtro 02, en la cual se tiene como variables de entrada la temperatura en dicho filtro y los niveles correspondientes; que son Nivel Muy Bajo, Nivel Bajo, Nivel Alto y Nivel Muy Alto. Para la salida de nuestro controlador será la frecuencia del variador.

Para este proyecto emplearemos el Método de Mamdani respecto a la fuzificación y las inferencias, para la salida de nuestro controlador será por el Método del Sugeno.

Empezamos con la Fuzificación ingresando las variables lingüísticas para las entradas, para que luego pasemos a la etapa de las inferencias o reglas de control que serán procesados por el controlador.

Identificado las variables de entrada, procedemos a darle los rangos de trabajo:

Para la entrada de temperatura serán con los siguientes rangos y su función de pertenencia es trapezoidal:

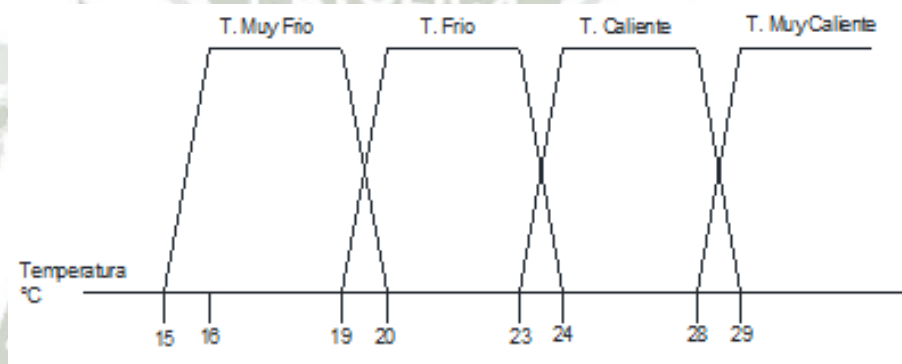


Fig. 3. 37 Entrada del Sistema - Rangos de Temperatura
(Fuente: Elaboración Propia)

Para la entrada de nuestro sensor de nivel son los siguientes rangos y su función de pertenencia es triangular.

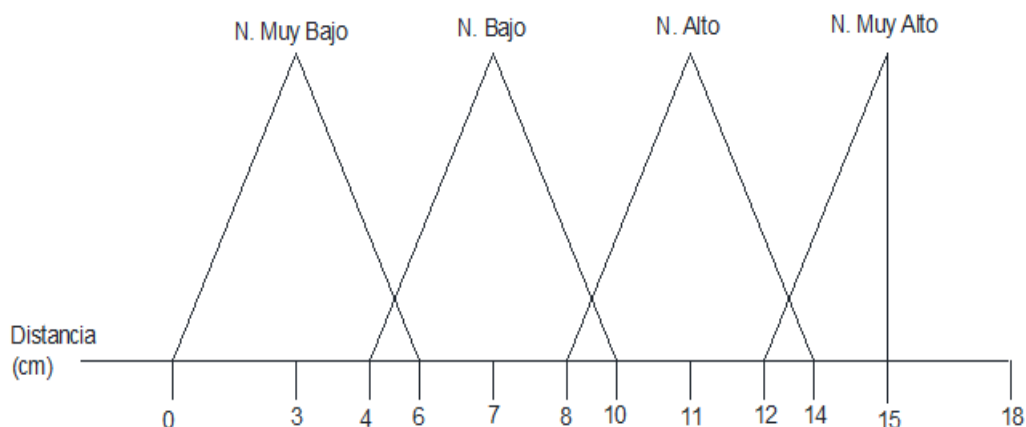


Fig. 3. 38 Entrada del Sistema - Rangos de Nivel
(Fuente: Elaboración Propia)

Para la salida de nuestro variador está relacionado con la distancia a medida que se vaya activando los sensores de nivel, de la misma forma la función de nuestro controlador es el siguiente.

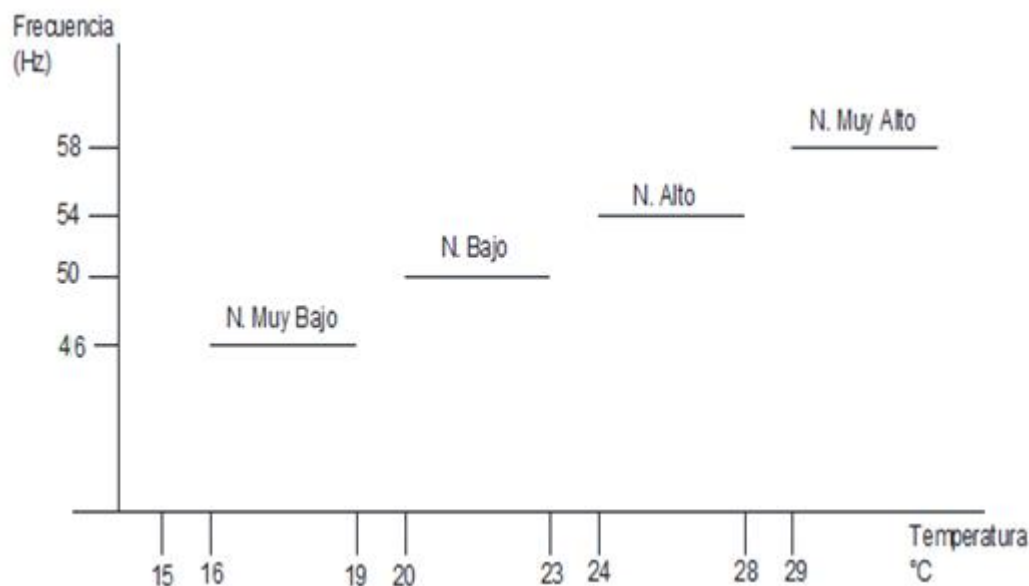


Fig. 3. 39 Salida del Sistema - Rangos de la Frecuencia
(Fuente: Elaboración Propia)

Una vez obtenido las variables lingüísticas pasamos a la inferencia o reglas que tendrá nuestro controlador; las reglas lo desarrollamos en el lenguaje de programación STL del PLC S7-1200. Para ello primero pasamos las reglas a un cuadro para poder entender lo que queremos realizar.

Nivel \ Temp	16<t<19	20<t<23	24<t<27	28<t<100
	Muy Frio	Frio	Caliente	Muy Caliente
N. Muy Bajo	46	46	46	46
N. Bajo	48	50	50	50
N. Alto	48	48	54	54
N. Muy Alto	48	48	48	58

Tabla 3. 10 Reglas para el Control Fuzzy
(Fuente: Elaboración Propia)

Como se puede observar en el cuadro, las frecuencias cambian a medida que pasan por cada nivel. Iniciar con una frecuencia baja se tomó por lo que cuando la temperatura aún es muy baja no necesitamos una frecuencia muy alta para que vaya llenando el filtro 02 ya que le baja la temperatura al filtro, esta lógica también se consideró ya que si iniciamos el arranque de la planta cuando la temperatura en el filtro 02 se encuentra mayor de los 28°C, por lo cual si mandamos a una frecuencia muy alta, el filtro presentara un cambio de temperatura muy brusca, por lo que se demorara en el proceso.

Lo que se encuentra resaltado en verde, es una frecuencia de 48 Hz es para que cuando haya cumplido el respectivo nivel a la temperatura indicada; entonces el cambio de frecuencia se realiza para que trabaje en la recirculación del agua para la etapa de enfriamiento y a la vez si la Bomba 01 aún se encuentra trabajando llenando agua al Tanque 02 este trabaje con la frecuencia también de 48 Hz.

Ahora pasamos al Controlador Difuso donde realizaremos las reglas de control, pero antes debemos de considerar que nuestro proceso tiene 2

entradas y 4 salidas, las reglas de control deberían de cumplir con la siguiente ecuación.

$$\#Reglas = (Salidas)^{Entradas}$$

Donde:

$$\#Reglas = (4)^2$$

$$\#Reglas = 16$$

Pero en este caso solo utilizaremos 10 reglas para poder controlar nuestro proceso. Para fines didácticos lo mostraremos en Matlab.

Ingresamos la entrada de la temperatura

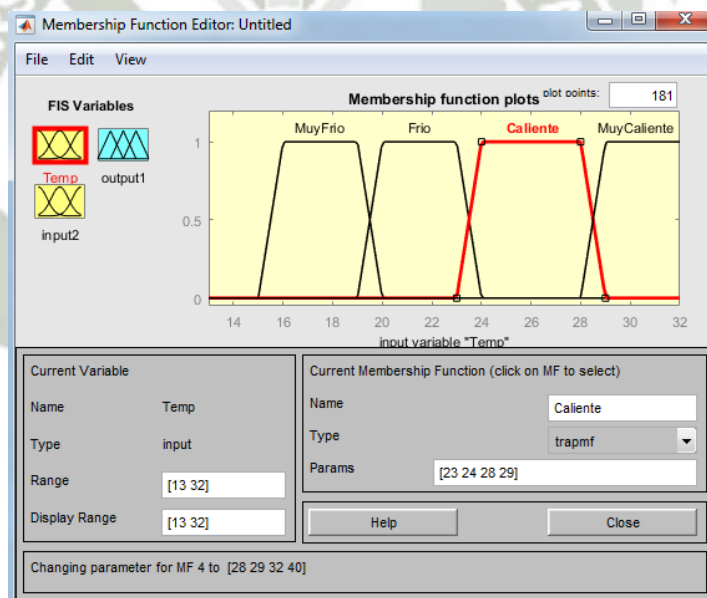


Fig. 3. 40 Entrada de Temperatura en fuzzyLogicDesigner
(Fuente: Matlab)

Ingresamos la entrada del nivel del Filtro 02

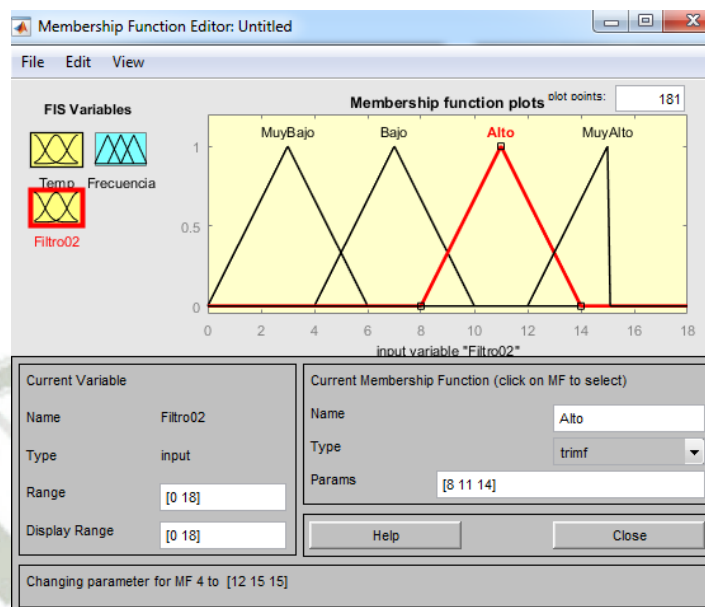


Fig. 3. 41 Entrada de Nivel Filtro 02 en fuzzyLogicDesigner
(Fuente: Matlab)

Ingresamos la salida de nuestro variador

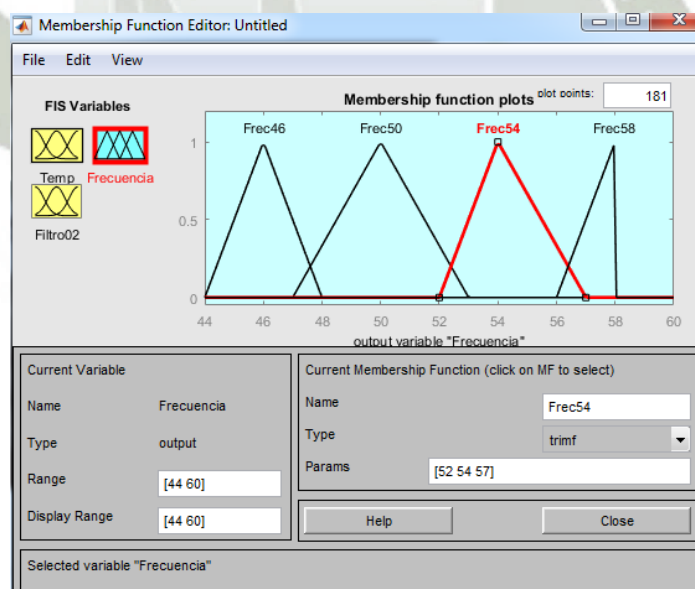


Fig. 3. 42 Salida del Variador en fuzzyLogicDesigner
(Fuente: Matlab)

Ingresamos las 10 reglas que me permitirán que trabaje mi controlador

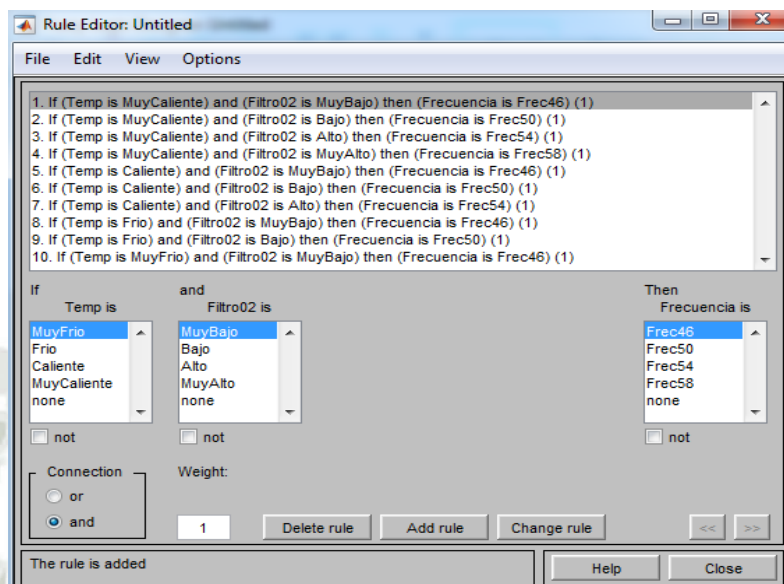


Fig. 3. 43 Reglas de Control en fuzzyLogicDesigner
(Fuente: Matlab)

Y como resultado aplicando el método de Mamdani.

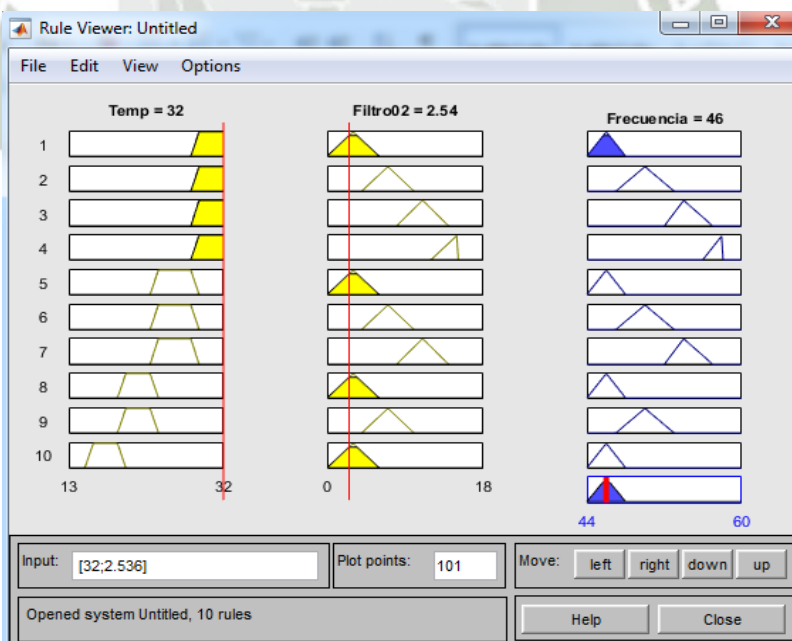


Fig. 3. 44 Resultado a una Frec 46 en fuzzyLogicDesigner
(Fuente: Matlab)

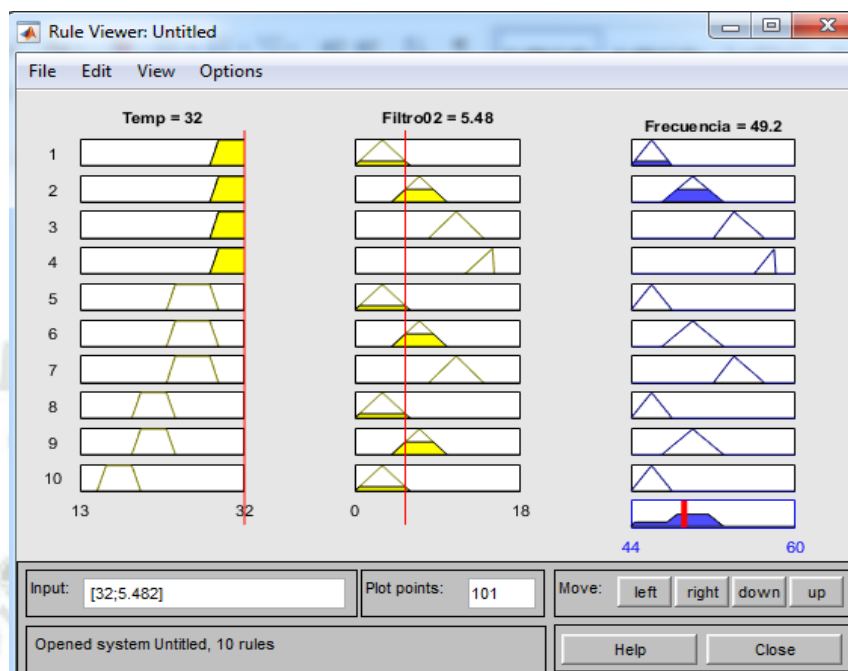


Fig. 3. 45 Resultado a una Frec 50 en fuzzyLogicDesigner
(Fuente: Matlab)

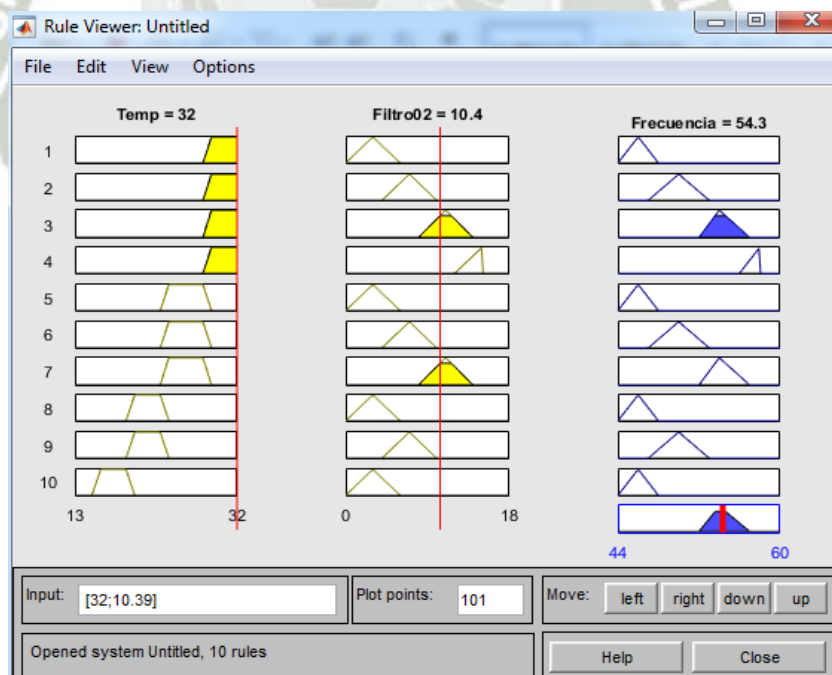


Fig. 3. 46 Resultado a una Frec 54 en fuzzyLogicDesigner
(Fuente: Matlab)

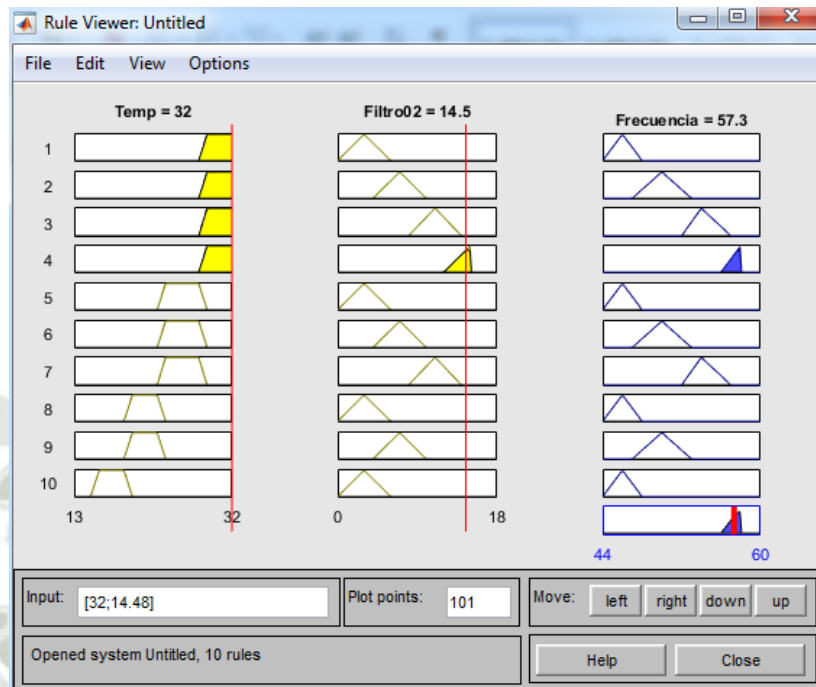


Fig. 3. 47 Resultado a una Frec 58 en fuzzyLogicDesigner
(Fuente: Matlab)

Ahora pasamos estas reglas de control al lenguaje de programación STL del PLC S7-1200. Para su programación constara en (IF – THEN). Debemos de considerar que solo para el lenguaje de programación STL se debe de considerar también el primer proceso del Filtro 01 y del Tanque 03 que es mi última etapa.

// REGLAS DE CONTROL FUZZY PARA EL FILTRO 02

///// Filtro 02

// %I0.2 %I0.3 %I0.4 %I0.5 ESTADO

```
// 0 0 0 0 Vacío "PLCVariables".Level_FI_02= 0
// 0 0 0 1 Muy Bajo "PLCVariables".Level_FI_02= 1
// 0 0 1 1 Bajo "PLCVariables".Level_FI_02= 2
// 0 1 1 1 Alto "PLCVariables".Level_FI_02= 3
// 1 1 1 1 Muy Alto "PLCVariables".Level_FI_02= 4
```

```
//No hay agua
////
"101-PU-001":="PLCVariables".PU_01;////enviamos la variable Bomba 01 a
la salida física Q0
"102-PU-002":="PLCVariables".PU_02; ////enviamos la variable Bomba 02 a
la salida física Q1
"102-XS-001":="PLCVariables".Valve_FI_2; ////enviamos la variable válvula
01 a la salida física Q2
"102-XS-002":="PLCVariables".Valve_Circulacion; ////enviamos la variable
válvula 02 a la salida física Q3
"101-VFD-001":="PLCVariables".VFD_Pum_01_02; ////enviamos la variable
FrecVariador a la salida física QW98
////////// INICIO DE PROGRAMA
IF ("PLCVariables".Level_Tank_02 > 0) THEN /// Nivel bajo tank2 esta
activo
    IF (("PLCVariables".Level_Tank_03 = 2) OR ("PLCVariables".Temp_Filtro
<= 15)) THEN ///temperatura muy baja no hay sol
        "PLCVariables".PU_02 := 0; ////apagamos bomba 2
        "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
    ELSE //Condiciones para sistema activo se cumplen
        ////////////INICIO de condiciones ACTIVA
        IF "PLCVariables".Level_FI_02 = 0 THEN /// Nivel FILTRO 02 esta vacio
            IF ((16 <= "PLCVariables".Temp_Filtro) AND
("PLCVariables".Temp_Filtro <= 150)) THEN///temperature mayor a 15
                "PLCVariables".Valve_FI_2 := 1; /// abrimos válvula de filtro 02
                "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
                "PLCVariables".FrecVariador := 46;///según tabla
```

```

        "PLCVariables".PU_02 := 1; ///encendemos la bomba 2
    END_IF;
END_IF;
IF "PLCVariables".Level_FI_02 = 1 THEN ///// Nivel FILTRO 02 esta
Muy Bajo
    IF ("PLCVariables".Temp_Filtro <= 15) THEN
        "PLCVariables".Valve_FI_2 := 0; /// abrimos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
        "PLCVariables".PU_02 := 0;
        "PLCVariables".FrecVariador := 0;////
    END_IF;
    IF ((16 <= "PLCVariables".Temp_Filtro) AND
("PLCVariables".Temp_Filtro <= 19)) THEN ///temperature 16 a 19

"PLCVariables".FrecVariador := 48; ///TABLA
        "PLCVariables".PU_02 := 1;
        "PLCVariables".Valve_FI_2 := 0; /// abrimos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 1; ///cerramos válvula de
recirculación

    END_IF;
    IF ((20 <= "PLCVariables".Temp_Filtro) AND
("PLCVariables".Temp_Filtro <= 23) AND ("OK_T20_100" = TRUE)) THEN
///temperature 20 a 23
        "PLCVariables".Valve_FI_2 := 1; /// abrimos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
        "PLCVariables".PU_02 := 1;
"PLCVariables".FrecVariador := 50;///según tabla

```

```

END_IF;
IF      ((24      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 27) AND ("OK_T24_100" = TRUE)) THEN
    "PLCVariables".Valve_FI_2 := 1; /// abrimos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
    "PLCVariables".PU_02 := 1;
    "PLCVariables".FrecVariador := 50; /////según tabla

END_IF;
IF      ((28      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 100) AND ("OK_T28_100" = TRUE)) THEN
    "PLCVariables".Valve_FI_2 := 1; /// abrimos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
    "PLCVariables".PU_02 := 1;
    "PLCVariables".FrecVariador := 50;//// ;////según tabla

END_IF;
END_IF;
IF "PLCVariables".Level_FI_02 = 2 THEN /// Bajo
    IF ("PLCVariables".Temp_Filtro <= 15) THEN
        "PLCVariables".Valve_FI_2 := 0; /// abrimos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
        "PLCVariables".PU_02 := 0;
        "PLCVariables".FrecVariador := 48;//// ;////según tabla
    END_IF;
    IF      ((16      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 19)) THEN

```

```

IF ("OK_T20_100" = FALSE) THEN
    "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 1; ///abrimos válvula de
recirculación
    "PLCVariables".PU_02 := 1;
    "PLCVariables".FrecVariador := 48; ///según tabla
END_IF;
END_IF;
IF      ((20      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 23)) THEN
    "PLCVariables".FrecVariador := 48;/// ///según tabla
    "PLCVariables".PU_02 := 1;///// activa bomba 2
    "PLCVariables".Valve_FI_2 := 0; /// abrimos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 1; ///cerramos válvula de
recirculación
    END_IF;
    IF      ((24      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 27) AND ("OK_T24_100" = TRUE)) THEN
        "PLCVariables".Valve_FI_2 := 1; /// abrimos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
        "PLCVariables".PU_02 := 1; ///// activa bomba 2
        "PLCVariables".FrecVariador := 54;/// ///según tabla /
    END_IF;
    IF      ((28      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 100) AND ("OK_T28_100" = TRUE)) THEN
        "PLCVariables".Valve_FI_2 := 1; /// abrimos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
        "PLCVariables".PU_02 := 1; ///// activa bomba 2

```

```

"PLCVariables".FrecVariador := 54;//// ////según tabla
END_IF;
END_IF;
IF "PLCVariables".Level_FI_02 = 3 THEN //Alto
    IF ("PLCVariables".Temp_Filtro <= 15) THEN
        "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
        "PLCVariables".PU_02 := 0; ///// apaga bomba 2
"PLCVariables".FrecVariador := 48;//// ////según tabla
END_IF;
IF      ((16      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 19)) THEN
"PLCVariables".FrecVariador := 48;////según Tabla
"PLCVariables".PU_02 := 1; ///// activa bomba 2
"PLCVariables".Valve_FI_2 := 0; /// abrimos válvula de filtro 02
"PLCVariables".Valve_Circulacion := 1; ///cerramos válvula de
recirculación

END_IF;
IF      ((20      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 23)) THEN
"PLCVariables".FrecVariador := 48;//// ////según tabla
"PLCVariables".PU_02 := 1; ///// activa bomba 2
"PLCVariables".Valve_FI_2 := 0; /// abrimos válvula de filtro 02
"PLCVariables".Valve_Circulacion := 1; ///cerramos válvula de
recirculación

END_IF;
IF      ((24      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 27)) THEN

```

```

IF ("OK_T24_100" = FALSE) THEN
    "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 1; ///abrimos válvula de
recirculación
    "PLCVariables".PU_02 := 1;
    "PLCVariables".FrecVariador := 48;/// ///según tabla
END_IF;
END_IF;
IF      ((28      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 100) AND ("OK_T28_100" = TRUE)) THEN
    "PLCVariables".Valve_FI_2 := 1; /// abrimos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
    "PLCVariables".PU_02 := 1;
    "PLCVariables".FrecVariador := 58;/// ///según tabla
END_IF;
END_IF;
IF "PLCVariables".Level_FI_02 = 4 THEN //Muy Alto
    IF ("PLCVariables".Temp_Filtro <= 15) THEN
        "PLCVariables".Valve_FI_2 := 0; /// abrimos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 0; ///cerramos válvula de
recirculación
        "PLCVariables".PU_02 := 0; ///// apaga  bomba 2
        "PLCVariables".FrecVariador := 48;/// ///según tabla
    END_IF;
    IF      ((16      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 19)) THEN
        "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 1; ///abrimos válvula de
recirculación

```

```

"PLCVariables".PU_02 := 1; ///// activa bomba 2
"PLCVariables".FrecVariador := 48;/// ///según tabla
END_IF;
IF      ((20      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 23)) THEN
    "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 1; ///abrimos válvula de
recirculación
    "PLCVariables".PU_02 := 1; ///// activa bomba 2
    "PLCVariables".FrecVariador := 48;/// ///según tabla
END_IF;
IF      ((24      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 27)) THEN
    "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 1; ///abrimos válvula de
recirculación
    "PLCVariables".PU_02 := 1; ///// activa bomba 2
    "PLCVariables".FrecVariador := 48;/// ///según tabla
END_IF;
IF      ((28      <=      "PLCVariables".Temp_Filtro)      AND
("PLCVariables".Temp_Filtro <= 100)) THEN
    IF ("OK_T28_100" = FALSE) THEN
        "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
        "PLCVariables".Valve_Circulacion := 1; ///abrimos válvula de
recirculación
        "PLCVariables".PU_02 := 1; ///// activa bomba 2
        "PLCVariables".FrecVariador := 48;/// ///según tabla
    END_IF;
END_IF;

```

END_IF;

```
"PLCVariables".PREFrecVariador :=
"PLCVariables".FrecVariador;///AQUI ENVIAMOS LA FREC AL VARIADOR
END_IF;
ELSE ///en caso que no haya nada de agua en el tank 2 y la temperatura
de filtro sea menor a 15
    "PLCVariables".Valve_FI_2 := 0; /// cerramos válvula de filtro 02
    "PLCVariables".Valve_Circulacion := 0;
    "PLCVariables".PU_02 := 0;
    "PLCVariables".PREFrecVariador := "PLCVariables".FrecVariador;
END_IF;
```

La salida se realizó por el Método de Takagi - Sugeno donde el valor que arroja de las reglas que se han activado en un momento determinado ya son valores numéricos por lo que no necesitan una etapa de defuzicación.

Las reglas de la base de conocimiento de un sistema Sugeno son diferentes a las de los sistemas Mandani, por lo que la salida no es una regla sino una función.

La salida es una combinación lineal de funciones con transiciones suaves. El propósito de este controlador es de suavizar las transiciones de una función a otra función.

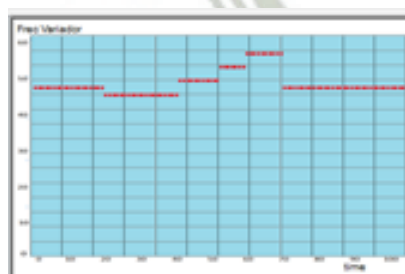


Fig. 3. 48 Método de Sugeno
(Fuente: Elaboración Propia)

3.7.5. SEGUNDA ETAPA: CONTROL CON GRAFCET Y LADDER PARA EL FILTRADO POR CONDENSACIÓN

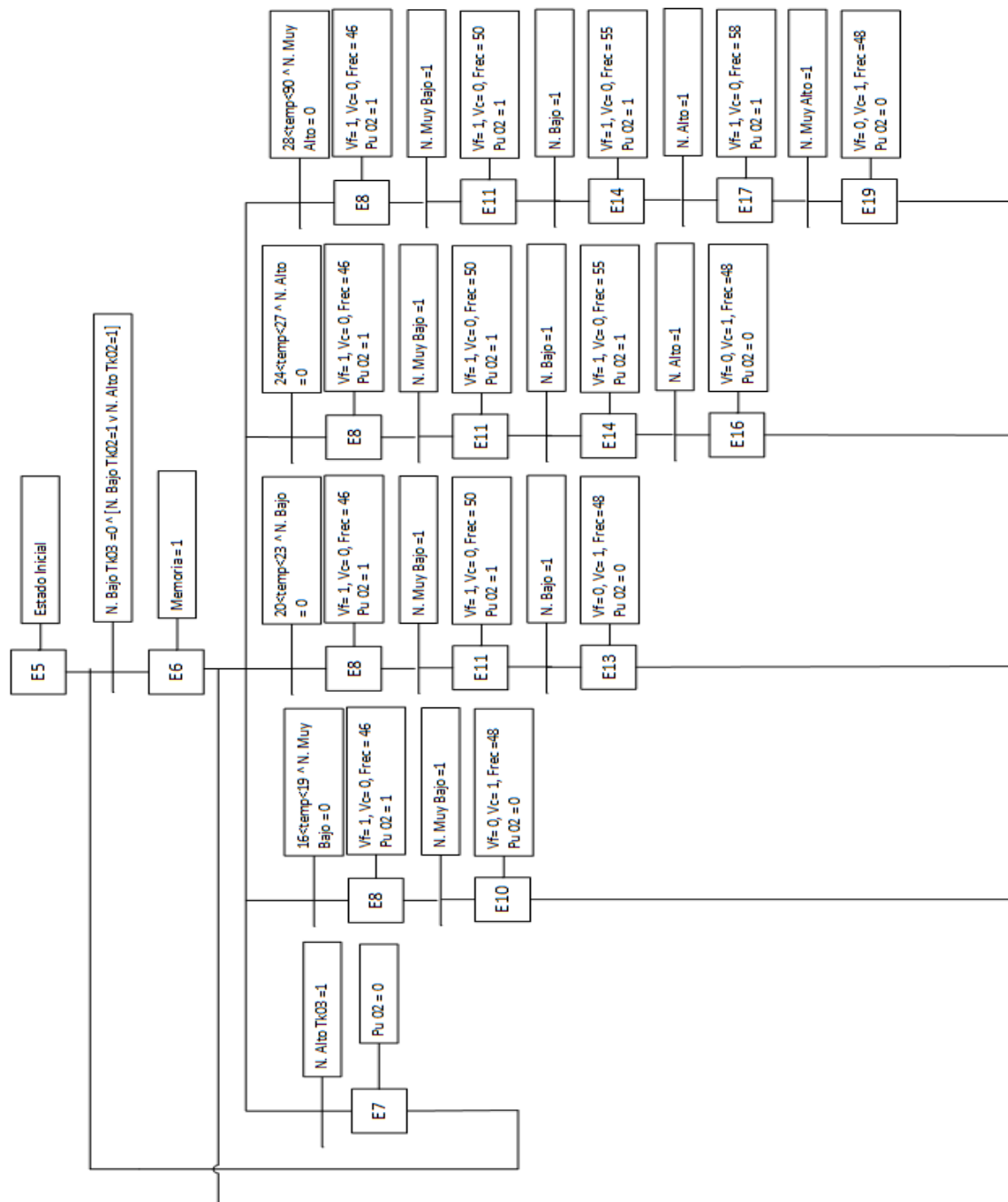
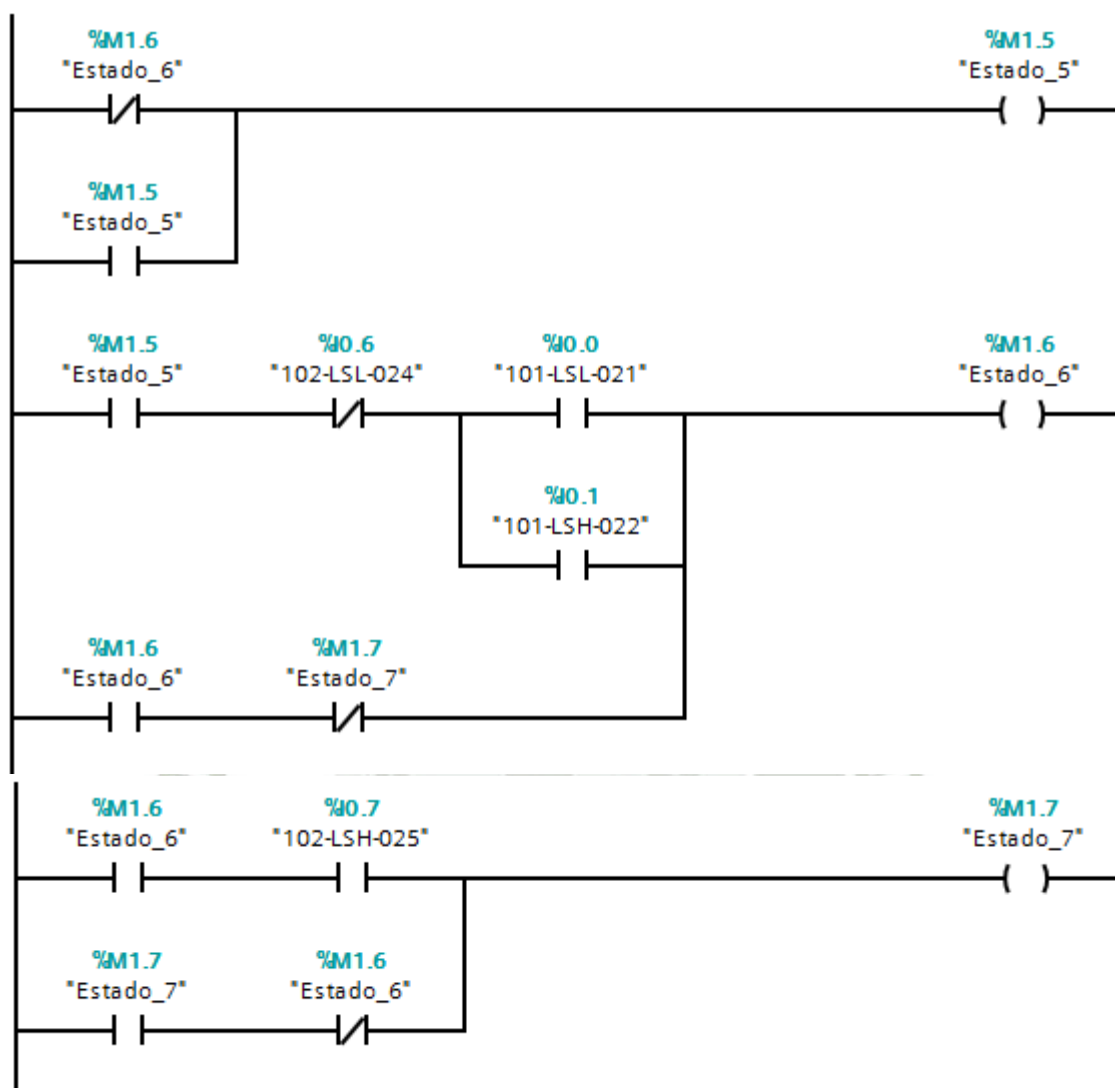
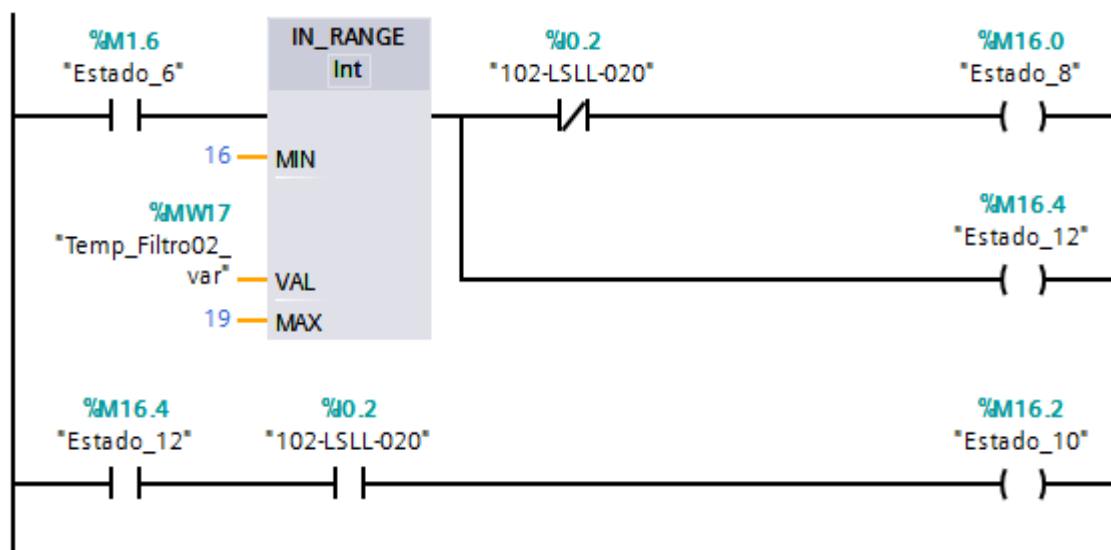


Fig. 3. 49 Lógica en Grafcet para la Etapa de Condensación
(Fuente: Elaboración Propia)

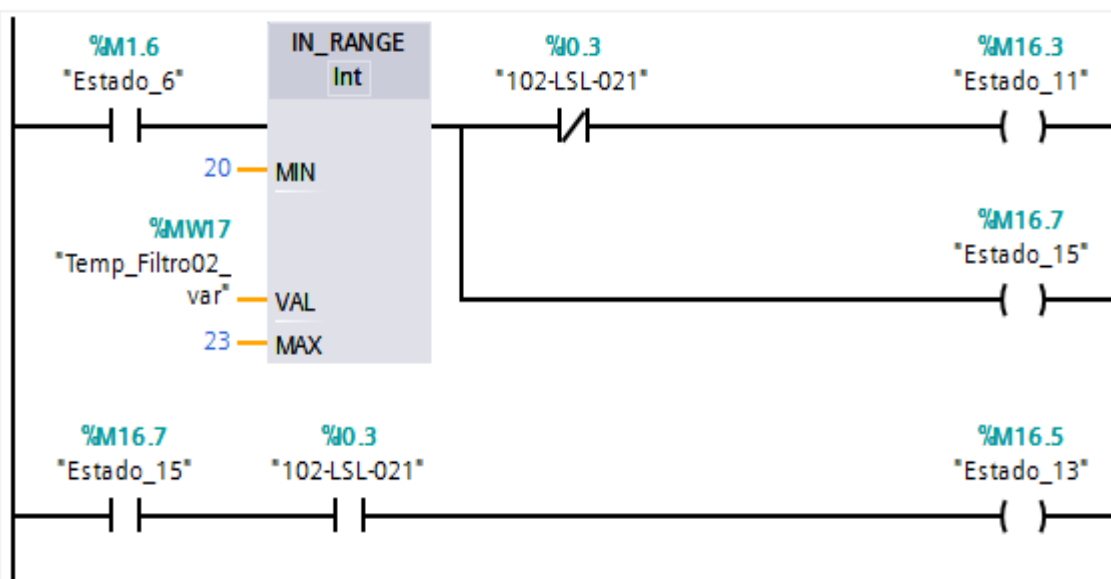
Ahora pasamos a Ladder



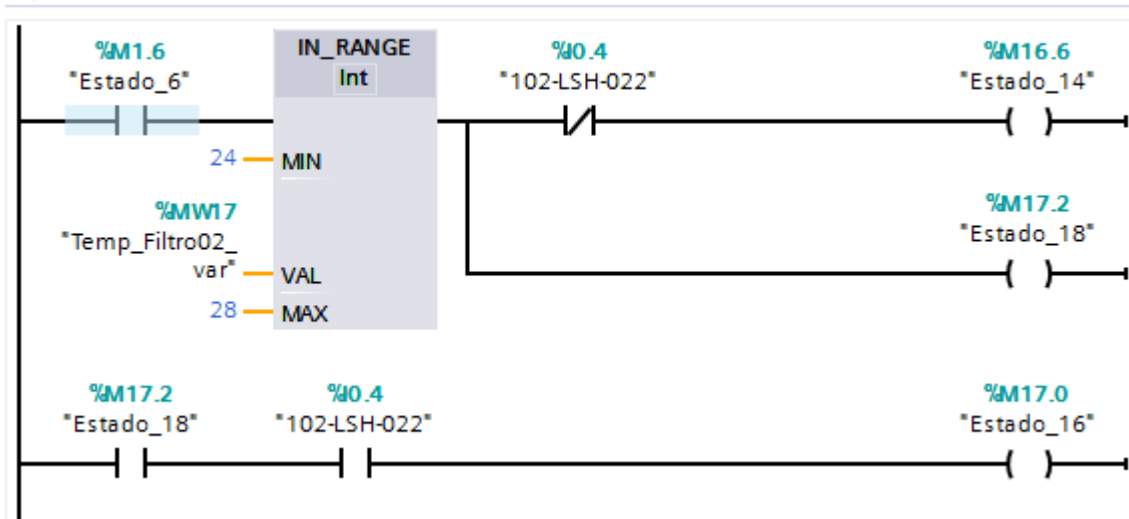
Logica para el Filtro 02 del nivel muy bajo con un rango de temperatura de $16 < t < 19$



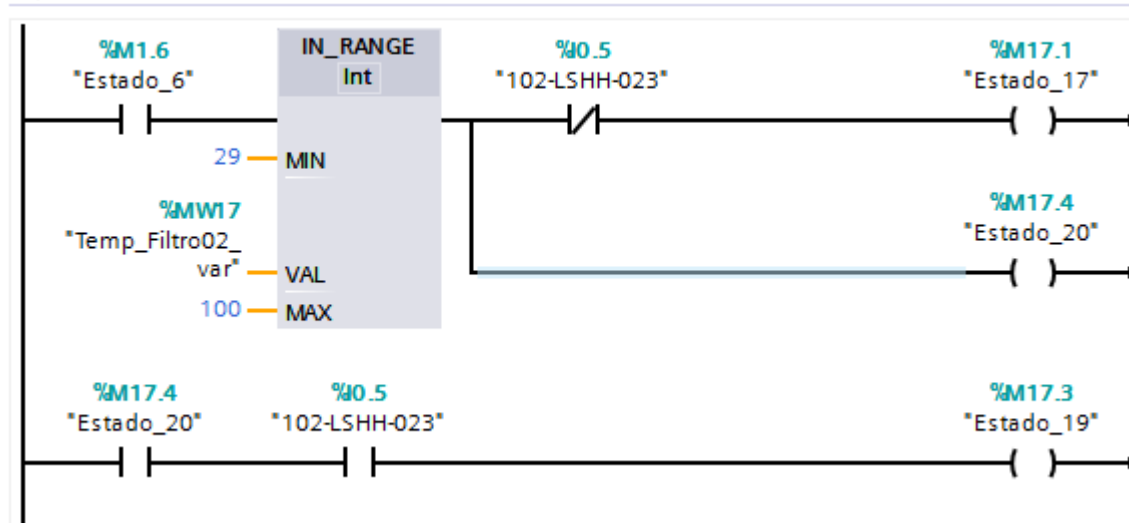
Logica para el Filtro 02 del nivel bajo con un rango de temperatura de $20 < t < 23$



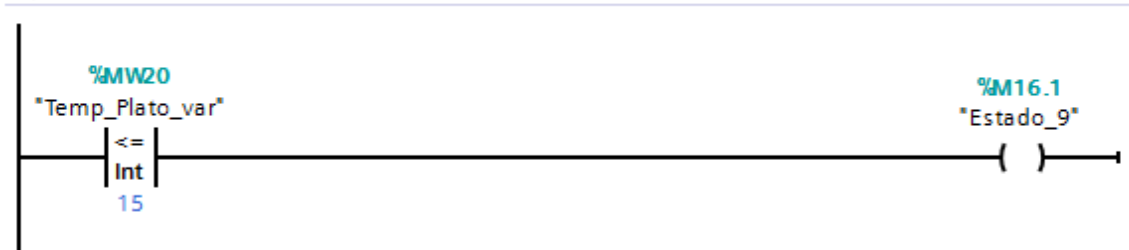
Logica para el Filtro 02 del nivel alto con un rango de temperatura de $24 < t < 28$



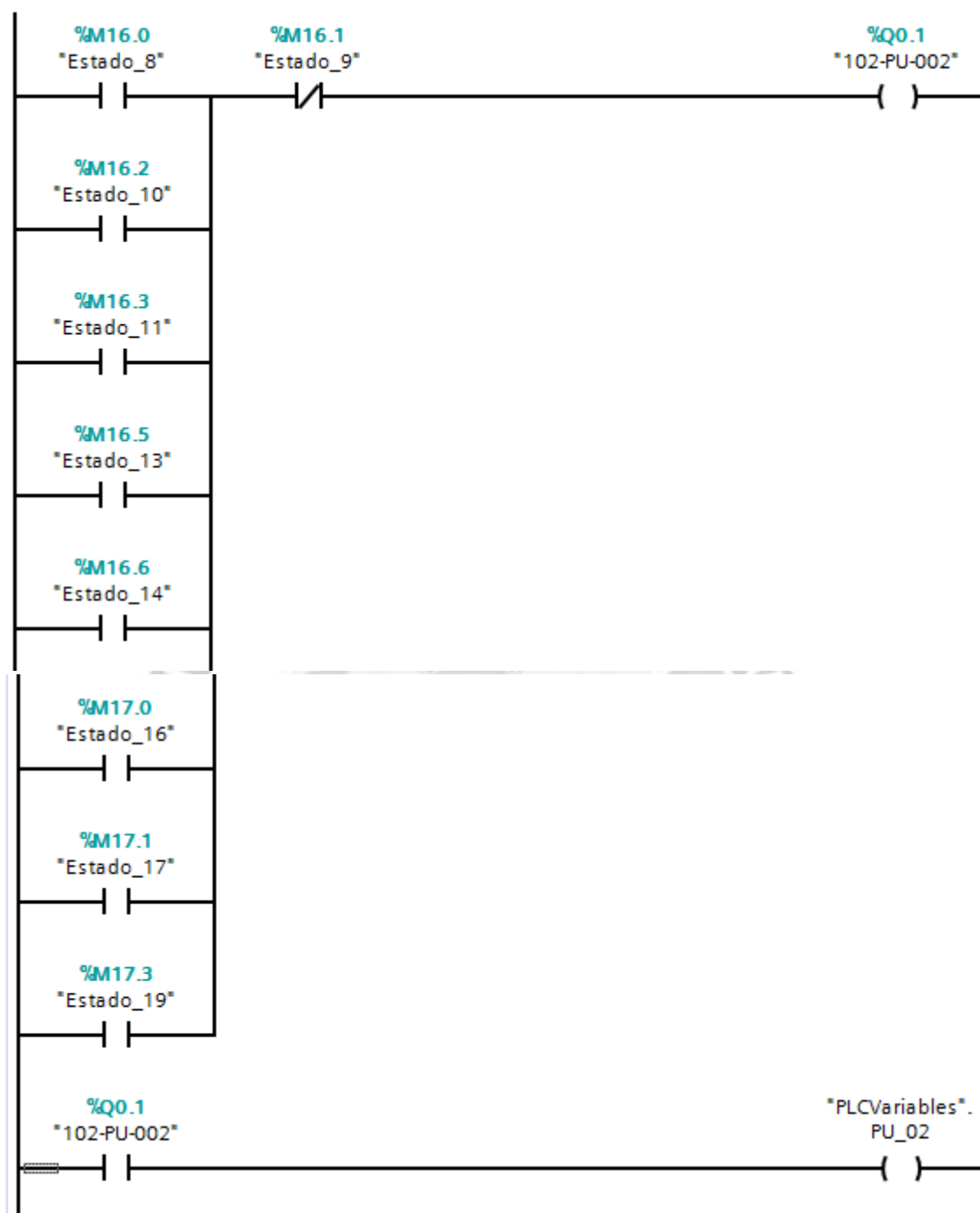
Logica para el Filtro 02 del nivel muy alto con un rango de temperatura de $29 < t < 100$



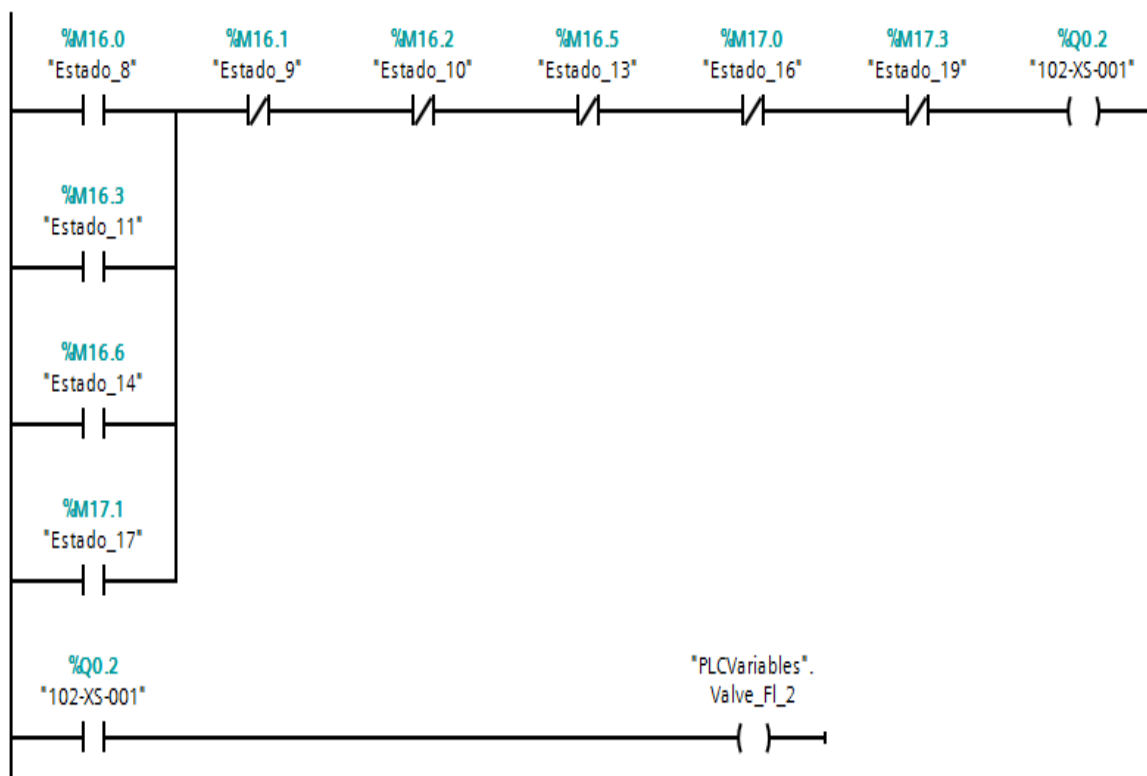
Temperatura menor que 15°C se aparara el sistema



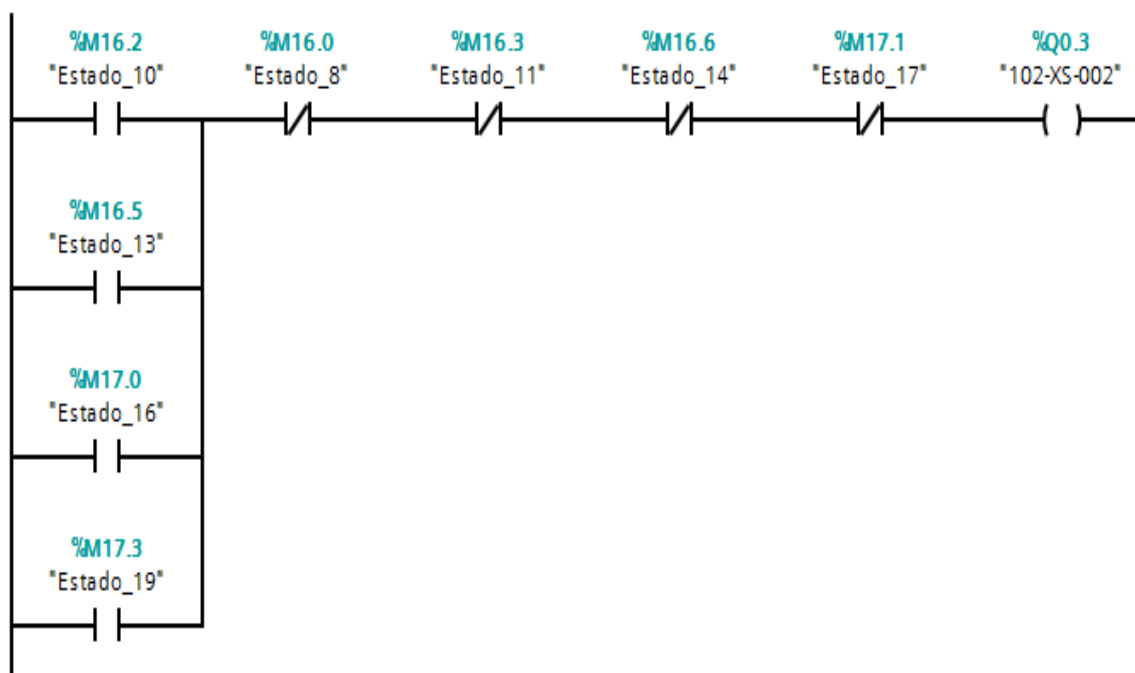
Accionamiento y desactivación de la Bomba 02



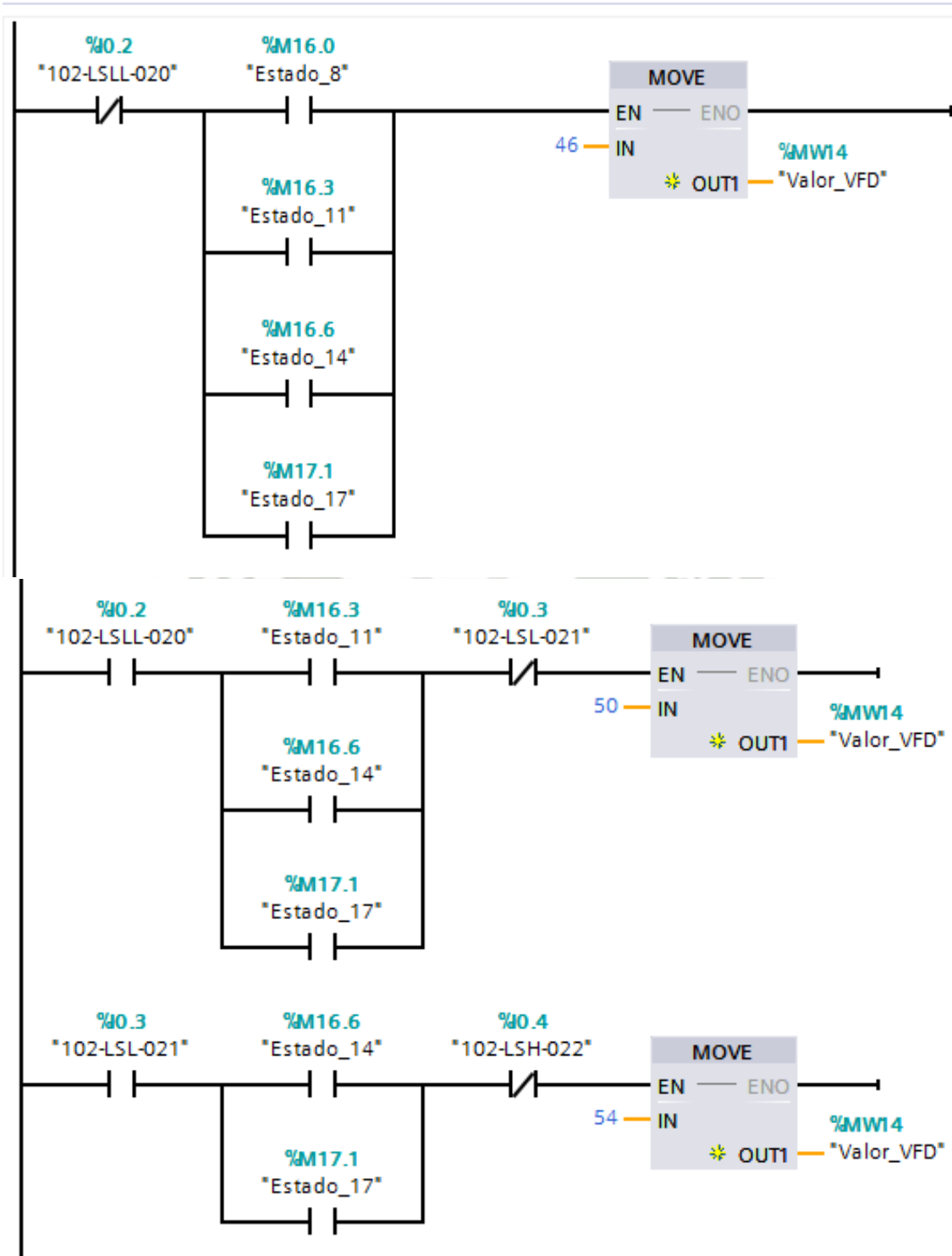
Activación y Desactivación Valvula de Filtro

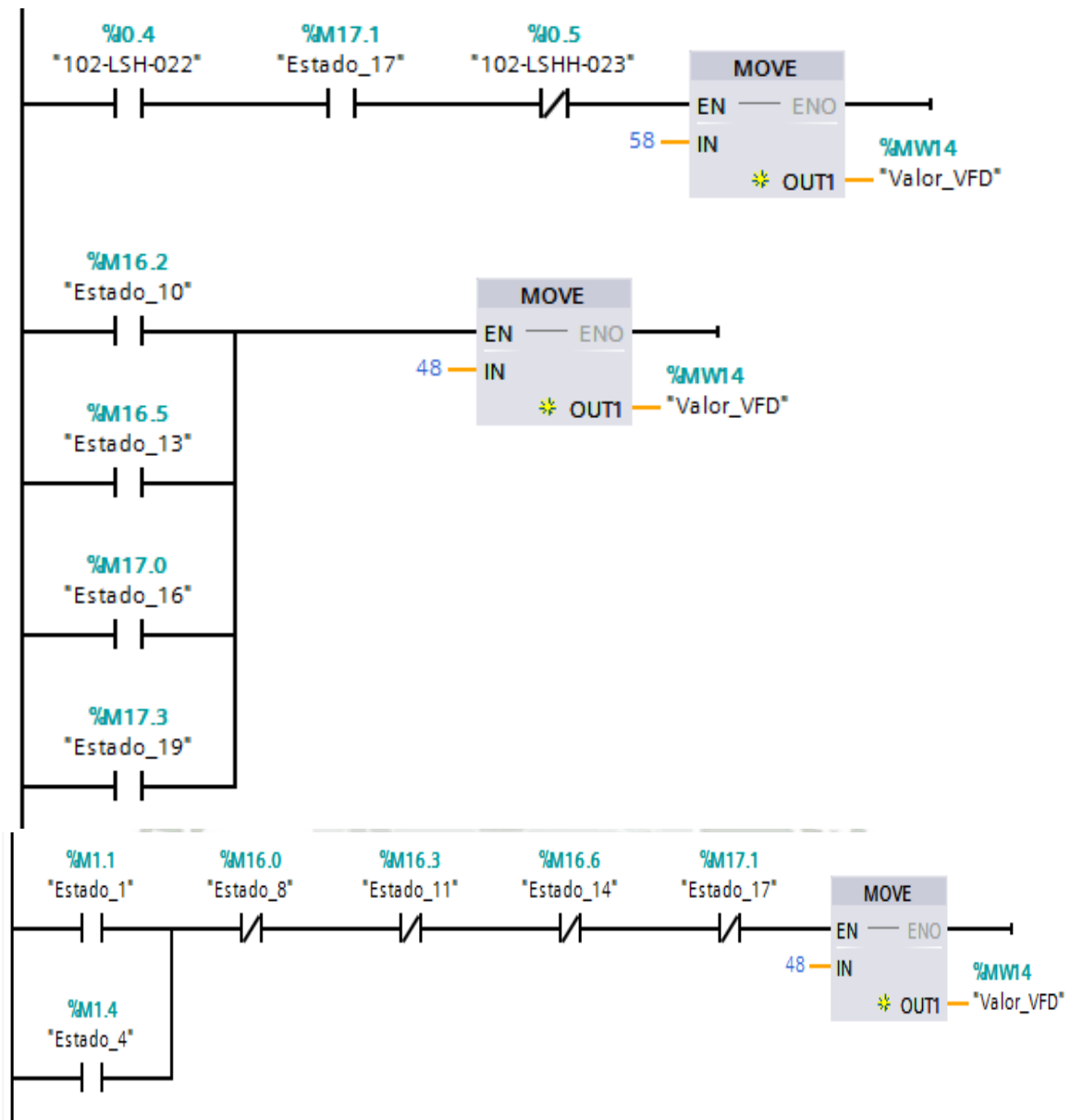


Activación y Desactivación Valvula de Circulación

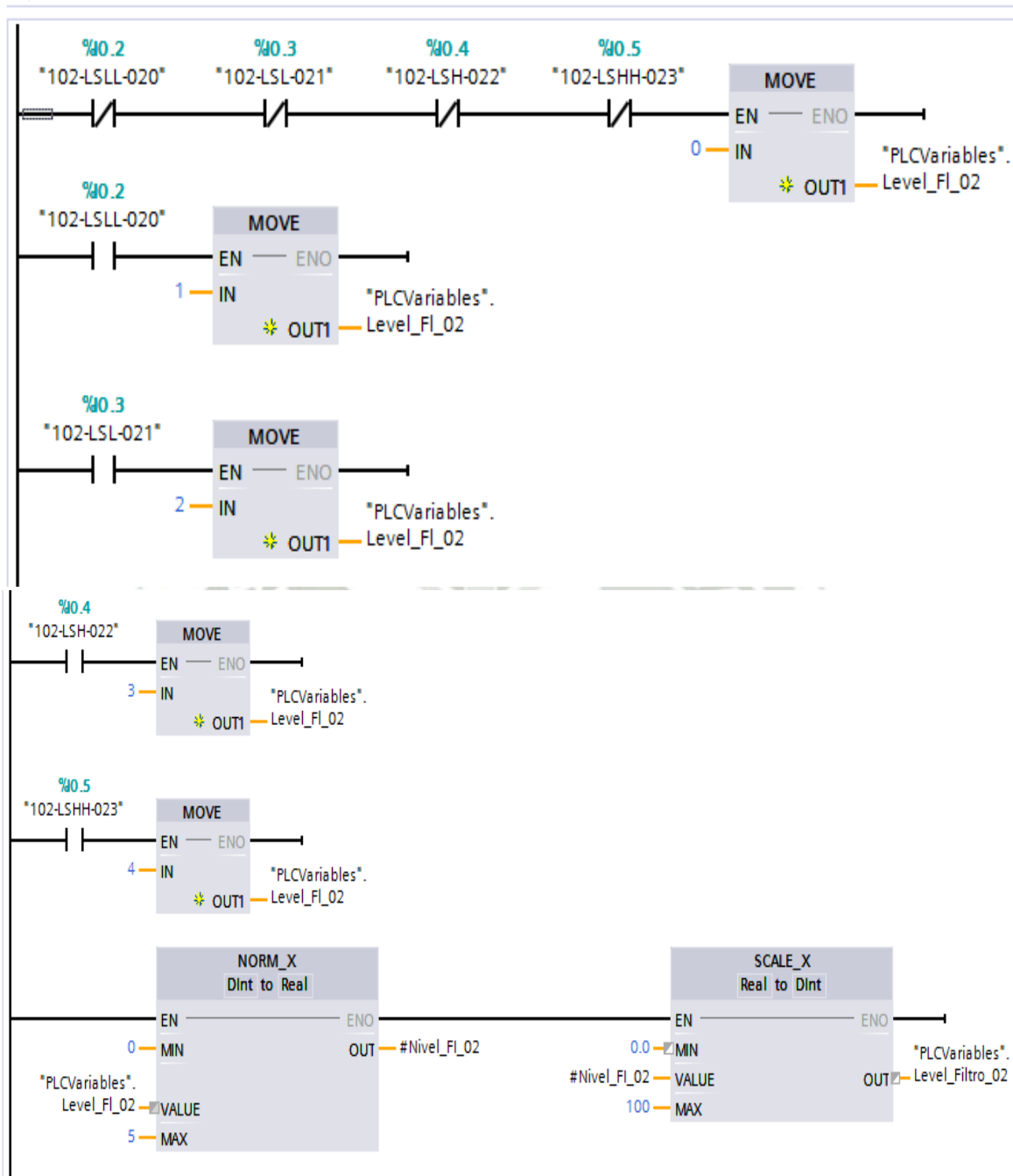


Logica para el Variador

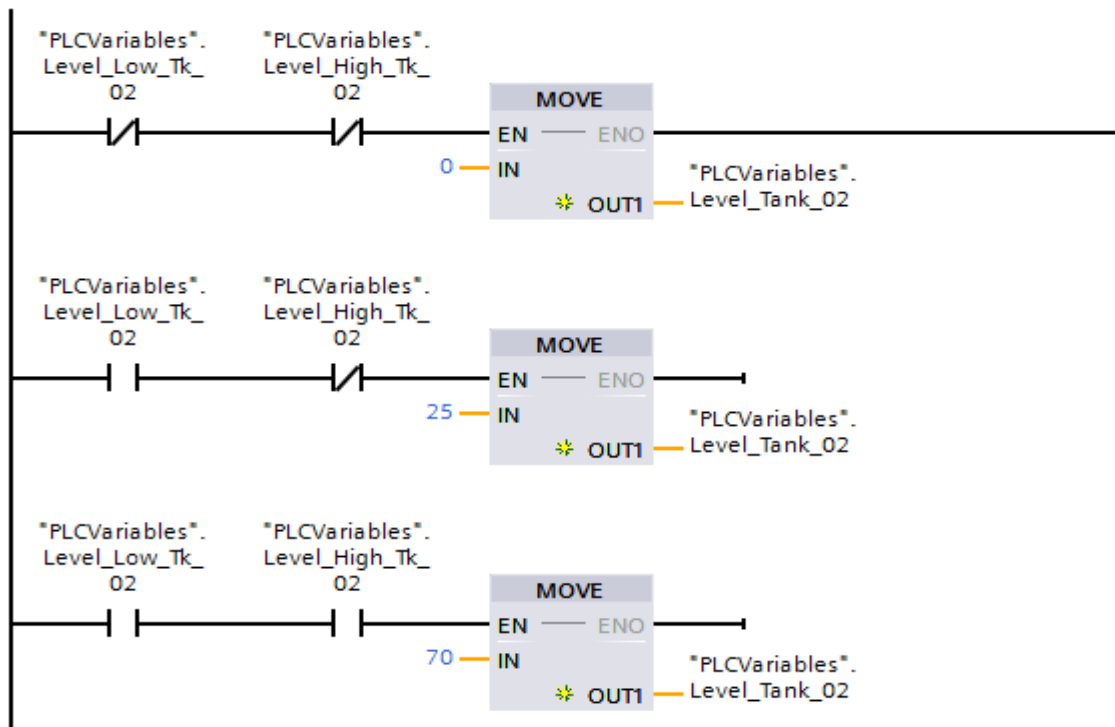




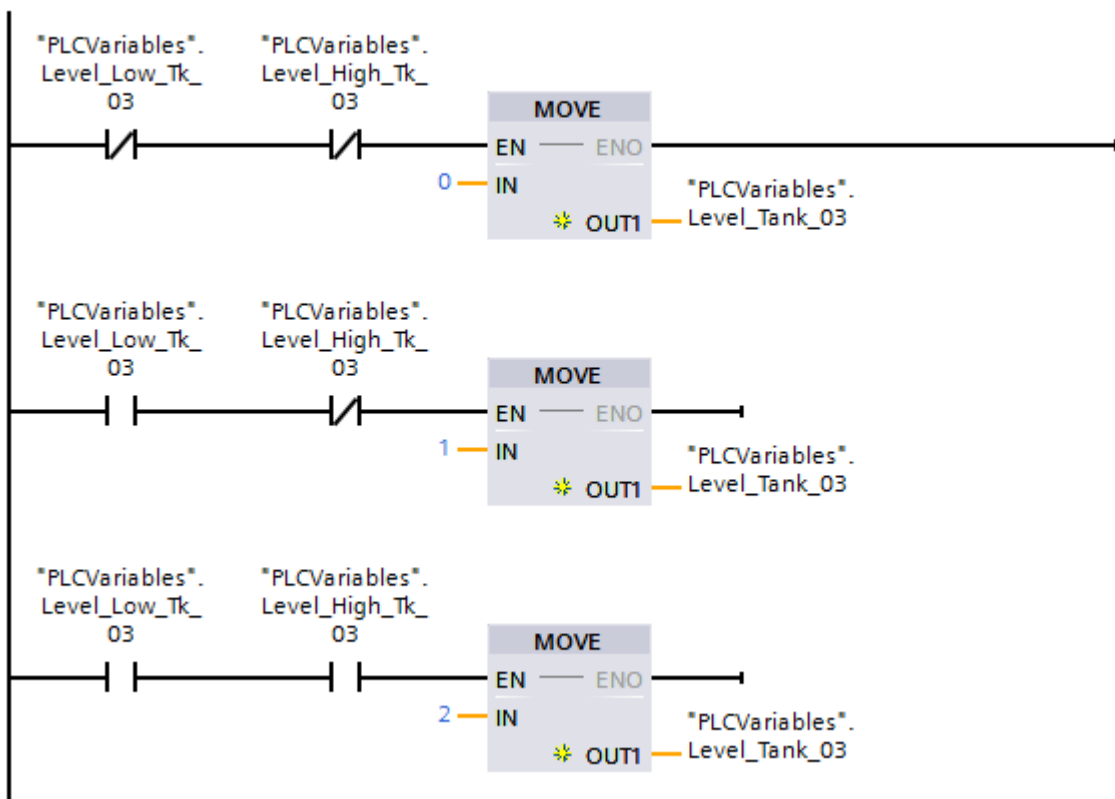
Niveles del Filtro 02



Niveles para el Tanque 02



Niveles para el Tanque 03



3.8. IMPLEMETACIÓN DEL SCADA EN HTML

Para la implementación del SCADA utilizamos el Servidor Web del PLC S7-1200, para ello lo que primero realizamos es la activación del Web Server del PLC.

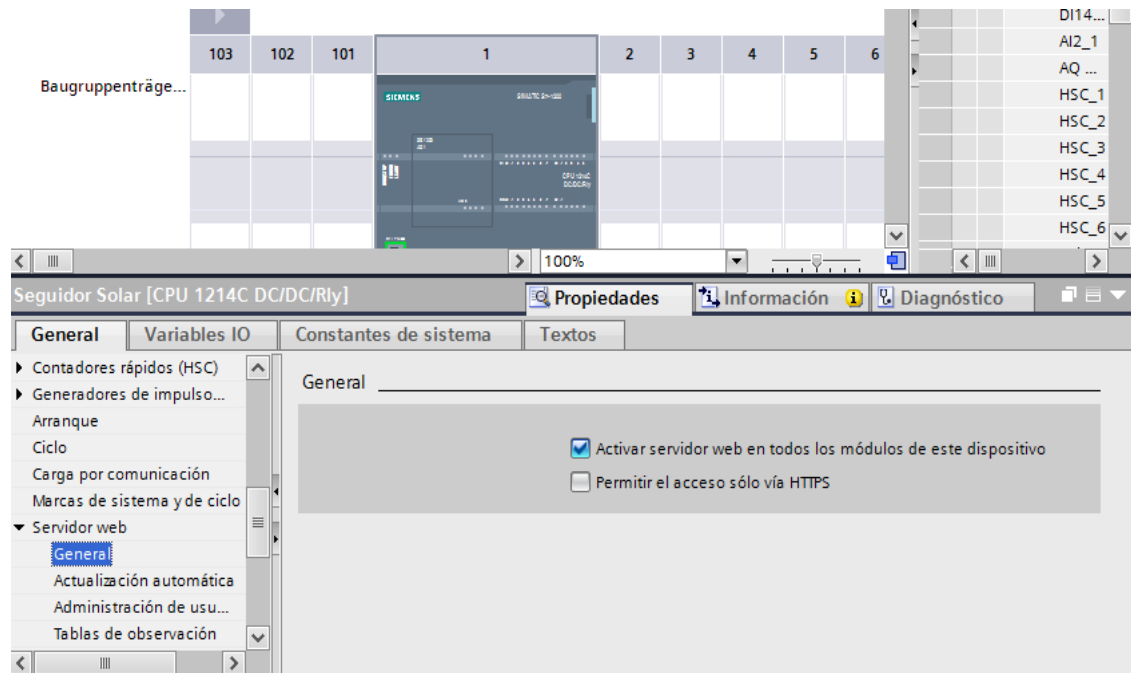


Fig. 3. 50 Activación del Servidor Web del PLC S7-1200
(Fuente: PLC Siemens S7-1200)

Para el desarrollo del SCADA, estaba basado en un web dinamica que soporta el Web Server del PLC S7-1200.

PROGRAMACION DEL HTML

Iniciamos la programacion del HTML en un block de nota donde se guardara con la extensión .html.

Primero declaramos todas las variables que sacaremos del PLC S7-120, con la instrucción AWP_In_Variable, que permite al HTML poder leer las variables del PLC.

```
<!-- AWP_In_Variable Name=""PLCVariables".Level_High_Tk_02' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_Low_Tk_02' -->
<!--
AWP_In_Variable
Name=""PLCVariables".Level_Very_Low_FI_02' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_Low_FI_02' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_High_FI_02' -->
<!--
AWP_In_Variable
Name=""PLCVariables".Level_Very_High_FI_02' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_Low_Tk_03' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_High_Tk_03' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_Tank_02' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_Tank_03' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_FI_01' -->
<!-- AWP_In_Variable Name=""PLCVariables".Level_Filtro_02' -->
<!-- AWP_In_Variable Name=""PLCVariables".PREFrecVariador' -->
<!-- AWP_In_Variable Name=""PLCVariables".LDR_Vertical' -->
<!-- AWP_In_Variable Name=""PLCVariables".LDR_Horizontal' -->
<!-- AWP_In_Variable Name=""PLCVariables".Temp_Filtro' -->
<!-- AWP_In_Variable Name=""PLCVariables".Temp_Plato' -->
<!-- AWP_In_Variable Name=""PLCVariables".Flujometro' -->

<!-- AWP_In_Variable Name=""PLCVariables".Valve_FI_2' -->
<!-- AWP_In_Variable Name=""PLCVariables".Valve_Circulacion' -->
<!-- AWP_In_Variable Name=""PLCVariables".PU_01' -->
<!-- AWP_In_Variable Name=""PLCVariables".PU_02' -->
```

Luego pasamos a la función de inicio, en la que me permitirá actualizar constantemente los datos que necesito visualizar con la instrucción “Update”, donde para ello declaro otras variables para luego utilizarlas en sus respectivas funciones de cada variable.

```
function Start()
{
    DetermineBrowser();
    Level_FI_01_Update(:="PLCVariables".Level_FI_01:);           //
    immediate initialization of the value visualization
    Level_FI_02_Update(:="PLCVariables".Level_Filtro_02:);       //
    immediate initialization of the value visualization
    Level_Tank_02_Update(:="PLCVariables".Level_Tank_02:);       //
    immediate initialization of the value visualization
    TemperaturaFiltro_Update(:="PLCVariables".Temp_Filtro:);     //
    immediate initialization of the value visualization
    TemperaturaPlato_Update(:="PLCVariables".Temp_Plato:);       //
    immediate initialization of the value visualization
    Flujometro_Update(:="PLCVariables".Flujometro:);             //
    immediate initialization of the value visualization
    FrecVariador_Update(:="PLCVariables".PREFrecVariador:);      //
    immediate initialization of the value visualization

    //Level_Low_Tk_02_Update(:="PLCVariables".Level_Low_Tk_02:);
    // immediate initialization of the value visualization
    valvula_filtro_Update(:="PLCVariables".Valve_FI_2:);

    valvula_recirculacion_Update(:="PLCVariables".Valve_Circulacion:);

    Bomba_01_Update(:="PLCVariables".PU_01:);
    Bomba_02_Update(:="PLCVariables".PU_02:);
    Level_Tank_03_Update(:="PLCVariables".Level_Tank_03:);       //
    immediate initialization of the value visualization
```

Luego activamos el muestreo de nuestro programa

```

        setTimeout("OnTimer()",200);/////iniciamos el muestreo a
traves de un timer
    }
    function OnTimer()
    {
        if (! g_bPageRequested)
        {
            g_bPageRequested = true;
            // request the update page asynchronously, function
            UpdateCallback is called if response
            // has been received
            DoHttpRequest(this, "update11.dat", UpdateCallback, true);
            // response is javascript
            // this asynchronous method does silently update the data
            within the browser
        }
        setTimeout("OnTimer()", 200); // the function OnTimer is to be
called every 200 ms
    }

```

Luego realizamos la llamada de las variables con la función "UpdateCallback", donde en Modbus se tiene las direcciones anidadas, donde llegaran los datos en orden del array.

```

function UpdateCallback(obj, response, status) {
    var ok;
    //Splitting the results
    var results = response.split(" ");
    //Splitting the results in single signs

```

```

var signs = results[0].split("");
var signs1 = results[1].split("");
var signs2 = results[2].split("");
var signs3 = results[3].split("");
var signs4 = results[4].split("");
var signs5 = results[5].split("");
var signs6 = results[6].split("");
var signs7 = results[7].split("");
var signs8 = results[8].split("");
var signs9 = results[9].split("");
var signs10 = results[10].split("");
var signs11 = results[11].split("");
var signs12 = results[12].split("");
var signs13 = results[13].split("");

var i;
var count = 0;
for (i = 0; i < signs.length; i++) {
    //Check if the first signs are numbers
    if (true == isNaN(signs[i])) {
        count = count + 1;
    }
    else {break;}
}
//delete signs which aren't numbers

Level_Fl_01= results[0].substr(count, signs.length);
Level_Fl_02= results[1].substr(count, signs1.length);
Level_Tank_02= results[8].substr(count, signs8.length);
Frec_Variador= results[2].substr(count, signs2.length);

```

```
//LDR_Vertical= results[6].substr(count, signs6.length);
//LDR_Horizontal= results[7].substr(count, signs7.length);
Temp_Filtro= results[3].substr(count, signs3.length);
Temp_Plato= results[4].substr(count, signs4.length);
Caudal_Flujometro= results[5].substr(count, signs5.length);
state_valvula_filtro= results[9].substr(count, signs9.length);
state_valvula_recirculacion=results[10].substr(count,
signs10.length);
```

```
state_bomba_01= results[11].substr(count, signs11.length);
state_bomba_02= results[12].substr(count, signs12.length);
Level_Tank_03= results[13].substr(count, signs13.length);
```

Respecto a la animación declaramos de la siguiente forma.

```
var Web_Level_FI_01 = parseInt(Level_FI_01);
var Web_Level_FI_02 = parseInt(Level_FI_02);
var Web_Level_Tank_02 = parseInt(Level_Tank_02);
var Web_TemperaturaFiltro_Update = parseInt(Temp_Filtro);
var Web_TemperaturaPlato_Update = parseInt(Temp_Plato);
var Web_Flujometro_Update = parseInt(Caudal_Flujometro);
var Web_FrecVariador_Update = parseInt(Frec_Variador);
var Web_state_valvula_filtro = parseInt(state_valvula_filtro);
varWeb_state_valvula_recirculacion =
parseInt(state_valvula_recirculacion);
var Web_state_bomba_01 = parseInt(state_bomba_01);
var Web_state_bomba_02 = parseInt(state_bomba_02);
var Web_Level_Tank_03 = parseInt(Level_Tank_03);
```

```
if (status < 300) { // check HTTP response status
```

```

        document.getElementById('NivelFiltro01').innerHTML      =
results[0];
        document.getElementById('NivelFiltro02').innerHTML      =
results[1];
        document.getElementById('FrecVariador').innerHTML      =
results[2];
        document.getElementById('TempFiltro').innerHTML        =
results[3];
        document.getElementById('TempPlato').innerHTML          =
results[4];
        document.getElementById('Flujometro').innerHTML        =
results[5];
        //document.getElementById('LDRVertical').innerHTML      =
results[6];
        //document.getElementById('LDRHorizontal').innerHTML    =
results[7];
        document.getElementById('NivelTank02').innerHTML        =
results[8];
        //document.getElementById('Nivel_Bajo_Tank02').innerHTML
= results[5];
        Level_FI_01_Update(Web_Level_FI_01);                  // update with
the provided value
        Level_FI_02_Update(Web_Level_FI_02);                  // update with
the provided value
        Level_Tank_02_Update(Web_Level_Tank_02);              // update
with the provided value
        TemperaturaFiltro_Update(Web_TemperaturaFiltro_Update);
        TemperaturaPlato_Update(Web_TemperaturaPlato_Update);
        Flujometro_Update(Web_Flujometro_Update);

```

```
FrecVariador_Update(Web_FrecVariador_Update);
valvula_filtro_Update(Web_state_valvula_filtro);    // update
with the provided value
```

```
valvula_recirculacion_Update(Web_state_valvula_recirculacion);
// update with the provided value
```

```
Bomba_01_Update(Web_state_bomba_01);
Bomba_02_Update(Web_state_bomba_02);
Level_Tank_03_Update(Web_Level_Tank_03);    // update
with the provided value
```

El desarrollo de las funciones para cada parte del SCADA considerando que visualizaremos los estados de nuestras variables y animaremos el llenado de los depósitos y la animación de la Grafica se encontraran en el Anexo 2.

Debido al desarrollo del programa en HTML, se generara las siguientes carpetas:

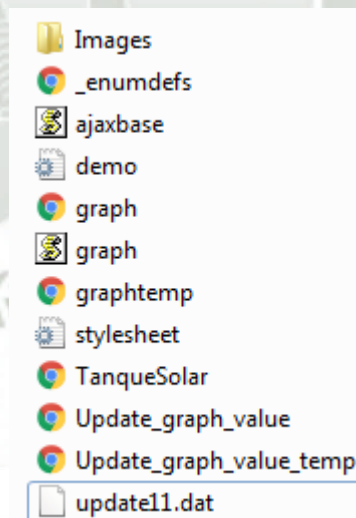


Fig. 3. 51 Generación de Carpetas, HTML
(Fuente: Elaboración Propia)

El archivo “ajaxbase”, me permite realizar la animación de manera dinámica y que la adquisición de datos sea en tiempo real, internamente contiene una programación en html.

El archivo “update11”, me permite almacenar los datos momentáneamente ya que el Web Server no te permite almacenar directamente desde el PLC, sino a través de este archivo. Con este tipo de archivo me permite la transferencia de datos entre el Servidor y el Cliente.

El archivo “TanqueSolar” es la página principal del html.

Los archivos “Graph_Temp”, “Update_graph_vale_temp”, son los que se realizó la parte de los gráficos para que se pueda visualizar la temperatura del Filtro 02 y el valor del Variador.

La carpeta de “Imagens” se almacena las imágenes que hemos utilizado para el SCADA.

Para cargar la página del html en el PLC, nos dirigimos a Páginas por el usuario y direccionamos la carpeta para luego direccionar el archivo del programa y realizamos Click en Generar Bloques y es así como hemos guardado el archivo en el Servidor Web del PLC S7-1200.

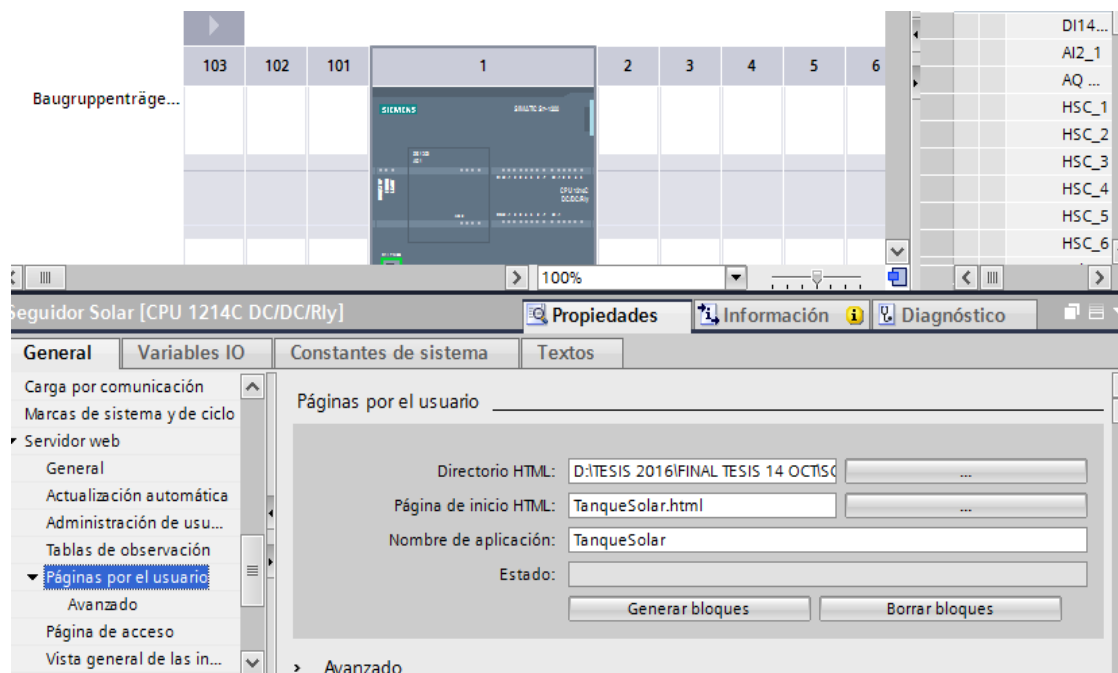


Fig. 3. 52 Ingreso de la Ruta para el SCADA
(Fuente: PLC Siemens S7-1200)

Para poder visualizar el SCADA ingresamos a un explorador de internet e ingresamos el IP del PLC: 192.168.0.1 y este fueron los resultados:



Fig. 3. 53 Visualización del SCADA
(Fuente: Elaboración Propia)

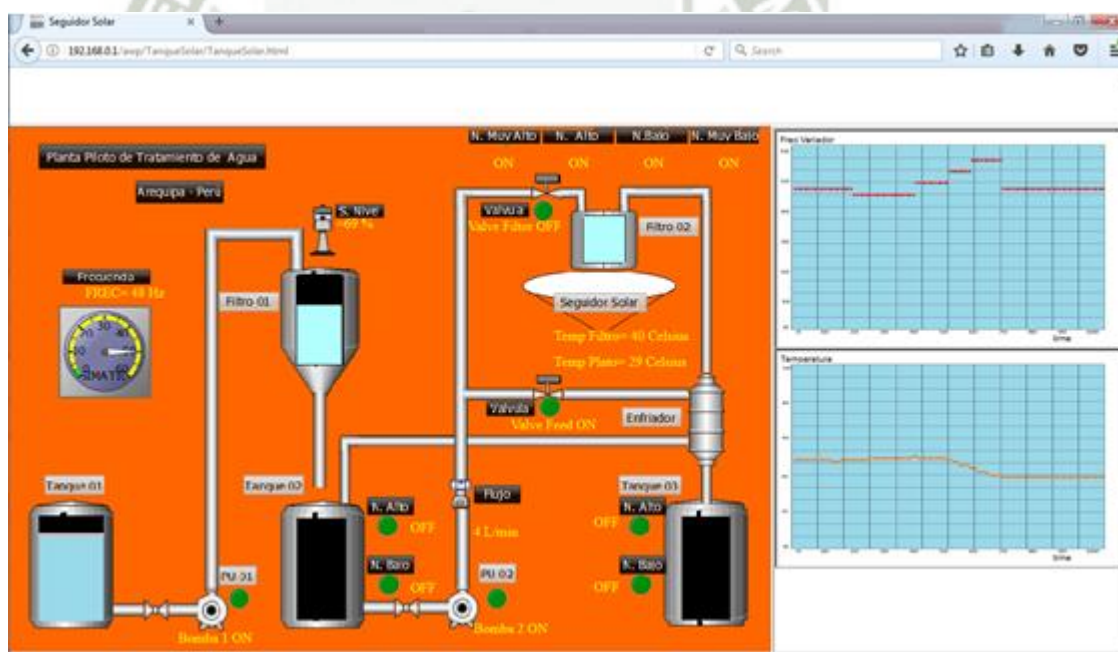


Fig. 3. 54 Visualización del SCADA en Funcionamiento
(Fuente: Elaboración Propia)

3.9. SISTEMA COMPLETO EN FUNCIONAMIENTO

Ensamblaje de la planta



Fig. 3. 55 Ensamblaje de la Planta Piloto
(Fuente: Elaboración Propia)

Planta en funcionamiento.



Fig. 3. 56 Funcionamiento de la Planta Piloto
(Fuente: Elaboración Propia)

CONCLUSIONES

1. Se logró la implementación de la planta piloto para la reutilización de agua mediante el filtrado y condensación, utilizando el Arduino y un PLC.
2. Respecto al control del seguidor solar se llegó a implementar con el microcontrolador Arduino cumpliendo satisfactoriamente la lectura de los datos como el de temperatura para luego ser enviados al PLC. Y de igual forma con el seguimiento al sol, siendo eficiente en la captación de energía.
3. Se logró la integración del Arduino con el PLC, mediante comunicación Modbus TCP/IP.
4. Se desarrolló el Control Fuzzy para la Etapa del Filtrado por evaporación utilizando lenguaje STL y adicionalmente se desarrolló el programa en Grafcet para luego pasar a Ladder.
5. Se desarrolló el SCADA mediante HTML, para luego ser visualizado por un explorador de internet.
6. Se utilizó los instrumentos y equipos adecuados, para poder lograr un correcto funcionamiento de la planta y así alcanzar los objetivos.

RECOMENDACIONES

1. Mejorar el SCADA para que pueda operar desde el monitor y mejorando las gráficas para un dinamismo más real.
2. Realizar una comparación con otro tipo de control para que cumpla el funcionamiento esperado.
3. Implementarle un tablero para su protección de la parte de control y así poder aislar con la etapa de potencia.
4. Añadir a la planta, sensores ultrasónicos para los tanques, sensores de flujo, sensor de PH y así mejorar la supervisión y control.
5. Utilizar otro tipo de comunicación ya sea por GSM para un control a distancias lejanas.

BIBLIOGRAFIA

- Andres, R. (04 de 05 de 2014). *Lógica Difusa Grupo 2*. Obtenido de SildeShare: <https://es.slideshare.net/AndresRodriguez18/logica-difusa-grupo-2>
- ⁶Arduino. (2017). *Entrada y Salida*. Obtenido de Arduino: <https://www.arduino.cc/en/Main/arduinoBoardMega>
- ¹⁵Barakat, J. (03 de 01 de 2017). *HTML*. Obtenido de MDN web docs moz://a: <https://developer.mozilla.org/es/docs/Web/HTML>
- ⁵Barney, B. (20 de 06 de 2007). *Computación Paralela*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Computaci%C3%B3n_paralela#cite_note-1lnltut-9
- ¹³Catarina UDLAP. (s.f.). *Lógico Difusa*. Obtenido de Sección del Controlador Difuso: http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/hernandez_b_ii/capitulo4.pdf
- Delta Electronics. (2008). *VFD - EL - Manual del Usuario*. Estados Unidos: Delta Electronics.
- ³Energía Nuclear. (22 de 11 de 2016). *Energía Térmica o Energía Calorífica. ¿Qué es?* Obtenido de Energía Nuclear: <https://energia-nuclear.net/definiciones/energia-calorifica.html>
- Guillermo, A. J. (2011). *Sistema, Control e Implementación de un Modulo de Planta de Tratamiento de Agua empleando un PLC y supervisado por un emulador de un terminal Grafico con Pantalla Tactil*. Arequipa: UCSM.
- Guillermo, I. T. (2000). *Tutorial de Lógica Difusa*. Lima: UNMSM.
- Jesús, G. M. (2011). *Control Difuso para un sistema de nivel implementado en un Autómata Programable*. Colombia.
- ⁹Llamas, L. (15 de 07 de 2015). *Que es un Sensor LM35*. Obtenido de Medir Temperatura con Arduino y Sensor LM35: <https://www.luisllamas.es/medir-temperatura-con-arduino-y-sensor-lm35/>

- ⁸NAYLAMP MECHATRONICS. (2017). *Tutorial Sensor de Flujo de Agua*. Obtenido de NAYLAMP MECHATRONICS: http://www.naylampmechatronics.com/blog/47_tutorial-sensor-de-flujo-de-agua.html
- ¹⁰NAYLAMP MECHATRONICS. (2017). *VÁLVULA SOLENOIDE 1/2" 12v DC*. Obtenido de NAYLAMP MECHATRONICS: <http://www.naylampmechatronics.com/valvulas/314-valvula-solenoide-12-12v.html>
- Nesmachnow, S. y. (2010). *Computación de Alta Performance*. Grupo de Procesamiento Parello Aplicado Centro de Cálculo.
- Prometec. (2017). *El Módulo Controlador de Motores L298N*. Obtenido de Prometec: <https://www.prometec.net/l298n/#>
- Ráez, M. (s.f.). *Breve Curso de Lógica Difusa y Control Difuso*. Obtenido de Redeya.com: <http://redeya.bytemaniacos.com/electronica/cursos/fuzzy/fuzzy.htm>
- Roberto, S. (16 de 01 de 2017). *ANÁLISIS TÉRMICO POR EL MÉTODO DE ELEMENTOS FINITOS*. Obtenido de ESSS: <http://www.esss.com.br/blog/es/2017/01/analisis-termico-por-el-metodo-de-elementos-finitos/>
- RobotShop. (s.f.). *Arduino Mega 2560 Datasheet*. Obtenido de RobotShop: <http://www.robotshop.com/media/files/pdf/arduinomega2560datasheet.pdf>
- Ruiz Olaya, A. (2002). *Implementación de una Red Modbus/TCP*. Santiago de Cali: Universidad del Valle.
- ¹Rural, G. d. (01 de 03 de 2017). *El desarrollo de la Energía Solar en el Perú*. Obtenido de Pontificia Universidad Católica del Perú: <http://gruporural.pucp.edu.pe/nota/el-desarrollo-de-la-energia-solar-en-el-peru/>
- Siemens. (2001). *The Probe Medidor de Nivel*. Canada: Siemens.
- Siemens. (12 de 08 de 2004). *Significado y funcionamiento del "Keep Alive" en los enlaces TCP*. Obtenido de Industry Online Support: <https://support.industry.siemens.com/cs/document/15014588/significado-y-funcionamiento-del-keep-alive-en-los-enlaces-tcp?dti=0&lc=es-WW>
- ¹¹Siemens. (11 de 2009). *Simatic S7 Controlador Programable S7-1200*. En Siemens, *Manual de Sistema* (pág. 378). Alemania.

- Siemens. (09 de 10 de 2015). *Creación y utilización de páginas web propias en el S7-1200*. Obtenido de Industry Online Support: <https://support.industry.siemens.com/cs/document/58862931/creaci%C3%B3n-y-utilizaci%C3%B3n-de-p%C3%A1ginas-web-propias-en-el-s7-1200-?dti=0&lc=es-WW>
- Siemens. (2015). *Creating User-defined Web Page on S7-1200 / S7-1500*. Siemens.
- Siemens. (2016). *S7-1500, ET 200SP, ET 200pro Servidor Web*. Alemania: Siemens.
- ¹⁴Takagi - Sugeno. (2017). *Reglas de Takagi - Sugeno*. Obtenido de Takagi - Sugeno: <https://takagi-sugeno.wikispaces.com/>
- Tomás, A. V. (2014). *Introducción a la Lógica Difusa*.
- ⁷Web - Robotica.com. (2017). *Descripción del módulo Arduino Ethernet Shield*. Obtenido de Como funciona el módulo Arduino Ethernet Shield: <http://www.web-robotica.com/arduino/como-funciona-el-modulo-arduino-ethernet-shield>
- ⁴Wiki Water. (s.f.). *El tratamiento del agua por filtración lenta en arena para uso familiar*. Obtenido de Wiki Water: <http://www.wikiwater.fr/e21-el-tratamiento-del-agua-por.html>
- ¹⁶Wikipedia - La enciclopedia libre. (11 de 11 de 2017). *JavaScript*. Obtenido de Wikipedia - La enciclopedia libre: <https://en.wikipedia.org/wiki/JavaScript>
- ¹²Wikipedia - La enciclopedia libre. (15 de 09 de 2017). *Modbus*. Obtenido de Modbus: <https://es.wikipedia.org/wiki/Modbus>
- ²Zeballos Paz, I. (2017). *Purificación del Agua por Condensación Provocada por Calentador Automatico Seguidor Solar con Supervisión de Mini- SCADA FAST - Tools*. Arequipa: UCSM.

ANEXOS

Anexo 1: Planos de la Planta Piloto

Anexo 2: Programa SCADA Planta Piloto en HTML

Anexo 3: Constancia de Funcionamiento por CETEMIN



Anexo 1

Planos de la Planta Piloto





TABLERO DE CONTROL

AUTOMATIZACIÓN DE PLANTA PILOTO PARA LA REUTILIZACIÓN DE AGUA POR FILTRADO Y CONDENSACIÓN; UTILIZANDO PROGRAMACIÓN PARALELA Y SECUENCIAL, MEDIANTE CONTROL FUZZY

UBICACION 961 VITOR

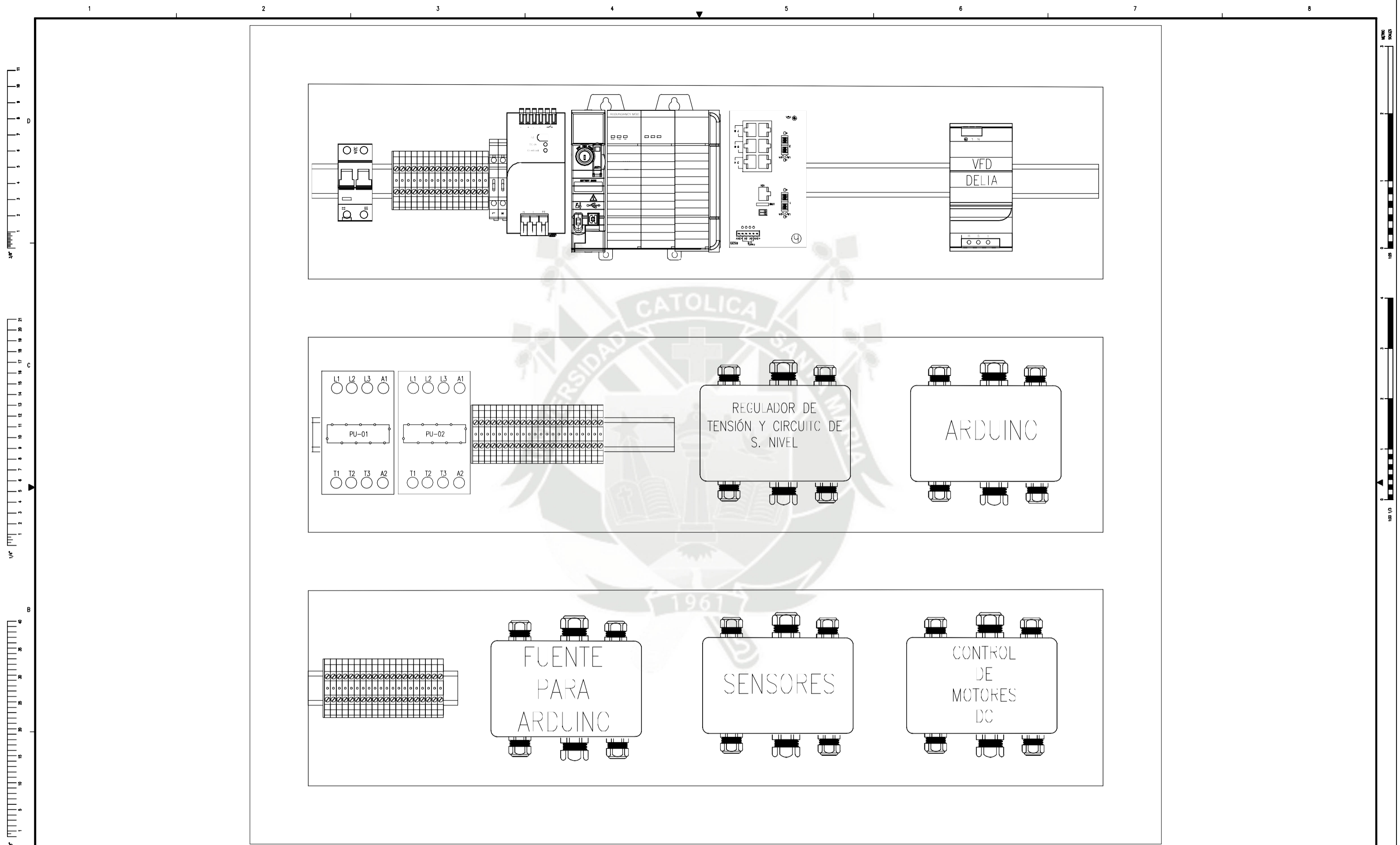
TENSION DE ALIMENTACION 220VAC

TENSION DE MANDO 220VAC

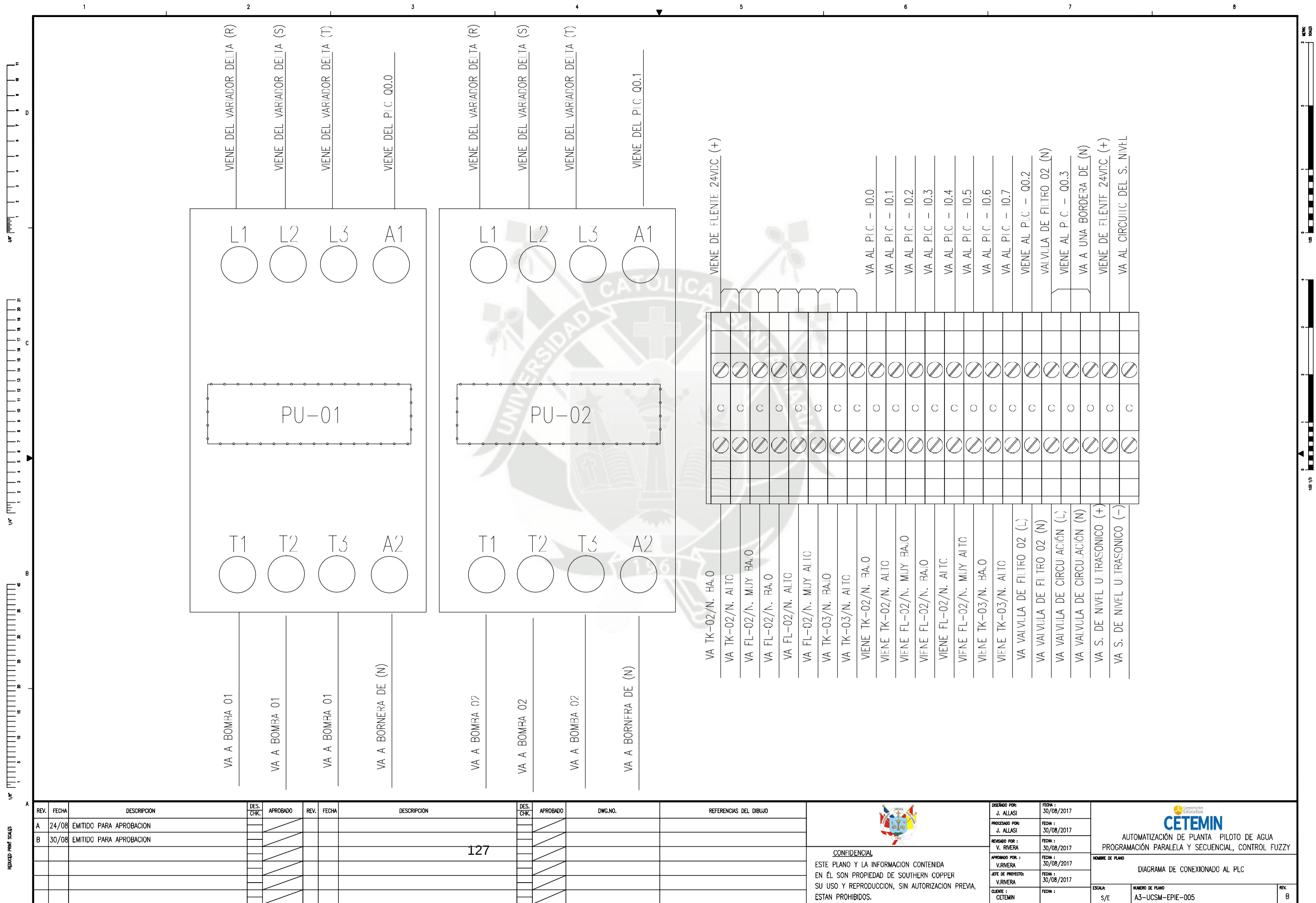
PLANOS PLANOS LAYOUT Y PLANOS CONEXIONADO

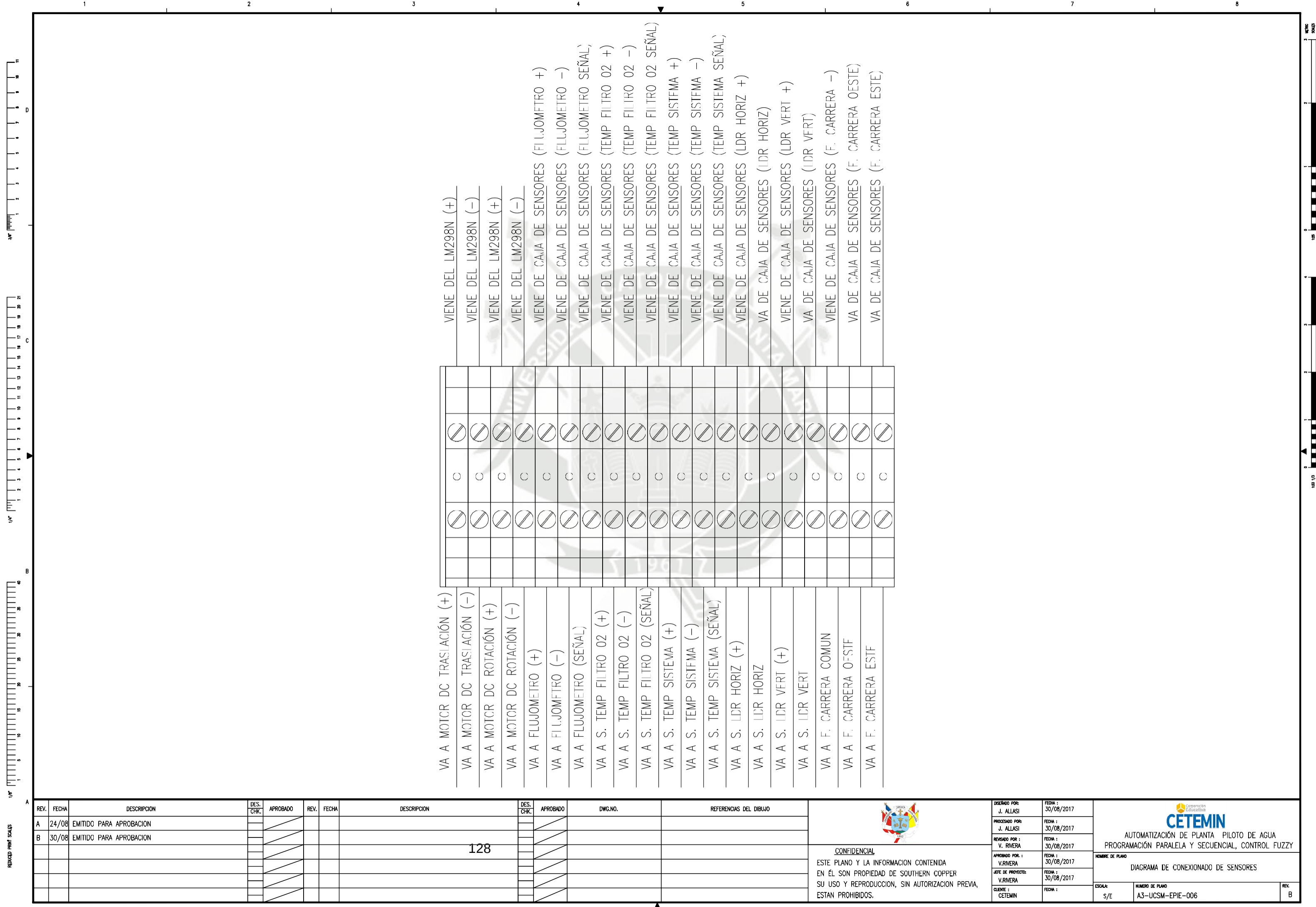
AÑO 2017

REV.	FECHA	DESCRIPCION	DES. CHK.	APROBADO	REV.	FECHA	DESCRIPCION	DES. CHK.	APROBADO	DWG.NO.	REFERENCIAS DEL DIBUJO	 CONFIDENCIAL ESTE PLANO Y LA INFORMACION CONTENIDA EN ÉL SON PROPIEDAD DE SOUTHERN COPPER SU USO Y REPRODUCCION, SIN AUTORIZACION PREVIA, ESTAN PROHIBIDOS.	DISEÑADO POR: J. ALLASI	FECHA : 30/08/2017	 AUTOMATIZACIÓN DE PLANTA PILOTO DE AGUA PROGRAMACIÓN PARALELA Y SECUENCIAL, CONTROL FUZZY			
A	24/08	EMITIDO PARA APROBACION											PROCESADO POR: J. ALLASI	FECHA : 30/08/2017				
B	30/08	EMITIDO PARA APROBACION											REVISADO POR : V. RIVERA	FECHA : 30/08/2017				
													APROBADO POR : V.RIVERA	FECHA : 30/08/2017				
													JEFE DE PROYECTO: V.RIVERA	FECHA : 30/08/2017				
													CLIENTE : CETEMIN	FECHA :	ESCALA: S/E	NUMERO DE PLANO A3-UCSM-EPIE-001	REV. B	



REV.	FECHA	DESCRIPCION	DES. CHK.	APROBADO	REV.	FECHA	DESCRIPCION	DES. CHK.	APROBADO	DWG.NO.	REFERENCIAS DEL DIBUJO	 <u>CONFIDENCIAL</u> ESTE PLANO Y LA INFORMACION CONTENIDA EN ÉL SON PROPIEDAD DE SOUTHERN COPPER SU USO Y REPRODUCCION, SIN AUTORIZACION PREVIA, ESTAN PROHIBIDOS.	DISEÑADO POR: J. ALLASI	FECHA : 30/08/2017	 AUTOMATIZACIÓN DE PLANTA PILOTO DE AGUA PROGRAMACIÓN PARALELA Y SECUENCIAL, CONTROL FUZZY DIAGRAMA DE DISTRIBUCIÓN		
A	24/08	EMITIDO PARA APROBACION											PROCESADO POR: J. ALLASI	FECHA : 30/08/2017			
B	30/08	EMITIDO PARA APROBACION											REVISADO POR : V. RIVERA	FECHA : 30/08/2017			
							125						APROBADO POR : V.RIVERA	FECHA : 30/08/2017			
													JEFE DE PROYECTO: V.RIVERA	FECHA : 30/08/2017			
													CLIENTE : CETEMIN	FECHA :	ESCALA: S/E	NUMERO DE PLANO A3-UCSM-EPIE-003	REV. B





REV.	FECHA	DESCRIPCION	DES. CHK.	APROBADO	REV.	FECHA	DESCRIPCION	DES. CHK.	APROBADO	DWG.NO.	REFERENCIAS DEL DIBUJO						
A	24/08	EMITIDO PARA APROBACION										 <u>CONFIDENCIAL</u> ESTE PLANO Y LA INFORMACION CONTENIDA EN ÉL SON PROPIEDAD DE SOUTHERN COPPER SU USO Y REPRODUCCION, SIN AUTORIZACION PREVIA, ESTAN PROHIBIDOS.	DISERADO POR: J. ALLASI	FECHA : 30/08/2017	 AUTOMATIZACIÓN DE PLANTA PILOTO DE AGUA PROGRAMACIÓN PARALELA Y SECUENCIAL, CONTROL FUZZY		
B	30/08	EMITIDO PARA APROBACION									PROCESADO POR: J. ALLASI		FECHA : 30/08/2017				
											REVISADO POR : V. RIVERA		FECHA : 30/08/2017				
											APROBADO POR : V.RIVERA		FECHA : 30/08/2017				
											JEFE DE PROYECTO: V.RIVERA		FECHA : 30/08/2017				
											CLIENTE : CETEMIN		FECHA :				
												NOMBRE DE PLANO	DIAGRAMA DE CONEXIONADO DE SENSORES				
												ESCALA: S/E				NUMERO DE PLANO A3-UCSM-EPIE-006	

Anexo 2
Programa SCADA Planta
Piloto en HTML

// PROGRAMA SCADA PLANTA PILOTO EN HTML

```

<!-- AWP_In_Variable Name="PLCVariables".Level_High_Tk_02' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_Low_Tk_02' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_Very_Low_FI_02' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_Low_FI_02' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_High_FI_02' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_Very_High_FI_02' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_Low_Tk_03' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_High_Tk_03' -->

<!-- AWP_In_Variable Name="PLCVariables".Level_Tank_02' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_Tank_03' -->

<!-- AWP_In_Variable Name="PLCVariables".Level_FI_01' -->
<!-- AWP_In_Variable Name="PLCVariables".Level_Filtro_02' -->
<!-- AWP_In_Variable Name="PLCVariables".PREFrecVariador' -->
<!-- AWP_In_Variable Name="PLCVariables".LDR_Vertical' -->
<!-- AWP_In_Variable Name="PLCVariables".LDR_Horizontal' -->
<!-- AWP_In_Variable Name="PLCVariables".Temp_Filtro' -->
<!-- AWP_In_Variable Name="PLCVariables".Temp_Plato' -->
<!-- AWP_In_Variable Name="PLCVariables".Flujometro' -->

<!-- AWP_In_Variable Name="PLCVariables".Valve_FI_2' -->
<!-- AWP_In_Variable Name="PLCVariables".Valve_Circulacion' -->

<!-- AWP_In_Variable Name="PLCVariables".PU_01' -->
<!-- AWP_In_Variable Name="PLCVariables".PU_02' -->

<!-- This AWP-Statement is necessary, if the variable "Velocity" will be written
within the HTML page -->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Language" content="en">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta http-equiv="Content-Script-Type" content="text/javascript" >
<link rel="stylesheet" type="text/css" href="stylesheet.css">

<title>Seguidor Solar</title>
<link rel="stylesheet" type="text/css" href="demo.css">
<script src="ajaxbase.js" type="text/javascript"></script>

```

```
<script type="text/javascript">

    var g_bPageRequested = false;
    var par=false; //parpadeo

    // a timer will be started, after one second the function OnTimer will be called
    function Start()
    {
        DetermineBrowser();
        Level_FI_01_Update(:="PLCVariables".Level_FI_01:); // immediate
        initialization of the value visualization
        Level_FI_02_Update(:="PLCVariables".Level_Filtro_02:); // immediate
        initialization of the value visualization
        Level_Tank_02_Update(:="PLCVariables".Level_Tank_02:); // immediate
        initialization of the value visualization
        TemperaturaFiltro_Update(:="PLCVariables".Temp_Filtro:); // immediate
        initialization of the value visualization
        TemperaturaPlato_Update(:="PLCVariables".Temp_Plato:); // immediate
        initialization of the value visualization
        Flujometro_Update(:="PLCVariables".Flujometro:); // immediate
        initialization of the value visualization
        FrecVariador_Update(:="PLCVariables".PREFrecVariador:); // immediate
        initialization of the value visualization
        //Level_Low_Tk_02_Update(:="PLCVariables".Level_Low_Tk_02:); //
        immediate initialization of the value visualization
        valvula_filtro_Update(:="PLCVariables".Valve_FI_2:);
        valvula_recirculacion_Update(:="PLCVariables".Valve_Circulacion:);

        Bomba_01_Update(:="PLCVariables".PU_01:);
        Bomba_02_Update(:="PLCVariables".PU_02:);
        Level_Tank_03_Update(:="PLCVariables".Level_Tank_03:); // immediate
        initialization of the value visualization

        setTimeout("OnTimer()",200);/////iniciamos el muestreo a traves de un timer

    }

    // The page update11.dat solely consists of a reference to the variable
    "Dynvalue".
    // <colon><equalsign><Variablenname><colon> is the common syntax for
    variable references.
    // Since this page is set "manual" (in AWP_Start_Fragment - statement at begin
    of the page), a dedicated
```

```
// user program within the PCL is triggered when this page is requested and will
then defer the HTTP response
// until the variables value has been changed
function OnTimer()
{
    if (! g_bPageRequested)
    {
        g_bPageRequested = true;
        // request the update page asynchronously, function UpdateCallback is
called if response
        // has been received
        DoHttpRequest(this, "update11.dat", UpdateCallback, true); // response
is javascript

        // this asynchronous method does silently update the data within the
browser
    }
    setTimeout("OnTimer()", 200); // the function OnTimer is to be called every
200 ms
}

// update11.dat has been received
function UpdateCallback(obj, response, status) {
    var ok;
    //Splitting the results
    var results = response.split(" ");
    //Splitting the results in single signs
    var signs = results[0].split("");
    var signs1 = results[1].split("");
    var signs2 = results[2].split("");
    var signs3 = results[3].split("");
    var signs4 = results[4].split("");
    var signs5 = results[5].split("");
    var signs6 = results[6].split("");
    var signs7 = results[7].split("");
    var signs8 = results[8].split("");
    var signs9 = results[9].split("");
    var signs10 = results[10].split("");
    var signs11 = results[11].split("");
    var signs12 = results[12].split("");
    var signs13 = results[13].split("");
    var i;
    var count = 0;
    for (i = 0; i < signs.length; i++) {
        //Check if the first signs are numbers
        if (true == isNaN(signs[i])) {
```

```

        count = count + 1;
    }
    else {break;}
}
//delete signs which aren't numbers

Level_FI_01= results[0].substr(count, signs.length);
Level_FI_02= results[1].substr(count, signs1.length);
Level_Tank_02= results[8].substr(count, signs8.length);
Frec_Variador= results[2].substr(count, signs2.length);
//LDR_Vertical= results[6].substr(count, signs6.length);
//LDR_Horizontal= results[7].substr(count, signs7.length);
Temp_Filtro= results[3].substr(count, signs3.length);
Temp_Plato= results[4].substr(count, signs4.length);
Caudal_Flujometro= results[5].substr(count, signs5.length);
state_valvula_filtro= results[9].substr(count, signs9.length);
state_valvula_recirculacion= results[10].substr(count, signs10.length);

state_bomba_01= results[11].substr(count, signs11.length);
state_bomba_02= results[12].substr(count, signs12.length);
Level_Tank_03= results[13].substr(count, signs13.length);

var Web_Level_FI_01 = parseInt(Level_FI_01);
var Web_Level_FI_02 = parseInt(Level_FI_02);
var Web_Level_Tank_02 = parseInt(Level_Tank_02);
var Web_TemperaturaFiltro_Update = parseInt(Temp_Filtro);
var Web_TemperaturaPlato_Update = parseInt(Temp_Plato);
var Web_Flujometro_Update = parseInt(Caudal_Flujometro);
var Web_FrecVariador_Update = parseInt(Frec_Variador);
var Web_state_valvula_filtro = parseInt(state_valvula_filtro);
var Web_state_valvula_recirculacion =
parseInt(state_valvula_recirculacion);
var Web_state_bomba_01 = parseInt(state_bomba_01);
var Web_state_bomba_02 = parseInt(state_bomba_02);
var Web_Level_Tank_03 = parseInt(Level_Tank_03);

if (status < 300) {// check HTTP response status
document.getElementById('NivelFiltro01').innerHTML = results[0];
document.getElementById('NivelFiltro02').innerHTML = results[1];
document.getElementById('FrecVariador').innerHTML = results[2];
document.getElementById('TempFiltro').innerHTML = results[3];
document.getElementById('TempPlato').innerHTML = results[4];
document.getElementById('Flujometro').innerHTML = results[5];
//document.getElementById('LDRVertical').innerHTML = results[6];

```

```

//document.getElementById('LDRHorizontal').innerHTML = results[7];
document.getElementById('NivelTank02').innerHTML = results[8];
//document.getElementById('Nivel_Bajo_Tank02').innerHTML = results[5];

Level_FI_01_Update(Web_Level_FI_01);    // update with the provided
value
Level_FI_02_Update(Web_Level_FI_02);    // update with the provided
value
Level_Tank_02_Update(Web_Level_Tank_02);    // update with the
provided value
TemperaturaFiltro_Update(Web_TemperaturaFiltro_Update);
TemperaturaPlato_Update(Web_TemperaturaPlato_Update);
Flujometro_Update(Web_Flujometro_Update);
FrecVariador_Update(Web_FrecVariador_Update);
valvula_filtro_Update(Web_state_valvula_filtro);    // update with the
provided value
valvula_recirculacion_Update(Web_state_valvula_recirculacion);    //
update with the provided value
Bomba_01_Update(Web_state_bomba_01);
Bomba_02_Update(Web_state_bomba_02);
Level_Tank_03_Update(Web_Level_Tank_03);    // update with the
provided value

g_bPageRequested = false;
setTimeout("OnTimer()", 200); // the function OnTimer is to be called in
200 ms
return;
}
if (status == 503) {    // service currently unavailable , server
overloaded
    ok = confirm(Web_Level_FI_01);
} else {
    ok = confirm("FAILED: HTTP error " + status);
}

g_bPageRequested = false;
if (ok) {
    setTimeout("OnTimer()", 200); // the function OnTimer is to be called in 1
sec
}
}

// Within the page update11.html or update11.js the function ForceUpdate is
called with the current value.
// This value (0..255) controls the width of table "table2" within the table "table1"

```

```
function Level_FI_01_Update(val)
{
    var width, barval;
    var talem;
    var Corriente;
    var Presion;
    var Nivel;
    var Altura;

    Nivel=val;

    talem = parent.document.getElementById("bar_Level_FI_01");
    // width = talem.parentNode.clientWidth;

    // barval = ((val*width)/256); // convert to percent, Firefox can
    display % only for integral numbers !
    barval=Nivel*0.67;
    if (Nivel < 0)
    {
        barval = 1;
        obj = document.getElementById('nosensor');
        obj.style.visibility = 'visible';
    }
    if (Nivel > 0)
    {
        obj = document.getElementById('nosensor');
        obj.style.visibility = 'hidden';
    }

    if (Nivel > 95)
    {
        obj = document.getElementById('limite');
        obj.style.visibility = 'visible';
    }

    if (Nivel < 95)
    {
        obj = document.getElementById('limite');
        obj.style.visibility = 'hidden';
    }

    if (Nivel > 102)
    {

```

```

obj = document.getElementById('alarma');
obj.style.visibility = 'visible';
obj = document.getElementById('limite');
obj.style.visibility = 'hidden';

document.getElementById('bar_Level_FI_01').style.backgroundColor='red';
    barval = 68;
}
    if (Nivel < 102)
    {
        obj = document.getElementById('alarma');
        obj.style.visibility = 'hidden';
document.getElementById('bar_Level_FI_01').style.backgroundColor="#B2FFFF";

    }

    tabelem.style.height = 1.5*Math.floor(barval)+"px";    // set table width

    var td = parent.document.getElementById("Ultrasonico");    // display value
numerically
    if (td.textContent)
        {
            // textContent is ok for Firefox, not for IE
            td.textContent = "=" + val + " %";
        }
        else
        {
            td.innerHTML = val+"data";
        }
    g_bPageRequested = false;
}

function TemperaturaFiltro_Update(temp_filtro)
{

    var temp = parent.document.getElementById("temperatura_filtro");    //
display value numerically
    if (temp.textContent)
        {
            // textContent is ok for Firefox, not for IE
            temp.textContent = "Temp Filtro= " + temp_filtro + " Celsius";
        }
        else
        {

```

```

        temp.innerHTML = temp_filtro +"data";
    }
    g_bPageRequested = false;

}

function TemperaturaPlato_Update(temp_plato)
{

    var temp1 = parent.document.getElementById("temperatura_plato");    //
display value numerically
    if (temp1.textContent)
    {
        // textContent is ok for Firefox, not for IE
        temp1.textContent ="Temp Plato= "+ temp_plato + " Celsius";
    }
    else
    {
        temp1.innerHTML = temp_plato +"data";
    }
    g_bPageRequested = false;

}

function Flujometro_Update(Flujometro1)
{

    var caudal;

    caudal=Flujometro1;

    var temp2 = parent.document.getElementById("text_flujometro");
    //.style.backgroundColor='red'; display value numerically
    if (temp2.textContent)
    {
        // textContent is ok for Firefox, not for IE

        temp2.textContent =caudal +" L/min";

    }
    else
    {
        temp2.innerHTML = caudal+"L/min";
    }
}

```

```

        g_bPageRequested = false;

    }

function FrecVariador_Update(variador)
{
    var temp3 = parent.document.getElementById("text_variador");
    //.style.backgroundColor='red'; display value numerically
    if (temp3.textContent)
    {
        // textContent is ok for Firefox, not for IE
        temp3.textContent = "FREC= " + variador + " Hz";
    }
    else
    {
        temp3.innerHTML = variador + "data";
    }
    g_bPageRequested = false;
}

function Level_FI_02_Update(val2)
{
    var width2, barval2;
    var talem2;
    var Nivel2;

    Nivel2=val2;

    talem2 = parent.document.getElementById("bar_Level_FI_02");
    barval2=Nivel2*0.45;

    var temp7 = parent.document.getElementById("text_muy_bajo_tk_02");
    var temp8 = parent.document.getElementById("text_bajo_tk_02");
    var temp9 = parent.document.getElementById("text_alto_tk_02");
    var temp10 = parent.document.getElementById("text_muy_alto_tk_02");

    if (Nivel2 < 0)
    {
        barval2 = 1;
        obj = document.getElementById('nosensor2');
    }
}

```

```

obj.style.visibility = 'visible';
}
if (Nivel2 == 0)
{
    // textContent is ok for Firefox, not for IE
    temp7.textContent = "___";
    temp8.textContent = "___";
    temp9.textContent = "___";
    temp10.textContent = "___";
}

if (Nivel2 == 20)
{
    temp7.textContent = "ON";
    temp8.textContent = "___";
    temp9.textContent = "___";
    temp10.textContent = "___";
}

if (Nivel2 == 40)
{
    temp7.textContent = "ON";
    temp8.textContent = "ON";
    temp9.textContent = "___";
    temp10.textContent = "___";
}

if (Nivel2 == 60)
{
    temp7.textContent = "ON";
    temp8.textContent = "ON";
    temp9.textContent = "ON";
    temp10.textContent = "___";
}

if (Nivel2 == 80)
{
    temp7.textContent = "ON";
    temp8.textContent = "ON";
    temp9.textContent = "ON";
    temp10.textContent = "ON";
}

if (Nivel2 < 102)
{
    obj = document.getElementById('alarma2');
    obj.style.visibility = 'hidden';
    document.getElementById('bar_Level_FI_02').style.backgroundColor="#B2FFFF";
}

```

```

    }
    tabelem2.style.height = 1.5*Math.floor(barval2)+"px";    // set table width

    g_bPageRequested = false;
}

function Level_Tank_02_Update(val3)
{
    var barval3;
    var tabelem3;
    var Nivel3;

    Nivel3=val3;

    tabelem3 = parent.document.getElementById("bar_Level_Tank_02");
    barval3=Nivel3;

    var temp5 = parent.document.getElementById("text_state_Low_tank_02");
    var temp6 = parent.document.getElementById("text_state_High_tank_02");

    if (Nivel3 < 0)
    {
        barval3 = 1;
        obj = document.getElementById('nosensor2');
        obj.style.visibility = 'visible';
    }
    if (Nivel3 == 0)
    {
        // textContent is ok for Firefox, not for IE
        temp5.textContent ="OFF";
        temp6.textContent ="OFF";
    }
    if (Nivel3 == 25)
    {
        // textContent is ok for Firefox, not for IE
        temp5.textContent ="ON ";
        temp6.textContent ="OFF ";
    }
    if (Nivel3 == 70)
    {
        temp5.textContent ="ON ";
        temp6.textContent ="ON ";
    }
    if (Nivel3 > 100)

```

```

    {
        obj = document.getElementById('alarma2');
        obj.style.visibility = 'visible';
        obj = document.getElementById('limite2');
        obj.style.visibility = 'hidden';

document.getElementById('bar_Level_Tank_02').style.backgroundColor='red';
        barval3 = 100;
    }
    if (Nivel3 < 102)
    {
        obj = document.getElementById('alarma2');
        obj.style.visibility = 'hidden';
document.getElementById('bar_Level_Tank_02').style.backgroundColor="#B2FFFF";
    }

    tabelem3.style.height = 1.5*Math.floor(barval3)+"px";    // set table width

    g_bPageRequested = false;
}

function valvula_filtro_Update(valfiltro)
{
    var temp11 = parent.document.getElementById("text_valvula_filtro");
    //.style.backgroundColor='red'; display value numerically

    if (valfiltro == 0)
    {
        // textContent is ok for Firefox, not for IE
        temp11.textContent = "Valve Filter OFF";
    }
    if (valfiltro == 1)
    {
        // textContent is ok for Firefox, not for IE
        temp11.textContent = "Valve Filter ON";
    }
    g_bPageRequested = false;
}

function valvula_recirculacion_Update(valrec)
{

```

```

var temp12 =
parent.document.getElementById("text_valvula_recirculacion");
//.style.backgroundColor='red'; display value numerically
if (temp12.textContent)
{
    // textContent is ok for Firefox, not for IE
    if( valrec == 1)
    {
        temp12.textContent = "Valve Feed ON";
    }
    else
    {
        temp12.textContent = "Valve Feed OFF";
    }
}
else
{
    temp12.innerHTML = "data";
}
g_bPageRequested = false;
}
function Bomba_01_Update(bomba01)
{
    var temp13 = parent.document.getElementById("text_bomba_01");
    //.style.backgroundColor='red'; display value numerically
    if (bomba01 == 0)
    {
        // textContent is ok for Firefox, not for IE
        temp13.textContent = "Bomba 1 OFF";
    }
    if (bomba01 == 1)
    {
        // textContent is ok for Firefox, not for IE
        temp13.textContent = "Bomba 1 ON";
    }
    g_bPageRequested = false;
}
function Bomba_02_Update(bomba02)
{
    var temp14 = parent.document.getElementById("text_bomba_02");
    //.style.backgroundColor='red'; display value numerically

```

```

if (bomba02 == 0)
{
    // textContent is ok for Firefox, not for IE
    temp14.textContent = "Bomba 2 OFF";
}
if (bomba02 == 1)
{
    // textContent is ok for Firefox, not for IE
    temp14.textContent = "Bomba 2 ON";
}
g_bPageRequested = false;
}
function Level_Tank_03_Update(val4)
{
    var barval4;
    var tabelle4;
    var Nivel4;

    Nivel4=val4;

    tabelle4 = parent.document.getElementById("bar_Level_Tank_03");
    barval4=Nivel4*35;

    var temp15 = parent.document.getElementById("text_state_Low_tank_03");
    var temp16 = parent.document.getElementById("text_state_High_tank_03");

    if (Nivel4 == 0)
    {
        // textContent is ok for Firefox, not for IE
        temp15.textContent = "OFF";
        temp16.textContent = "OFF";
    }

    if (Nivel4 == 1)
    {
        // textContent is ok for Firefox, not for IE
        temp15.textContent = "ON ";
        temp16.textContent = "OFF ";
    }

    if (Nivel4 == 2)
    {
        temp15.textContent = "ON ";
        temp16.textContent = "ON ";
    }
}

```

```

parpadeo() {
  col=par ? 'red' : 'black';
  document.getElementById('txtlimite').style.color=col;
  par = !par;
  setTimeout("parpadeo()",500); //500 = medio segundo
}

>
>

////////////////////NICIO////////////////////////////////////

onload="Start()" >

<div class="demohead">
<tr>
  <td class="democoltitle" >TESIS 2017:</td>
  <td class="democoltext" >Planta de Recuperacion de Aguas</td>
</tr>
</div>

<div>
  el Filtro 01 = <span
  elFiltro01">:"PLCVariables".Level_FI_01:</span></p>

```

```
<p>LDR Vertical=    <span
id="LDRVertical">:="PLCVariables".LDR_Vertical:</span></p>
<p>LDR Horizontal=    <span
id="LDRHorizontal">:="PLCVariables".LDR_Horizontal:</span></p>
<p>Nivel Tank 02=    <span
id="NivelTank02">:="PLCVariables".Level_Tank_02:</span></p>
<p>Nivel Bajo Tank 02=<span
id="Nivel_Bajo_Tank02">:="PLCVariables".Level_Low_Tk_02:</span></p>
```

<!-- writing of the value "Ultrasonico" requires the AWP_In_Variable - statement at
begin of this page -->

```
<table align="center" border="1" width="100%" id="table1">
<tr>
```

```
<div id="alarma" style="visibility:hidden ; color:#FF0000">
El Filtro 01 esta Apunto de rebalzar <span id="txtalarma">Alarma</span>
</div>
```

```
<div id="limite" style="visibility:hidden ; color:#FF0000">
El Filtro 01 esta completamente lleno <span id="txtlimite">Limite
Alcanzado</span>
</div>
```

```
<div id="alarma2" style="visibility:hidden ; color:#FF0000">
El Filtro 02 esta Apunto de rebalzar <span id="txtalarma">Alarma</span>
</div>
```

```
<div id="limite2" style="visibility:hidden ; color:#FF0000">
El Filtro 02 esta completamente lleno <span id="txtlimite">Limite
Alcanzado</span>
</div>
```

```
<td width="750px" height="450px" rowspan="6" valign="top"
background="Images\SCADA.png"
style="background-repeat:no-repeat; background-position:top left">
```

```
<TABLE BORDER=0 WIDTH=900 HEIGHT=450>
```

```
<TD VALIGN=TOP WIDTH=300>
```

```
<TABLE BORDER=0 ALIGN=CENTER VALIGN=BOTTOM
WIDTH=300 HEIGHT=200>
<tr>
```

```
<td id="text_variador" VALIGN=BOTTOM ALIGN=CENTER
style="color:#FFFA00">
xxxx
```

```
</td>
```

```

</tr>
</TABLE>
<TABLE BORDER=0 VALIGN=TOP WIDTH=300 HEIGHT=400>
<tr>
<td>
</td>
<td id="text_bomba_01" style="color:#FFFA00" ALIGN=RIGHT
VALIGN=BOTTOM HEIGHT=100>
    Bomba 01
</td>
</tr>
</TABLE>
</TD>
<TD WIDTH=300 VALIGN=TOP ALIGN=CENTER>
    <TABLE BORDER=0 VALIGN=BOTTOM WIDTH=100
HEIGHT=120>
<tr>
<td id="Ultrasonico" VALIGN=BOTTOM ALIGN=CENTER
style="color:#FFFA00">
    xxxxxxxxxxxx
</td>
</tr>
</TABLE>
<TABLE BORDER=0 VALIGN=BOTTOM WIDTH=200
HEIGHT=200>
<tr>
<td VALIGN=BOTTOM>

    <div id="bar_Level_Fl_01" style="margin-bottom:46; margin-left:30px;
width:52px; height:100px; border:1px; background-color:#B2FFFF;" ></div>

</td>
</tr>
</TABLE>
<TABLE BORDER=0 VALIGN=BOTTOM WIDTH=200
HEIGHT=250>
<tr>
<td VALIGN=BOTTOM>
    <div id="bar_Level_Tank_02" style="margin-bottom:0; margin-
left:31px; width:55px; height:20px; border:1px; background-color:#B2FFFF;"
></div>

</td>
<td VALIGN=BOTTOM ALIGN=RIGHT>
    <TABLE BORDER=0 VALIGN=BOTTOM WIDTH=80
HEIGHT=80>

```

```

        <tr>
        <td id="text_state_High_tank_02" VALIGN=BOTTOM
style="color:#FFFA00" ALIGN=RIGHT>

        fffffff

        </td>
        </tr>
        </TABLE>
        <TABLE BORDER=0 VALIGN=BOTTOM WIDTH=80
HEIGHT=70>
        <tr>
        <td id="text_state_Low_tank_02" VALIGN=BOTTOM
style="color:#FFFA00" ALIGN=RIGHT>
        ddddddd

        </td>
        </tr>
        </TABLE>
        <TABLE BORDER=0 VALIGN=BOTTOM WIDTH=80 HEIGHT=25>
        <tr>
        <td VALIGN=CENTER ALIGN=RIGHT>
        </td>
        </tr>
        </TABLE>
        </td>
        </tr>
        </TABLE>

</TD>
<TD WIDTH=400 VALIGN=TOP ALIGN=RIGHT>
        <TABLE BORDER=0 ALIGN=BOTTOM WIDTH=350 HEIGHT=50>
        <tr>
        <td id="text_muy_alto_tk_02" VALIGN=BOTTOM
style="color:#FFFA00" ALIGN=RIGHT>
        AAA
        </td>
        <td id="text_alto_tk_02" VALIGN=BOTTOM style="color:#FFFA00"
ALIGN=RIGHT>
        BBB
        </td>
        <td id="text_bajo_tk_02" VALIGN=BOTTOM style="color:#FFFA00"
ALIGN=RIGHT>
        CCC
        </td>

```

```

        <td id="text_muy_bajo_tk_02" VALIGN=BOTTOM
style="color:#FFFA00" ALIGN=CENTER>
        DDD
    </td>
</tr>
</TABLE>

<TABLE BORDER=0 ALIGN=BOTTOM WIDTH=350 HEIGHT=120>
    <tr>
        <td id="text_valvula_filtro" style="color:#FFFA00" WIDTH=110
VALIGN=CENTER ALIGN=LEFT>
        valvula filtro
    </td>
    <td VALIGN=BOTTOM>
        <div style="margin-bottom:14; margin-left:20px; width:48px;
height:50px; border:1px; background-color:#B2FFFF;"
id="bar_Level_FI_02"></div>
    </td>
    </tr>
</TABLE>
<TABLE BORDER=0 WIDTH=250 HEIGHT=80>
    <tr>
        <td id="temperatura_filtro" VALIGN=BOTTOM
style="color:#FFFA00">
        0000
    </td>
    </tr>
</TABLE>
<TABLE BORDER=0 WIDTH=250 HEIGHT=30>
    <tr>
        <td id="temperatura_plato" VALIGN=BOTTOM
style="color:#FFFA00">
        xxxx
    </td>
    </tr>
</TABLE>
<TABLE BORDER=0 WIDTH=300 HEIGHT=120 >
    <tr>
        <td id="text_valvula_recirculacion" VALIGN=CENTER
style="color:#FFFA00">
        Valvula recirculacion
    </td>
    <td>

```

```
</td>
</tr>
</TABLE>
```

```
<TABLE BORDER=0 WIDTH=350 HEIGHT=200>
<tr>
<td VALIGN=BOTTOM>
<td WIDTH=250 HEIGHT=50 VALIGN=TOP>
<TABLE BORDER=0 WIDTH=100 HEIGHT=70>
<tr>
<td id="text_flujometro" HEIGHT=70 VALIGN=BOTTOM
style="color:#FFFA00">
xxxx
</td>
</tr>
<tr>
<td id="text_bomba_02" HEIGHT=110 VALIGN=BOTTOM
style="color:#FFFA00">
bomba 2
</td>
</tr>
</TABLE>
</td>
<td WIDTH=250 HEIGHT=50 VALIGN=TOP>

<TABLE BORDER=0 WIDTH=40 HEIGHT=100 >
<tr>
<td id="text_state_High_tank_03" VALIGN=CENTER
style="color:#FFFA00">
ALTO
</td>
<td>
</td>
</tr>
</TABLE>
<TABLE BORDER=0 WIDTH=40 HEIGHT=50 >
<tr>
<td id="text_state_Low_tank_03" VALIGN=CENTER
style="color:#FFFA00">
Bajo
</td>
<td>
</td>
</tr>
</TABLE>
```

```

        </TABLE>
        </td>
        <td WIDTH=240 VALIGN=BOTTOM>
        <div id="bar_Level_Tank_03" style="margin-bottom:30; margin-
left:14px; width:54px; height:100px; border:1px; background-color:#B2FFFF;"
></div>
        </td>
        </td>
        </tr>
        </TABLE>
        </TD>
    </table>

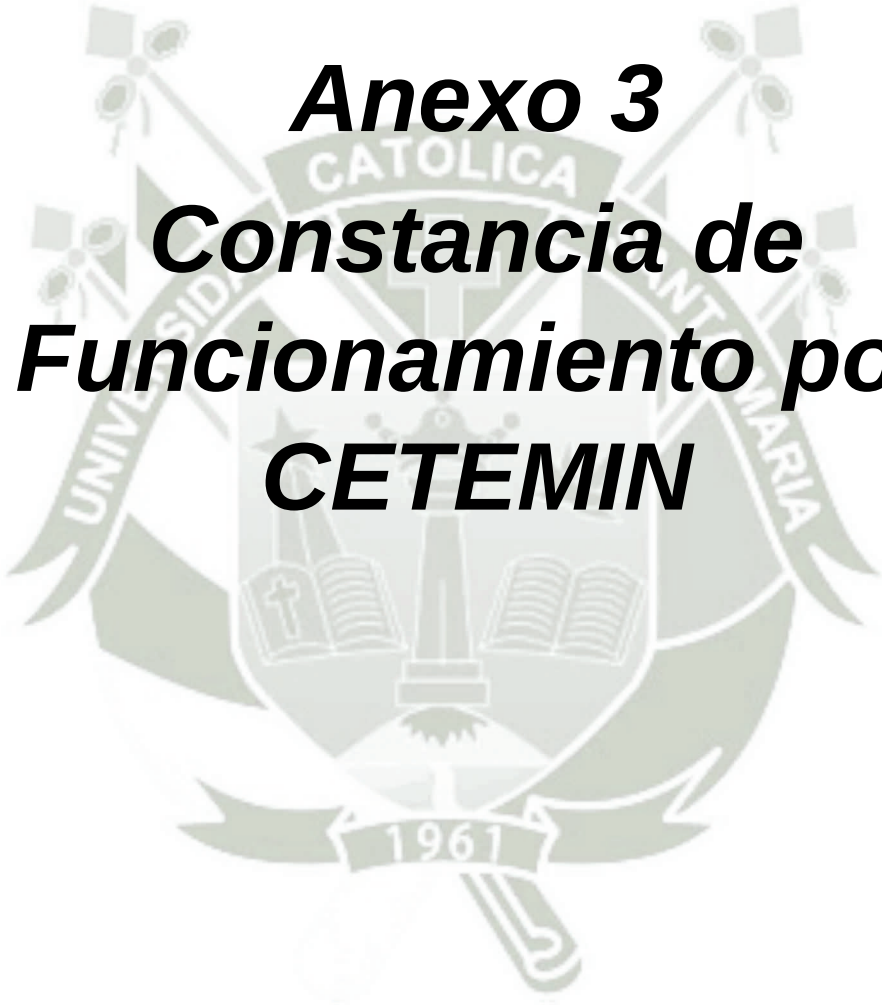
    </td>
    </tr>
    <div id="nosensor" style="visibility:hidden ; color:#FF0000">
    El sensor del Filtro 01 esta descalibrado o PLC no esta conectado <span
id="txtsensor">No hay sensor</span>
    </div>
    <div id="nosensor2" style="visibility:hidden ; color:#FF0000">
    El sensor del Filtro 02 esta descalibrado o PLC no esta conectado <span
id="txtsensor">No hay sensor</span>
    </div>
    <td id="td1" style="width:50px" VALIGN= TOP>

    <div id="divGraph" style="position:relative; z-index:99">
        <iframe src="graph.html" width="400px" height="250px">
            iframes are not supported by your browser
        </iframe>
        <iframe src="graphtemp.html" width="400px" height="250px">
            iframes are not supported by your browser
        </iframe>
    </div>

    </td>

</table>
</body>
</html>

```



Anexo 3

Constancia de Funcionamiento por CETEMIN



"Año del buen Servicio al Ciudadano"

N° 024-2017/CO

CONSTANCIA

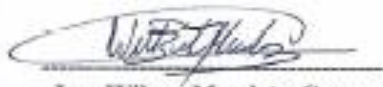
La Dirección Académica del Centro Tecnológico Minero (CETEMIN) del Distrito de Vitor de la Provincia y Departamento de Arequipa.

HACE CONSTAR:

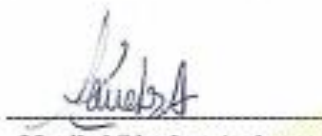
Que el señor: **ALLASI GONZALES, JONATHAN RICHARD** identificado con DNI 46319083, entrego satisfactoriamente el proyecto denominado: **"AUTOMATIZACIÓN DE PLANTA PILOTO PARA LA REUTILIZACIÓN DE AGUA POR FILTRADO Y CONDENSACIÓN; UTILIZANDO PROGRAMACIÓN PARALELA Y SECUENCIAL, MEDIANTE CONTROL FUZZY"**. Cumpliendo con su funcionamiento y objetivos.

Se extiende el presente documento para los fines que el interesado estime convenientes.

Vitor, 19 de octubre de 2017



Ing. Wilson Mendoza Ceon
Docente Académico
CETEMIN-AREQUIPA



Maribel Sánchez Anderson
Directora Académica
CETEMIN-AREQUIPA

Arequipa
Oficina Comercial: Av. Aviación 616-A, Cerro Colorado (frente a Metros)
Campus Vitor: Fundo La Munilla s/n - Vitor
Telf: (084) 755-025 / 619-750
www.cetemin.edu.pe

