



“MONITOREO DE SEÑALES DE UNA PELETIZADORA DE ALIMENTO
BALANCEADO UTILIZANDO DISPOSITIVOS IOT CON CONTROL DIFUSO
DE TEMPERATURA PARA AMBIENTES INDUSTRIALES”

TRABAJO DE TITULACIÓN

Previo a la obtención del Título de:

MAGISTER EN AUTOMATIZACIÓN Y CONTROL INDUSTRIAL

Presentado por:

Ing. Anthony Barona Espinar

Guayaquil – Ecuador

2020

AGRADECIMIENTOS

Agradezco a Dios, por ser incondicional y brindarme salud, capacidad, motivación y sobre todo perseverancia para emprender nuevos objetivos personales y profesionales día a día y cumplirlos.

A mi profesor y colega el Magister Ronald Solis, que con su paciencia y conocimientos supo solventar mis dudas y hacer que este proyecto haya sido posible de desarrollar y culminar.

A mis profesores de la MACI, porque supieron guiarnos como docentes y compartir sus conocimientos y vivencias como profesionales y como amigos haciendo que la experiencia de la maestría sea la mejor de todas en mi formación profesional.

A mis compañeros de curso y colegas que a lo largo del tiempo en la maestría se convirtieron en excelentes amigos formando un excelente grupo de estudios y sobre todo formando lazos de amistad que prevalecen en el tiempo.

DEDICATORIA

Dedico este proyecto a mi Dios. Siempre estas primero en mi vida personal y profesional y aunque no soy el mejor practicante y seguidor de tus doctrinas soy consciente de que sin ti no somos nada y contigo tenemos todo. Gracias Dios por llenarme de capacidad y actitud para cumplir un nuevo objetivo.

A mis padres Sixto y Carmen, porque desde que me trajeron al mundo supieron cobijarme bajo sus cuidados preparándome para convertirme en la excelente persona y profesional que considero ser en la actualidad. Gracias, papá, gracias, mamá por su amor incondicional y por enseñarme a ser mejor hombre cada día de mi vida. Los amo.

A ti Yanira, por haberte convertido en lo que siempre busque y hasta ahora no había encontrado, la mujer de mi vida, este logro también es para ti y para el resto de nuestras vidas. Te amo.

TRIBUNAL DE EVALUACIÓN

.....
Ph.D. Juan Carlos Avilés
SUBDECANO DE LA FIEC

.....
Ms.C. Ronald Solis
DIRECTOR DEL TRABAJO DE TITULACIÓN

.....
Ph.D. Wilton Agila
MIEMBRO PRINCIPAL DEL TRIBUNAL

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me(nos) corresponde exclusivamente; y doy(damos) mi(nuestro) consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

Ing. Anthony Barona Espinar

RESUMEN

El presente proyecto tiene como principal objetivo realizar el monitoreo de las señales de una peletizadora de alimento balanceado para camarón durante todo el tiempo de producción para que dicha información sea analizada por los operadores y supervisores de planta y mantenimiento eléctrico y poder garantizar el rendimiento del equipo y la calidad de la producción debido a que este equipo produce el pellet final que es despachado a los clientes consumidores del alimento.

Para realizar el monitoreo de las señales se emplearán dispositivos electrónicos IOT como el Arduino UNO y la Raspberry PI 3 B+, convertidores de corriente a voltaje de 4-20 mA a 0-5 V y se implementará la conexión al módulo de comunicaciones del PLC ubicado en el panel de automatización de la planta de producción donde se encuentra conectada la peletizadora a estudiar.

El proyecto consta de un sitio web sencillo y muy práctico para el acceso de los usuarios involucrados en el proceso de monitoreo. Así podrán ingresar al aplicativo con su cuenta y revisar la información registrada en la Base de Datos del sistema que es enviada por medio de los dispositivos IOT instalados en la planta de producción. Para el fin lo más importantes son los Reportes Detallados y Gráficos de variables que posee el programa.

Estos dispositivos se colocarán en el interior de una caja de control IOT para garantizar el confort del ambiente de trabajo en el interior de la caja que permita el correcto funcionamiento de los equipos a 23 °C, para regular la temperatura se utilizará un controlador difuso implementado en un Arduino UNO con una Celda de Peltier como elemento actuador y conectado a través de un transistor Mosfet lógico para comunicar con el Arduino a través de voltajes pequeños.

Los programas para el monitoreo de señales son desarrollados en lenguaje Arduino y Python, la Raspberry utiliza como Sistema Operativo Raspbian. El controlador difuso es diseñado en Matlab y la implementación del controlador difuso se realizó en lenguaje Arduino utilizando las librerías difusas para trabajar con las herramientas que corresponden a este tipo de controlador.

Se pudo obtener excelentes resultados respecto a la facilidad de monitoreo de las señales desde el sistema web implementado para el fin, los operadores pudieron experimentar el cambio en cuanto al tiempo de respuesta para obtener dicha información haciéndolo incluso desde sus casas comprobando así la efectividad de la solución y aplicando inmediatamente correctivos sencillos al proceso de peletizado debido a que no existieron mayores novedades durante el tiempo de despliegue del proyecto.

El controlador difuso funciono sin mayor inconveniente debido a que el esfuerzo que debe realizar para mantener la temperatura del ambiente de trabajo en la caja de control IOT no es tan alto, según el experimento la temperatura en el interior de la caja oscila entre 22 °C y 27 °C y el Set Point del controlador para validar el error está en 23 °C. Los resultados obtenidos a partir del muestreo fueron los adecuados para el fin del proyecto.

INDICE GENERAL

AGRADECIMIENTOS	ii
DEDICATORIA	iii
TRIBUNAL DE EVALUACIÓN	iv
DECLARACIÓN EXPRESA	v
RESUMEN	vi
ÍNDICE GENERAL	viii
ÍNDICE DE FIGURAS	x
ÍNDICE DE TABLAS	xv
INTRODUCCIÓN	xvi
CAPÍTULO 1	1
1. ANÁLISIS DEL PROBLEMA	1
1.1 Antecedentes	1
1.2 Descripción del Problema	2
1.3 Solución Propuesta	4
1.4 Objetivos	6
1.4.1 Objetivos Generales	6
1.4.2 Objetivos Específicos	6
1.5 Metodología	7
CAPÍTULO 2	9
2. ESTADO DEL ARTE	9
2.1 Introducción	9
2.2 La industria del alimento balanceado en Ecuador	10
2.3 La peletización en la producción de alimento balanceado	12
2.4 El Internet de las Cosas (IOT)	18
2.5 Raspberry PI 3 B+	21
2.6 Arduino UNO	26
2.7 Celda de Peltier	29
2.8 Lógica Difusa (LD)	34
CAPÍTULO 3	46
3. DESARROLLO	46
3.1 Diseño del proceso recolector de datos de la peletizadora	46
3.1.1 Programa en Arduino UNO	50

3.1.2 Programa en Raspberry PI 3 B+	53
3.2 Diseño del portal web para presentación de datos	58
3.2.1 Diagrama Entidad-Relación de la Base de Datos	59
3.2.2 Interfaz de usuario final	65
3.3 Diseño de las comunicaciones del controlador difuso de temperatura	75
3.3.1 Definición de variables	60
3.3.2 Diseño del controlador en Matlab	84
3.3.3 Programa en Arduino UNO	91
3.3.4 Interfaz web para revisión de temperatura	96
CAPÍTULO 4	98
4. ANÁLISIS DE RESULTADOS	98
4.1 Análisis de valores de variables obtenidos de la peletizadora	98
4.2 Análisis del desempeño del controlador difuso de temperatura	105
CONCLUSIONES Y RECOMENDACIONES	110
BIBLIOGRAFÍA	113
ANEXOS	115

INDICE DE FIGURAS

Figura 2.1 Componentes de una Peletizadora	14
Figura 2.2 Acondicionador de una Peletizadora	15
Figura 2.3 Datos de una Peletizadora	16
Figura 2.4 Rodillos de una Peletizadora	17
Figura 2.5 Raspberry Pi 3 B+	23
Figura 2.6 Características de la Raspberry Pi 3 B vs. B+	24
Figura 2.7 Componentes de la Placa Arduino UNO	27
Figura 2.8 Célula Peltier	31
Figura 2.9 Diagrama del Efecto Peltier	32
Figura 2.10 Comparación entre Lógica Clásica y Lógica Difusa	36
Figura 2.11 Funciones características habituales: (a) triangular, (b) trapezoidal, (c) gaussiana y (d) sigmoideal	37
Figura 2.12 Función de pertenencia triangular	39
Figura 2.13 Función de pertenencia trapezoidal	39
Figura 2.14 Función de pertenencia Gaussiana	40
Figura 2.15 Función de pertenencia de Bell	40
Figura 2.16 Función de pertenencia Sigmoideal	41
Figura 2.17 Operaciones Difusas	41
Figura 2.18 Definición de la variable lingüística Temperatura	42
Figura 2.19 Sistema de Lógica Difusa	44
Figura 3.1 Diseño de comunicaciones del proyecto	46
Figura 3.2 Convertidor DROK de corriente al voltaje de 4-20 mA a 0-5 V	48

Figura 3.3 Conexión de los convertidores con el panel eléctrico	48
Figura 3.4 Conexión de los convertidores con el Arduino UNO	49
Figura 3.5 Definición de puertos y arreglo de variables en Arduino UNO	50
Figura 3.6 Definición de la velocidad de trabajo del puerto serial	51
Figura 3.7 Lectura, preparación y envío de datos al puerto serial	51
Figura 3.8 Proceso de lectura de variables de la peletizadora en Arduino UNO	52
Figura 3.9 Definición de librerías del programa en Python	53
Figura 3.10 Configuración de medidas de trabajo de la Raspberry	54
Figura 3.11 Lectura y envío de datos a la nube	55
Figura 3.12 Proceso de lectura de variables de la peletizadora en Raspberry	55
Figura 3.13 Configuración de recepción de variables y acceso a Base de Datos	56
Figura 3.14 Conversión de datos según escala de variables	57
Figura 3.15 Inserción de datos en la Base de Datos del Portal Web	57
Figura 3.16 Proceso de inserción de variables de la peletizadora en la nube	58
Figura 3.17 Estructura general de la Base de Datos del Portal Web	59
Figura 3.18 Tabla “pwiot_tblempresa”	60
Figura 3.19 Tabla “pwiot_tbldepartamento”	60
Figura 3.20 Tabla “pwiot_tblusuario”	61
Figura 3.21 Tabla “pwiot_tblpersona”	61
Figura 3.22 Tabla “pwiot_tbltipopersona”	62
Figura 3.23 Tabla “pwiot_tblopcion”	62
Figura 3.24 Tabla “pwiot_tbltipopersonaopcion”	62
Figura 3.25 Tabla “pwiot_tblvariables”	63
Figura 3.26 Tabla “pwiot_tblrangovaloracionvariables”	63

Figura 3.27 Tabla “pwiot_ tblregistrovariables”	64
Figura 3.28 Procedimiento Almacenado “pwiot_ sprvariables” (1)	64
Figura 3.29 Procedimiento Almacenado “pwiot_ sprvariables” (2)	64
Figura 3.30 Diagrama Entidad-Relación de la Base de Datos PwIOT	65
Figura 3.31 Ventana principal del Sistema de Monitoreo de Variables	66
Figura 3.32 Opción para la creación de nuevos usuarios	66
Figura 3.33 Opción para la recuperación de acceso al sistema	67
Figura 3.34 Opción para acceder al sistema	67
Figura 3.35 Ventana de sesión en el sistema	68
Figura 3.36 Opciones de configuraciones en el sistema	68
Figura 3.37 Opción de actualización de datos generales del usuario	69
Figura 3.38 Opción de actualización de la clave del usuario	70
Figura 3.39 Opción de cancelación del usuario del sistema	70
Figura 3.40 Opciones de consultas del sistema	71
Figura 3.41 Opción de consulta de usuarios del sistema	71
Figura 3.42 Opción de consulta de empresas del sistema	71
Figura 3.43 Opción de consulta de variables del sistema	72
Figura 3.44 Opciones de reportes del sistema	72
Figura 3.45 Generación de reportes detallados del sistema	73
Figura 3.46 Generación de reportes gráficos del sistema	73
Figura 3.47 Muestra de reporte detallado del sistema	74
Figura 3.48 Muestra de reporte gráfico del sistema	74
Figura 3.49 Sensor de temperatura LM35	75
Figura 3.50 Programa en Arduino para lectura de temperatura ambiente (1)	76

Figura 3.51 Programa en Arduino para lectura de temperatura ambiente (2)	77
Figura 3.52 Proceso de lectura de temperatura en la caja de control IOT	77
Figura 3.53 Programa en Python para la lectura de la temperatura ambiente	78
Figura 3.54 Proceso de envío de temperatura de la caja de control IOT a la nube	78
Figura 3.55 Programa en PHP para el registro de la temperatura ambiente	79
Figura 3.56 Proceso de registro de temperatura de la caja IOT en la BD	79
Figura 3.57 Celda de Peltier TEC1-12706	80
Figura 3.58 Transistor Mosfet lógico IRLB8748PBF	81
Figura 3.59 Características del Transistor Mosfet lógico IRLB8748PBF	82
Figura 3.60 Circuito del controlador difuso de temperatura	83
Figura 3.61 Controlador Fuzzy en Matlab	85
Figura 3.62 Función de Membresía NG (Error_Temperatura)	85
Figura 3.63 Función de Membresía NP (Error_Temperatura)	86
Figura 3.64 Función de Membresía Z (Error_Temperatura)	86
Figura 3.65 Función de Membresía PP (Error_Temperatura)	87
Figura 3.66 Función de Membresía PG (Error_Temperatura)	87
Figura 3.67 Función de Membresía NULO (Frecuencia_Peltier)	88
Figura 3.68 Función de Membresía BAJO (Frecuencia_Peltier)	88
Figura 3.69 Función de Membresía MEDIO (Frecuencia_Peltier)	89
Figura 3.70 Función de Membresía ALTO (Frecuencia_Peltier)	89
Figura 3.71 Reglas del controlador difuso de temperatura (1)	90
Figura 3.72 Reglas del controlador difuso de temperatura (2)	90
Figura 3.73 Defusificación del controlador por el método del centroide	91
Figura 3.74 Programa del controlador difuso de temperatura (1)	92

Figura 3.75 Programa del controlador difuso de temperatura (2)	93
Figura 3.76 Programa del controlador difuso de temperatura (3)	93
Figura 3.77 Programa del controlador difuso de temperatura (4)	94
Figura 3.78 Programa del controlador difuso de temperatura (5)	94
Figura 3.79 Programa del controlador difuso de temperatura (6)	95
Figura 3.80 Proceso del controlador difuso de temperatura	95
Figura 3.81 Reporte detallado de temperatura	96
Figura 3.82 Reporte gráfico de temperatura	97
Figura 4.1 Reporte detallado de variables de la peletizadora (1)	99
Figura 4.2 Reporte detallado de variables de la peletizadora (2)	100
Figura 4.3 Reporte detallado de variables de la peletizadora (3)	100
Figura 4.4 Reporte detallado de variables de la peletizadora (4)	101
Figura 4.5 Reporte detallado de variables de la peletizadora (5)	101
Figura 4.6 Reporte gráfico de variables de la peletizadora	102
Figura 4.7 Reporte detallado de la Temperatura Ambiente (1)	103
Figura 4.8 Reporte detallado de la Temperatura Ambiente (2)	104
Figura 4.9 Reporte gráfico de la Temperatura Ambiente	105
Figura 4.10 Rendimiento del controlador difuso de temperatura	108
Figura 4.11 Rendimiento del controlador difuso de temperatura (Promedio)	109

INDICE DE TABLAS

Tabla 1: Áreas y conceptos de la Industria 4.0	19
Tabla 2: Relación de pines análogos de Arduino UNO y variables de la Peletizadora	47
Tabla 3: Definición de variable lingüística de entrada del controlador difuso	84
Tabla 4: Definición de variable lingüística de salida del controlador difuso	84
Tabla 5: Reglas del controlador difuso	84
Tabla 6: Relación del Ciclo de Trabajo PWM y el voltaje de salida del Arduino UNO	106

INTRODUCCIÓN

En este proyecto se plantea el diseño de un esquema de monitoreo de señales de una peletizadora que forma parte de una planta de producción de alimento balanceado para camarón, la solución planteada a la necesidad se implementará con dispositivos IOT como la placa Arduino UNO y la Raspberry PI 3 B+ en combinación con acceso a un hosting vía Internet para registrar la información recolectada y a su vez facilitando como herramienta para consultas un sistema web con los reportes de las variables monitoreadas. Paralelamente se ha realizado un controlador difuso para gestionar la temperatura ambiente en el interior de la caja de control que alberga a los dispositivos IOT del proyecto y así mantener el ambiente de confort para el trabajo.

El Capítulo 1 del documento contiene la fundamentación del problema, se indica los objetivos de la investigación y el estado actual de la situación referente a la lectura de las variables a estudiar y justamente se comprenderá el motivo por el cual se dese mejorar el actual proceso de monitoreo.

El Capítulo 2 contiene la fundamentación teórica relevante y referente a los elementos involucrados en el problema y la solución de este, de esta forma se obtiene una idea clara y formal acerca de los recursos involucrados en el proyecto y que se describen el Capítulo 3 donde se exponen los detalles de la solución aplicada para solucionar el problema planteado y que se indicó al inicio de esta introducción.

El Capítulo 4 contiene el análisis de los resultados obtenidos producto de la implementación de la solución tanto a nivel de monitoreo de variables como a nivel del controlador difuso de temperatura.

Finalmente, el documento se cierra con las conclusiones y recomendaciones que el autor refiere en base a la experiencia durante el proceso de desarrollo de la investigación.

CAPÍTULO 1

1. ANÁLISIS DEL PROBLEMA.

1.1 Antecedentes

La industria camaronesa en el Ecuador se ha convertido actualmente en la segunda fuente generadora de ingresos por exportaciones en el país siendo en los últimos años la más estable y superando con creces en algunos meses a las exportaciones de banano y petróleo quienes siempre fueron las industrias más importantes a considerar dentro de la balanza comercial ecuatoriana.

Debido al crecimiento de la industria camaronesa se incrementó la producción y el comercio de los insumos que complementan a este sector productivo siendo uno de los elementos más importantes o en muchos de los casos considerado el más importante entre los productores camaroneseros y el cuál es el alimento balanceado que es utilizado en las diferentes dietas recomendadas por los asesores técnicos y administradores de las instalaciones equipadas para ejecutar el proceso de siembra, seguimiento y cosecha del camarón.

La industria del alimento balanceado para camarón al igual que la producción camaronesa ha alcanzado niveles altos de desarrollo en el país sobre todo en los últimos 10 años. En este lapso de tiempo las empresas dedicadas a esta actividad han realizado grandes inversiones en recursos tecnológicos con el objetivo de generar grandes cantidades de alimento de alta calidad que garantice excelentes resultados en la producción de camarón, esto implica simplificar grandes procesos que anteriormente se realizaban manualmente para ser más efectivos a la hora de cumplir con las metas planteadas respondiendo a la vez con las grandes exigencias requeridas por los clientes que a su vez también han realizado grandes inversiones en la obtención de recursos técnicos-científicos y tecnológicos todos orientados a la obtención de producto de primer nivel que luego será exportado a mercados muy exigentes como Estados Unidos, Europa y China.

Sin embargo, a pesar de las grandes inversiones realizadas en automatización con el objetivo de simplificar procesos manuales y ser más rápidos y efectivos a la hora de producir balanceado, existen brechas por cubrir que aún en la actualidad se siguen manejando con poca eficiencia y que forman parte del proceso de transformación de la industria actual a la denominada Industria 4.0 y que en breves palabras se refiere a la conexión de todos los equipos industriales que componen las grandes líneas de producción con la red de datos que poseen las mismas organizaciones dedicadas a esta actividad para facilitar el acceso a la información en tiempo real al personal administrativo y técnico que requiere la misma para la toma de decisiones.

El principal equipo que forma parte de las líneas de producción de alimento balanceado en todas las organizaciones dedicadas a esta actividad industrial se denomina "PELETIZADORA" y su función se explica brevemente como el equipo que convierte las grandes mezclas de ingredientes de las recetas de alimento como lo son la Soya, Trigo, Harinas de Pescado, Aceites y otros en el grano que luego es comercializado entre las empresas camaroneras y que se conoce en el sector como "pellet".

El proceso de transformación de las mezclas de ingredientes en el pellet final es bastante complejo y se requiere que el equipo que ejecuta esta tarea cumpla con la configuración adecuada de las variables que componen al equipo a través de sus partes físicas las mismas que son: Amperaje, Frecuencia de alimentación, Presión y Temperatura.

1.2 Descripción del Problema

En el punto anterior indicamos el crecimiento de la industria camaronera en el Ecuador y el impacto que ha generado en la producción de insumos para sostener su desarrollo siendo el más importante el alimento balanceado que constituyen las dietas especiales que los biólogos marinos recomiendan se cumplan en las diferentes organizaciones camaroneras para alcanzar los estándares de calidad que el mercado internacional y local exige.

El equipo principal dentro de la línea de producción de alimento balanceado es la Peletizadora, este equipo convierte las grandes mezclas de ingredientes sólidos y

líquidos en el grano o también conocido como pellet que luego será consumido por los camarones sembrados en las haciendas camaroneras.

La peletizadora debe trabajar cumpliendo con configuraciones adecuadas en sus parámetros para la correcta producción del pellet, las señales de las variables que se deben monitorear constantemente son: El Amperaje, Frecuencia de Alimentación, Presión y Temperatura.

Se puede apreciar que la peletizadora es un equipo crítico dentro del proceso de producción de balanceado, si no se hace un seguimiento acorde a las necesidades de la organización se pueden perder grandes cantidades de producto lo que se traduce en pérdida de dinero y a la vez se arriesga la calidad del producto y la imagen de la empresa generando desconfianza entre los clientes por alimento con problemas que puede generar inconvenientes en la alimentación del camarón.

El problema que se plantea en esta tesis está relacionado justamente con el seguimiento para el análisis de las variables antes mencionadas, como se indicó al inicio, se requiere que el personal técnico de supervisión de la producción revise constantemente los valores de los parámetros del equipo en dos situaciones, sea como medida de prevención para garantizar el rendimiento correcto del equipo y la calidad de la mezcla generada previamente, así como también en los momentos de generación de informes posterior a medidas correctivas tomadas por problemas generados durante el proceso de producción y que estén relacionados principalmente con el equipo.

Sin embargo, esta información muchas veces se puede analizar solamente a través de bases de datos que contienen la información almacenada con frecuencia de grabado de cada segundo, lo que hace que la tarea de seguimiento en tiempo de corrección de errores se vuelva muy compleja de realizar, de igual forma acceder a la información en tiempo real o el solo hecho de necesitar acceder requiere de autorizaciones mediante licencias de parte de los propietarios del software de automatización en donde está la base de datos que aloja esta información. La mayor parte de las ocasiones los proveedores de la tecnología de automatización de las grandes industrias del Ecuador son extranjeras y están asentadas en Europa o Asia

lo que dificulta la coordinación del soporte técnico y vuelve muy lento el proceso de acceso a la información.

Toda la logística indicada en el párrafo anterior relacionada a la obtención de la información de las variables de la peletizadora y que está involucrada con los permisos del propietario para acceder a estos datos más la coordinación del soporte técnico extranjero considerando las diferencias horarias entre continentes y la criticidad que conlleva el obtener la información a destiempo representan el principal problema a resolverse en este proyecto.

Como aporte complementario a la información previa es importante indicar que las implementaciones que se realicen en este proyecto para obtener información de la peletizadora se deberán ejecutar en el interior de la planta de producción donde se encuentran los equipos que conforman las líneas de trabajo y esto se traduce a que existirá de cierta forma un ambiente de trabajo con temperaturas altas para los posibles equipos que componen la implementación de la solución, de esta forma como parte del planteamiento del problema se debe tener en cuenta el entorno de trabajo al cuál se someterá la solución aplicada.

1.3 Solución Propuesta

La peletizadora está compuesta por varios equipos que ejecutan el proceso de conversión de la mezcla de macro y micro ingredientes en el grano o pellet del alimento balanceado que posteriormente es comercializado a los camaroneros para la alimentación del camarón en el proceso de siembra y crecimiento de este.

Para garantizar una producción acorde a los estándares de calidad se debe monitorear en tiempo real o en tiempo de corrección de errores las variables relacionadas al equipo: Amperaje de motores, Frecuencia de Alimentación de la mezcla, Presión y Temperatura.

La Peletizadora se encuentra conectada a los equipos de las líneas de producción a través de un PLC que controla todo el proceso automático de la planta.

La solución planteada al problema de seguimiento de la información generada por las variables de la peletizadora consiste en la implementación de un sistema de lectura de señales utilizando dispositivos IOT como Raspberry PI 3 B+ y Arduino UNO

conectados al panel del PLC a través de módulos de conversión de señales de 4-20 mA a 0-5 V con el objetivo de enviar la información receptada en tiempo real desde el equipo hacia la nube de datos que luego podrá ser visualizada en un portal web acondicionado para la presentación de datos de forma detallada o gráfica y a la cuál tendrán acceso los usuarios encargados de la administración de la peletizadora.

Además, considerando el entorno de trabajo en el que se encontrarán los equipos de monitoreo de las señales de la peletizadora y que consiste en un entorno con temperaturas relativamente altas debido a que forman parte de un ambiente industrial y a esto se debe indicar que los equipos IOT considerados como herramientas para la lectura de datos deben trabajar en un ambiente de confort a temperatura de un centro de datos, se plantea como complemento de la solución en este proyecto la implementación de un controlador difuso de temperatura desplegado a través de una Celda de Peltier en combinación con un sistema de lectura de temperatura ambiente que también enviara este dato a la nube a través de un equipo Raspberry PI 3 B+ y un Arduino UNO.

Todos los equipos IOT y complementarios utilizados para la lectura de datos de la peletizadora serán colocados al interior de una caja de control junto con los dispositivos para el levantamiento del controlador de difuso que actuará sobre la temperatura al interior de la caja.

La lectura de datos de la peletizadora se realizará a través del Arduino UNO utilizando un programa que trabajará con los puertos analógicos de la placa, estos puertos estarán conectados a los convertidores de señal de 4-20 mA a 0-5 V y estos a su vez toman datos desde el módulo de señales del PLC conectado en el panel de trabajo de la línea de producción. El Arduino UNO estará conectado a la placa Raspberry PI 3 B+ a través del puerto serial USB que ambos dispositivos poseen y el Raspberry leerá los datos del puerto serial para mediante programación enviarlos a la nube (hosting) donde serán almacenados en la base de datos del portal web al que se accederán los usuarios finales para las respectivas revisiones de la información.

El software utilizado para el desarrollo del código fuente lector de datos en Arduino es el IDE Genuino 1.8.1.

La Raspberry PI 3B+ debido a ser una minicomputadora utiliza la versión de Linux para Raspberry conocida como Raspbian y el programa fuente desarrollado para la lectura de datos del puerto serial y el posterior envío a la nube está basado en Python.

El portal web estará desarrollado en la versión de XAMPP 7.3.0 y utiliza la versión de PHP 7.3.0, Apache versión 2.4.37 y base de datos MariaDB 10.1.37.

El controlador difuso será implementado en el Arduino UNO a través de las librerías para Lógica Difusa del IDE Genuino de Arduino 1.8.1.

El análisis del funcionamiento del controlador se realizará en Matlab utilizando la herramienta "Fuzzy" a partir de los datos recolectados de la implementación en el sitio de trabajo.

Para la lectura de la temperatura del ambiente al interior de la caja de control se utilizará el sensor de temperatura LM35.

Al igual que en la lectura de datos de la peletizadora, los valores de la temperatura en la caja se enviarán a la nube (hosting) a través de un programa levantado en la minicomputadora Raspberry PI 3 B+ desarrollado en Python y ejecutado sobre Raspbian.

1.4 Objetivos

1.4.1 Objetivos Generales

- a. Implementar un sistema de monitoreo de señales de las variables principales de una peletizadora de producción de alimento balanceado para camarón y un controlador difuso de temperatura para el ambiente de trabajo de los equipos de monitoreo.

1.4.2 Objetivos Específicos

- a. Acondicionar los valores de las señales de la peletizadora que serán monitoreadas para facilitar la lectura de valores del usuario final.
- b. implementar los programas que se utilizarán en el proceso de monitoreo de las señales de la peletizadora.

- c. Implementar el portal web para la presentación de los valores de las señales receptados durante el proceso de monitoreo de la peletizadora.
- d. Instalar la caja de control para alojar los elementos de monitoreo de las señales de la peletizadora y el sistema de control difuso de temperatura.
- e. Desarrollar los programas que se utilizarán en el proceso de implementación del controlador difuso de temperatura.
- f. Realizar las conclusiones y recomendaciones del proyecto a partir de las experiencias de los usuarios en el uso del portal web.

1.5 Metodología

Para la implementación del proyecto propuesto en este documento se recomienda la utilización de los siguientes métodos tanto en su fase de análisis del problema y planteamiento de la solución como en la fase de implementación de la solución:

Análisis-Síntesis: Este método de investigación muy conocido permite descomponer procesos o sistemas sin importar su complejidad en sus partes más elementales, de esta forma el investigador podrá determinar con mayor claridad y precisión todas las características del proceso o sistema estudiado.

Hipotético-Deductivo: Este método de investigación se utiliza por parte de los investigadores para establecer la estructura de investigación basada en las fases de Planteamiento del Problema, Planteamiento de la Solución, Delimitación, entre otros puntos muy conocidos y necesarios para establecer un proceso investigativo.

La observación: Este método se utilizará para realizar el levantamiento de información en campo (planta), así se podrá conocer el funcionamiento de la peletizadora y el impacto que genera el comportamiento de las variables estudiadas en la producción del pellet. Además, se podrá conocer el entorno de trabajo de los equipos involucrados en el proceso de implementación de la solución recomendada.

La medición: Este método se utilizará para establecer el proceso de lectura de señales de la peletizadora a través de los equipos de IOT y la preparación de las escalas de interpretación de los valores obtenidos, así se facilitará la presentación de los datos al usuario final. También permitirá conocer los valores de las temperaturas

de trabajo del entorno donde se aplicará la solución, lo que aportará para el desarrollo del controlador difuso en la caja de control de los equipos.

La experimentación: Este método se utilizará comprobar los resultados obtenidos en el proceso de lectura e interpretación de señales de las variables de la peletizadora y la evaluación del controlador difuso de temperatura que ayudará a mantener el ambiente de confort en la caja de control lo que garantizará el rendimiento correcto de los equipos de medición.

CAPÍTULO 2

2. ESTADO DEL ARTE.

2.1 Introducción

El presente capítulo tiene como objetivo exponer la fundamentación teórica de la investigación planteada.

Ahora conocemos el problema a resolver que consiste en la falta de acceso a la información de los valores de trabajo de los parámetros de una peletizadora de alimento balanceado sea en tiempo real o luego de la existencia de un problema crítico en el proceso de producción.

Para resolver el problema de plantea la implementación de un sistema de lectura de datos desde el panel de control del PLC donde se encuentra conectada la peletizadora y que estará constituido por elementos IOT como la placa Arduino UNO y la minicomputadora Raspberry PI 3 B+ los mismos que tomaran datos a través de conversores de 4-20 mA a 0-5 V y enviaran la información en tiempo real hacia la nube (hosting) para que posteriormente los usuarios finales supervisores del proceso puedan acceder a la información.

Adicional a lo indicado todos los elementos se ubicarán en una caja de control para garantizar el cuidado de los equipos dentro del ambiente de trabajo donde se encuentra ubicado el panel eléctrico (planta) y para garantizar el ambiente de confort que requieren los equipos de lectura de datos se implementará un controlador difuso de temperatura a partir de una Celda de Peltier y un sistema de lectura de datos, ambos procesos se desarrollaran sobre la placa Arduino UNO y se complementaran con una minicomputadora Raspberry PI 3 B+ para el envío de la información del ambiente a la nube.

Cómo lo indicamos previamente este capítulo explicará la fundamentación teórica del problema de investigación planteado y los diferentes elementos que constituyen la solución propuesta al problema, el desarrollo de estos conceptos, definiciones y métodos los veremos a continuación.

2.2 La industria del alimento balanceado en el Ecuador

En el Ecuador la industria de alimento balanceado se originó a partir de la necesidad de satisfacer los requerimientos nutricionales sugeridos por los investigadores para mejorar la calidad de las carnes de los animales como aves, ganado vacuno y porcino para el consumo local y de exportación.

Esta necesidad de obtener los mejores estándares de calidad propuestos en el sector de la producción de animales conlleva a que se realicen diferentes tipos de dietas según el tipo de animal a alimentar.

Anteriormente el país no era autosuficiente en la producción de alimento balanceado y se realizaban importaciones del producto principalmente de Estados Unidos y Panamá. Actualmente el país posee una excelente infraestructura en esta industria generando la cantidad necesaria de alimento para el consumo local y siendo en los últimos años uno de los principales exportadores de balanceado para camarón trabajando en gran medida con países de Centroamérica.

Según los antecedentes, en el país la industria camaronera tuvo sus inicios en la década del sesenta e inicios del setenta, esto generó que a partir de los años ochenta el país alcanzara una excelente infraestructura en la industria del alimento balanceado para satisfacer las necesidades de este sector de la producción.

Sin embargo, en el año 1998 la macha blanca causó grandes estragos en el sector camaronero generando un alto impacto en la producción de alimento debido a que en épocas de crisis en la industria camaronera el principal insumo afectado siempre es el balanceado por ser el de más alto consumo y el más costoso. A partir de la recuperación del sector se presentaron grandes mejorías en la industria de alimento y los principales fabricantes realizaron grandes inversiones en tecnología para automatizar y simplificar los procesos de producción de sus plantas.

Con el ingreso de la tecnología a gran escala en la industria de alimento balanceado se incrementó la capacidad de respuesta a las necesidades del sector camaronero quienes a su vez aumentaron su producción generando hasta ahora que el camarón se convierta en el segundo recurso más importante de exportación del país y en muchas ocasiones siendo el primero dentro de un año de trabajo.

Es conocido en la industria alimenticia del camarón que el grano producido por las organizaciones dedicadas a esta tarea se puede generar en dos presentaciones: peletizado y extrusado.

Básicamente el proceso de peletización se realiza con vapor y alta presión. Su principal característica es que es un alimento sumergible. El camarón necesita que el alimento realice este proceso de sumergimiento en el agua para que pueda ser digerido por dicho animal. [1]

En el caso del proceso de extrusión (aplicando específicamente al caso de peces) se añade aceite a ingredientes secos y esta mezcla húmeda se extruye a alta temperatura y presión. Las partículas que finalmente salen de la extrusora dan al pellet un aspecto poroso y un menor peso por volumen con respecto al alimento compuesto. Estas características otorgan al alimento extruídos dos ventajas principales: [1]

- a) es un alimento que flota, con lo cual los peces lo comen más fácilmente.
- b) Permite incluir mayor porcentaje de aceite de pescado como truchas, que se dan en aguas frías. [1]

Debido a que existe gran variedad de requerimientos en el mercado del camarón ecuatoriano y dependiendo de las diferentes recomendaciones de los expertos o las necesidades nutricionales que los productores de camarón requieran para su producción existe gran variedad de productos balanceados con diferentes proteínas, siendo las más comercializadas: 22%, 25%, 28% y 35%. Estos porcentajes varían entre las condiciones ya mencionadas también por la etapa de crecimiento del camarón.

El proceso de producción de alimento balanceado para camarón se divide en las siguientes etapas:

- Compra de Materia Prima.
- Análisis Organoléptico y bromatológico.
- Formulación.
- Pesaje de Macros y Micros.
- Molienda.

- Mezclado.
- Peletizado.
- Enfundado.
- Etiquetado.
- Sellado.
- Análisis Bromatológico. [1]

La tecnificación obliga a emprender nuevos caminos, se necesita profesionales capacitados para dirigir procesos automáticos a gran escala, personal técnico especialista en nutrición y el equipo de investigadores que sugiera mejoras continuas en los procesos de las plantas con el objetivo de generar siempre un producto de alta calidad.

2.3 La peletización en la producción de alimento balanceado

La peletización es un proceso crítico dentro de la producción de alimento balanceado para camarón, es prácticamente la fase de transformación de toda la materia prima mezclada en el grano o producto terminado que se comercializará entre los consumidores de alimento en el sector camaronero.

El proceso de peletización consiste en la compresión de cualquier masa de alimentos previamente acondicionada, para obtener el pellet se requiere que aquella masa de alimentos circule a través de la prensa peletizadora y por medio de la acción de los rodillos la misma atraviese las perforaciones del dado o matriz.

Se puede definir al proceso de peletización desde el punto de vista mecánico como una tarea sencilla donde se puede transformar la mezcla de alimento previamente acondicionada en el pellet con la capacidad de ajustar el largo o ancho del mismo según las diferentes necesidades del mismo fabricante.

El proceso de peletizado mucho depende del operador de la peletizadora, la experiencia de este determinará la calidad del pellet final, dicha experiencia radica en el conocimiento que se tenga principalmente del tratamiento que se deba dar a las materias primas que participan en el proceso debido a que las materias primas presentan diferentes características en su composición y se debe conocer cuales

presentas mayor capacidad para ser peletizadas al igual que la mezcla de las mismas para ser acondicionadas por medio del vapor.

El proceso de peletizado de los alimentos para camarones considera de vital importancia el texturizado de las materias primas previo al sometimiento del acondicionamiento y mucho más considerable aún es la humedad debido a que esta trabaja como el lubricante que facilita la acción fricción entre el rodillo y los dados de la matriz permitiendo el filtrado de la mezcla y obteniendo el pellet.

El proceso de peletizado trabaja por lo general con diferentes temperaturas a las que somete la mezcla de ingredientes en un determinado lapso. Esto la hace diferente del proceso de extrusión el cuál es otro método muy conocido de la producción de alimento para camarón. A partir de este principio muchas organizaciones dedicadas a esta labor suelen sacrificar la calidad del proceso el cual requiere de mucho análisis y seguimiento por la producción generada en toneladas/horas. Es decir que ciertos administradores de plantas prefieren obtener grandes cantidades de producto a costas de una excelente calidad esto debido a la gran demanda que existe en el mercado por la obtención del alimento en menos tiempo para llegar a lugares lejanos donde se encuentran las hectáreas de siembra y cosecha del camarón.

Existe otro error común dentro de la producción del balanceado para camarón y consiste en peletizar a bajas temperaturas, esto puede ser útil en el alimento para aves, pero para el camarón se requiere mayor tiempo y potencia de cocción con el fin de garantizar la compresión correcta de los ingredientes en el grano final.

El alimento para camarones es conocido por tener altos niveles de proteína y poca grasa en su composición, por este motivo uno de sus principales ingredientes es la Harina de Pescado pues esta posee cantidades altas de proteína. Sin embargo, a más de la dificultad natural que presenta el proceso de peletizado se debe sumar la inclusión de grasa en el proceso debido a lo indicado previamente con relación a la harina y su composición netamente proteica, pues mientras más grasa se adiciona más complejo resulta filtrar el pellet entre los dados de la peletizadora.

Para realizar el proceso de peletizado que permitirá obtener el producto terminado o pellet para la alimentación del camarón entre todas las condiciones indicadas previamente es muy importante indicar que este proceso lo realiza uno de los

principales equipos dentro de la industria alimenticia de balanceados y que se conoce como "Peletizadora".

La Peletizadora está constituida por seis elementos muy importantes para su correcto funcionamiento y estos son: el alimentador, la cámara de acondicionamiento, rodillos, dados, motor y carcasa. La parte más importante del proceso de peletizado se realiza en la cámara de peletización formada por los rodillos y el dado matriz.

La formación correcta del grano o pellet para el camarón tienen relación directa con el desempeño de las partes mecánicas y el ajuste que se necesita para mediante la presión ejercida realizar extrudir la mezcla a través del dado.

Las partes que componen a una peletizadora son las siguientes:

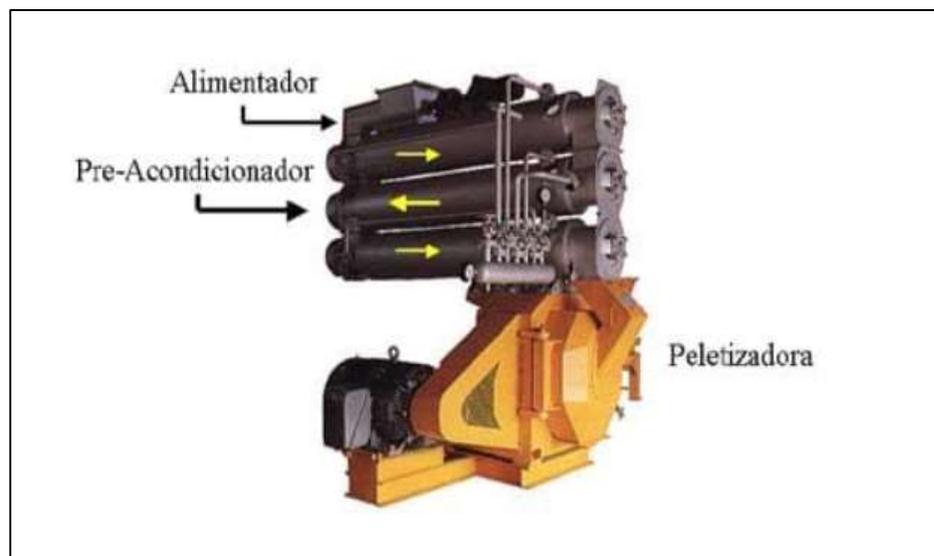


Figura 2.1 Componentes de una Peletizadora. [2]

El alimentador es un sinfín que procesa la mezcla de la tolva del acondicionador. Las espas de gusano tienen un ángulo de trabajo el cual debe estar preparado correctamente para proveer de la mezcla constantemente y sin causar mayores movimientos de fluctuaciones hacia el acondicionador. El alimentador también actúa como sellador evitando que el vapor del acondicionador escape por la vía menos protegida. Es muy necesario que el acondicionador y alimentador entreguen la mezcla de manera uniforme, de no ser así las perturbaciones en el flujo de la mezcla pueden

ocasionar capas gruesas en la pista y por ende fallas en el trabajo del rodillo y atoramiento en el equipo

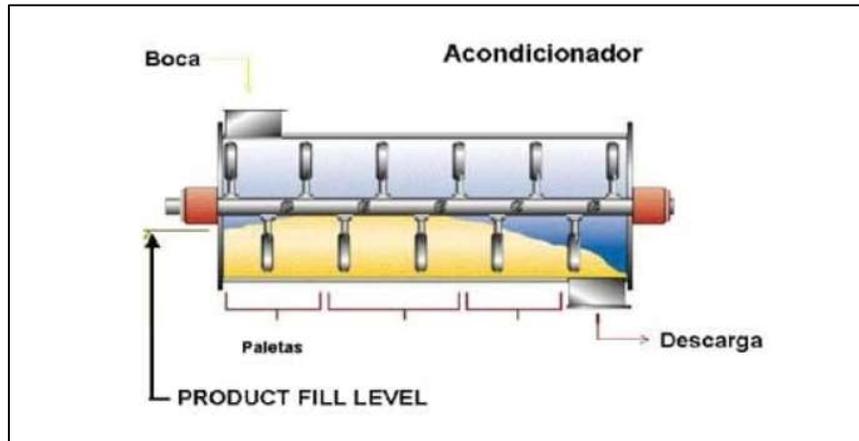


Figura 2.2 Acondicionador de una Peletizadora. [2]

El acondicionador juega un papel extremadamente importante en la estabilidad final del pellet de camarón. Un acondicionador enchaquetado con inyección de vapor vivo en la mezcla aumenta la gelatinización de los almidones de la mezcla, y ayuda en el desarrollo de las propiedades funcionales de los ingredientes proteicos, como por ejemplo el gluten. La combinación de humedad, tiempo de residencia y temperatura son factores determinantes para alcanzar altas hidroestabilidades de los pellets de camarón. El acondicionador también se puede utilizar para activar aglutinantes comerciales. [2]

A continuación, se presentan las recomendaciones para producir alimentos peletizados de camarón con alta estabilidad en el agua:

- Tiempo de acondicionamiento mínimo de 90 segundos, pero es preferible 270 a 350 segundos dependiendo de los ingredientes que se incorporan en la fórmula. El tiempo de residencia se puede ajustar cambiando el ángulo de las paletas y/o disminuyendo la velocidad del acondicionador.
- Aplicar vapor saturado de 1-2 bar (15-30 psi) de presión. Los puntos de inyección de vapor deben ser a la parte inicial del acondicionador para permitir más tiempo de contacto con la mezcla.

- La temperatura de la mezcla acondicionada debe ser no menor a 90 grados centígrados.
- La humedad de la mezcla a la salida del acondicionador debe estar entre 16-18%.

A medida que la humedad incrementa se corre el riesgo de causar atascaduras en el dado. También para poder controlar la peletización a altos niveles de humedad se debe sacrificar la productividad ya que de otra manera es imposible lograr extrudir alimento en dados con relaciones de compresión como las que se usan para alimentos de camarón. [2]

La cámara de peletización está formada por los dados y rodillos. Los dados para fabricar alimentos de camarón deben ser de acero inoxidable altos en cromo. Esto se debe a que las fórmulas de camarón son bastante ácidas lo que podría causar la corrosión. Si no se puede lograr la calidad del pellet con este tipo de dados la segunda opción es utilizar dados de acero inoxidable carburizado. Por el tipo de aleación el coeficiente de fricción es mayor que el de acero al cromo lo que produce una mayor resistencia a la mezcla que se está peletizando. Con este tipo de dados se pueden esperar reducciones en productividad y mayor desgaste no solo por la mayor fricción sino también por la menor resistencia a agentes acídicos que se incluyen en las fórmulas. [2]



Figura 2.3 Dados de una Peletizadora. [2]

El grosor de los dados para el pellet de camarón debe ser entre 45 y 50 mm, este grosor se conoce como grosor o área de trabajo y se lo debe diferenciar completamente de los alivios.

Debido a la naturaleza del camarón, el alimento procesado debe ser peletizado en dados con orificios de entre 1.8 a 2.2 mm de diámetro y su compresión debe trabajar entre 18-22. La relación entre el espesor del dado (50 mm) y el diámetro del agujero (2.2 mm) se conoce relación de compresión.

La relación entre los rodillos y el dado de una peletizadora es muy importante para el y trabajo de este equipo, si el rodillo presenta inconvenientes el dado fallará. El rodillo tiene como principal objetivo brindar fuerza de resistencia y esta depende de su espesor, diámetro del orificio y coeficiente de fricción. Se recomienda como regla general para el correcto funcionamiento del dado y larga duración de este utilizar juegos de rodillos nuevos para nuevos dados.



Figura 2.4 Rodillos de una Peletizadora. [2]

Uno de los problemas más comunes que se observa en los dados de plantas en donde se fabrican alimentos para camarón es el desgaste de la pista del dado. Este desgaste se observa como un planchado que causa el bloqueo de los agujeros por donde entra la mezcla. Este fenómeno es común debido al sobre ajuste de los rodillos ya que los operadores tratan de obtener mayor tracción posible con la creencia que de esta manera van a lograr mayores producciones. Lamentablemente esta práctica produce efectos contrarios ya que al bloquear la entrada de los orificios la productividad y la vida del dado disminuyen considerablemente. [2]

Luego del proceso de peletizado los granos entran a la fase de post-acondicionamiento y que consiste principalmente en retener los pellets calientes por más de cinco minutos en el post-acondicionador. La temperatura a la que se los mantiene en esta cámara esta entre 90-100 grados centígrados. Esta fase es importante para mantener al pellet caliente por más tiempo y así logra la mayor gelatinización de los almidones.

Se complementa el proceso anterior con el proceso de rompimiento de pellets para producir migajas que permiten separar el pellet de mejor calidad de los productos iniciadores para camarones de período larval. Este proceso se ejecuta con rodillos trituradores o desmoronadores.

2.4 El Internet de las Cosas (IOT)

Para definir el término IOT orientado a la industria y conocido como IIOT que significa Internet Industrial de las Cosas, primeramente, debemos conocer más a profundidad el termino Industria 4.0 y que corresponde a un salto más en la evolución de la industria manufacturera y que tiene como objetivo crear procesos inteligentes para cumplir con el ciclo de producción de esta forma las nuevas fábricas que den este salto evolutivo serán capaces de gestionar la complejidad y fabricar productos de manera más eficiente.

Ahora bien, para poder desplegar los fundamentos de Industria 4.0 se requieren de tecnologías que permitan complementar el funcionamiento automatizado de los equipos de las plantas y la conexión entre sí para trabajar como un solo elemento comunicándose entre sí y con el exterior, esta nueva tecnología es lo que se conoce como Internet de las Cosas.

La inclusión de estos nuevos recursos tecnológicos en las empresas para implementar el despliegue de las comunicaciones entre los equipos industriales a través de los sensores de las máquinas, piezas de trabajo e incluso los sistemas informáticos permiten no solo mantener comunicación en una sola empresa sino también dependiendo de las necesidades del negocio se podría establecer comunicaciones entre varias plantas.

Las plantas inteligentes recogen y analizan mucha información de sus procesos para poder conocer acerca de las bondades y debilidades de sus productos y de esta forma poder mejorar la calidad de los procesos orientados siempre a satisfacer las necesidades de sus clientes.

Las áreas y conceptos tecnológicos clave de la Industria 4.0 son las siguientes:

Áreas	Conceptos
Internet de las Cosas (IoT, Internet of Things).	Conexión de los dispositivos a Internet.
Integración de los sistemas horizontales y verticales.	Sistemas departamentales integrados.
Sistemas Ciberfísicos y Ciberseguridad.	Unión de los sistemas físicos y digitales.
La Nube o Cloud Computing.	Almacenamiento de la información.
Big Data.	Análisis y procesamiento de los datos de la información.
Simulación.	Modelación virtual de los escenarios.
Fabricación con impresión 3D.	Diseños personalizados y complejos.
Realidad virtual.	Aprendizaje mutuo entre personas y máquinas.
Robótica.	Fabricación flexible.
Gestión de la energía.	Ahorro de consumo de energía.

Tabla 1: Áreas y conceptos de la Industria 4.0. [3]

Luego de revisar las definiciones de Industria 4.0 cuyo concepto es bastante global se puede revisar con más profundidad las cualidades del Internet de las Cosas y cuya definición trata acerca de una red global para el intercambio de información y establecimiento de comunicaciones a través de Internet, la finalidad es permitir que cualquiera de los elementos conectados en la red pueda comunicarse entre sí en cualquier momento y en cualquier lugar e incluso se los pueda controlar. El Internet desde sus inicios ha sido bastante importante en el desarrollo de las comunicaciones de usuario final, pero con la introducción de este en la industria se busca cerrar la brecha de interacción entre el mundo físico y los sistemas informáticos que rodean a los procesos de producción.

La implantación de IOT en cualquier nivel requiere de recursos que permitan el correcto despliegue de las conexiones, los recursos son los siguientes:

- Nivel de Tecnología, desafíos vinculados a la integración de los objetos de las redes inteligentes bajo fuertes restricciones de energía y medio ambiente.
- Comunicación y trabajo en red, desafíos vinculados a la seguridad de la red y flexibilidad en la prestación de servicios en todas partes.

- Nivel de Inteligencia, desafíos vinculados con la fusión de datos y descubrimiento de servicios donde los datos son recogidos por los objetos inteligentes para ser examinados y distribuidos. [3]

Internet de las Cosas tiene principalmente las siguientes características:

- El IoT hace que cualquier objeto pueda conectarse a Internet para intercambiar información y comunicarse entre sí, con una identificación inteligente, obteniendo la localización, el seguimiento, el control y la gestión de red (Tian et al., 2011).
- La fiabilidad y precisión en la entrega de la información a través de la integración de las telecomunicaciones y redes.
- El análisis inteligente (Big Data) de la gran cantidad de datos que proporcionan los objetos, los cuáles se almacenan en el Cloud Computing. [3]

El Internet de las Cosas tiene una amplia aplicación en diferentes campos de trabajo por mencionar los más relevantes: Industria, Transporte, Seguridad Nacional, Medicina, Hogares, entre otros. Sin embargo, se debe tener en consideración que es una tecnología muy compleja y que se requiere de numerosos dispositivos para ser integrados entre ellos para ello a más del excelente servicio de comunicaciones se requiere de personal técnico especializado que garantice el correcto funcionamiento y la constante evolución de los servicios desplegados.

A pesar de que esta tecnología en los últimos años se ha desarrollado a pasos agigantados, aún se requieren de muchos años más para que todos los elementos posibles de conectar en el mundo ingresen a la red global de comunicaciones que se desea finalmente establecer.

Para alcanzar el desarrollo total de esta tecnología se requiere que las siguientes áreas claves sean consideradas:

- La privacidad, gestión de identidades, seguridad y control de acceso. El IoT presenta importantes desafíos en términos de quién puede ver qué credenciales (recordando que las entidades ya no son sólo las personas, pero podría ser cualquier forma de objeto de IoT).

- La normalización e interoperabilidad. Garantizar que cada vez que se desarrolla una nueva aplicación no haya ningún problema en la conexión.
- La gran cantidad de datos. El sistema debe poseer de una infraestructura que permita el flujo de datos de millones de objetos, asegurándose de que sea utilizable para las generaciones futuras. [3]

2.5 Raspberry PI 3 B+

Unos de los elementos de nueva generación muy utilizados en la implementación de comunicaciones IOT para extraer información de diferentes entornos tecnológicos y enviar dicha información hacia algún repositorio de datos local o externo es la placa Raspberry PI, a continuación, se presentan antecedentes y el desarrollo de la placa a través de los últimos años para conocer acerca de uno de los recursos que utilizaremos en la implementación de este proyecto que será descrita más adelante.

Raspberry PI es un ordenador de placa reducida, ordenador de placa única u ordenador de placa simple (SBC) de bajo costo desarrollado en el Reino Unido por la Raspberry Pi Foundation, con el objetivo de estimular la enseñanza de informática en las escuelas. El modelo original se convirtió en más popular de lo que se esperaba, hasta incluso vendiéndose afuera del mercado objetivo para usos como robótica. No incluye periféricos (como teclado y ratón) o carcasa. [4]

Aunque no se indica expresamente si es hardware libre o con derechos de marca, en su web oficial explican que disponen de contratos de distribución y venta con dos empresas, pero al mismo tiempo cualquiera puede convertirse en revendedor o redistribuidor de las tarjetas Raspberry Pi, por lo que da a entender que es un producto con propiedad registrada, manteniendo el control de la plataforma, pero permitiendo su uso libre tanto a nivel educativo como particular. [4]

Sin embargo, la plataforma de aplicaciones es de código abierto, el sistema operativo principal se conoce como Raspbian y se trata de una versión del Debian, pero preparada para trabajar sobre Raspberry, adicional a esto se podrían utilizar otras versiones de sistemas operativos incluso existe una distribución de Windows 10 para esta placa. El equipo posee en todas sus versiones un procesador Broadcom, RAM, GPU, USB, HDMI, Ethernet, 40 pines GPIO y conector para cámara. Las versiones más actuales requieren de tarjetas SD y MicroSD.

La fundación da soporte para las descargas de las distribuciones para arquitectura ARM, Raspbian (derivada de Debian), RISC OS 5, Arch Linux ARM (derivado de Arch Linux) y Pidora (derivado de Fedora) y promueve principalmente el aprendizaje del lenguaje de programación Python. Otros lenguajes también soportados son Tiny BASIC, C, Perl y Ruby. [4]

La organización detrás de la Raspberry Pi consiste en dos brazos. Los primeros modelos fueron desarrollados por la Raspberry Pi Foundation. Después de que la Raspberry Pi 1 Modelo B fuese lanzado. La fundación creó Raspberry Pi Trading, con Eben Upton como CEO, para desarrollar el tercer modelo, el Raspberry Pi Modelo 1 B+. Raspberry Pi Trading es responsable de desarrollar la tecnología, mientras que la fundación es una caridad educativa que tiene como objetivo promocionar la enseñanza de informática en escuelas y países en desarrollo. [4]

De acuerdo con la Raspberry Pi Foundation, más de cinco millones de Raspberry Pi fueron vendidas en febrero 2015, haciéndola la computadora británica mejor vendida. En noviembre de 2016 vendieron 11 millones de unidades y 12.5 millones en marzo de 2017, haciéndolo el tercer mejor vendido “computador de propósito general”. En julio de 2017, las ventas alcanzaron cerca de 15 millones. En marzo de 2018, las ventas alcanzaron 19 millones. [4]

La mayoría de Raspberry Pis son hechas en una fábrica de Sony en Pencoed, Wales; algunas son hechas en países asiáticos como China o Japón. [4]

Los modelos lanzados a la fecha de Raspberry Pi son los siguientes:

- Raspberry Pi 1 modelo A (descontinuada), año 2012.
- Raspberry Pi 1 modelo B (descontinuada), año 2012.
- Raspberry Pi 2 modelo B, año 2014.
- Raspberry Pi 3 modelo B, año 2016.
- Raspberry Pi 3 modelo B+, año 2018.
- Raspberry Pi 3 modelo A+, año 2018.
- Raspberry Pi 4 modelo B, año 2019. [4]

La versión de la placa Raspberry utilizada en la solución de este proyecto será la versión Pi 3 B+, esta placa surgió en el año 2018 y respecto a su antecesora la

principal diferencia radica en que consiste en una CPU “acelerada”, soporte Wifi de doble banda y también soporte Gigabit Ethernet. Esas mejoras proporcionan mayores prestaciones con la ventaja de que continúan siendo baratas.



Figura 2.5 Raspberry Pi 3 B+. [5]

La placa que contiene los diferentes circuitos, chips y conectores es prácticamente idéntica a su antecesora, sin embargo, debido a la evolución de los diseños de estos equipos existen cambios significativos.

Uno de los principales cambios o tal vez el más relevante consiste en la ubicación de un disipador por encima del SoC, este dispositivo permite regular las temperaturas que el CPU genera en gran medida debido a su nueva capacidad de trabajo.

Existe además una placa de metal que sirve como protección adicional del chip de la conexión inalámbrica (Wifi y Bluetooth). En la protección se presenta el logotipo en forma de frambuesa de la Raspberry PI, el célebre logo de la marca que no podría dejar de estar grabado en esta placa.

La ubicación de los conectores es similar a la del modelo predecesor y se mantienen los cuatro puertos USB 2.0, el puerto RJ45 Ethernet, la toma de auriculares, el

conector HDMI, el puerto MicroUSB para la alimentación o el ya clásico puerto GPIO. Las diferencias, eso sí, vienen en el hardware. [5]

A continuación, veremos la comparación de las características y precios entre las placas Raspberry Pi 3 B+ y su inmediata antecesora:

	RASPBERRY PI 3 MODEL B	RASPBERRY PI 3 MODEL B+
PROCESADOR	Broadcom BCM2837, Cortex-A53 (ARMv8) 64-bit SoC	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC
FRECUENCIA DE RELOJ	1,2 GHz	1,4 GHz
GPU	VideoCore IV 400 MHz	
MEMORIA	1GB LPDDR2 SDRAM	1GB LPDDR2 SDRAM
CONECTIVIDAD INALÁMBRICA	2.4GHz IEEE 802.11.b/g/n Bluetooth 4.1	2.4GHz / 5GHz IEEE 802.11.b/g/n/ac Bluetooth 4.2, BLE
CONECTIVIDAD DE RED	Fast Ethernet 10/100 Gbps	Gigabit Ethernet over USB 2.0 (300 Mbps de máximo teórico)
PUERTOS	GPIO 40 pines HDMI 4 x USB 2.0 CSI (cámara Raspberry Pi) DSI (pantalla tácil) Toma auriculares / video compuesto Micro SD Micro USB (alimentación)	GPIO 40 pines HDMI 4 x USB 2.0 CSI (cámara Raspberry Pi) DSI (pantalla tácil) Toma auriculares / video compuesto Micro SD Micro USB (alimentación) Power-over-Ethernet (PoE)
FECHA DE LANZAMIENTO	29/2/2016	14/3/2018
PRECIO	33,95 euros	39,75 euros

Figura 2.6 Características de la Raspberry Pi 3 B vs. B+. [5]

Los cambios que la Fundación Raspberry Pi ha realizado en esta nueva versión de su pequeño miniPC son muy interesantes, y más que los referentes a su potencia llaman la atención los que afectan a su conectividad. [5]

El procesador, por ejemplo, es una iteración mínima del Broadcom BCM2837 que llega con algo de overclocking: pasamos de una frecuencia de 1,2 GHz a una de 1,4 GHz para sus cuatro núcleos, lo que sin duda ayudará a obtener mejores rendimientos en muchas de las tareas que le propongamos a este miniPC. [5]

No hay cambios en la GPU, la cantidad de memoria RAM o, lamentablemente, los puertos USB. Eso significa que de momento no contamos con puertos USB 3.0, algo que quizás sea una de las novedades de esa hipotética y futura Raspberry Pi 4. [5]

Refiriéndonos al disipador de la CPU, esto ayuda a suavizar las temperaturas del procesador al llevarlo a su máxima capacidad de trabajo, además, disminuye las operaciones de throttling que a su vez reducen la frecuencia de reloj al detectarse grandes cambios en el comportamiento de la temperatura. Estas cualidades permiten que el microcomputador pueda trabajar de manera constante con las altas frecuencias y es sin duda una excelente mejora referente al modelo anterior.

Los cambios más relevantes y llamativos están aplicados a la conectividad. Las versiones modernas de Raspberry Pi 3 Model B+ poseen soporte Wifi 802.11ac de doble banda, así podremos conectarnos a redes con bandas de trabajo de 2.4 y 5 GHz. De esta manera se podrá aprovechar la mayor velocidad de las redes 802.11ac para las transferencias de datos a través del Wifi.

A través de la conectividad inalámbrica se obtienen mejoras en el soporte de Bluetooth 4.2 frente al 4.1 de las versiones anteriores de Raspberry Pi 3 B. Esto se traduce en que entre varios nuevos beneficios el rendimiento será un poco más rápido y seguro.

A todas las bondades ya revisadas hasta el momento se le debe complementar dos más en el aspecto de conectividad de red, esto es que en las RPI 3 B+ existe la conectividad Gigabit Ethernet sobre USB 2.0.

Esto significa que podremos acceder a velocidades de transferencia mayores a las que teníamos en las versiones anteriores. Sin embargo, no todas las noticias son buenas, porque ese soporte tiene un cuello de botella llamado USB 2.0 que hará que las velocidades máximas que podamos obtener sean de 300 Mbps, y no de esos 1.000 Mbps que alcanzaríamos en condiciones óptimas.

Aun así, ese cambio es desde luego muy bienvenido, pero es que a él se le añade otro igualmente llamativo: la presencia de un nuevo conector en la placa con cuatro pines que ofrece soporte para el estándar Power-over-Ethernet (PoE). [5]

Esta interesante opción seguro que llama la atención de ciertos usuarios, aunque para aprovecharla es necesario adquirir de forma separada el llamado "PoE Hat", un accesorio en forma de placa adicional que se conecta a esos cuatro pines de los que hablábamos y que permiten usar el conector de la propia RPi3B+ de ese modo. [5]

2.6 Arduino UNO

Arduino es una plataforma de hardware de código abierto, basada en una sencilla placa de circuito impreso que contiene un microcontrolador de la marca "ATMEL" que cuenta con entradas y salidas analógicas y digitales en un entorno de desarrollo que esta que se sustenta en el lenguaje de programación processing. El dispositivo conecta el mundo físico con el mundo virtual o el mundo analógico con el digital controlando sensores, alarmas, sistemas de luces, motores, sistemas de comunicaciones y actuadores físicos. [6]

En el mercado existen una gran variedad de otras plataformas o microcontroladores que pueden ser utilizados en proyectos de computación pero que suelen presentar complicaciones a la hora de programar, Arduino ofrece la oportunidad de simplificar dicho proceso de desarrollo con la facilidad que presenta para realizar código para diversos proyectos.

Arduino es muy accesible de obtener en relación con otras placas de microcontroladores tanto en costo como en variedad. El software que se utiliza para el desarrollo de programas es multiplataforma y compatible con las principales versiones de sistemas operativos que actualmente se utilizan como Windows y Linux.

El entorno de programación de Arduino está basado en processing lo que facilita el aprendizaje de los usuarios acerca de cómo programar sus aplicaciones. Este entorno es software publicado bajo una licencia libre para ser adquirida y mejorada por una comunidad de programadores dedicados a esta labor. Este lenguaje puede complementarse con librerías de C++ y ser modificado a través del lenguaje de programación AVR C en el que está diseñado.

Los microcontroladores utilizados por Arduino son ATMEGA168, ATMEGA328 y ATMEGA1280. Estos módulos pueden ser ampliados u optimizados por parte de los

diseñadores de circuitos debido a la facilidad para acceder a los planos de los módulos publicados bajo licencia creative commons.

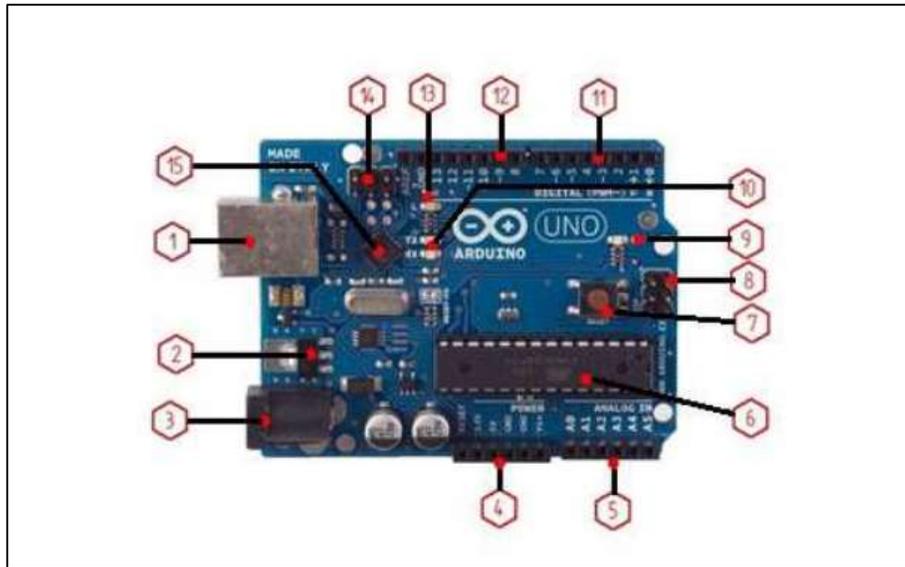


Figura 2.7 Componentes de la Placa Arduino UNO. [6]

Las placas Arduino han ido evolucionando al igual que su plataforma de software, en sus inicios las placas usaban un chip FTDI "FT232RL" para comunicarse vía USB con el computador, de esta forma se podría programar el procesador, luego se utilizó un microcontrolador especial para cumplir con esta función. Actualmente, se utiliza un solo microcontrolador para realizar la comunicación y a su vez carga las instrucciones a ejecutar.

Los componentes de la placa Arduino UNO señalados en la Figura 2.7 se describen a continuación:

1. **Conector USB**, proporciona la comunicación para la programación y la toma de datos, también provee una fuente de 5V DC para alimentar al Arduino, pero de baja corriente por lo que no sirve para alimentar motores de gran potencia.
2. **Regulador de voltaje de 5V**, se encarga de convertir el voltaje ingresado por el plug 3 en un voltaje de 5V regulado necesario para el funcionamiento de la placa y para alimentar circuitos externos.

3. **Plug de conexión para fuente de alimentación externa**, es el voltaje que se suministra que debe ser directo y estar entre 6V y 18V o hasta 20V, generalmente se debe tener cuidado de que el terminal del centro del plug quede conectado a positivo ya que algunos adaptadores traen la opción de intercambiar la polaridad de los cables.
4. **Puerto de conexiones**, está constituido por 6 pines de conexión con las funciones de RESET que permite resetear el microcontrolador al enviarle un cero lógico. Pin 3.3 V, provee de una fuente de 3.3 V DC para conectar dispositivos externos como en la protoboard. Pin 5V, es una fuente de 5 V DC para conectar dispositivos externos. Pin Vin, está conectado con el positivo del plug 3 por lo que se usa para conectar la alimentación de la placa con una fuente externa de entre 6 y 12 V DC en lugar del plug 3 o la alimentación por el puerto USB.
5. **Puertos de entradas análogas**, lugar donde se conectan las salidas de los sensores análogos. Estos pines solo funcionan como entradas recibiendo voltajes entre cero y cinco voltios directos.
6. **Microcontrolador Atmega 328**, implementado con los Arduino UNO. En la versión SMD del Arduino UNO R2, se usa el mismo microcontrolador, pero en montaje superficial, en este caso las únicas ventajas son la reducción del peso y ganar un poco de espacio.
7. **Botón reset**, permite resetear el microcontrolador haciendo que reinicie el programa.
8. **Pines de programación ICSP**, son usados para programar microcontroladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.
9. **Led ON**, enciende cuando el Arduino este encendido.
10. **Leds de recepción y transmisión**, se encienden cuando la tarjeta se comunica con el PC. El Tx indica transmisión de datos y el Rx recepción.
11. **Puertos de conexiones de pines de entradas o salidas digitales**, su configuración como entrada o salida debe ser incluida en el programa. Cuando se usa la terminal serial es conveniente no utilizar los pines cero (Rx) y uno

(Tx). Los pines 3, 5 y 6 están precedidos por el símbolo ~, lo que indica que permiten su uso como salidas controladas por ancho de pulso PWM.

12. Puerto de conexiones 5 entradas o salidas adicionales, las salidas 9, 10 y 11 permiten el control por ancho de pulso; la salida 13 es un poco diferente pues tiene conectada una resistencia en serie, lo que permite conectar un led directamente entre ella y tierra. Finalmente hay una salida a tierra GND y un pin AREF que permite ser empleado como referencia para las entradas análogas.

13. Led pin 13, indica el estado en que se encuentra.

14. Pines de programación ICSP, son usados para programar microcontroladores en protoboard o sobre circuitos impresos sin tener que retirarlos de su sitio.

15. Chip de comunicación, permite la conversión de serial a USB. [6]

Es importante tener en consideración que debido a que Arduino contienen un microcontrolador de la marca Atmel de nominado AVR 8-Bit RISC, esta línea de microcontroladores está formada por varios grupos entre los cuales se encuentra ATMEGA. La diferencia entre las diferentes familias es que para cada una de ellas puede variar el número y tipo de periféricos que incluyen la cantidad de memoria de programa y de datos. [6]

2.7 Celda de Peltier

En 1834 el físico francés Jean Charles Peltier descubrió este efecto termoeléctrico, en el curso de sus investigaciones sobre la electricidad. Este interesante fenómeno se mantuvo reducido a algunas pequeñas aplicaciones hasta ahora, época en que se comienza a utilizar sus posibilidades con más frecuencia. [7]

En la naturaleza, los materiales están formados por moléculas compuestas por átomos enlazados entre sí. Según el tipo de enlace atómico y molecular, los electrones exteriores de cada átomo tienen mayor o menor posibilidad de moverse alrededor de los núcleos. En los conductores, metales puros y aleaciones, los electrones exteriores menos ligados, pueden moverse en todo el material como si no pertenecieran a ningún átomo. Estos "electrones libres" tienen una distribución de

energía que depende principalmente de la temperatura y del tipo de átomos que compone el metal. [7]

La fundamentación teórica explicada previamente es lo que origina lo que se conoce como los 3 "Efectos Termoeléctricos" que fueron estudiados por Seebeck, Peltier y Thomson. Una de las primeras investigaciones que se hizo a partir de este efecto fue el estudio del famoso físico italiano Volta quién realizó un manuscrito donde recogía los puntos de su investigación para desarrollar la muy conocida Pila de Volta.

Dentro de la importante serie de descubrimientos de esos años, en 1834 Peltier descubre el fenómeno inverso al Seebeck, el "Efecto Peltier", por el cual, una corriente eléctrica que atraviesa las uniones de un lazo formado por dos metales diferentes, dependiendo del sentido de la corriente, genera calor en una unión y lo absorbe en la otra. Entonces, concretamente, el principio físico del "Efecto Peltier" es que al conectar una fuente de corriente a un lazo formado por 2 conductores A y B, en una unión la corriente que va desde A hacia B es favorecida por el potencial de contacto, y en la otra, la corriente que va desde B hacia A debe vencer una barrera de energía debida al potencial de contacto opuesto. Por lo tanto, la corriente al atravesar las uniones, en una libera calor, y en la otra lo absorbe del medioambiente. [7]

Debido a que los metales tienen distribuciones electrónicas similares, los potenciales de contacto son muy bajos (del orden de 100 mV) y el bombeo de calor mediante el efecto Peltier entre metales es muy pequeño. Esta es la razón por la que no se utilizó este fenómeno en refrigeradores hasta la segunda mitad del Siglo XX. [7]

Para conocer de forma más específica la operación de una célula Peltier primeramente debemos tener claro en que consiste este dispositivo.

Bien, una célula Peltier consiste en una bomba de calor activa de estado sólido que transfiere calor de un lado del dispositivo al otro consumiendo energía eléctrica en el proceso, el enfriamiento o calentamiento generado depende de la dirección de la corriente. Este dispositivo también es conocido como bomba de calor Peltier, refrigerador de estado sólido o refrigerador termoeléctrico (TEC).

Este dispositivo se puede usar para generar calor como calefacción o para generar frío como acondicionador de aire, pero la mayor práctica que se le da es la

refrigeración. En muchos casos se utiliza como controlador de temperatura para acondicionar el aire del lugar donde se encuentra activo.



Figura 2.8 Célula Peltier. [8]

Como lo hemos visto la operación de la Célula Peltier se basa en la conversión termoeléctrica en la que el calor se transforma en electricidad mediante el uso de equipos de transformación calor-eléctricos. A partir de este concepto existen los módulos termoeléctricos reversibles que pueden trabajar como enfriadores cuando se les aplica energía o generadores al aplicar calor.

Un termoeléctrico reversible está formado por un conjunto alternante de termoelementos semiconductores tipo n y p conectados en serie, en medio de dos metales hay dos aislantes eléctricos pero conductores térmicamente que pueden ser placas de cerámica.

Cuando un conductor es recorrido por una corriente eléctrica, hay electrones que se mueven por el conductor. Esos electrones no son libres: si quisiéramos arrancarlos del conductor, haría falta que gastásemos energía para liberarlos. Dependiendo de la naturaleza del conductor, los electrones de su interior están más o menos “atrapados” en el material. [7]

Se puede imaginar de esta manera: un electrón en el interior de un conductor es como una canica en el interior de un cuenco. Para sacar el electrón del material (o la canica

del cuenco) hace falta realizar un trabajo. Algunos materiales son “cuencos profundos”, y los electrones en su interior están muy “amarrados”: tienen muy poca energía, y hace falta mucho trabajo para liberarlos. Otros son “cuencos llanos”, con dar un poco de energía a sus electrones, escapan del conductor. [7]

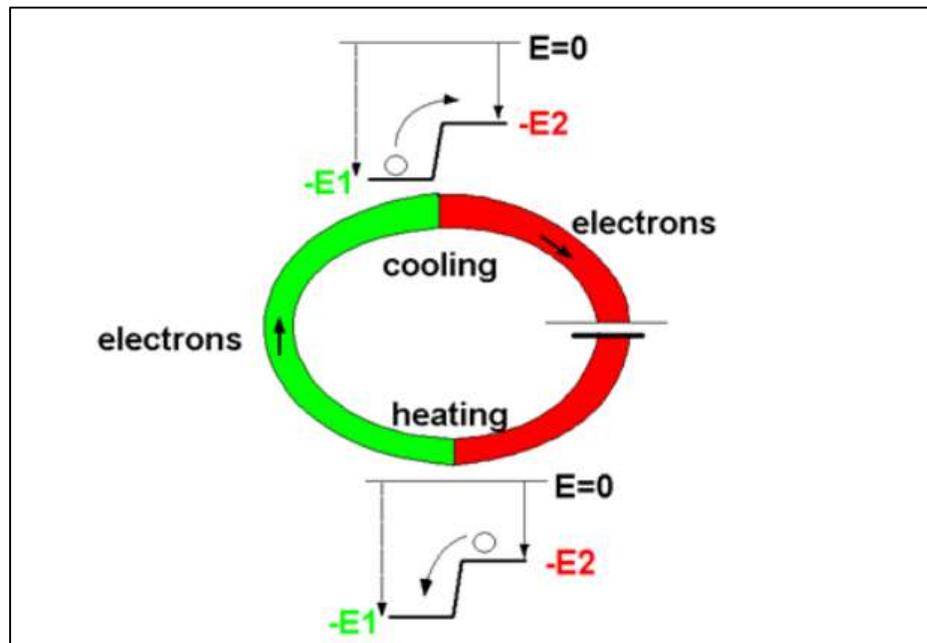


Figura 2.9 Diagrama del Efecto Peltier. [8]

Bien, ahora se piensa en un conductor por el que circula una corriente eléctrica: los electrones se están moviendo. En la analogía de la canica, ahora el conductor no es un cuenco: es una especie de surco en el suelo, por el que se mueven los electrones. Que los electrones se muevan por él no quiere decir que sean libres: no pueden salir del conductor salvo que alguien les dé energía (los “saque del surco” de un empujón). Llegamos ahora al quid de la cuestión. Se supone que no tiene un solo conductor sino dos conductores diferentes, el conductor “verde” y el conductor “rojo” (Figura 2.9). Y se supone que ambos conductores no “amarran” los electrones igual de intensamente: el conductor verde es un cuenco (o un surco, cuando los electrones se mueven) muy profundo, es decir, los electrones en el conductor verde tienen muy poca energía. Pero el conductor rojo es menos avaro con sus electrones, es un

“cuenca poco profundo”, y los electrones que circulan por él tienen más energía haría falta poco trabajo para arrancarlos de él. [7]

Si se conectan estos dos conductores (uno cuyos electrones tienen muy poca energía, y otro cuyos electrones tienen más energía) uno a continuación del otro, y a una pila, de modo que por ellos circule la corriente eléctrica, ocurre una circulación lineal. Un electrón que circula por el conductor verde tiene muy poca energía, y llega un momento en el que tiene que pasar al conductor rojo. Es como si una canica fuera por un surco muy profundo y se encontrase con una “cuesta arriba” que lo conecta con un surco menos profundo. Lo que sucede entonces es lo mismo que sucedería con la canica: según ésta sube la cuesta, gana energía potencial, pero pierde energía cinética, es decir, cuando llega arriba se mueve más despacio de lo que hacía abajo. Al electrón le sucede exactamente lo mismo: cuando pasa del conductor verde (donde su energía potencial eléctrica es muy pequeña) al conductor rojo (donde tiene más energía potencial eléctrica) se mueve más despacio. Pero, puesto que la temperatura es una medida de la energía cinética media de las partículas que componen un material (y los electrones son esas partículas), como consecuencia lógica e inevitable la temperatura de la zona de transición de un conductor a otro desciende. ¡Se enfría! El efecto contrario, cuando el electrón que se mueve despacio y va por un “surco poco profundo” pase del conductor rojo al verde, evidentemente, justo lo contrario: según “baja la cuesta” y cae hacia el conductor verde, donde su energía potencial eléctrica es menor, se acelera y como consecuencia, la temperatura de la “cuesta abajo” aumenta. Al final lo que sucede es que se tiene un circuito cerrado, una de cuyas mitades está más caliente que la otra; de hecho, una está más fría que la temperatura ambiente y la otra está más caliente (cuantos más electrones recorran el circuito, mayor diferencia de temperatura). Al igual que en los sistemas de refrigeración por compresión de los que hablamos antes, ocurre algo muy raro y que parece antinatural: Donde antes no había una diferencia de temperatura, ahora la hay, como consecuencia de que la temperatura no sea otra cosa que una medida de cómo de rápido se mueven las partículas de un material. [7]

En los últimos años estos dispositivos han sido utilizados en varias investigaciones para explicar diferentes efectos termoeléctricos y sus aplicaciones se han ido

incrementando en el tiempo debido a que no presentan partes que causen ruido o vibraciones y son muy amigables con el medio ambiente.

Entre las aplicaciones que se pueden mencionar tenemos investigaciones para generación de energía, sistemas de enfriamiento y calentamiento utilizando agua como fluido de trabajo, remover calor de celdas fotovoltaicas, estudio de procesos para enfriamiento y calentamiento del aire, se han desarrollado prototipos en el área de refrigeración doméstica y así existen otras aplicaciones en el área de investigación que están preparando el futuro de esta tecnología.

2.8 Lógica Difusa (LD)

La Lógica Difusa también conocida como Lógica Borrosa es en realidad una rama de la Inteligencia Artificial que debido a sus fundamentos teóricos permite manejar información vaga o inexacta para poder con esta información ajustar el funcionamiento o el estado de un sistema específico. El término “borroso” proviene de la palabra inglesa “fuzzy” que significa “confuso, difuso o indefinido”.

A partir de esta definición se puede con la LD controlar un sistema por medio de reglas establecidas por el “sentido común” que hacen referencia a valores indefinidos.

Se fundamenta en la teoría de conjuntos y un sistema de inferencia borroso basado en reglas de la forma “SI... ENTONCES...”, para imitar el comportamiento de la lógica humana, donde los valores lingüísticos de la premisa y el consecuente están definidos por conjuntos borroso, de esta forma las reglas forman un conjunto borro en otro. [9]

Se conoce que históricamente La Lógica Difusa se comenzó a investigar en los años sesenta en la Universidad de Berkeley en California por el famoso ingeniero Lofty A. Zadeh al descubrir lo que el denominó principio de incompatibilidad, lo que concluye que, de acuerdo con el aumento de la complejidad de un sistema, nuestra capacidad humana para describir a gran precisión su comportamiento disminuye hasta más allá del límite donde la precisión y el significado son características excluyentes. A partir de esta premisa introdujo la definición de conjunto difuso, esta definición indica que las bases del pensamiento humano no son números sino etiquetas lingüísticas.

La Lógica Difusa permite representar el conocimiento común, que es mayoritariamente del tipo lingüístico cualitativo y no necesariamente cuantitativo, en

un lenguaje matemático a través de la teoría de conjuntos difusos y funciones características asociadas a ellos. Permite trabajar a la vez con datos numéricos y términos lingüísticos; los términos lingüísticos son inherentemente menos preciso que los datos numéricos, pero en muchas ocasiones aportan una información más útil para el razonamiento humano. [9]

Esta teoría nos permite manejar y procesar ciertos tipos de información en los cuales se manejen términos inexactos, imprecisos y subjetivos. De una manera similar a como lo hace el cerebro humano, es posible ordenar un razonamiento basado en reglas imprecisas y en datos incompletos. [9]

Las definiciones de Lógica Difusa permiten combinar la teoría de conjuntos y la lógica booleana de tal forma que una entidad pueda formar parte parcialmente en muchos casos de más de un conjunto, esto se puede lograr a partir de los conjuntos nebulosos y que las operaciones lógicas tengan valores más allá de ceros y unos. Ahora a partir de las reglas difusas se puede procesar las variables difusas y obtener una salida difusa referente a la relación de las variables de entrada, a partir de esas salidas se pueden generar valores que permita ejercer una acción sobre cualquier elemento del mundo real.

Las reglas de un sistema difuso se pueden crear a partir del conocimiento humano facilitando el aprendizaje directamente desde los expertos en los procesos a controlar o también por medio de sistemas adaptativos que aprenden observando las operaciones de las personas.

La Lógica Difusa es muy flexible y no tan complicada como la Lógica Clásica, permite trabajar con diferentes grados de verdad simulando el pensamiento y dejando espacio a la imprecisión cuando se debe representar un problema y a pesar de aquello se puede obtener una solución correcta debido a que maneja la incertidumbre e imprecisión adecuadamente y utiliza diferentes grados de veracidad o falsedad en contraste con los valores cerrados de la lógica tradicional (verdadero o falso).

Los sistemas difusos tienen muchas aplicaciones y se aplican para controlar aplicaciones de ingeniería, física, biología o economía, entre otros.

Las aplicaciones más conocidas de este tipo de sistemas van desde la estimación de parámetros, toma de decisiones, sistemas mecánicos de control como aires acondicionados o lavadoras automáticas, control de automóviles o casas inteligentes. A partir de esta lógica se ha tratado de aplicar una forma más humana de pensar en la programación de computadoras.

Los conjuntos difusos difieren de los conjuntos clásicos en el aspecto de que la teoría de conjuntos clásicos considera la pertenencia total o no de un elemento dentro de un conjunto, mientras que la teoría difusa considera cierto grado de pertenencia de un elemento en un conjunto difuso, es decir cualquier elemento no se encuentra obligado a pertenecer totalmente o no a un conjunto.

El grado de pertenencia de un elemento en un determinado conjunto difuso se define mediante la función característica asociada al conjunto difuso.

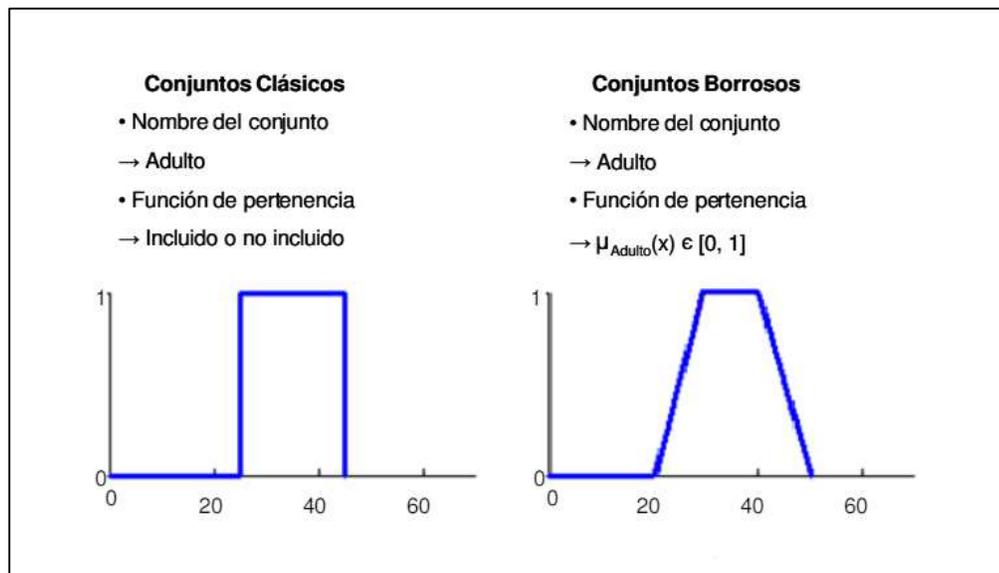


Figura 2.10 Comparación entre Lógica Clásica y Lógica Difusa. [9]

La forma de la función característica utilizada para definir el comportamiento de un elemento en un conjunto difuso depende de varios aspectos relacionados al conocimiento y entorno siendo el más importante el punto de vista del diseñador del sistema que a su vez debería apoyar en la experiencia del usuario clave del proceso,

la principal condición que debe cumplir la función característica es tomar valores entre 0 y 1.

Los tipos de funciones de membresía o características de mayor uso debido a la facilidad en su manejo matemático son: triangular y trapezoidal, en muchos casos se considera además las gaussianas, sigmoideal, gamma, pi y campana. La definición de una función se basa en dos criterios muy importantes siendo el primero el conocimiento de los expertos humanos y segundo la colección de datos.

El número de funciones que se deberán elegir para una variable depende solamente del criterio del experto, sin embargo, por definición se conoce que mientras más funciones de membresía se utilicen los resultados serán más efectivos, pero esto ocasiona que se incremente la complejidad del proceso computacional, pues las variables pueden estar traslapadas formando parte de varios conjuntos difusos.

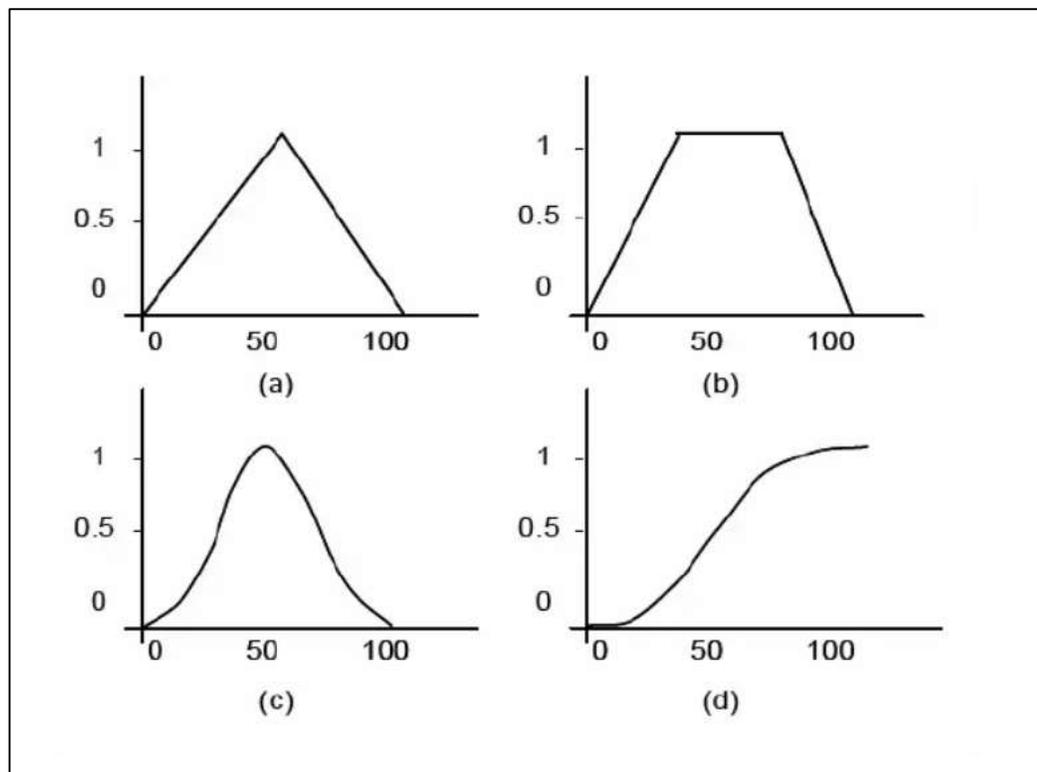


Figura 2.11 Funciones características habituales: (a) triangular, (b) trapezoidal, (c) gaussiana y (d) sigmoideal. [9]

Las leyes y propiedades que cumplen los conjuntos clásicos no siempre se cumplen en el caso de los conjuntos borrosos. A continuación, se verificarán las leyes que se cumplen para los conjuntos borrosos y cuales no:

1. **Propiedad Conmutativa**, siempre se verifica, debido a que las t-normas y las t-conormas son conmutativas por definición.
2. **Propiedad Asociativa**, también se verifica, debido a que las t-norma y las t-conormas también son asociativas por definición.
3. **Leyes de Idempotencia**, se cumplen si se elige el mínimo y el máximo como operadores para la intersección y la unión respectivamente. Pero si se escoge por ejemplo el producto y la suma algebraicos, no se cumplen.
4. **Leyes de Absorción**, se cumplen si se elige el mínimo y el máximo como operadores para la intersección y la unión respectivamente. Con otras normas no ocurre necesariamente lo mismo.
5. **Propiedad Distributiva**, se cumple si se elige el mínimo y el máximo como operadores para la intersección y la unión respectivamente. Con otras normas no ocurre necesariamente lo mismo.
6. **Propiedad de Absorción e Identidad**, siempre se cumplen por la última propiedad de t-norma y t-conorma.
7. **Involución del Complemento**, se cumple si el operador elegido es el estándar.
8. **Leyes de Morgan**, se garantiza su cumplimiento si la t-norma y la s-norma elegidas derivan una de la otra.
9. **Leyes Complementarias**, no se cumplen, Es quizás la consecuencia más clara de introducir la borrosidad a los conjuntos.

Se puede comprobar fácilmente que en el caso de que los conjuntos sean clásicos (función de pertenencia restringida a 0 o 1) las diferencias entre las diversas normas desaparecen, convirtiéndose en los operadores intersección y unión clásicos. [9]

Las funciones de membresía representan la forma gráfica de un conjunto borroso, son expresadas por fórmulas matemáticas y definidas en el espacio unidimensional, bidimensional y multidimensional.

Las más utilizadas son las triangulares, trapezoidales, gaussiana, sigmoideal, entre otras.

La función triangular está representada por los vértices de un triángulo a través de tres literales a, b, y c.

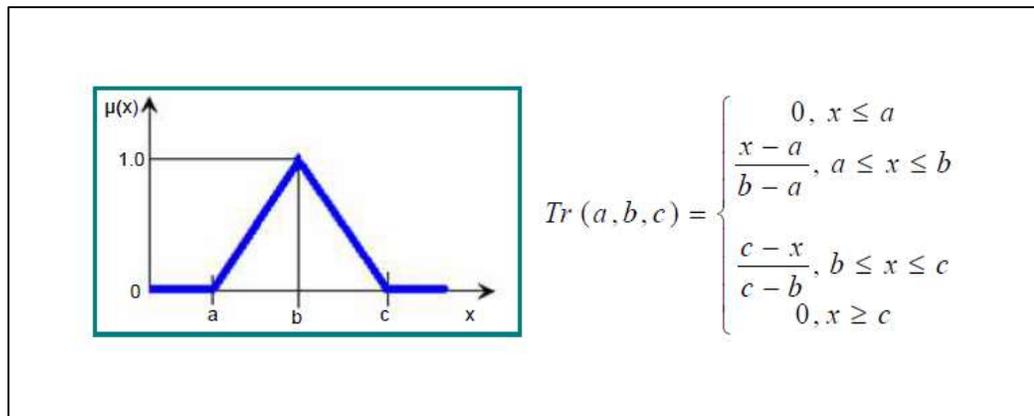


Figura 2.12 Función de pertenencia triangular. [9]

La función trapezoidal está representada en función a cuatro vértices de los parámetros a, b, c y d.

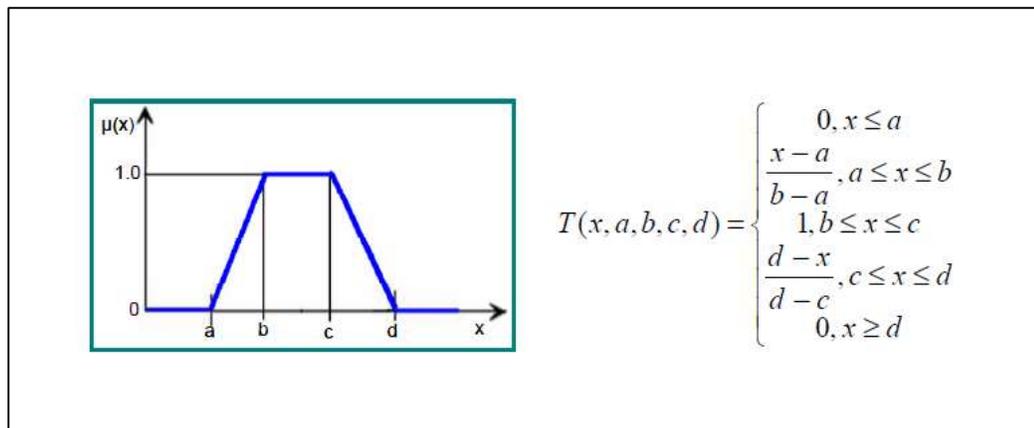


Figura 2.13 Función de pertenencia trapezoidal. [9]

Las funciones de pertenencia más utilizadas en aplicaciones industriales son justamente la triangular y la trapezoidal, pero tienen la desventaja de no ser muy suaves en los puntos donde se forman los ángulos. [9]

La función de pertenencia tipo Gaussiana se define totalmente por los literales c y σ , aquí c es el centro y σ el ancho de la función de pertenencia.

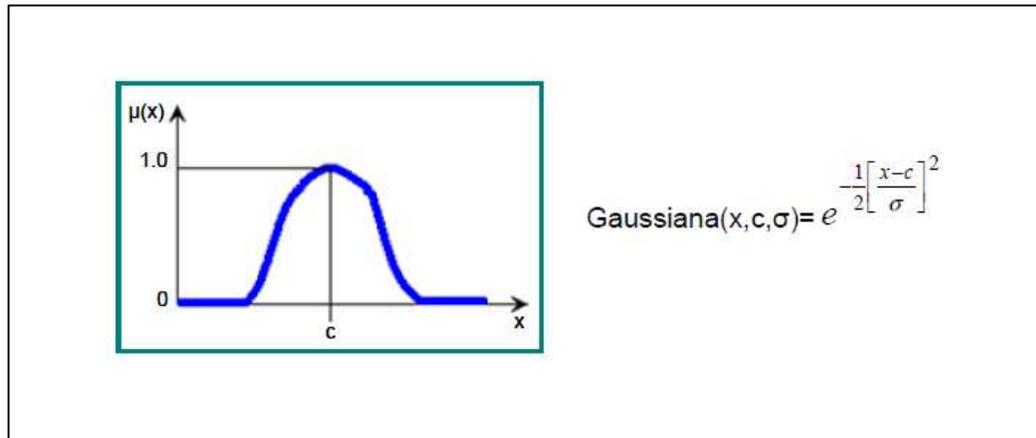


Figura 2.14 Función de pertenencia Gaussiana. [9]

La función de Bell está definida por los parámetros a , b y c de la siguiente forma:

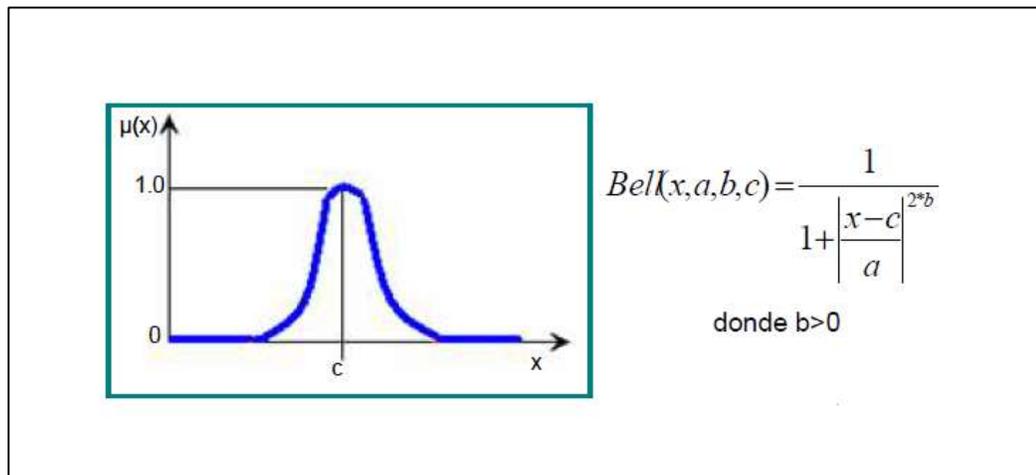


Figura 2.15 Función de pertenencia de Bell. [9]

Esta función también es conocida como función de pertenencia de Cauchy debido a que está relacionada con la distribución de Cauchy de la Teoría de las Probabilidades.

La función de pertenencia Sigmoidal se utiliza en redes neuronales como función de activación y simular el comportamiento del sistema de inferencia borroso. [9]

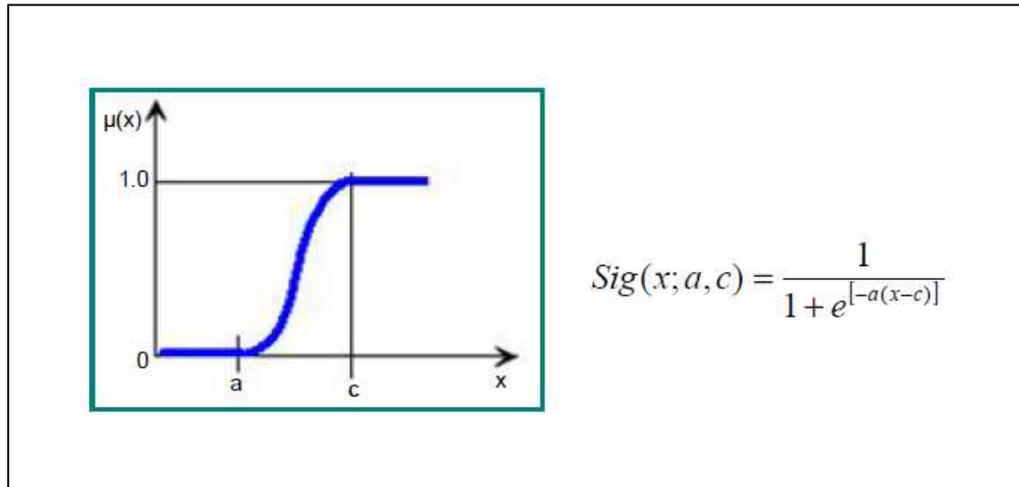


Figura 2.16 Función de pertenencia Sigmoidal. [9]

La teoría de conjuntos borrosa al igual que la teoría de conjuntos clásica presenta operaciones básicas como la intersección, unión, complemento, inclusión entre otras, siendo las más conocidas y utilizadas las tres primeras:

- Operación de Intersección.
- Operación de Unión.
- Complemento Borroso.

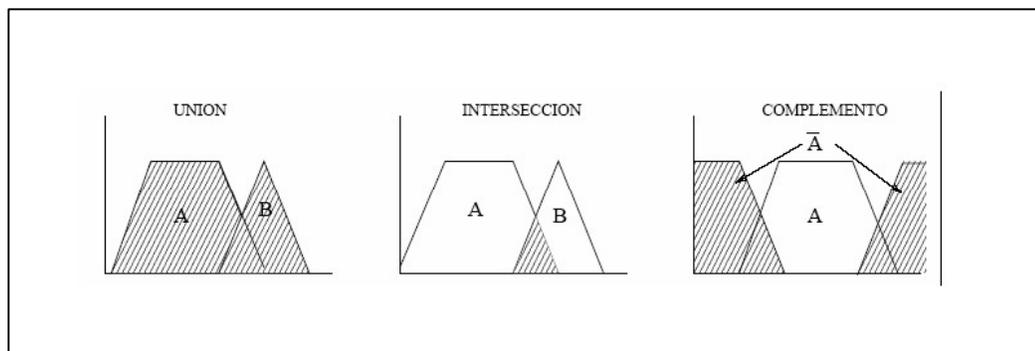


Figura 2.17 Operaciones Difusas. [9]

Las variables lingüísticas son sentencias que se enmarcan en lenguaje natural y se conocen como etiqueta lingüística y está representada por medio de un conjunto borroso definido a partir del universo de discurso de la variable. [9]

Una variable lingüística se caracteriza por una quintupla $(A; T(A); U; G; M)$ en la que:

- **A**, es el nombre de la variable.
- **T(A) o solo T**, denota el conjunto de términos de A, es decir, el conjunto de nombres de los valores lingüísticos de A.
- **X**, es el universo de discurso de la variable A, que toma valores en un universo de discurso U asociada a una variable base u.
- **G**, es una regla sintáctica para términos lingüísticos (que suele tomar la forma de una gramática) para los nombres de los valores de A.
- **M**, es una regla semántica para asociar su significado a cada A, $M(X)$, que es un subconjunto borroso de U. [9]

Por ejemplo, la temperatura del cuerpo humano puede catalogarse como “baja”, “normal”, “alta” o “muy alta”, cada uno de esos términos es una etiqueta lingüística que puede definirse como un conjunto borroso. [9]

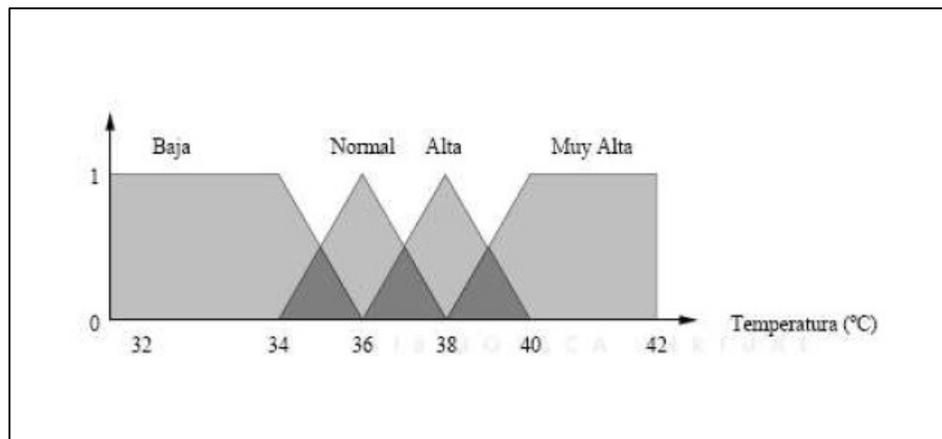


Figura 2.18 Definición de la variable lingüística Temperatura. [9]

En la Lógica Borrosa el razonamiento no tiene precisión, al contrario, su respuesta suele ser una aproximación (Razonamiento Aproximado), se puede inferir la conclusión obtenida y la misma se parece más a la conclusión formal de la regla original cuanto mayor sea el grado de cumplimiento de la regla por parte del hecho. [9]

El razonamiento aproximado se puede organizar en esquemas modus ponens y modus tollens, esto conlleva a los siguientes razonamientos:

- Proposiciones Borrosas.
- Modus Ponens Generalizado (MPG).
- Modus Tollens Generalizado (MTG).
- Razonamiento Aproximado Multicondicional. [9]

Los sistemas de inferencia borrosa corresponden a la aplicación de la inferencia borrosa en la automatización de procesos. Su estructura está constituida por La Base de Conocimiento o Diccionario donde se definen las funciones de pertenencia de las reglas borrosas, La Base de Reglas, que contiene el conjunto de acciones a seguir en función al estado y el Mecanismo de Inferencia que realiza procedimiento de inferencia sobre las reglas borrosas para producir una conclusión.

Existen tres tipos de sistemas basados en reglas borrosas:

- **Sistemas Puros**, que poseen conjuntos borrosos como entrada y salida y debido a que no realizan ninguna transformación sobre las mismas tienen solo dos componentes primordiales: una base de conocimiento y un motor de inferencia.
- **Sistemas Borrosos Tipo Mamdani**, está compuesto por una base de conocimientos, un motor de inferencias y las interfaces de fusificación y defusificación. Pueden ser usados en aplicaciones en el mundo real por la facilidad para manejar entradas y salidas reales. Permiten incluir conocimiento experto en forma de reglas lingüísticas con facilidad. Tienen libertad a la hora de elegir las interfaces de fusificación y defusificación.
- **Sistemas Borrosos Tipo Takagi-Sugeno-Kang**, no trabaja con reglas lingüísticas, sino que se basa en reglas donde el antecedente esta forma por variables lingüísticas y el consecuente se representa como una función de las variables de entrada. [9]

Los elementos de un Sistema de Lógica Difusa tipo Mamdani (más usado) son cuatro:

- Interfaz de Fusificación.

- Base de Conocimiento.
- Motor de Inferencias.
- Interfaz de Defusificación. [9]

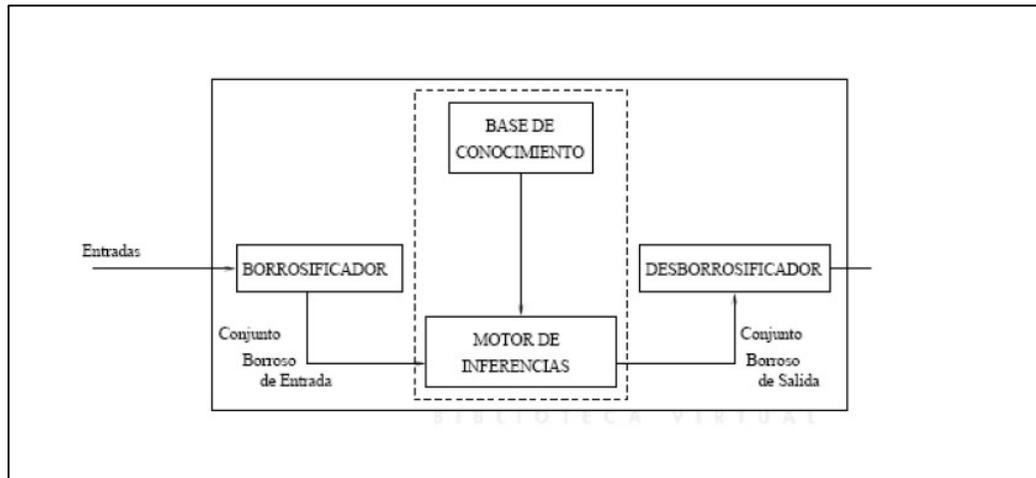


Figura 2.19 Sistema de Lógica Difusa. [9]

- **La Interfaz de Fusificación**, realiza el escalado de los valores de las entradas para adecuarlos a valores típicos para los que se define el sistema.
- **La Base de Conocimiento**, formada por una “base de datos” que recoge la definición de las funciones de pertenencia de las entradas y el sistema y una base de reglas que resume objetivos de control de un experto por medio de las reglas lingüísticas de control.
- **El Motor de Inferencias**, que trabaja las acciones del sistema empelando alguna representación de la implicación difusa.
- **Interfaz de Defusificación**, que convierte la acción “difusa” inferida en una acción concreta susceptible de aplicación sobre el proceso. [9]

La experiencia del ser humano experto en cualquier proceso que se desee controlar es primordial para formar la base de reglas (base de conocimiento).

Las reglas de una base de conocimientos pueden a más de obtenerse del experto en el proceso también obtenerse de las acciones de control de un operador en función de la observación o basados en un modelo difuso del proceso obtenido previamente, finalmente también se puede formar las reglas basado en aprendizaje.

Para definir conjuntos borrosos se utilizan dos métodos dependiendo de si los universos de discurso son discretos o continuos:

- **Definición Numérica**, en el caso de los universos discretos, la función de pertenencia de un conjunto borroso se representa como un vector de números cuya dimensión depende del grado de discretización.
- **Definición Funcional**, en el caso de los universos continuos, las funciones de pertenencia se expresan por medio de funciones regulares, típicamente triangulares o trapezoidales. [9]

La inferencia difusa es el proceso mediante el cual se obtiene como consecuente un conjunto difuso a partir de unos antecedentes también difusos.

Para obtener la señal de salida que se debe aplicar al proceso a controlar se puede recurrir a diferentes métodos siendo los más utilizados los siguientes:

- **Método del centroide o centro de gravedad**, es el más utilizado de todos los métodos debido a que proporciona variaciones suaves y continuas de los valores de salida. Se calcula en el centro de gravedad de la función de pertenencia del conjunto de salida.
- **Método de máxima pertenencia**, aquí el valor de salida es aquel cuyo grado de pertenencia al conjunto de salida inferido sea máximo.
- **Método del máximo indexado**, calcula el centro de gravedad del subconjunto difuso del consecuente inferido, correspondiente a los puntos cuyo valor de pertenencia al consecuente es superior a un umbral dado. [9]

CAPÍTULO 3

3. DESARROLLO.

En este capítulo se presentará la metodología de solución aplicada al problema planteado en el Capítulo 1. El objetivo es conocer de forma detallada los pasos implementados a lo largo del proyecto para conocer de mejor manera las ideas generales y específicas aplicadas durante el desarrollo de la solución.

3.1 Diseño de proceso recolector de datos de la peletizadora

Para plantear la solución al problema de recolección de datos de las señales de la peletizadora se escogió dispositivos IOT que permitan recolectar los valores de las señales desde el panel eléctrico y enviarlos directamente al hosting del programa desarrollado para presentar los datos finales.

El esquema de solución aplicado sería el siguiente:

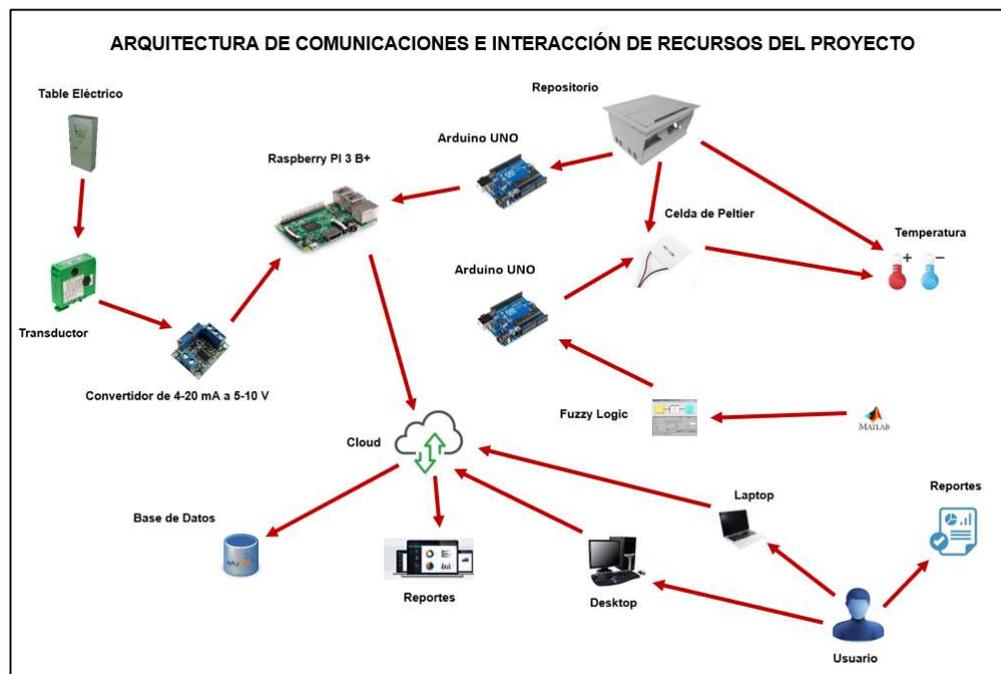


Figura 3.1 Diseño de comunicaciones del proyecto.

En la Figura 3.1 podemos observar todos los elementos utilizados en la implementación de la solución; es muy importante concentrar la atención en el objeto “Repositorio”, esta definición engloba físicamente la caja de control que albergará a todos los equipos que forman parte del proceso recolector de datos y regulador de la temperatura ambiente al interior de la misma caja.

Dentro de la caja de control tendremos el diseño de las conexiones para el monitoreo de señales de la peletizadora que se conecta al panel eléctrico. Este circuito está formado por un Arduino UNO conectado en dos vías. Por una parte, utiliza sus puertos análogos desde el A0 al A4 para obtener las señales de 5 convertidores de 4-20 mA a 0-5 V que a su vez están conectados al módulo de comunicaciones del PLC del tablero eléctrico que controla a los equipos industriales del proceso de producción de balanceado entre ellos la peletizadora.

Las variables por monitorear desde el panel eléctrico en orden y los pines análogos asignados para la recepción en Arduino UNO son las siguientes:

Pines	Variable
A0	Amperaje
A1	Frecuencia
A2	Presión
A3	TemperaturaDADO
A4	TemperaturaDDC

Tabla 2: Relación de pines análogos de Arduino UNO y variables de la Peletizadora.

Estas variables han sido definidas por parte del personal técnico de producción y complementadas por el personal técnico de mantenimiento de la planta como las principales variables a monitorear de la peletizadora, esto debido a que tal como se indicó en el Capítulo 1 y 2 de este documento, el equipo necesita mantener rangos de valores adecuados para la correcta transformación de la mezcla de materias primas en el pellet final.

Para obtener los valores de las señales desde el panel eléctrico se utilizaron cinco convertidores de miliamperios a voltaje (un convertidor por cada señal monitoreada), esto a partir de que las señales desde el PLC se presentan en valores en miliamperios

entre 4–20 mA y el dispositivo Arduino UNO lee señales en voltaje dentro del rango de 0–5 V.



Figura 3.2 Convertidor DROK de corriente al voltaje de 4-20 mA a 0-5 V.

Estos convertidores se conectan al módulo de comunicaciones del PLC en el panel eléctrico a través de cables de conexión tomando los polos positivos y negativos asignados a los pines del módulo para cada señal y configurados en la programación del PLC.

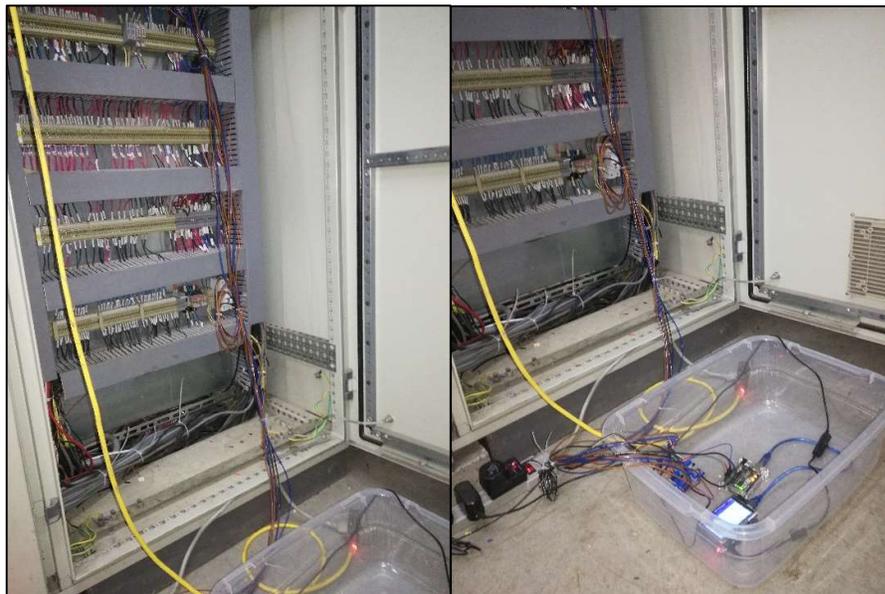


Figura 3.3 Conexión de los convertidores con el panel eléctrico.

La configuración para la obtención de estas señales consiste en crear variables paralelas a las variables principales dentro del PLC para no interrumpir el funcionamiento normal de los equipos de la planta, a la programación del PLC no se puede acceder por motivos de confidencialidad. De esta forma las señales a revisar se copian desde el módulo y se envían hacia los convertidores de señales de corriente a voltaje y es el Arduino UNO que lee este valor para luego enviar el mismo por conexión serial (USB) hacia el dispositivo Raspberry PI 3 B+ seleccionado para enviar finalmente el dato a la Base de Datos en la nube donde se realiza la conversión del valor leído es los dispositivos IOT en el valor final de lectura.

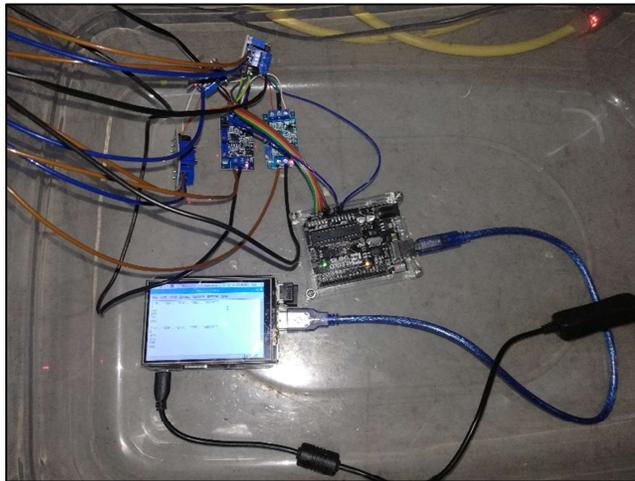


Figura 3.4 Conexión de los convertidores con el Arduino UNO.

La conversión se realiza a través de una tabla de datos configurada previamente por parte del técnico eléctrico que indico los rangos mínimo y máximo de trabajo de cada variable y a partir de esta información se realizó la respectiva escala de calibración de cada valor.

Los valores detallados de la configuración de la escala para las variables se presentarán en los Anexos a este documento.

A continuación, veremos los detalles de las configuraciones realizadas en los dispositivos IOT Arduino UNO y Raspberry PI 3 B+ para recolectar los valores de las señales desde el panel eléctrico y ser enviados hacia la nube de datos del proyecto para su posterior presentación en el portal web de información.

3.1.1 Programa en Arduino UNO

Para leer la información desde el panel se utilizaron convertidores de corriente a voltaje conectados entre el panel eléctrico y el Arduino UNO. Los pines asignados para la recepción de los datos en Arduino son los análogos desde el A0 al A4 en orden tal como se especifica en la Tabla 2.

Las conexiones de los convertidores hacia el Arduino se especifican en la Figura 3.4 en el mismo orden que se especificó en la Tabla 2.

Para poder leer los valores de las señales monitoreadas se creó un programa en Arduino que luego de leer los pines análogos realiza un arreglo de valores que luego será enviado al puerto serial del dispositivo. La razón de crear un arreglo de valores está relacionada con la forma en que la Raspberry Pi 3 B+ receptorá luego los valores de las señales, debido a que la Raspberry recibe la información a manera de arreglo para un mejor tratamiento de los datos.

El programa se denomina **“RegistroVariablesPeletizadoraAr.ino”** consta de tres partes. La primera contiene la definición de los pines análogos a utilizar en el proceso de lectura de datos y la definición del arreglo de datos para encapsular los valores leídos.

```
// Declaracion de variables globales

int pinA0 = 0; //Variable del pin de entrada del sensor (A0)
int pinA1 = 1; //Variable del pin de entrada del sensor (A1)
int pinA2 = 2; //Variable del pin de entrada del sensor (A2)
int pinA3 = 3; //Variable del pin de entrada del sensor (A3)
int pinA4 = 4; //Variable del pin de entrada del sensor (A4)
String variables_peletizadora; //Variable contenedora de los valores de las variables de la peletizadora
```

Figura 3.5 Definición de puertos y arreglo de variables en Arduino UNO.

La segunda parte consiste en la definición típica de la velocidad de trabajo del puerto serial que para este caso se definió en el valor de 9600 bps tal como se muestra en la Figura 3.6.

En la tercera parte del código en Arduino se realiza la lectura de los pines análogos de la placa y a la vez que se forma el arreglo de datos para ser enviados al puerto serial del equipo. La lectura de esta información se realiza cada 10 segundos, esto debido a que los técnicos de producción y mantenimiento que controlan el proceso

requieren que la información se reciba en ese lapso, aunque la presentación final de la información en el portal web se realice en dos formas que se verán más adelante.

La programación para la recepción de datos y formación del arreglo con su posterior envío al puerto serial de la placa se puede ver en la Figura 3.7

```
void setup()
{
    //Configuramos el puerto serial a 9600 bps
    Serial.begin(9600);
}
```

Figura 3.6 Definición de la velocidad de trabajo del puerto serial.

```
void loop()
{
    //Con analogRead leemos las variables de la peletizadora
    variables_peletizadora =
    String(analogRead(pinA0)) + "|"
    + String(analogRead(pinA1)) + "|"
    + String(analogRead(pinA2)) + "|"
    + String(analogRead(pinA3)) + "|"
    + String(analogRead(pinA4));

    //Envia el dato al puerto serial
    Serial.print(variables_peletizadora);

    //Imprime un salto de línea
    Serial.print("\n");

    //Esperamos un tiempo de 10 segundos para repetir el loop
    delay(10000);
}
```

Figura 3.7 Lectura, preparación y envío de datos al puerto serial.

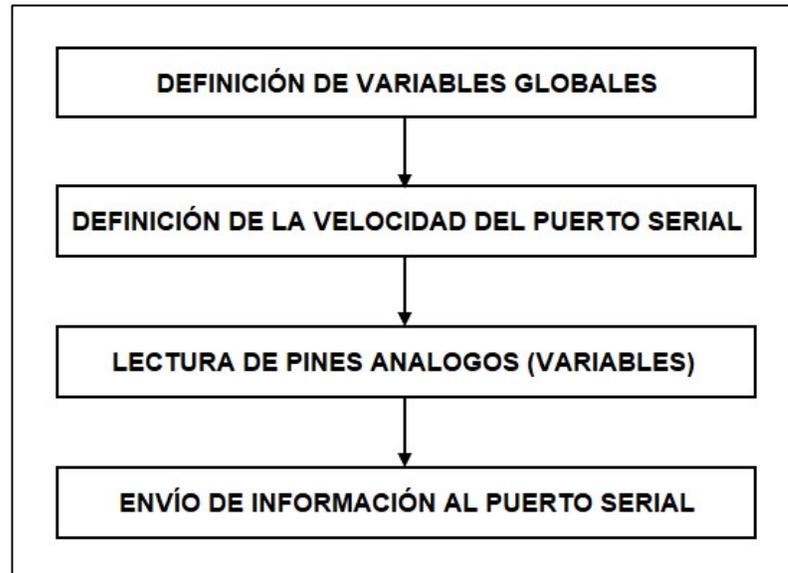


Figura 3.8 Proceso de lectura de variables de la peletizadora en Arduino UNO.

El programa en Arduino descrito previamente en diferentes partes se puede resumir en el diagrama de procesos de la Figura 3.8. Aquí claramente se observan las fases de trabajo del código para cumplir el objetivo de lectura de variables de la peletizadora.

Se puede observar que el trabajo que realiza el Arduino es sencillo pero muy importante pues es el medio de conexión entre el panel eléctrico y la Raspberry PI 3 B+. Esto se da debido a que este último dispositivo no puede leer valores de voltaje superiores a 3.3 voltios, y como vimos anteriormente las señales desde el panel eléctrico son presentadas en valores de 5 voltios aunque el convertidor utilizado se maneje entre 0–5 V para la salida, por lo general las lecturas tienden a presentar los valores más altos y no se puede arriesgar la placa Raspberry a sufrir problemas de sobrecarga en sus pines de lecturas e incluso a sufrir daños en su integridad por un exceso de voltaje lo que puede sobre todo causar una mala lectura de la información recolectada.

Al momento de la implementación no se pudo obtener convertidores de señal para usar desde el panel y que trabajen con 4-20 mA a 0–3.3 V que trabajen eficientemente pues la lectura de valores no era totalmente confiable debido a que por valores

bastante bajos se presentaba mucha variación en su contenido y así esta información trabajada con esta medida no presentaba garantías para la obtención de los resultados del proyecto.

El Arduino UNO a través de la comunicación serial (USB) establecida con la Raspberry PI 3 B+ realiza la conversión de voltajes desde 5 a 3.3 V y de esta forma se protege y garantiza la información procesada durante la lectura de las señales de la Peletizadora desde el panel eléctrico.

3.1.2 Programa en Raspberry PI 3 B+

En este proyecto mientras que el Arduino UNO trabaja del lado del panel eléctrico, la Raspberry Pi 3 B+ que esté conectada por puerto USB al Arduino para obtener los datos leídos por este último trabaja también conectada a la nube.

Debido a las bondades de la microcomputadora Raspberry que permite conectarse a través de su dispositivo Wifi con cualquier red inalámbrica, se ha conectado a la red interna de la empresa con acceso a Internet para enviar la información hacia el hosting (nube) del proyecto.

Raspberry PI utiliza la versión de Linux para Raspberry originaria de Debian y que en este caso es Raspbian, desde esta plataforma se puede utilizar la herramienta de programación Python para diseñar el código que permite conectar con la nube y enviar los datos leídos por el Arduino desde el panel eléctrico.

El programa desarrollado en Raspberry y ejecutado desde la misma se denomina “**RegistroVariablesPeletizadoraPy.py**” y se puede explicar en tres partes muy importantes.

La primera parte del programa contiene la definición de las librerías que el programa importara para poder ejecutar las instrucciones que permitan procesar la información recolectada.

```
import serial
import time
import MySQLdb
from urllib.parse import urlencode
from urllib.request import Request, urlopen
```

Figura 3.9 Definición de librerías del programa en Python.

La segunda parte del programa contiene la configuración de la velocidad de trabajo del puerto serial, se debe considerar que debe trabajar a la misma velocidad del Arduino UNO que en este caso es 9600 bps, adicional a aquello se debe configurar el tiempo de lectura de los datos, recordar que Arduino toma las muestras cada 10 segundos por tal motivo se ha establecido el tiempo de lectura para la Raspberry dos segundos más tarde es decir en 12 segundos para dejar un espacio de tiempo pertinente entre las lecturas de ambos dispositivos.

```
arduino=serial.Serial ("/dev/ttyACM0",baudrate=9600,timeout=12.0)

arduino.setDTR(False)
time.sleep(1)
arduino.flushInput ()
arduino.setDTR(True)

url = "http://www.iotsignalcontrol.com/iotsignalcontrol/Data/RegistroVariablesPeletizadoraBd.php"
```

Figura 3.10 Configuración de medidas de trabajo de la Raspberry.

Adicional a lo mencionado se configura la limpieza del arreglo de datos posible información basura que puede filtrarse producto del ruido alrededor de los equipos durante la lectura de información.

Finalmente, en esta segunda parte del código se configura la conexión con el hosting del portal web para poder acceder al archivo gestor de la inserción de datos en la Base de Datos del sistema previa interpretación y conversión de los datos utilizando la escala de valores de las variables que se ubicará en los Anexos a este documento.

La dirección web del sitio en Internet es “www.iotsignalcontrol.com” y el archivo de procesamiento de datos utilizado en la nube y que está desarrollado en lenguaje de programación PHP se denomina “**RegistroVariablesPeletizadoraBD.php**”, este archivo lo revisaremos más adelante.

En la tercera parte del programa en Python se realiza finalmente la lectura de datos del puerto serial de la placa tomando la información que es enviada hacia la nube que en si consiste en el portal web que preparará los datos que luego serán presentados al usuario final.

Como se puede ver en la Figura 3.11 los datos leídos del puerto serial de la Raspberry y que a su vez vienen desde el Arduino son enviados al hosting (nube) por el método

“POST” de esta forma se protegen los datos durante el trayecto y se comienza a interactuar con la programación web que en este caso se ejecuta a través del lenguaje PHP.

```

while True:
    val=arduino.readline()
    if val != '\n':
        try:
            data = val.decode('utf-8').split("|")
            print (data)
            print (data[0])#Amperaje
            print (data[1])#Frecuencia
            print (data[2])#Presion
            print (data[3])#TemperaturaDADO
            print (data[4])#TemperaturaDDC
            post_fields = {'Amperaje':data[0],
                'Frecuencia':data[1],
                'Presion':data[2],
                'TemperaturaDADO':data[3],
                'TemperaturaDDC':data[4]}
            request = Request(url, urlencode(post_fields).encode())
            json = urlopen(request).read().decode()
        except ValueError:
            print ("")
    arduino.close()

```

Figura 3.11 Lectura y envío de datos a la nube.

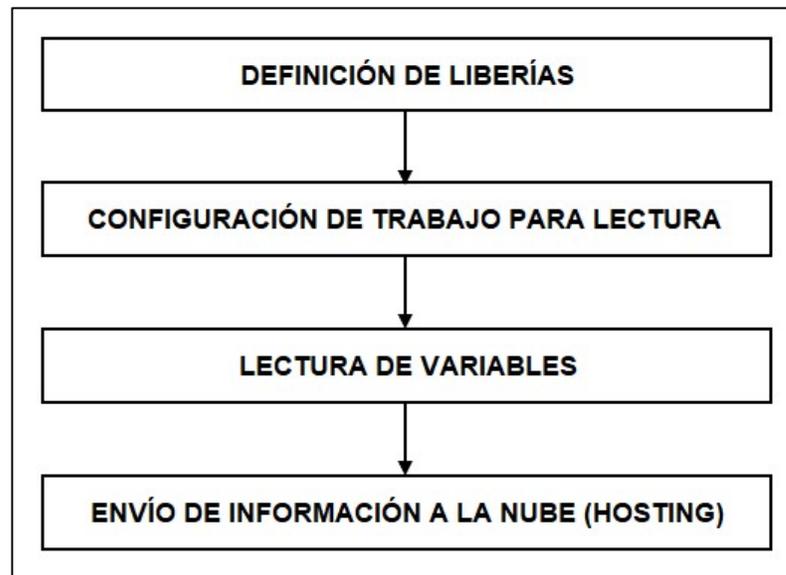


Figura 3.12 Proceso de lectura de variables de la peletizadora en Raspberry.

Del lado del hosting existe un sistema de archivos que contiene la programación de todo el portal web que se presentará al usuario final al momento de realizar las respectivas consultas. En esta estructura existe un archivo de interacción con la Raspberry que permite primeramente convertir el dato tomado desde la Raspberry en un valor real a partir de una escala de valores definida previamente por los administradores del proceso de monitoreo de las variables, luego de aquello procede a la respectiva inserción en la Base de Datos del sistema de la información procesada.

El archivo denominado **“RegistroVariablesPeletizadoraBD.php”** y que ya lo indicamos previamente consta de tres partes, una parte general para todo el proyecto y que contiene la configuración de las variables en método “POST” para poder recibir las mismas desde la Raspberry y a su vez contiene la configuración del acceso a la Base de Datos del proyecto tal como se ve en la Figura 3.13.

```

<?php session_start();

//OBTENCIÓN DE DATOS DESDE PYTHON (RASPBERRY PI 3 B+)

$Amperaje = $_POST['Amperaje'];
$Frecuencia = $_POST['Frecuencia'];
$Presion = $_POST['Presion'];
$TemperaturaDADO = $_POST['TemperaturaDADO'];
$TemperaturaDDC = $_POST['TemperaturaDDC'];

//ASIGNACIÓN DE LA FECHA DEL DÍA

$FechaActual = date ('Y/m/d H:i:s');

//INICIALIZACIÓN DE CONEXIÓN PARA INTERACTUAR CON LA BD

include_once('../Connections/TransaccionBd.php');

$Bd = new TransaccionBd();
$Bd->AbrirConexion();

```

Figura 3.13 Configuración de recepción de variables y acceso a Base de Datos.

La segunda parte del programa se repite por cada variable a procesar y contiene la conversión de los datos leídos del panel eléctrico en el valor que le corresponde adoptar a cada variable en base a la escala de equivalencias calibrada por el criterio del experto en automatización de la planta. En total son cinco variables para procesar. Esto lo podemos ver en la Figuras 3.14.

```

if($Amperaje > 0)
{
    //OBTENCIÓN DEL VALOR DE LA VARIABLE AMPERAJE SEGÚN LA ESCALA DE VALORACIÓN
    $SqlSelect = "SELECT Valor_real
                FROM      pwiot_tblrangovaloracionvariables
                WHERE     Valor_muestra = '". $Amperaje.'"
                AND       Id_variable = 1";

    $Resultado = $Bd->ConsultarInformacion($SqlSelect);

    if($Bd->ContarRegistro($Resultado))
    {
        if($Row = $Bd->RecorrerConsulta($Resultado))
        {
            do
            {
                $AmperajeReal = $Row[0];

                }while($Row = $Bd->RecorrerConsulta($Resultado));
            }
        }
    else
    {
        $AmperajeReal = -1;
    }
}

```

Figura 3.14 Conversión de datos según escala de variables.

La tercera parte del programa al igual que la segunda, se repite por todas las variables y contiene el proceso de inserción de los valores de las variables ya convertidos en el dato final dentro la Base de Datos del sistema web para el posterior seguimiento del usuario final.

```

//CREACIÓN DE SCRIPT PARA INSERCIÓN DEL VALOR DE LA VARIABLE AMPERAJE
$SqlInsert = "INSERT INTO pwiot_tblrregistrovariables (Id_variable, Fecha_registro, Valor) ";
$SqlInsert.= "VALUES ('1', '$FechaActual', '$AmperajeReal')";

$Resultado = $Bd->GrabarInformacion($SqlInsert);

```

Figura 3.15 Inserción de datos en la Base de Datos del Portal Web.

Luego de realizar los procesos indicados previamente para poder registrar la información que es el fin de la primera parte del proyecto el programa procede a cerrar la conexión a la Base de Datos y de ahí en adelante se debe complementar esta fase con la presentación de los datos al usuario, eso lo podremos ver en el siguiente apartado del documento.

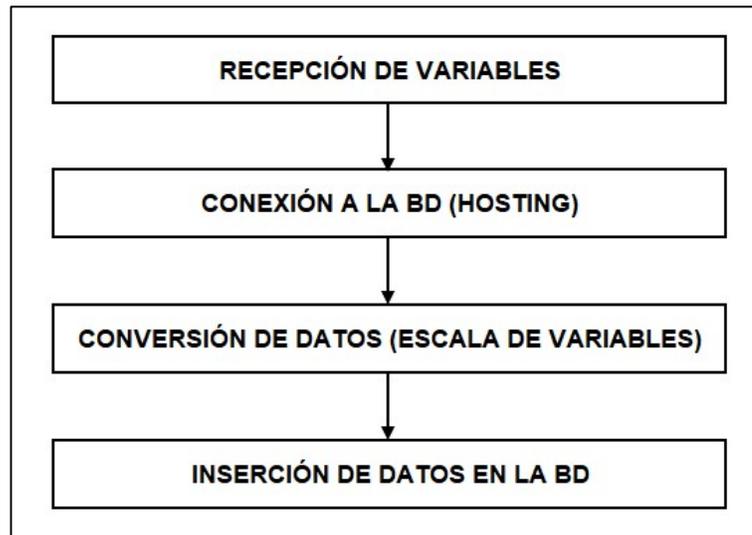


Figura 3.16 Proceso de inserción de variables de la peletizadora en la nube.

La Figura 3.16 muestra el diagrama de procesos que resume el programa que permite registrar la información de las variables de la peletizadora en la Base de Datos de la nube donde se aloja el sistema de monitoreo.

3.2 Diseño del portal web para presentación de datos

Luego de obtener los datos desde el panel eléctrico a través del diseño de comunicaciones conformado por la placa Arduino UNO y la placa Raspberry PI 3 B+ esta información es enviada a la Base de Datos del sistema que permitirá al usuario revisar los datos de las variables a través de las consultas y reportes que el programa posee como opciones para presentación de información.

El programa se encuentra alojado en la nube (hosting) y está diseñado en lenguaje de programación PHP 7.3.0, Apache versión 2.4.37 y base de datos MariaDB 10.1.37.

Esta aplicación utiliza tipo de programación modular y arquitectura en 3 capas, esto quiere decir que existe la interacción con el usuario a través del gestor Ajax combinado con HTML y JavaScript y las peticiones al servidor se realizan a través del lenguaje PHP.

La Base de Datos utilizada es la nueva aplicación distribuida con PHP en el paquete XAMPP 3.2.2 y cuyo nombre es MariaDB, sin embargo, esta base de datos es una versión mejorada de la anterior MySQL, de hecho, se puede utilizar las aplicaciones

de gestión que se utilizaban con MySQL para administrar MariaDB como lo son PhpMyAdmin (web) o MySQL Workbench (desktop).

Para generar la nube de datos se utilizó un hosting del gestor Godaddy y se escogió el dominio “**www.iotsignalcontrol.com**” para mostrar el sitio web alojado en el repositorio de los archivos de la página web.

La aplicación desarrollada es muy intuitiva para el usuario, se necesita un usuario para acceder a la misma y este usuario deberá ser un correo electrónico, luego se podrán realizar ciertas actualizaciones del perfil, consultas de parámetros generales y finalmente consultas y ver reportes detallados y gráficos de información referente a las variables de monitoreo así como la temperatura del interior de la caja de control que servirá para estudiar el controlador difuso de temperatura que forma parte de la segunda parte del proyecto que revisaremos más adelante.

3.2.1 Diagrama Entidad-Relación de la Base de Datos

La Base de Datos del sistema fue diseñada en MariaDB y consta de 10 tablas para almacenar la información relevante del aplicativo y un Procedimiento Almacenado que da como resultado la información de las variables consultadas tabuladas según el tipo de reporte solicitado.

La Base de Datos del programa se denomina PwIOT y su estructura general se muestra en la Figura 3.17.

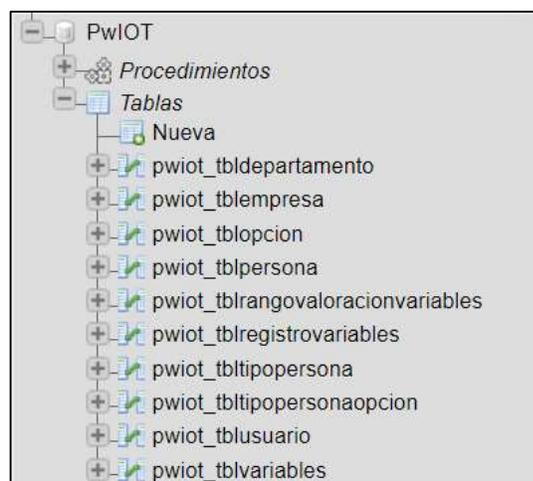


Figura 3.17 Estructura general de la Base de Datos del Portal Web.

Los nombres definidos para las tablas están asignados en base a la entidad que manejan, son intuitivos de reconocer tal como se muestra en la Figura 3.17.

Al realizar el diseño del modelo Entidad-Relación se pudo determinar que no era necesario extender el uso de tablas para anidar información, debido a la facilidad y sencillez del sistema.

Para generar las consultas y reportes si fue necesario la inclusión de un Procedimiento Almacenado que permita consolidar los valores de todas las variables a consultar y debido al volumen de datos que la lectura de la peletizadora genera (cada 10 segundos) los tiempos de respuesta pueden ser bastante largos si se realiza consultas generales, pero también está a disposición la consulta de promedios por hora que permite ver el promedio de la información recolectada por cada hora del día lo que resuelve el problema de la lentitud y a la vez permite mejor lectura de datos para el usuario pues facilita la presentación por ser resumida.

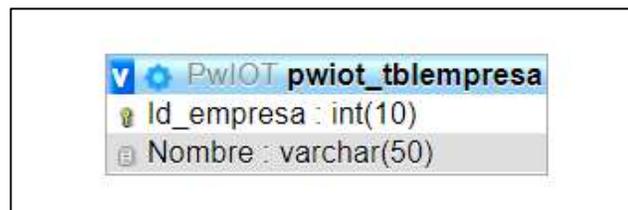


Figura 3.18 Tabla “pwiot_tblempresa”.

Esta tabla contiene información de las empresas que participan con sus usuarios en el proceso de registro de información de la peletizadora, para este caso solo es una empresa.

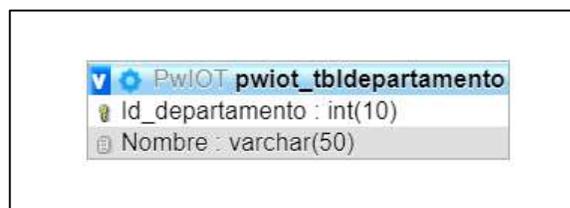
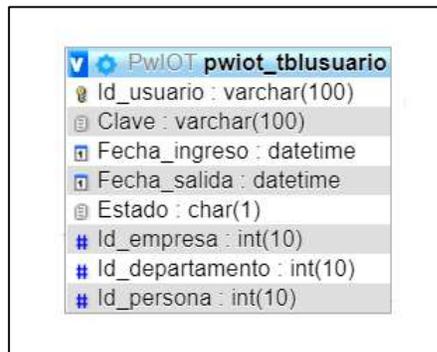


Figura 3.19 Tabla “pwiot_tbldepartamento”.

Al igual que la tabla anterior, esta table es general y aloja información relacionada a los departamentos de la empresa que están relacionados con el proceso de toma de datos de la peletizadora siendo los más relevantes TI y Producción.

La tabla de usuarios “**pwio_tblusuario**” aloja toda la información del usuario de la persona que tiene acceso al sistema.

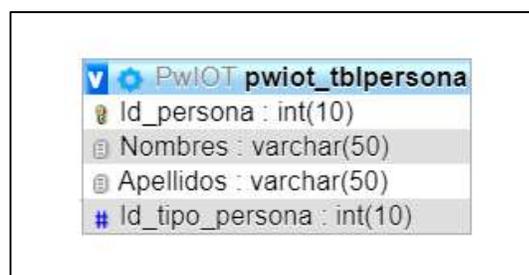
La información de esta tabla es específica y tiene significado técnico relacionado directamente a la habilitación o baja del usuario en el portal, relación con empresas o departamentos de donde viene el usuario, así como su estado actual en el programa para poder realizar la correcta administración de las cuentas registrada en el aplicativo. Esta tabla la podemos observar en la Figura 3.20.



Column Name	Data Type	Constraints
Id_usuario	varchar(100)	Primary Key
Clave	varchar(100)	
Fecha_ingreso	datetime	
Fecha_salida	datetime	
Estado	char(1)	
Id_empresa	int(10)	Foreign Key
Id_departamento	int(10)	Foreign Key
Id_persona	int(10)	Foreign Key

Figura 3.20 Tabla “**pwiot_tblusuario**”.

Para registrar información general del usuario, es decir su información personal más básica se creo la tabla “**pwiot_tblpersona**”, la vemos en la Figura 3.21.

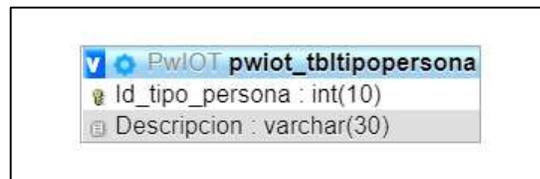


Column Name	Data Type	Constraints
Id_persona	int(10)	Primary Key
Nombres	varchar(50)	
Apellidos	varchar(50)	
Id_tipo_persona	int(10)	Foreign Key

Figura 3.21 Tabla “**pwiot_tblpersona**”.

En el sistema es necesario la existencia de un usuario administrador para gestionar los servicios del aplicativo de forma general y a su vez se requiere diferenciar este

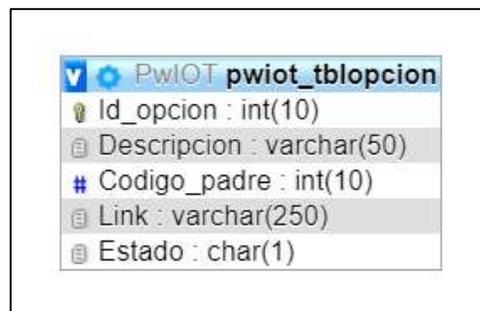
administrador de los demás usuarios por este motivo existe la tabla “**pwiot_tbltipopersona**”.



Column Name	Data Type
Id_tipo_persona	int(10)
Descripcion	varchar(30)

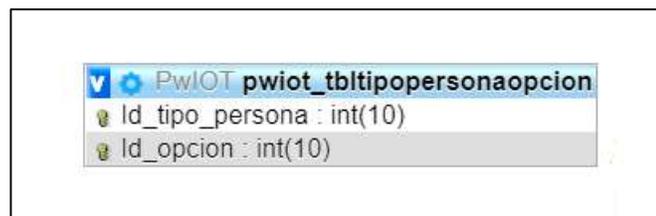
Figura 3.22 Tabla “**pwiot_tbltipopersona**”.

Las diferentes opciones que maneja el sistema y que ofrece a los usuarios se han registrado en la tabla “**pwio_tblopcion**”, aquí se puede enlazar estas opciones que se presentan en el menú del programa al momento de realizar el login en el mismo junto con el tipo de usuario que accede al aplicativo, esto debido a que se debe diferenciar el administrador de los usuarios normales del sistema, para esto existe la tabla “**pwiot_tbltipopersonaopcion**”. Ambas tablas se pueden ver en las Figuras 3.23 y 3.24 respectivamente.



Column Name	Data Type
Id_opcion	int(10)
Descripcion	varchar(50)
Codigo_padre	int(10)
Link	varchar(250)
Estado	char(1)

Figura 3.23 Tabla “**pwiot_tblopcion**”.



Column Name	Data Type
Id_tipo_persona	int(10)
Id_opcion	int(10)

Figura 3.24 Tabla “**pwiot_tbltipopersonaopcion**”.

The screenshot shows a table definition for 'pwiot_tblvariables' in a database. The table has five columns: 'Id_variable' (int(10)), 'Nombre' (varchar(50)), 'Unidad' (varchar(20)), 'Valor_minimo' (decimal(18,8)), and 'Valor_maximo' (decimal(18,8)).

Column Name	Data Type
Id_variable	int(10)
Nombre	varchar(50)
Unidad	varchar(20)
Valor_minimo	decimal(18,8)
Valor_maximo	decimal(18,8)

Figura 3.25 Tabla “pwiot_tblvariables”.

Las variables estudiadas en este proyecto y que son seis de las cuales cinco están relacionadas a la peletizadora y una tiene que ver con la temperatura ambiente de la caja de control que aloja a los dispositivos IOT se encuentran parametrizadas en la tabla “pwiot_tblvariables” tal como se presenta en la Figura 3.25.

Para interpretar adecuadamente la información de las variables que son recolectadas desde el panel eléctrico se mencionó que existe una escala de equivalencias desarrollada por los expertos en automatización de la organización. Estos valores están parametrizados en la tabla “pwiot_tblrangovaloracionvariables”, así se muestra en la Figura 3.26.

The screenshot shows a table definition for 'pwiot_tblrangovaloracionvariables' in a database. The table has five columns: 'Id_rango_valoracion' (int(10)), 'Valor_mA' (decimal(18,8)), 'Valor_muestra' (decimal(18,8)), 'Valor_real' (decimal(18,8)), and 'Id_variable' (int(10)).

Column Name	Data Type
Id_rango_valoracion	int(10)
Valor_mA	decimal(18,8)
Valor_muestra	decimal(18,8)
Valor_real	decimal(18,8)
Id_variable	int(10)

Figura 3.26 Tabla “pwiot_tblrangovaloracionvariables”.

Finalmente, una de las tablas más relevantes por no indicar que es la más relevante del sistema, es aquella que justamente contiene el detalle de todos los valores receptados por parte del circuito IOT formado por los elementos electrónicos que ya hemos mencionado previamente donde los más relevantes son el Arduino UNO y la Raspberry PI 3 B+. Esta table se muestra en la Figura 3.27 y es “pwiot_tblrregistrovariables”.

Column Name	Data Type
Id_registro_variables	int(255)
Id_variable	int(10)
Fecha_registro	datetime
Valor	decimal(18,8)

Figura 3.27 Tabla “pwiot_tblrregistrovariables”.

A partir de estas tablas estudiadas y como ya se indicó se fundamenta toda la estructura del sistema web del usuario final, pero se debe recordar que existe además un Procedimiento Almacenado utilizado específicamente para mostrar los resultados de las consultas de que realizan los usuarios de todas las variables del sistema. Un parte del contenido de este programa se la puede ver en las Figura 3.28 y 3.29.

```

DELIMITER //
DROP PROCEDURE IF EXISTS 'pwiot_sprvariables' //
CREATE PROCEDURE 'pwiot_sprvariables'(Id_variable INT(10), Fecha_inicial DATETIME, Fecha_final DATETIME, promedio INT)
BEGIN
DECLARE UltimaFila      INT DEFAULT 0   ;
DECLARE IdVariable      INT(10)        ;
DECLARE FechaRegistro  DATETIME       ;
DECLARE Valor           DECIMAL(18,8)  ;

/*CREACIÓN DE CURSOR PARA VARIABLES NO PROMEDIOS*/
DECLARE RegistroVariables CURSOR FOR
SELECT a.Id_variable ,
       b.Fecha_registro ,
       b.Valor
FROM   pwiot_tblrregistrovariables a ,
       pwiot_tblrregistrovariables b
WHERE  a.Id_variable = b.Id_variable
AND    b.Id_variable <> 0
AND    b.Fecha_registro >= Fecha_inicial
AND    b.Fecha_registro <= Fecha_final
ORDER BY b.Fecha_registro ASC, a.Id_variable ASC;

/*FIN DE CREACIÓN DE CURSOR PARA VARIABLES NO PROMEDIOS*/

```

Figura 3.28 Procedimiento Almacenado “pwiot_sprvariables” (1).

```

/*CREACIÓN DE CURSOR PARA VARIABLES PROMEDIOS*/
DECLARE RegistroVariablesPromedio CURSOR FOR
SELECT *
FROM   pwiot_tmregistrovariables0
ORDER BY Id_variable, Fecha_registro ASC;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET UltimaFila = 1;

/*FIN DE CREACIÓN DE CURSOR PARA VARIABLES PROMEDIOS*/

/*OBTENCIÓN DE INFORMACIÓN PARA CURSOR DE VARIABLES PROMEDIOS*/
DROP TABLE IF EXISTS pwiot_tmregistrovariables0;
CREATE TEMPORARY TABLE pwiot_tmregistrovariables0
(
  Id_variable      INT(255)      ,
  Fecha_registro  DATETIME       ,
  Valor           DECIMAL(18,8)  ,
) ENGINE = InnoDB DEFAULT CHARSET = utf8 COLLATE = utf8_general_ci;

INSERT INTO pwiot_tmregistrovariables0
SELECT a.Id_variable ,
       CONVERT(CONCAT(CONVERT(YEAR(b.Fecha_registro),CHAR(50)),'-',CASE WHEN MONTH(b.Fecha_registro) < 10
AVG(b.Valor)

```

Figura 3.29 Procedimiento Almacenado “pwiot_sprvariables” (2).

El Diagrama Entidad-Relación de la Base de Datos del sistema se presenta en la Figura 3.30.

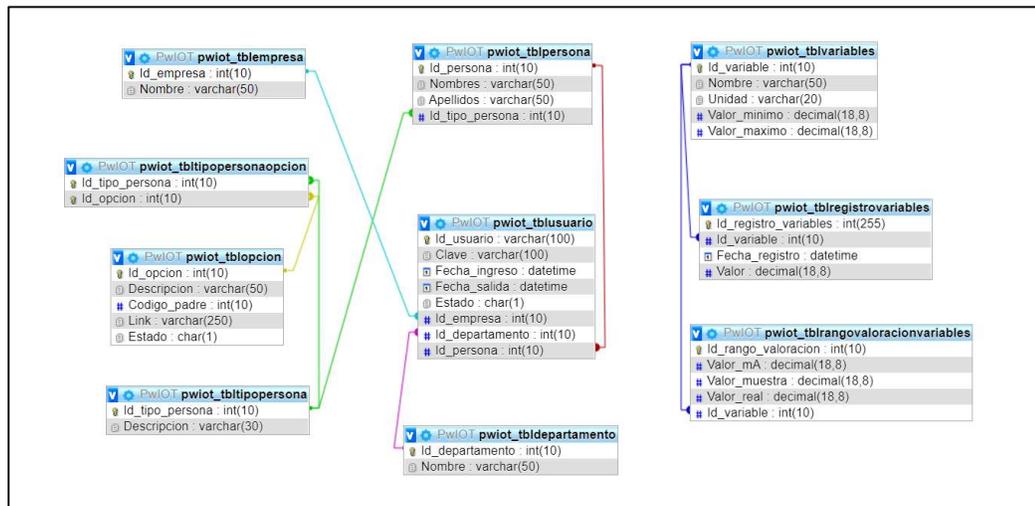


Figura 3.30 Diagrama Entidad-Relación de la Base de Datos PwIoT.

3.2.2 Interfaz de usuario final

El sistema de presentación de información de las variables de la Peletizadora estudiada posee pocas interfaces, pero a su vez son muy intuitivas y fáciles de utilizar.

Todo parte del acceso al sitio web “www.iotsignalcontrol.com”, aquí se muestra la ventana principal tal como en la Figura 3.31.

Para acceder al sistema primeramente se debe crear una cuenta de usuario en la opción pertinente para aquella acción, esta se ubica en la ventana principal en el enlace “**Crear Nueva Cuenta**”, esta opción lleva a la ventana de gestión de usuarios donde puede registrar la cuenta de correo electrónico que solicita el sistema para poder dar de alta a un nuevo usuario, colocar la contraseña, nombres ya apellidos y finalmente escoger la empresa y el departamento de origen para hacer el respectivo monitoreo de las señales.

Una vez que se ha procedido a realizar el registro completo el sistema enviara un mensaje de correo electrónico para confirmar la creación del nuevo usuario y habilitar así el acceso al sistema. La presentación de la opción de registro se puede ver en la Figura 3.32.



Figura 3.31 Ventana principal del Sistema de Monitoreo de Variables.

Crear su cuenta

Usuario

Email

Clave

El mínimo de caracteres es: 6

Datos generales

Nombres

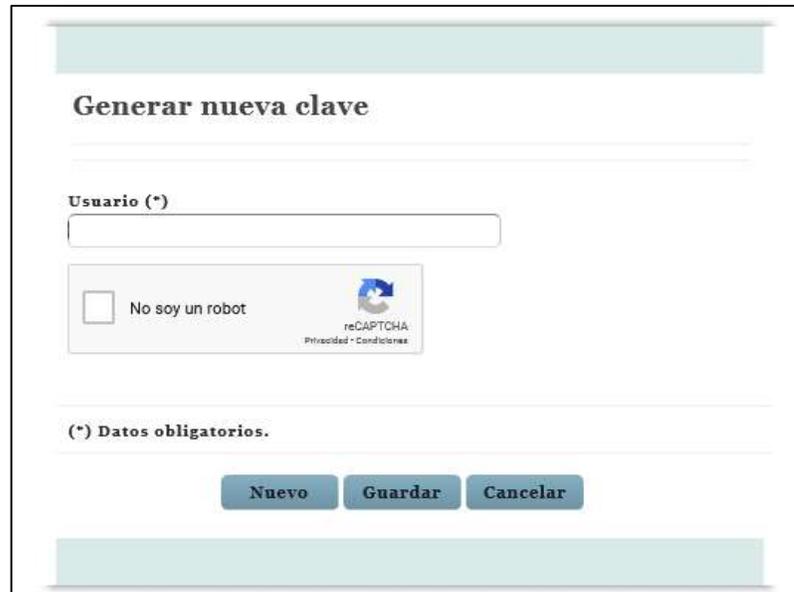
Apellidos

Empresa

Departamento

Figura 3.32 Opción para la creación de nuevos usuarios.

En caso de pérdida del acceso al sistema por olvido de la contraseña, se puede generar una nueva clave a través de la opción “**Recuperar Acceso**” de la ventana principal. Tal como se muestra en la Figura 3.33 se debe ingresar la cuenta de usuario y validar el mismo para obtener una clave temporal en un mensaje de correo electrónico enviado inmediatamente.



El formulario, titulado "Generar nueva clave", contiene un campo de texto para el "Usuario (*)". Debajo de este campo se encuentra un widget reCAPTCHA con el texto "No soy un robot" y un icono de un robot. En la parte inferior del formulario, hay una leyenda "(*) Datos obligatorios." y tres botones: "Nuevo", "Guardar" y "Cancelar".

Figura 3.33 Opción para la recuperación de acceso al sistema.

Para ingresar al sistema se colocan las credenciales en el lugar asignado para el login del sistema que se muestra en la Figura 3.34.



El formulario, titulado "Ingreso al Sistema", contiene un campo de texto para "Email:" con el valor "leonardtony_86@hotmail.com". Debajo de este campo se encuentra un campo de texto para "Clave:" con caracteres ocultos por puntos y un ícono de ojo para alternar la visibilidad. En la parte inferior del formulario, hay un botón "Ingresar" y dos enlaces de texto: "Recuperar Acceso" y "Crear Nueva Cuenta".

Figura 3.34 Opción para acceder al sistema.

Una vez que se ha ingresado al sistema aparecerán en el lado izquierdo de la pantalla la información general del usuario y hacia el centro superior de la misma el menú de opciones, recordar que existe una ligera diferencia entre usuario administrador y usuario normal la cuál explicaremos pronto. La ventana de acceso se puede ver en la Figura 3.35.



Figura 3.35 Ventana de sesión en el sistema.

La opción de inicio permite volver a la venta de sesión iniciada en el sistema en cualquier momento que sea presionada.

Mientras que las opciones de configuración permiten trabajar la gestión de usuarios en todas sus fases dependiendo si el usuario logueado es de tipo administrador o normal, tal como se ve en la Figura 3.36.



Figura 3.36 Opciones de configuraciones en el sistema.

La opción para creación de usuarios es prácticamente la misma que se utiliza cuando se crea un usuario por primera vez y se la puede ver en la Figura 3.32.

La opción para actualización de datos de usuario puede ser accedida por todos los tipos de usuarios y permite actualizar los datos generales del usuario que se ven en la Figura 3.37.



El formulario muestra un encabezado con el título "Datos generales". A continuación, hay un campo de texto para "Nombres" con el valor "Anthony" y un ícono de "x" para borrar. Debajo, un campo de texto para "Apellidos" con el valor "Barona". Luego, un menú desplegable para "Empresa" con "BioMar Ecuador" seleccionado. Finalmente, un menú desplegable para "Departamento" con "TI" seleccionado. En la parte inferior del formulario, hay tres botones: "Nuevo", "Guardar" y "Cancelar".

Figura 3.37 Opción de actualización de datos generales del usuario.

Las configuraciones que complementan el menú de opciones permiten actualizar la clave del usuario en caso de ser requerido, esta opción se puede ver en la Figura 3.38.

Otra de las opciones del menú de configuraciones permite cancelar la cuenta de usuario, es decir un usuario administrador puede dar de baja a cualquier cuenta de usuario normal del sistema o el mismo usuario normal puede darse de baja así mismo, pero siempre existe la opción de reactivarse y esto se lo hace en la opción de creación de cuentas de usuario siempre que ya haya estado registrado previamente por obvias razones.

La última opción se puede ver en las Figura 3.39. Se debe tener en consideración que todas las opciones que involucran afectaciones en la información del usuario requieren de la validación del usuario ejecutante por tal motivo se habilitó la opción de Recaptcha para cumplir con este fin.

Cambiar clave

Usuario
leonardtony_86@hotmail.com

Clave actual (*)

Nueva clave (*)

El mínimo de caracteres es: 6

Confirmar nueva clave (*)

No soy un robot  reCAPTCHA
Privacidad · Condiciones

(*) Datos obligatorios.

Nuevo Guardar Cancelar

Figura 3.38 Opción de actualización de la clave del usuario.

Cancelar cuenta de usuario

Usuario
[Elija una opción](#)

Datos generales

Nombres

Apellidos

Empresa

Departamento

Fecha ingreso

No soy un robot  reCAPTCHA
Privacidad · Condiciones

Nuevo Guardar Cancelar

Figura 3.39 Opción de cancelación del usuario del sistema.

El menú de “Consultas” permite obtener información general de los usuarios, empresas, departamentos y variables registrados en las plantillas del sistema.

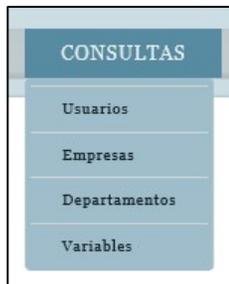


Figura 3.40 Opciones de consultas del sistema.

La opción de consulta de usuarios muestra el detalle de los usuarios registrados en el sistema. A esta opción solo tiene acceso el usuario Administrador.

The screenshot shows the 'Usuarios' query page. It features a table with the following data:

Nombres	Apellidos	Usuario	Empresa	Estado
Anthony	Barona	leonardtony_s6@hotmail.com	BioMar Ecuador	A

Below the table, it indicates 'Total Usuario(s) = 1' and provides 'PDF' and 'Cancelar' buttons.

Figura 3.41 Opción de consulta de usuarios del sistema.

Las opciones de consultas de empresas y departamentos muestran la información general de esas entidades.

The screenshot shows the 'Empresas' query page. It features a table with the following data:

Identificador	Empresas
1	BioMar Ecuador
2	ESPOL
3	FIEC
4	MACI V
5	Invitado

Below the table, it indicates 'Total Empresa(s) = 5' and provides 'PDF' and 'Cancelar' buttons.

Figura 3.42 Opción de consulta de empresas del sistema.

La opción para consulta de variables muestra el detalle de las variables de monitoreo y control que son utilizadas en el proyecto, tal como se ve en la Figura 3.43.

Identificador	Variables	Unidad	Valor Mínimo	Valor Máximo
1	Amperaje	A	100.00000000	250.00000000
2	Frecuencia	Hz	10.00000000	25.00000000
3	Presion	PSI	20.00000000	35.00000000
4	TemperaturaDADO	°C	74.00000000	95.00000000
5	TemperaturaDDC	°C	74.00000000	95.00000000
6	TemperaturaAmbiente	°C	0.00000000	40.00000000

Total Variable(s) = 6

PDF Cancelar

Figura 3.43 Opción de consulta de variables del sistema.

La opción de “Reportes” del sistema presenta dos tipos de resultados: Detallados y Gráficos.



Figura 3.44 Opciones de reportes del sistema.

Las ventanas previas a la obtención de los resultados para reportes detallados o gráficos son muy similares, solo difieren en los tipos de reporte a mostrar por parte de la opción de gráficos debido a que maneja tres tipos de presentaciones: “Lineal”, “Barras” y “Áreas”.

Las ventanas de generación de reportes poseen una opción para escoger el tipo de consulta la misma que puede ser: “General” o “Promedios”, es importante considerar que la opción “General” realiza la consulta de toda la información registrada en el sistema por cada variable y el tiempo de respuesta de la consulta puede ser muy

largo, en promedio un mes de consulta puede tardar 15 minutos en procesar la presentación de datos, solo es recomendable utilizar en caso de ser necesario.

La opción recomendada para el manejo de la consulta es la de “Promedios”, esta presentación obtiene los datos registrados, pero promediando los mismos por cada hora de proceso de esta forma el volumen de resultados es menor y se puede obtener el reporte rápidamente, así como también leer de mejor manera los resultados.

Las pantallas de generación de reportes se pueden ver en las Figuras 3.45 y 3.46.

The screenshot shows a web form titled "Reporte Detallado de Variables". It contains the following fields and controls:

- Variable(s)**: A dropdown menu with "Todas las variables" selected, followed by an asterisk (*) indicating it is required.
- Fecha inicial**: A date input field with a calendar icon, followed by an asterisk (*). Below the field is the label "día/mes/año".
- Fecha final**: A date input field with a calendar icon, followed by an asterisk (*). Below the field is the label "día/mes/año".
- Tipo de consulta**: A dropdown menu with "Escoja un tipo" selected, followed by an asterisk (*).

Below the fields, there is a note: "(*) Datos obligatorios." and three buttons: "Nuevo", "Consultar", and "Cancelar".

Figura 3.45 Generación de reportes detallados del sistema.

The screenshot shows a web form titled "Reporte Gráfico de Variables". It contains the following fields and controls:

- Variable(s)**: A dropdown menu with "Todas las variables" selected, followed by an asterisk (*) indicating it is required.
- Fecha inicial**: A date input field with a calendar icon, followed by an asterisk (*). Below the field is the label "día/mes/año".
- Fecha final**: A date input field with a calendar icon, followed by an asterisk (*). Below the field is the label "día/mes/año".
- Tipo de consulta**: A dropdown menu with "Escoja un tipo" selected, followed by an asterisk (*).
- Tipo de reporte**: A dropdown menu with "Escoja un tipo" selected, followed by an asterisk (*).

Below the fields, there is a note: "(*) Datos obligatorios." and three buttons: "Nuevo", "Consultar", and "Cancelar".

Figura 3.46 Generación de reportes gráficos del sistema.

Los reportes detallados del sistema se muestran en formato PDF como en la Figura 3.47.

Impreso el: 28/01/2020 12:31:43
Por: linaarroy_96@hcmex.com

PORTAL WEB DE MONITOREO DE SEÑALES
REPORTE DETALLADO DE REGISTRO DE VARIABLES
DATOS PROMEDIADOS POR HORA
PERIODO DEL: 09/01/2020 AL: 09/01/2020

VARIABLE: AMPERAJE

Identificador	Fecha Registro	Valor
1	2020-01-09 00:00:00	135.18393714
1	2020-01-09 01:00:00	122.54319373
1	2020-01-09 02:00:00	118.95647076
1	2020-01-09 03:00:00	124.12360151
1	2020-01-09 04:00:00	126.27260193
1	2020-01-09 05:00:00	110.59659125
1	2020-01-09 06:00:00	117.26570512
1	2020-01-09 07:00:00	121.28823315
1	2020-01-09 08:00:00	133.44396969
1	2020-01-09 09:00:00	126.50877327
1	2020-01-09 10:00:00	124.20423185
1	2020-01-09 11:00:00	111.14374712
1	2020-01-09 12:00:00	110.44112174
1	2020-01-09 13:00:00	116.93060961
1	2020-01-09 14:00:00	125.96915616
1	2020-01-09 15:00:00	118.73909681
1	2020-01-09 16:00:00	119.19128456
1	2020-01-09 17:00:00	124.69777601
1	2020-01-09 18:00:00	113.01014210
1	2020-01-09 19:00:00	115.39930593

Figura 3.47 Muestra de reporte detallado del sistema.

Los reportes gráficos del sistema se muestran en la ventana de este tal como en la Figura 3.48.

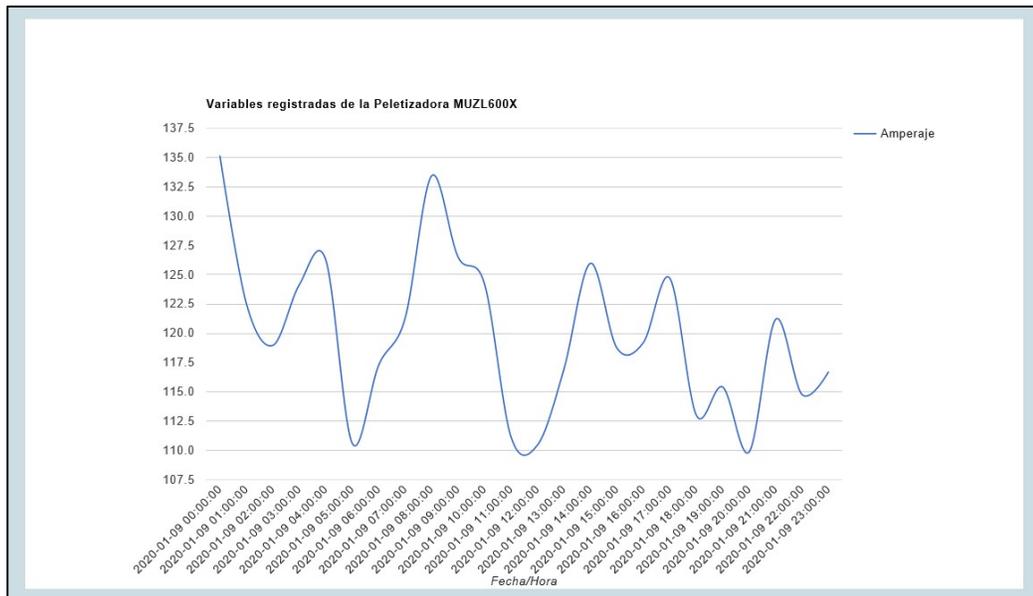


Figura 3.48 Muestra de reporte gráfico del sistema.

Finalmente, la opción “Salir” permite cerrar la sesión del usuario en el sistema para seguridad de su información.

3.3 Diseño de las comunicaciones del controlador difuso de temperatura

El presente proyecto tiene como objetivo monitorear las señales de una peletizadora de una planta de producción de alimento balanceado para camarón. Para cumplir con el fin requiere de la instalación de dispositivos IOT conectados al panel eléctrico de la planta donde se encuentra el PLC que controla la automatización de los equipos.

Debido al entorno de trabajo se colocará en una caja de control sencilla los dispositivos para el monitoreo de la peletizadora y dentro de la caja de control se requiere controlar la temperatura ambiente de la misma para poder mantenerla en 23 grados centígrados que corresponde a la temperatura recomendada por los expertos en administración de Centros de Datos informáticos y que se aplicará para este proyecto.

Para diseñar el controlador se utilizará un esquema de conexión que permita medir la temperatura ambiente a través de un sensor de temperatura LM35.

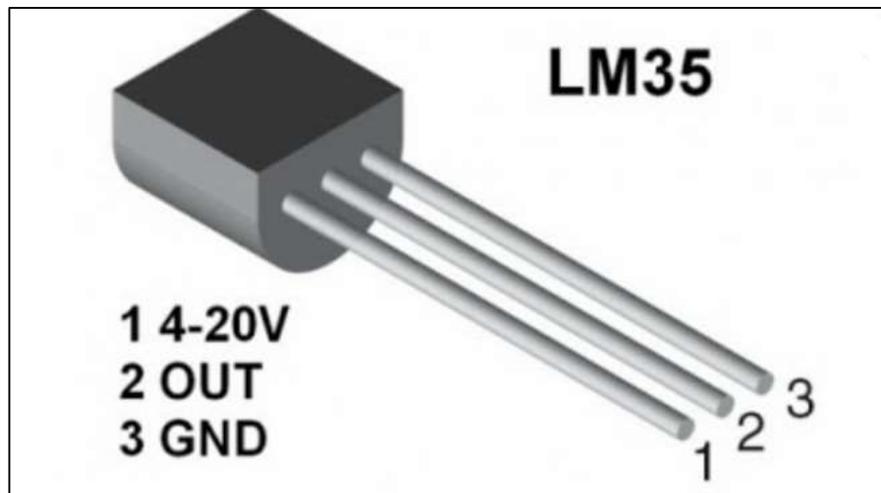


Figura 3.49 Sensor de temperatura LM35.

El sensor LM35 de temperatura, es un sensor de precisión en un circuito integrado que tiene salida de voltaje lineal y directamente proporcional a la temperatura en grados centígrados. Este sensor tiene como principal ventaja sobre otros sensores de escala en grados Kelvin pues no requiere de calibración externa.

Tiene un rango de medición de +2 °C hasta +150 °C, precisión de +/- 0.9 °C y una ganancia de +10 mv/°C. [10]

Como complemento a las definiciones del sensor LM35 es importante tener en consideración que actualmente este dispositivo puede medir temperaturas entre -55 °C y 150 °C lo que lo convierte en un sensor muy utilizado en la industria para diferentes objetivos debido a su fácil manejo y excelentes resultados a la hora de leer las temperaturas requeridas.

Este sensor estará conectado a un Arduino UNO utilizando el pin análogo A0 para receptor por medio de un programa hecho en Arduino el valor de la temperatura al interior de la caja de control, esta información es enviada a través del puerto serial (USB) para que luego sea leída por la microcomputadora Raspberry PI 3 B+ y ser enviada a la Base de Datos en la nube donde se encuentra alojado el sistema de monitoreo de señales de la peletizadora.

Esto significa que se utilizará el mismo esquema de comunicaciones empleado en la lectura de señales de la peletizadora, la diferencia radica en que la información aquí será tomada del ambiente a partir del sensor de temperatura.

El programa realizado en Arduino UNO se denomina “**ControladorFuzzyTemperaturaAr.ino**”, este archivo contiene el código del controlador difuso y para la lectura de la temperatura presenta dos partes de trabajo en su contenido.

```
// Declaracion de variables globales

int pinA0 = 0; //Asigna 0 a la variable pinA0. A este pin se conecta el voltaje de entrada del sensor de temperatura
int val;      //int representa un entero (integer, val es el valor leído por el sensor

//Esta función corre una sola vez, al inicio del controlador

void setup()
{
    //Inicia el puerto serial para posteriormente observar la temperatura (serial monitor)
    Serial.begin(9600);
}
}
```

Figura 3.50 Programa en Arduino para lectura de temperatura ambiente (1).

La primera parte se observa en la Figura 3.50 y consiste en la definición de la variable para inicializar el pinA0 donde se recibirá la temperatura ambiente y la configuración de la velocidad de trabajo del puerto Serial del Arduino.

La segunda parte consiste en el proceso de lectura de la temperatura ambiente y su envío al puerto serial del Arduino. Esta parte del programa se puede observar en la Figura 3.51.

La lectura de esta variable debido a que es considerada una variable de cambio lento en el tiempo y por la toma de muestra en campo se ha definido como tiempo de muestreo 1 minuto.

```

//Esta función hace un bucle. Cuando se termina la rutina, vuelve al principio
void loop()
{
  //Obtiene los datos del sensor de temperatura
  val = analogRead(pinA0);           //Valor leído por el sensor de temperatura. Entre 0 y 1024
  float mv = (val/1024.0)*5000;      //Fórmula para transformar el valor "val" anterior a milivoltios (mv)
  float cel = mv/10;                //Fórmula que asigna la temperatura en Celsius (cel) al relacionarlo con los milivoltios (mv).
  float temperatura = cel;          //Asignación de la temperatura final calculada a la variable general de temperatura

  //Envía el dato final procesado de temperatura al puerto serial
  Serial.print(temperatura);
  //Imprime un salto de línea
  Serial.print("\n");
  //Esperamos un tiempo 1 minuto para repetir el loop
  delay(60000);
}

```

Figura 3.51 Programa en Arduino para lectura de temperatura ambiente (2).

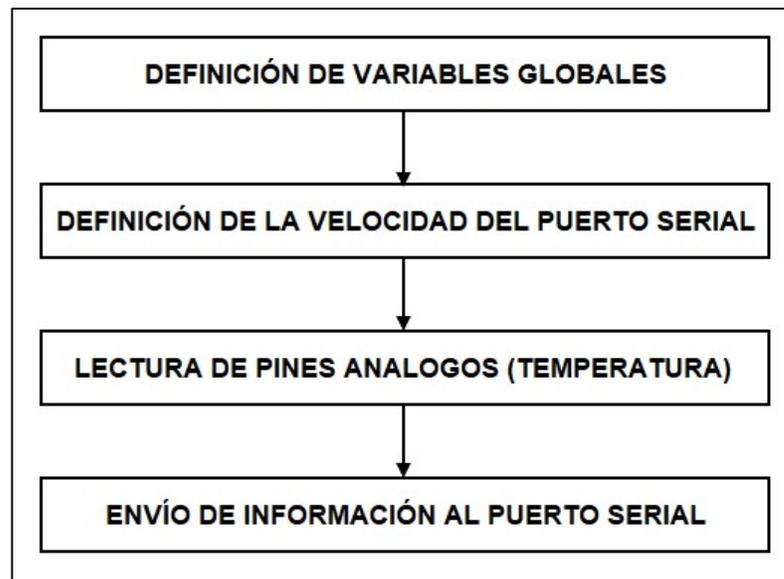


Figura 3.52 Proceso de lectura de temperatura en la caja de control IOT.

El programa desarrollado en Python y que trabaja sobre la plataforma Raspbian de la Raspberry PI 3 B+ presenta tres partes que corresponde a la declaración e importación de las librerías a utilizar en el proceso de lectura de datos del puerto serial, la configuración de la velocidad de trabajo y el tiempo de toma de lectura del puerto serial y finalmente el proceso de envío de la información a la nube para el registro en la base de datos.

```
import serial
import time
import MySQLdb
from urllib.parse import urlencode
from urllib.request import Request, urlopen

arduino=serial.Serial("/dev/ttyACM0",baudrate=9600,timeout=65.0)

arduino.setDTR(False)
time.sleep(1)
arduino.flushInput()
arduino.setDTR(True)

url = "http://www.iotsignalcontrol.com/iotsignalcontrol/Data/RegistroTemperaturaAmbienteBd.php"

while True:
    val=arduino.readline()
    if val != '\n':
        try:
            data = val.decode('utf-8')
            print (data)#TemperaturaAmbiente
            post_fields = {"TemperaturaAmbiente":data}
            request = Request(url, urlencode(post_fields).encode())
            json = urlopen(request).read().decode()
        except ValueError:
            print ("")
    arduino.close()
```

Figura 3.53 Programa en Python para la lectura de la temperatura ambiente.

El programa desarrollado en Python se denomina “ResgistroTemperturaAmbientePy.py” y se muestra su contenido en la Figura 3.53. En la Figura 3.54 se presenta el resumen del proceso de envío de información a la nube como diagrama de proceso.

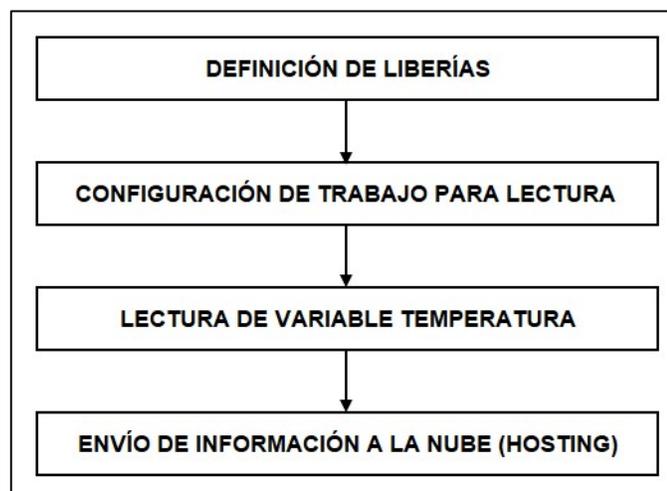


Figura 3.54 Proceso de envío de temperatura de la caja de control IOT a la nube.

Finalmente, para completar el registro de la información en la Base de Datos del sistema se utiliza un archivo desarrollado en PHP y que se aloja en la nube al mismo que recibe los datos desde la Raspberry por el método “POST”.

El archivo se denomina “**RegistroTemperaturaAmbienteBd.php**” y contiene tres partes, primero la recepción de la variable que envía la Raspberry, segundo la configuración de la conexión a la Base de Datos y finalmente el proceso de inserción de la variable en la tabla correspondiente. Esto se puede ver en la Figura 3.55.

```

<?php session_start();

//OBTENCIÓN DE DATOS DESDE PYTHON (RASPBERRY PI 3 B+)
$TemperaturaAmbiente = $_POST['TemperaturaAmbiente'];

//ASIGNACIÓN DE LA FECHA DEL DÍA
$FechaActual = date ('Y/m/d H:i:s');

//PROCESO DE INSERCIÓN DEL VALOR DE LA TEMPERATURA AMBIENTE
if($TemperaturaAmbiente > 0)
{
    //INICIALIZACIÓN DE CONEXIÓN PARA INTERACTUAR CON LA BD
    include_once('../Connections/TransaccionBd.php');

    $Bd = new TransaccionBd();
    $Bd->AbrirConexion();

    //CREACIÓN DE SCRIPT PARA INSERCIÓN DEL VALOR DE LA TEMPERATURA AMBIENTE
    $SqlInsert = "INSERT INTO pwiot_tblregistrovariables (Id_variable,Fecha_registro,Valor) ";
    $SqlInsert.= "VALUES ('6','$FechaActual','$TemperaturaAmbiente')";

    $Resultado = $Bd->GrabarInformacion($SqlInsert);

    $Bd->CerrarConexion();
}
-?>

```

Figura 3.55 Programa en PHP para el registro de la temperatura ambiente.

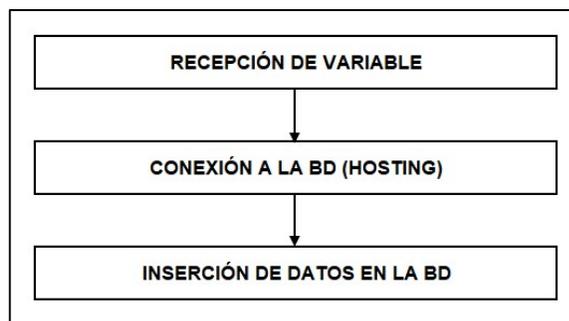


Figura 3.56 Proceso de registro de temperatura de la caja IOT en la BD.

Con la información registrada en el sistema para el seguimiento de parte del usuario se procederá a utilizar como elemento para levantar la lógica del controlador difuso de temperatura a la misma placa Arduino UNO que realiza la lectura de la temperatura ambiente al interior de la caja de control y se utilizará como elemento actuador para regular la temperatura ambiente una Placa Peltier que realizará el efecto de un aire acondicionado.

El modelo escogido para este proyecto es el TEC1-12706, para este trabajo estará equipado con un disipador y los elementos extractores del aire frío y caliente como los coolers adicionales que para el fin trae el dispositivo Peltier.



Figura 3.57 Celda de Peltier TEC1-12706.

El Efecto Peltier bajo el que se basa el funcionamiento de este dispositivo fue revisado en el Capítulo 2.

Las especificaciones técnicas de este equipo indican que sus dimensiones físicas son 40*40*3.75 mm, su potencia de trabajo requerida es de 72W y la corriente que necesita para rendir correctamente es de 6A y 12 VDC.

Este pequeño aire acondicionado estará conectado en un circuito electrónico con un transistor Mosfet lógico IRLB8748PBF, este transistor puede ser activado con voltajes pequeños que fluctúan entre 0-5 V como los que provee el Arduino, el transistor a su vez será activado en modo interruptor desde un pin PWM de la placa Arduino, específicamente para este proyecto se utilizará el pin 9. De por medio entre el Arduino y el transistor se utilizará una resistencia de 100K para regular el proceso de activación del transistor y evitar que se caliente rápidamente. El transistor utilizará un disipador de calor para contrarrestar aún más el efecto de la corriente de 6A proveniente de una fuente de alimentación a modo de cargador eléctrico.

El objetivo del controlador es a través del circuito ya descrito, regular la potencia de trabajo del Peltier por medio de anchos de pulsos variados que enviará el pin PWM del Arduino al transistor, así y en base a la lógica implementada en el controlador difuso según las muestras y la evaluación en el experimento previo en el ambiente de trabajo determinado en el proyecto se podrá mantener los valores adecuados de temperatura al interior de la caja de control.

En este circuito uno de los principales elementos para cumplir con el trabajo del controlador difuso es el transistor Mosfet lógico IRLB8748PBF que a la larga es quién regulará la potencia de uso del Peltier para enfriar o no el ambiente de la caja de control.

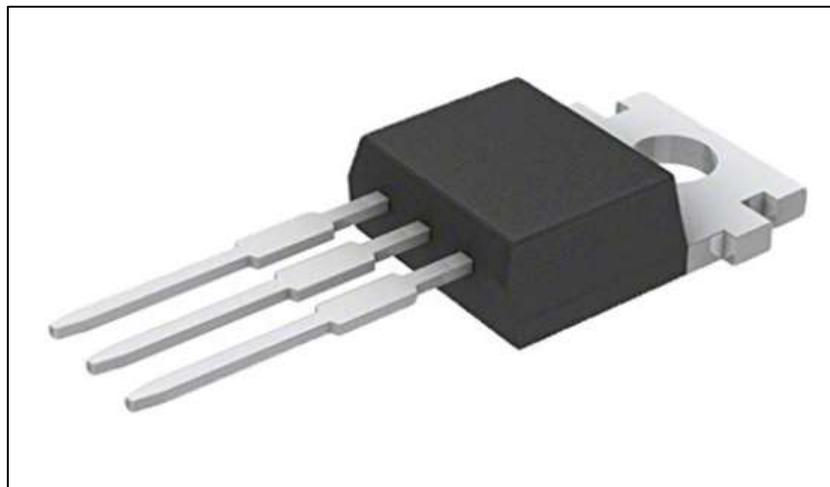


Figura 3.58 Transistor Mosfet lógico IRLB8748PBF.

Este transistor se define como lógico debido a que puede ser activado utilizando voltajes muy pequeños que van desde los 0.6 V hasta los 12 V. En el caso específico de este transistor, la transmisión de corriente se activa desde 4.5 V lo que lo hace ideal para trabajar con el circuito de control planteado y que tiene como plataforma para la implementación de la lógica la placa Arduino que trabaja con rangos desde 3.3 V a 12 V.

Soporta además corrientes de hasta 92A a 25 °C, pero para el experimento solo se necesita utilizar 6A que es la corriente que emplea el Peltier para funcionar a su máxima capacidad.

Según la International Rectifier (importante empresa norteamericana que fabricaba circuitos integrados para la tecnología) las características principales del Mosfet lógico IRLB8748PbF son las que se muestran en la Figura 3.59.

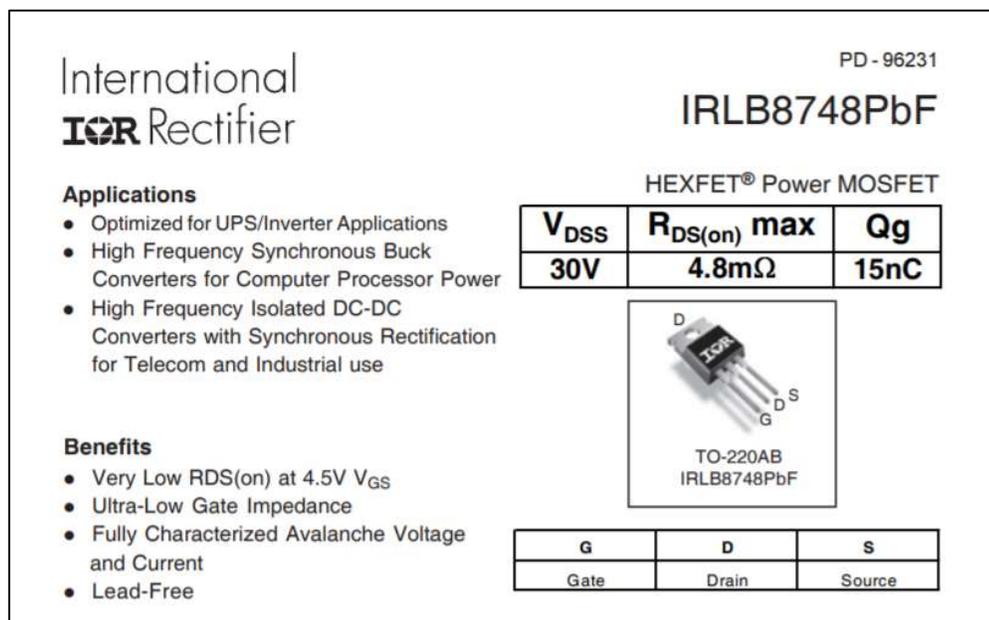


Figura 3.59 Características del Transistor Mosfet lógico IRLB8748PbF. [11]

Con toda la información indicada previamente podemos claramente tener ya la idea de la forma del circuito que se realizará para operar el controlador y las acciones que se deriven del mismo.

Ahora veremos una imagen real de la forma en que se conectaron los dispositivos según el diseño del circuito para operar el controlador difuso de temperatura. Figura 3.60.

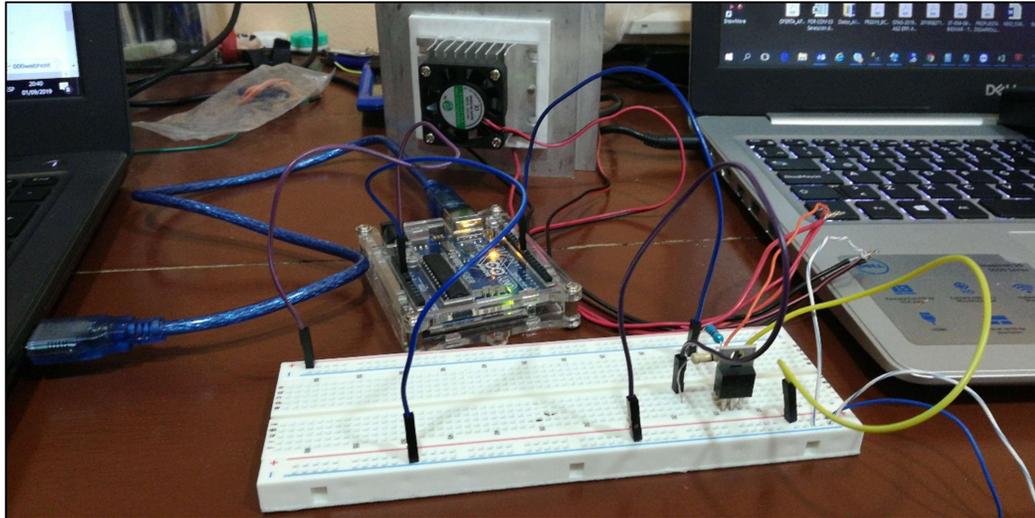


Figura 3.60 Circuito del controlador difuso de temperatura.

De esta forma se cerrará el ciclo de trabajo del proyecto y se cumplirá con los objetivos planteados en el mismo: monitorear las señales de la peletizadora y mantener la temperatura ambiente en zona de confort de la caja de control de los dispositivos IOT que monitorean la peletizadora.

3.3.1 Definición de variables

Para diseñar el controlador difuso de temperatura de este proyecto se utilizará como variables lingüísticas del sistema una variable de entrada y una de salida.

La variable de entrada considerada para el controlador es la diferencia entre el Set Point de la temperatura ambiente en el interior e la caja de control que esta parametrizado en 23 °C por estándar de temperaturas dentro de Centro de Datos y el valor de la temperatura medido por el Sensor LM35. Esto da como resultado el valor del error que será la primera variable.

La segunda variable lingüística del sistema será la salida del controlador y que en este caso debido a que se utilizará a la placa Arduino UNO como plataforma de control

actuando sobre una Placa Peltier por medio de un pin PWM, la segunda variable de trabajo queda definida como el Ancho de Pulso suministrado por la placa.

Set Point	Temperatura en Caja de Control	Rangos del Error	Función de Membresía
23 °C	11 °C - 3 °C	De +12 a +20	PG - Mucho Frío
23 °C	21 °C - 11 °C	De +2 a +12	PP - Poco Frío
23 °C	25 °C - 21 °C	De -2 a +2	Z - Set Point
23 °C	35 °C - 25 °C	De -12 a -2	NP - Poco Calor
23 °C	43 °C - 35 °C	De -20 a -12	NG - Mucho Calor

Tabla 3: Definición de variable lingüística de entrada del controlador difuso.

% Ciclo de Trabajo	Salida PWM Arduino UNO	Voltaje Suministrado	Función de Membresía
0% - 10%	0 - 25.5	0 - 0.5	Nulo
10% - 30%	25.5 - 76.5	0.5 - 1.5	Bajo
30% - 70%	76.5 - 178.5	1.5 - 3.5	Medio
70% - 100%	178.5 - 255	3.5 - 5	Alto

Tabla 4: Definición de variable lingüística de salida del controlador difuso.

A partir de la definición de las variables del controlador se pueden definir las reglas del diseño para mediante la combinación de las variables de entrada y salida obtener los resultados deseados de ajuste de la temperatura ambiente en el interior de la caja de control. Esto se puede ver en la Tabla 5.

Error Temperatura	PG	PP	Z	NP	NG
Ancho de Pulso	Nulo	Nulo	Bajo	Medio	Alto

Tabla 5: Reglas del controlador difuso.

3.3.2 Diseño del controlador en Matlab

Luego de tener definidas las variables a utilizar en el diseño del controlador, tanto las variables de entrada y salida, así como las reglas de combinación de las variables para obtener los resultados del controlador se puede utilizar Matlab para realizar el diseño del controlador difuso y obtener una idea de cómo puede funcionar el mismo.

En este diseño se ha utilizado la herramienta Fuzzy de Matlab 2018A.

El archivo que contiene el diseño del controlador difuso en Matlab se denomina “**Controlador_Fuzzy_Temperatura.fis**”. En este archivo se han incluido las variables lingüísticas y las reglas indicadas previamente, de tal forma que tenemos a continuación las diferentes definiciones de esas variables y los resultados de las reglas del controlador en las Figuras que siguen en el documento.

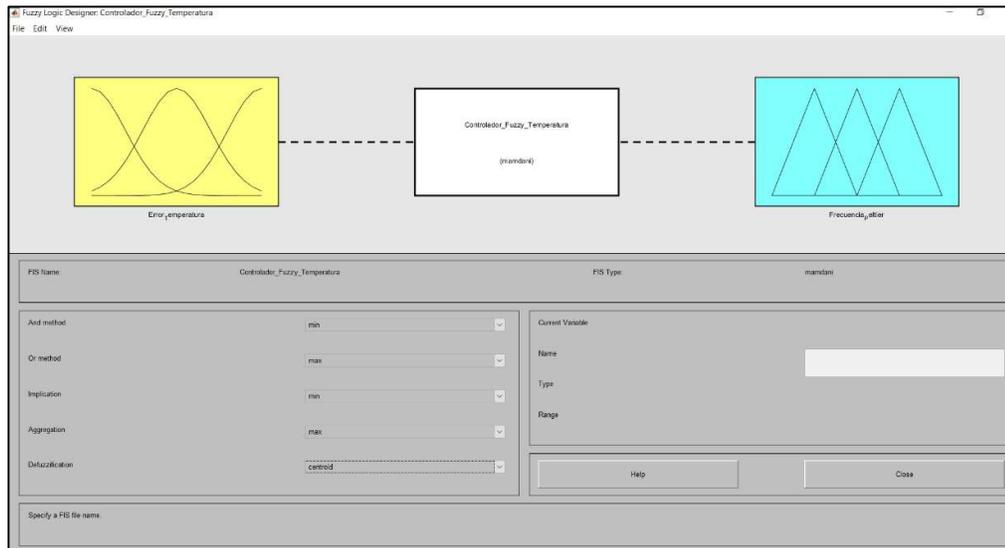


Figura 3.61 Controlador Fuzzy en Matlab.

En la Figura 3.61 podemos observar la información general del controlador difuso, la variable de entrada ha sido denominada “**Error_Temperatura**”, la variable de salida es “**Frecuencia_Peltier**” y se ha escogido al método del centroide para realizar el proceso de defusificación.

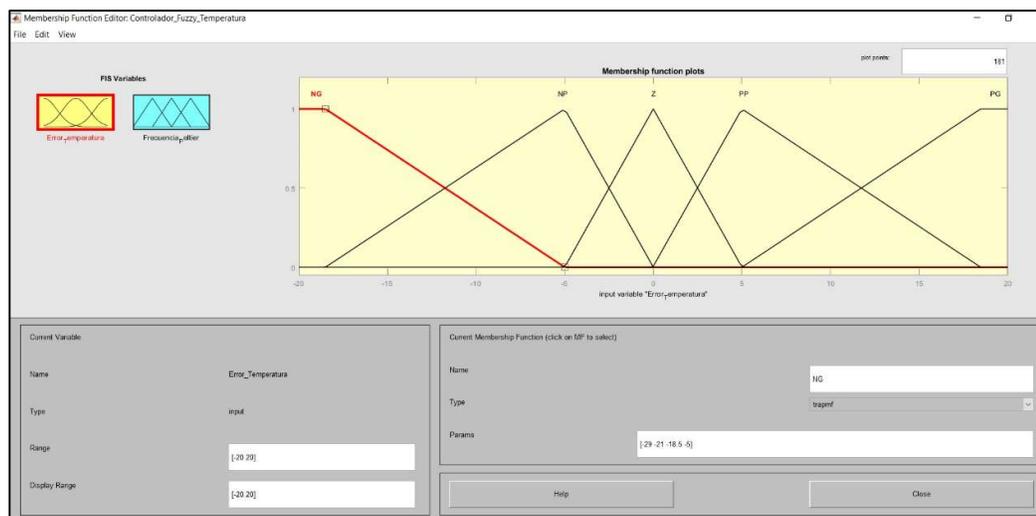


Figura 3.62 Función de Membresía NG (Error_Temperatura).

La función de membresía NG para la variable temperatura trabaja en el rango entre -29 y -5 y tiene forma de trapecio.

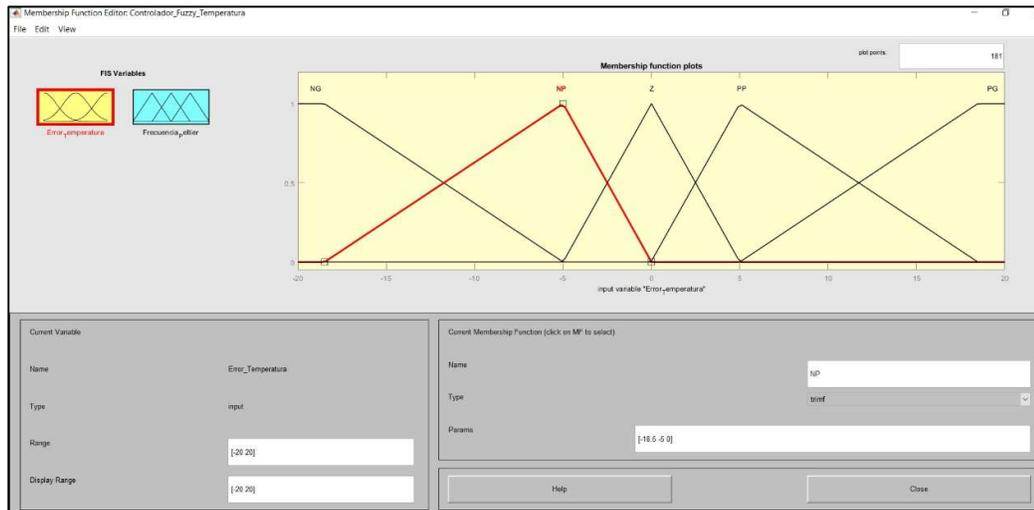


Figura 3.63 Función de Membresía NP (Error_Temperatura).

La función de membresía NP para la variable temperatura trabaja en el rango entre -18.5 y 0 y tiene forma de triángulo.

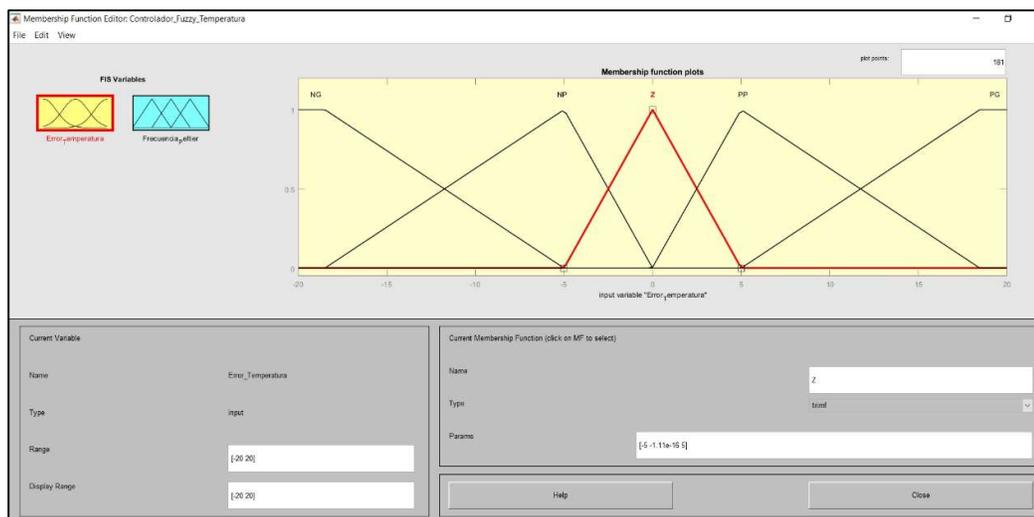


Figura 3.64 Función de Membresía Z (Error_Temperatura).

La función de membresía Z para la variable temperatura trabaja en el rango entre -5 y 5 y tiene forma de triángulo.

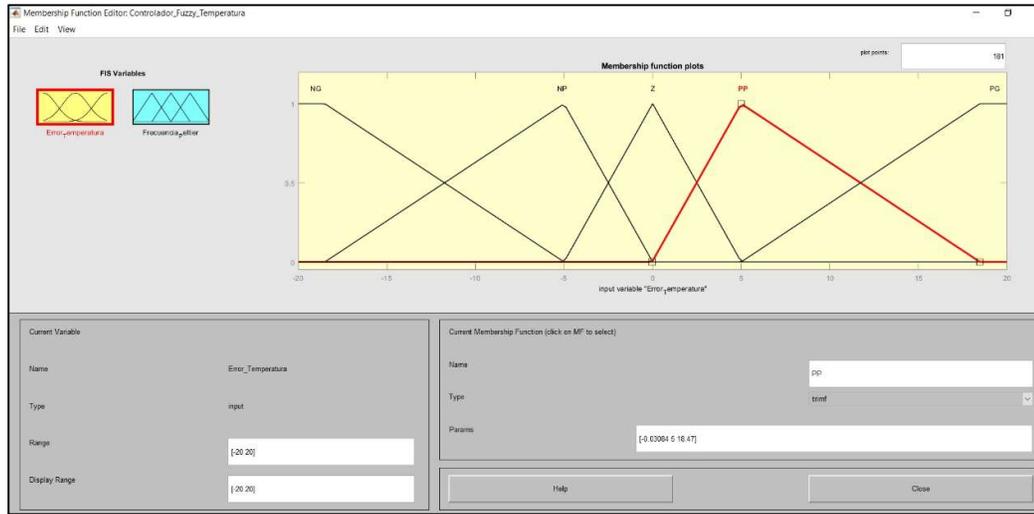


Figura 3.65 Función de Membresía PP (Error_Temperatura).

La función de membresía PP para la variable temperatura trabaja en el rango entre 0 y 18.47 y tiene forma de triángulo.

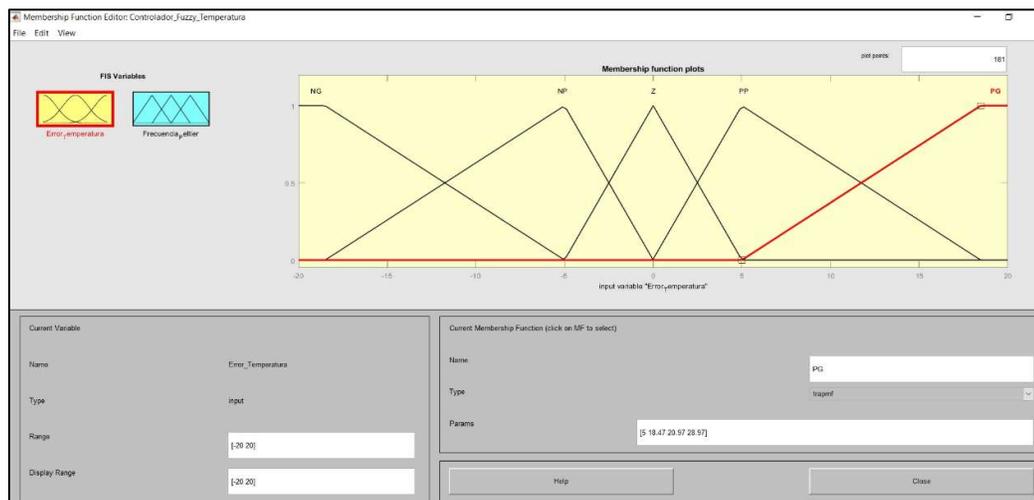


Figura 3.66 Función de Membresía PG (Error_Temperatura).

La función de membresía PG para la variable temperatura trabaja en el rango entre 5 y 28.97 y tiene forma de trapecio.

Ahora revisaremos las definiciones de las funciones de membresía de la variable de salida del controlador.

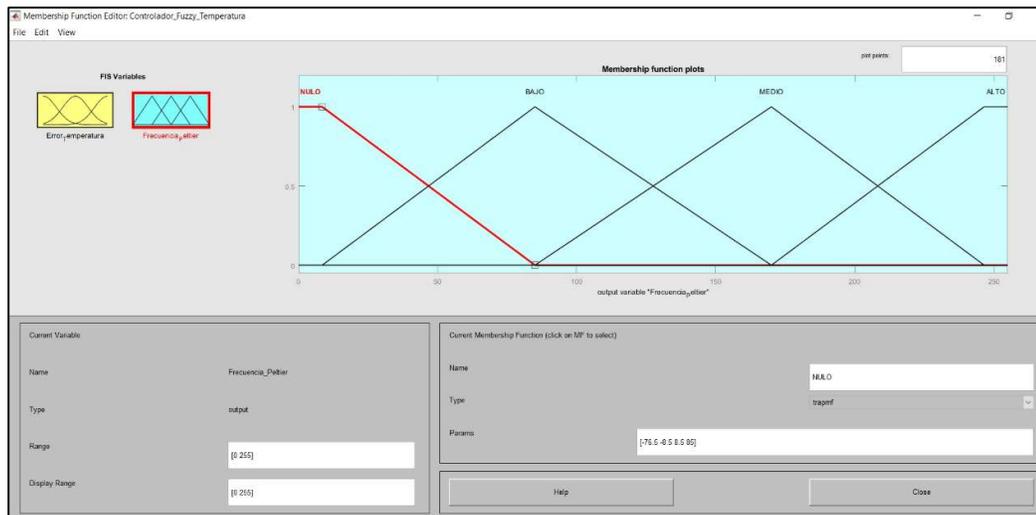


Figura 3.67 Función de Membresía NULO (Frecuencia_Peltier).

La función de membresía NULO para la variable de frecuencia trabaja en el rango entre -76.5 y 85 y tiene forma de trapecio.

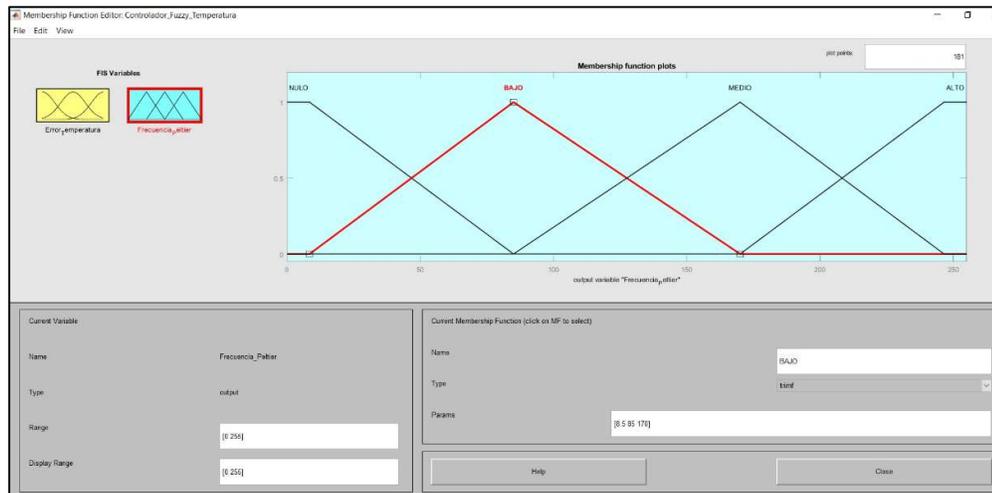


Figura 3.68 Función de Membresía BAJO (Frecuencia_Peltier).

La función de membresía BAJO para la variable de frecuencia trabaja en el rango entre 8.5 y 170 y tiene forma de triángulo.

La función de membresía MEDIO para la variable de frecuencia trabaja en el rango entre 85 y 246.5 y tiene forma de triángulo.

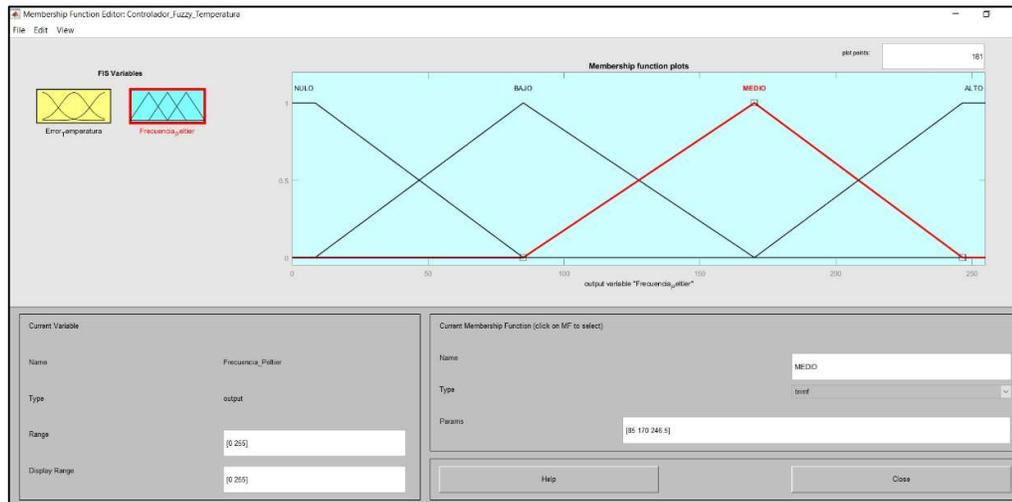


Figura 3.69 Función de Membresía MEDIO (Frecuencia_Peltier).

La función de membresía ALTO para la variable de frecuencia trabaja en el rango entre 170 y 331.5 y tiene forma de trapecio.

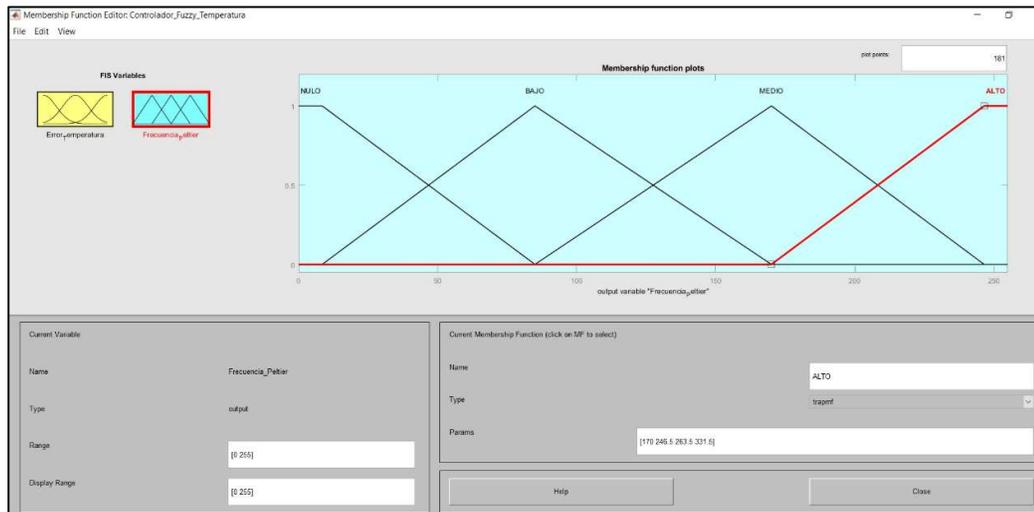


Figura 3.70 Función de Membresía ALTO (Frecuencia_Peltier).

A partir de las variables de entrada y salida ya definidas y estudiadas previamente se proceden a realizar las reglas del controlador difuso de temperatura para a partir de allí evaluar los resultados de este y que persiguen como objetivo principal regular la temperatura ambiente de la caja de control IOT. Esto lo veremos en las siguientes figuras del documento.

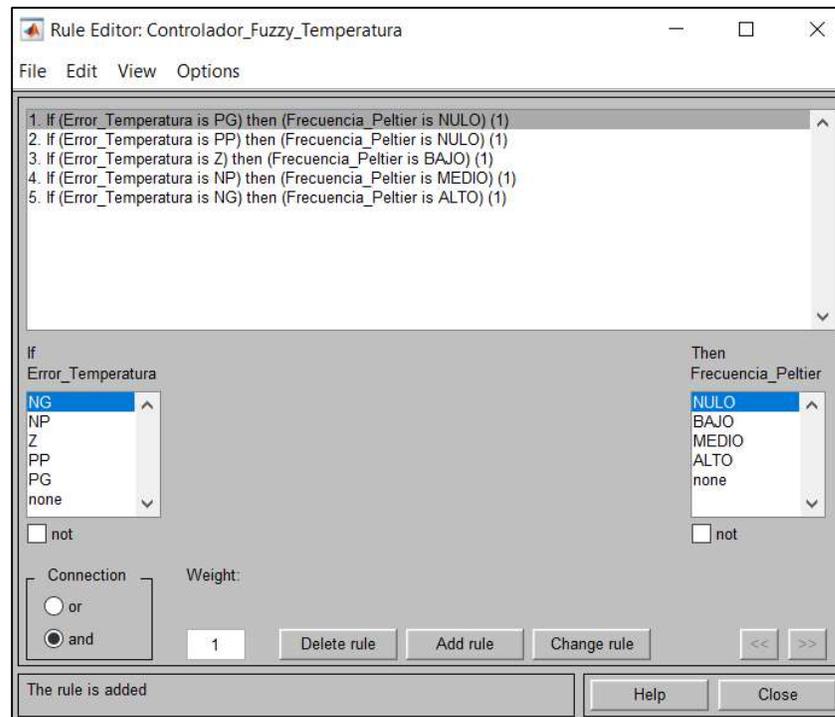


Figura 3.71 Reglas del controlador difuso de temperatura (1).

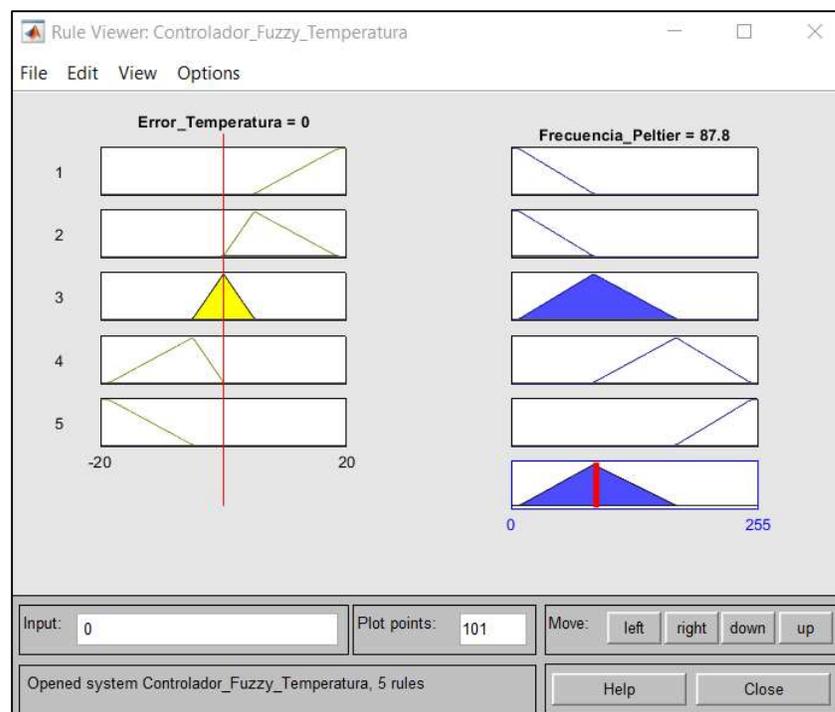


Figura 3.72 Reglas del controlador difuso de temperatura (2).

Finalmente, a través del método del centroide se realiza la defusificación y se obtiene el resultado deseado para que el controlador difuso tenga efecto sobre la temperatura ambiente de la caja de control IOT.

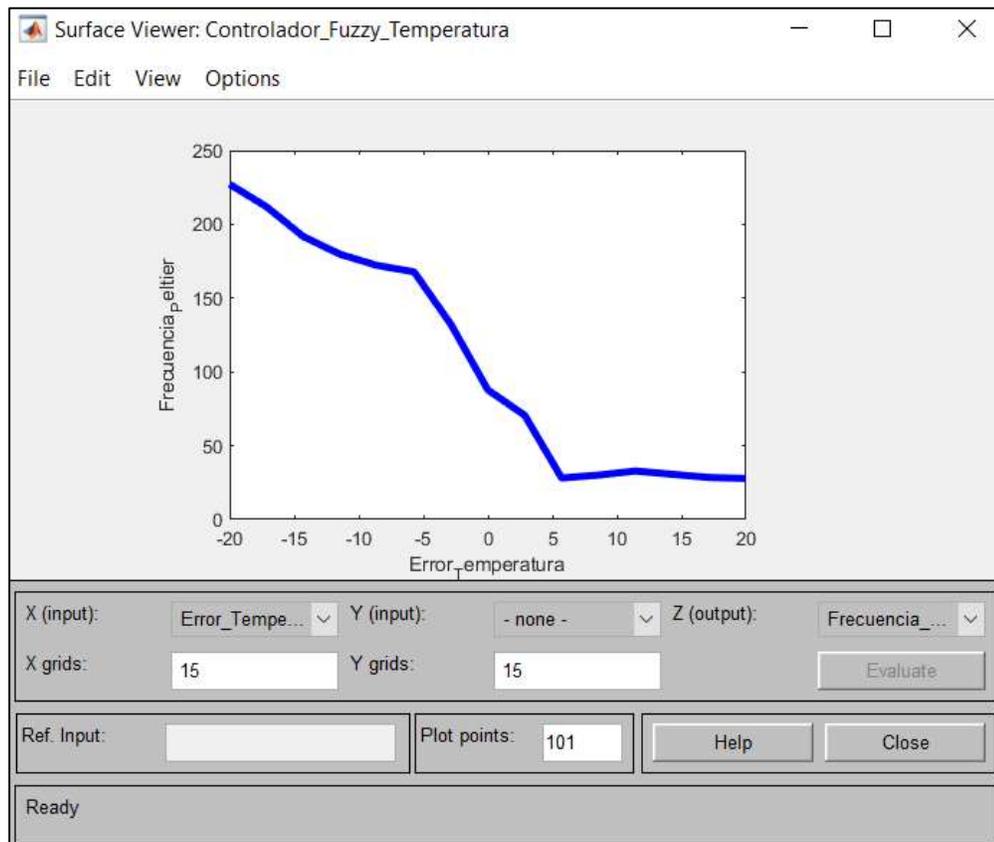


Figura 3.73 Defusificación del controlador por el método del centroide.

3.3.3 Programa en Arduino UNO

El diseño del controlador en Matlab permite determinar los rangos de trabajo de cada función de membresía para las variables lingüísticas, a partir de aquí y en conjunto con la combinación de las reglas definidas en el mismo diseño se puede realizar el programa en código Arduino que permitirá ejecutar las acciones del controlador difuso de temperatura para trabajar a través de la Celda de Peltier la regulación de la temperatura ambiente al interior de la caja de control de los dispositivos IOT.

El programa diseñado ya lo revisamos previamente en este documento para explicar la lectura de la temperatura a través del sensor LM35 y se denomina “ControladorFuzzyTemperaturaAr.ino”.

La primera parte del programa del controlador difuso en Arduino contiene la importación de las librerías difusas que permitirán ejecutar el controlador, así como la definición de las variables globales del programa y la configuración de la velocidad de trabajo del puerto serial. La figura 3.74 muestra el detalle indicado.

```
//Controlador Difuso de Temperatura utilizando Celda de Peltier
#include <Fuzzy.h>
#include <FuzzyComposition.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzyOutput.h>
#include <FuzzyRule.h>
#include <FuzzyRuleAntecedent.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzySet.h>

// Declaracion de variables globales

//Temperatura
int pinA0 = 0; //Asigna 0 a la variable pinA0. A este pin se conecta el voltaje de entrada del sensor de temperatura
int val; //int representa un entero (integer, val es el valor leído por el sensor

//Controlador
Fuzzy*control = new Fuzzy(); //Se declara el objeto fuzzy, se llamará, control
int set_point; //Variable desde la que se controla el set point del sistema
int ERROR; //Es la resta entre el Set point y la temperatura leída por el sensor
int POTENCIA; //Variable en la cual se guarda el nivel de potencia del PWM

void setup()
{
  Serial.begin(9600); //Se inicia la transmisión serial
  pinMode(9,OUTPUT); //Declaro los pines de salida del PWM
```

Figura 3.74 Programa del controlador difuso de temperatura (1).

Luego se definen las variables de entrada del sistema que en este caso se trata del error de las temperaturas entre el Set Point y la temperatura ambiente leída por el sensor. Figura 3.75.

Se definen de igual manera las variables de salida (Figura 3.76) que para el caso del controlador es la frecuencia moduladora (PWM) de la potencia de trabajo de la Celda de Peltier que estará conectada al pin 9 del Arduino y que regulará la corriente suministrada al dispositivo Peltier a través de un transistor Mosfet de características lógicas indicado previamente en este documento.

Cada variable presenta sus rangos de trabajo en base al diseño del controlador previo.

```

//Se declara el universo de entrada
FuzzyInput*error = new FuzzyInput(1); //El universo de entrada será el error, el error se define como Set Point-Temperatura
//Se establecen los conjuntos del universo ERROR con sus respectivas funciones de membresia
////////////////////////////////////
FuzzySet*ng = new FuzzySet(-29, -21, -18.5, -5); //Negativo Grande, trapecio.
error->addFuzzySet(ng); //Se añade el conjunto al universo de entrada.
////////////////////////////////////
FuzzySet*np = new FuzzySet(-18.5, -5, 0); //Negativo Pequeño, triangulo.
error->addFuzzySet(np); //Se añade el conjunto al universo de entrada.
////////////////////////////////////
FuzzySet*cero = new FuzzySet(-5, 0, 5); //Cero, triangular.
error->addFuzzySet(cero); //Se añade el conjunto al universo de entrada.
////////////////////////////////////
FuzzySet*pp = new FuzzySet(0, 5, 18.47); //Positivo Pequeño, triangulo.
error->addFuzzySet(pp); //Se añade el conjunto al universo de entrada.
////////////////////////////////////
FuzzySet*pg = new FuzzySet(5, 18.47, 20.97, 28.97); //Positivo Grande, trapecio.
error->addFuzzySet(pg); //Se añade el conjunto al universo de entrada.
////////////////////////////////////

control->addFuzzyInput(error); //Se agrega al sistema fuzzy el universo con sus conjuntos.

```

Figura 3.75 Programa del controlador difuso de temperatura (2).

```

//Se declara la salida del sistema, en este caso será la POTENCIA
FuzzyOutput*potencia = new FuzzyOutput(1);
//Se crean los conjuntos pertenecientes a este universo, junto a sus funciones de membresia
////////////////////////////////////
FuzzySet*nula = new FuzzySet(-76.5, -8.5, 8.5, 85); //Potencia nula, trapecio.
potencia->addFuzzySet(nula); //Se añade el conjunto al universo de salida.
////////////////////////////////////
FuzzySet*baja = new FuzzySet(8.5, 85, 170); //Potencia baja, trapecio.
potencia->addFuzzySet(baja); //Se añade el conjunto al universo de salida.
////////////////////////////////////
FuzzySet*media = new FuzzySet(85, 170, 246.5); //Potencia media, triangular.
potencia->addFuzzySet(media); //Se añade el conjunto al universo de salida.
////////////////////////////////////
FuzzySet*alta = new FuzzySet(170, 246.5, 263.5, 331.555); //Potencia alta, trapecio.
potencia->addFuzzySet(alta); //Se añade el conjunto al universo de salida.
////////////////////////////////////

control->addFuzzyOutput(potencia); //Se agrega el universo de salida con sus conjuntos al Sistema Fuzzy

```

Figura 3.76 Programa del controlador difuso de temperatura (3).

Luego de trabajar con la definición de las variables del controlador se procede realizar la configuración de las reglas. Figuras 3.77 y 3.78

Para este programa se han utilizado cinco reglas que permiten relacionar las variables de entrada y salida.

```

//Se establecen las reglas, cada regla debe tener un antecedente, una consecuencia y una asignación.

////////////////////////////////////
////////////////////////////////////Si error es PG(POSITIVO GRANDE), entonces potencia es NULA////////////////////////////////////
FuzzyRuleAntecedent*si_error_es_pg = new FuzzyRuleAntecedent();
si_error_es_pg->joinSingle(pg); //Antecedente

FuzzyRuleConsequent*entonces_potencia_es_nula = new FuzzyRuleConsequent();
entonces_potencia_es_nula->addOutput(nula); //Consecuencia

FuzzyRule*regla_1 = new FuzzyRule(1, si_error_es_pg,entonces_potencia_es_nula);
control->addFuzzyRule(regla_1); //Asignación
////////////////////////////////////

////////////////////////////////////Si error es PP(POSITIVO PEQUEÑO), entonces potencia es NULA////////////////////////////////////
FuzzyRuleAntecedent*si_error_es_pp = new FuzzyRuleAntecedent();
si_error_es_pp->joinSingle(pp); //Antecedente

FuzzyRuleConsequent*entonces_potencia_es_nula = new FuzzyRuleConsequent();
entonces_potencia_es_nula->addOutput(nula); //Consecuencia

FuzzyRule*regla_2 = new FuzzyRule(2, si_error_es_pp,entonces_potencia_es_nula);
control->addFuzzyRule(regla_2); //Asignación
////////////////////////////////////

```

Figura 3.77 Programa del controlador difuso de temperatura (4).

```

////////////////////////////////////Si error es Z(CERO), entonces potencia es BAJA////////////////////////////////////
FuzzyRuleAntecedent*si_error_es_cero = new FuzzyRuleAntecedent();
si_error_es_cero->joinSingle(cero); //Antecedente

FuzzyRuleConsequent*entonces_potencia_es_baja = new FuzzyRuleConsequent();
entonces_potencia_es_baja->addOutput(baja); //Consecuencia

FuzzyRule*regla_3 = new FuzzyRule(3, si_error_es_cero,entonces_potencia_es_baja);
control->addFuzzyRule(regla_3); //Asignación
////////////////////////////////////

////////////////////////////////////Si error es NP(NEGATIVO PEQUEÑO), entonces potencia es MEDIA////////////////////////////////////
FuzzyRuleAntecedent*si_error_es_np = new FuzzyRuleAntecedent();
si_error_es_np->joinSingle(np); //Antecedente

FuzzyRuleConsequent*entonces_potencia_es_media = new FuzzyRuleConsequent();
entonces_potencia_es_media->addOutput(media); //Consecuencia

FuzzyRule*regla_4 = new FuzzyRule(4, si_error_es_np,entonces_potencia_es_media);
control->addFuzzyRule(regla_4); //Asignación
////////////////////////////////////

////////////////////////////////////Si error es NG(NEGATIVO GRANDE), entonces potencia es ALTA////////////////////////////////////
FuzzyRuleAntecedent*si_error_es_ng = new FuzzyRuleAntecedent();
si_error_es_ng->joinSingle(ng); //Antecedente

FuzzyRuleConsequent*entonces_potencia_es_alta = new FuzzyRuleConsequent();
entonces_potencia_es_alta->addOutput(alta); //Consecuencia

FuzzyRule*regla_5 = new FuzzyRule(5, si_error_es_ng,entonces_potencia_es_alta);
control->addFuzzyRule(regla_5); //Asignación
////////////////////////////////////

```

Figura 3.78 Programa del controlador difuso de temperatura (5).

Finalmente, luego de evaluar las reglas del controlador se procede a realizar el código para la ejecución de las acciones de revisión y despliegue de las tareas del controlador que permitirán ajustar la temperatura ambiente. El procedimiento se ejecuta cada minuto debido a que la variable temperatura es de respuesta lenta.

```

void loop()
{
  //Set Point de la temperatura

  set_point=23; // Es la temperatura ambiente donde se quiere estabilizar

  //Obtiene los datos del sensor de temperatura

  val = analogRead(pinA0); //Valor leído por el sensor de temperatura. Entre 0 y 1024
  float mv = (val/1024.0)*5000; //Fórmula para transformar el valor "val" anterior a milivoltios (mv)
  float cel = mv/10; //Fórmula que asigna la temperatura en Celsius (cel) al relacionarlo con los milivoltios (mv). En las especificaciones se puede ver la relación voltaje-temperatura
  float temperatura = cel; //Asignación de la temperatura final calculada a la variable general de temperatura

  //Ejecución del controlador

  ERROR=set_point-temperatura; //Se establece el valor del error restando el set point con la temperatura que lee el sensor
  control->setInput(1,ERROR); //Se asigna al sistema Fuzzy llamado control una entrada con el valor de ERROR
  control->fuzzify(); //Se fuzzifica el sistema
  POTENCIA = control->defuzzify(); // Se defuzzifica el sistema, guardando la salida en la variable potencia.
  analogWrite(5,POTENCIA); //Se establece la potencia del PWM

  //Envía el dato final procesado de temperatura al puerto serial
  Serial.print(temperatura);

  //Imprime un salto de línea
  Serial.print("\n");

  //Esperamos un tiempo 1 minuto para repetir el loop
  delay(60000);
}

```

Figura 3.79 Programa del controlador difuso de temperatura (6).

La Figura 3.80 muestra el proceso general del programa en Arduino con el que se implementó el control difuso de temperatura de la caja de control IOT.



Figura 3.80 Proceso del controlador difuso de temperatura.

3.3.4 Interfaz web para revisión de temperatura

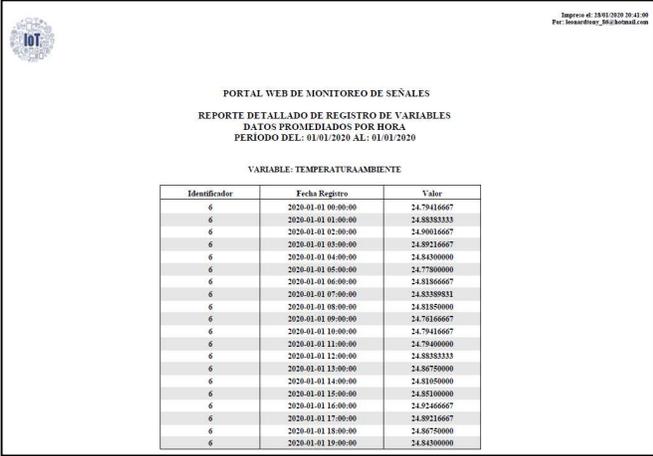
El sistema diseñado para el monitoreo de las señales de la peletizadora presenta entre sus consultas y reportes como una variable de estudio a la temperatura ambiente que podrá ser monitoreada para establecer las acciones del controlador difuso de la temperatura ambiente al interior de la caja de control.

Esta variable “TemperaturaAmbiente” se almacena en la misma tabla de variables de la Base de Datos del sistema llamada “PwIoT” y cuya entidad es “pwiot_tblvariables”.

Su rango de trabajo oscila entre 0 °C y 43 °C y el método de lectura aplicado para el registro de esta variable es el mismo que se utilizó para la toma de información de las variables de la peletizadora, es decir se implementó un programa en Arduino UNO y en Raspberry PI 3 B+ pero para la lectura de la temperatura se utilizó un sensor LM35.

En el sistema se puede acceder a la revisión de los valores de la temperatura en las opciones de “Reportes”, como ya se vio previamente en la presentación del aplicativo.

Existen dos tipos de reportes, el “Reporte Detallado” y el “Reporte Gráfico”. En ambos tipos de reportes se manejan las consultas generales y la consulta por promedios bajo el mismo criterio de las otras variables debido a la gran cantidad de información registrada para evitar tiempos de respuesta largos en las consultas, a pesar de que la temperatura se registra por cada minuto a diferencia de las otras variables que se registran cada 10 segundos.



Impreso el: 15/01/2020 10:41:00
Por: temandrey_36@hotmail.com

PORTAL WEB DE MONITOREO DE SEÑALES
 REPORTE DETALLADO DE REGISTRO DE VARIABLES
 DATOS PROMEDIADOS POR HORA
 PERIODO DEL: 01/01/2020 AL: 01/01/2020

VARIABLE: TEMPERATURAAMBIENTE

Identificador	Fecha Registro	Valor
6	2020-01-01 00:00:00	24.79416667
6	2020-01-01 01:00:00	24.88383333
6	2020-01-01 02:00:00	24.99016667
6	2020-01-01 03:00:00	24.99216667
6	2020-01-01 04:00:00	24.84300000
6	2020-01-01 05:00:00	24.77800000
6	2020-01-01 06:00:00	24.81866667
6	2020-01-01 07:00:00	24.83389333
6	2020-01-01 08:00:00	24.82350000
6	2020-01-01 09:00:00	24.76366667
6	2020-01-01 10:00:00	24.79416667
6	2020-01-01 11:00:00	24.79400000
6	2020-01-01 12:00:00	24.88383333
6	2020-01-01 13:00:00	24.80750000
6	2020-01-01 14:00:00	24.81050000
6	2020-01-01 15:00:00	24.85100000
6	2020-01-01 16:00:00	24.92466667
6	2020-01-01 17:00:00	24.99216667
6	2020-01-01 18:00:00	24.80750000
6	2020-01-01 19:00:00	24.84300000

Figura 3.81 Reporte detallado de temperatura.

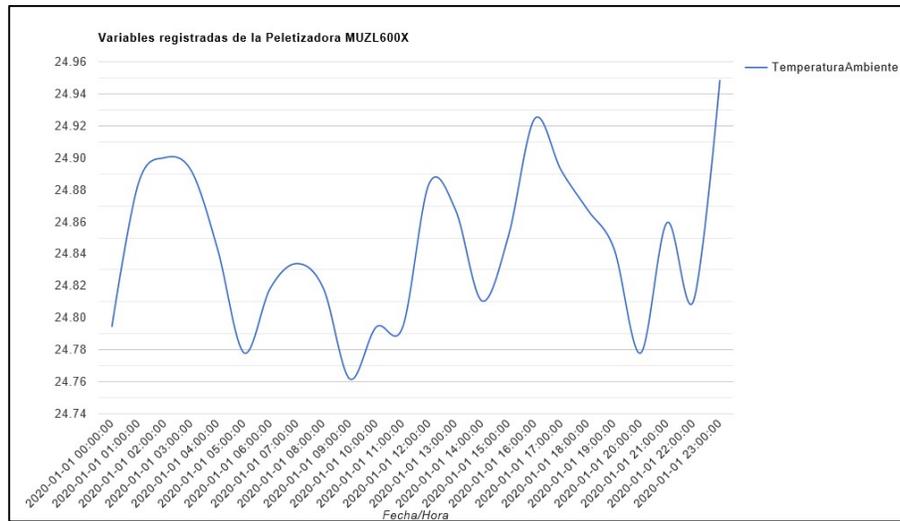


Figura 3.82 Reporte gráfico de temperatura.

CAPÍTULO 4

4. ANÁLISIS DE RESULTADOS.

En este capítulo se presentará el análisis de los resultados obtenidos durante el proceso de implementación del proyecto planteado en este documento. Es decir, se revisará los resultados del proceso de lectura de las variables monitoreadas de la peletizadora confirmando el correcto funcionamiento de los programas. De igual forma se revisarán los resultados del controlador difuso de temperatura para conocer acerca de rendimiento dentro del proceso de regulación de temperatura en la caja de control IOT.

4.1 Análisis de valores de variables obtenidos de la peletizadora

El objetivo de la lectura de datos del panel eléctrico para monitorear las señales que posee una peletizadora de alimentos balanceado para camarón es poder finalmente revisar esta información en cualquier momento del día de trabajo para hacer seguimiento al comportamiento del equipo respecto a la calidad de la producción en tiempo real o para conocer acerca de los inconvenientes que pueden existir en el proceso de producción, estos inconvenientes pueden ser estudiados a partir de las variables pues se podría reconocer luego mediante patrones los posibles momentos de falla futuras que pueden generarse.

Las variables que son monitoreadas en este proyecto son las siguientes:

- Amperaje.
- Frecuencia.
- Presión.
- Temperatura DADO.
- Temperatura DDC.
- Temperatura Ambiente.

Previamente se ha indicado en los Capítulos 1 y 2 de este documento para que se requiere medir estas variables y su relación directa con el proceso de producción, sin embargo, podemos explicar brevemente que estas variables permiten interpretar el proceso de peletización de la mezcla de materia prima y así determinar si se está

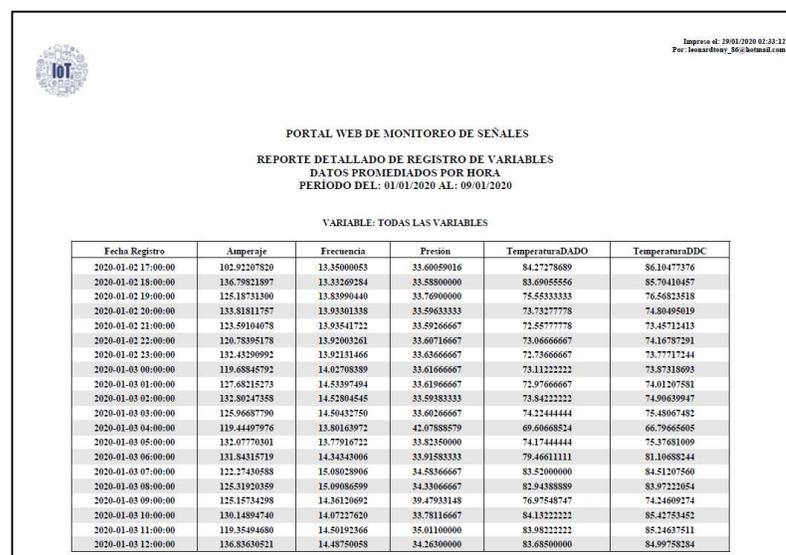
trabajando correctamente o comprender acerca de los problemas que pueden ocasionar inconvenientes en la producción.

Además, se debe considerar que la variable “Temperatura Ambiente” está relacionada con la temperatura al interior de la caja de control IOT que será controlada por el sistema difuso diseñado para ese fin.

En la Figura 3.40 podemos observar los rangos de trabajo normales que cada variable debe tener durante el proceso de peletización. La tarea de monitoreo consiste como se lo indico previamente en revisar en tiempo real o tiempo de corrección las señales de la peletizadora y determinar si existieron variación en sus valores durante un lapso determinado y así concluir mediante el razonamiento y fundamentos técnicos el estado del proceso.

Para este apartado del documento se considerará como rango de tiempo los primeros nueve días del mes de enero, aquí se comparará los valores de las variables de la peletizadora durante este tiempo y el valor de la Temperatura Ambiente la cuál fue medida solamente por tres días en ese transcurso de tiempo para el estudio del controlador difuso.

El Reporte Detallado de variables de la peletizadora consultado entre el 1 y 9 de enero de este año muestra los resultados en las Figuras 4.1 a la 4.5.



Impreso el: 29/01/2020 02:33:12
Por: tecnoartmay_89@hotmail.com

PORTAL WEB DE MONITOREO DE SEÑALES

REPORTE DETALLADO DE REGISTRO DE VARIABLES
DATOS PROMEDIADOS POR HORA
PERIODO DEL: 01/01/2020 AL: 09/01/2020

VARIABLE: TODAS LAS VARIABLES

Fecha Registro	Amperaje	Frecuencia	Presión	TemperaturaDADO	TemperaturaDDC
2020-01-02 17:00:00	102.92207820	13.35000053	33.60059016	84.27178689	86.10477376
2020-01-02 18:00:00	136.79821897	13.33269254	33.58900000	83.69055556	85.70410457
2020-01-02 19:00:00	125.18731300	13.83990440	33.76900000	75.55333333	76.56823518
2020-01-02 20:00:00	133.81811757	13.93301338	33.59633333	73.73277778	74.80495019
2020-01-02 21:00:00	123.59104078	13.93541722	33.59266667	72.55777778	73.45712413
2020-01-02 22:00:00	120.78395178	13.92003261	33.60716667	73.06666667	74.16787291
2020-01-02 23:00:00	132.43290992	13.92131466	33.63666667	72.73666667	73.77712144
2020-01-03 00:00:00	119.68845792	14.02708389	33.61666667	73.11222222	73.87318693
2020-01-03 01:00:00	127.68215273	14.83397494	33.61966667	72.97666667	74.01207553
2020-01-03 02:00:00	132.80247358	14.52804548	33.59383333	73.84222222	74.90639947
2020-01-03 03:00:00	125.96687790	14.50432750	33.60266667	74.24444444	75.48067482
2020-01-03 04:00:00	119.44497976	13.80163972	42.07888579	69.6968524	66.79665605
2020-01-03 05:00:00	132.0770301	13.77916722	33.82350000	74.17444444	75.37681009
2020-01-03 06:00:00	131.84315719	14.34343006	33.91583333	79.46011111	81.10688244
2020-01-03 07:00:00	122.27430588	15.08028906	34.58366667	83.52000000	84.51207560
2020-01-03 08:00:00	125.31920359	15.09086599	34.33066667	82.94388889	83.97222654
2020-01-03 09:00:00	125.15734298	14.36120692	39.47933148	76.97548747	74.24609274
2020-01-03 10:00:00	130.14894740	14.07227620	33.78116667	84.13222222	85.41753452
2020-01-03 11:00:00	119.35494680	14.50192366	35.01100000	83.98222222	85.24637511
2020-01-03 12:00:00	136.33630521	14.48750658	34.26300000	83.65000000	84.99758284

Figura 4.1 Reporte detallado de variables de la peletizadora (1).

Fecha Registro	Amperaje	Frecuencia	Presión	TemperaturaDADO	TemperaturaDDC
2020-01-03 13:00:00	130.35817062	14.32003262	38.97183333	82.10888889	83.59093214
2020-01-03 14:00:00	133.74627452	12.31041716	37.96466667	78.78722222	77.01267962
2020-01-03 15:00:00	137.59518541	12.10753254	56.58000000	49.65055556	43.78203966
2020-01-03 16:00:00	133.22842985	13.40181112	51.25587744	52.44512535	49.39203802
2020-01-03 17:00:00	140.91416436	10.87179531	38.15450000	76.42000000	79.03743803
2020-01-03 18:00:00	123.48785503	10.89326967	33.93333333	76.03444444	77.58152019
2020-01-03 19:00:00	133.21551302	10.89967992	36.28050000	75.57611111	77.81038492
2020-01-03 20:00:00	132.22787115	10.94038505	35.32800000	77.83055556	79.87741386
2020-01-03 21:00:00	121.61470759	10.93205172	33.65050000	80.98388889	83.11594037
2020-01-03 22:00:00	131.78961432	10.93750044	35.92833333	82.35944444	84.31581957
2020-01-03 23:00:00	124.38672471	11.87211586	34.14533333	83.53833333	85.49758283
2020-01-04 00:00:00	127.01307459	13.30208387	33.59816667	84.95888889	86.02137508
2020-01-04 01:00:00	140.63776850	13.31923130	33.57783333	84.40944444	85.81159249
2020-01-04 02:00:00	120.44686765	13.30416720	34.84466667	84.13500000	85.55615771
2020-01-04 03:00:00	128.54366064	13.59278901	34.02633333	83.87111111	85.42028835
2020-01-04 04:00:00	119.44731435	14.17543446	34.20367688	84.63899721	86.18989776
2020-01-04 05:00:00	122.74369448	14.50016084	34.62033333	85.13888889	86.82789681
2020-01-04 06:00:00	127.08180627	13.80320568	39.04216667	82.52277778	81.89794520
2020-01-04 07:00:00	131.05609700	13.90528902	37.37066667	83.71666667	84.21074711
2020-01-04 08:00:00	132.79897219	13.90128261	36.46866667	78.41333333	77.73490183
2020-01-04 09:00:00	127.91852984	13.66875055	36.91000000	81.28833333	82.34178579
2020-01-04 10:00:00	132.48054623	13.61442362	36.13783333	82.73833333	83.13405631
2020-01-04 11:00:00	105.98718489	14.49903904	33.43500000	82.84166667	84.43719638
2020-01-04 12:00:00	76.47095984	14.49070571	34.05000000	83.37222222	85.22101779
2020-01-04 13:00:00	126.53409124	14.49519289	33.68750000	83.71333333	85.67089201
2020-01-04 14:00:00	127.10916010	14.49567366	33.40066667	83.95555556	85.96074707
2020-01-04 15:00:00	125.89129652	14.50000058	33.40016667	85.11833333	86.83272773
2020-01-04 16:00:00	127.94624650	14.49201901	33.40963788	85.38384401	86.88385438
2020-01-04 17:00:00	122.48064466	14.49359032	33.65516667	85.49388889	87.06280019
2020-01-04 18:00:00	124.23514018	14.50288520	49.76183333	85.72888889	85.63647226
2020-01-04 19:00:00	134.89583366	14.52003263	34.67350000	84.80277778	86.45833160
2020-01-04 20:00:00	128.43094436	14.50593007	33.63500000	78.08666667	79.13224479
2020-01-04 21:00:00	130.60461988	14.51073776	33.87716667	72.85555556	74.26267967
2020-01-04 22:00:00	127.46440340	14.48381468	34.11400000	71.09055556	72.31944300
2020-01-04 23:00:00	151.67850420	12.24700070	47.80245125	57.52311978	54.21581580

Figura 4.2 Reporte detallado de variables de la peletizadora (2).

Fecha Registro	Amperaje	Frecuencia	Presión	TemperaturaDADO	TemperaturaDDC
2020-01-05 00:00:00	115.62015785	13.53429541	33.63783333	72.85833333	74.40760721
2020-01-05 01:00:00	129.84758331	13.19006463	38.93216667	69.18444444	68.12862183
2020-01-05 02:00:00	125.13974234	14.16362236	34.28916667	75.46555556	77.23309024
2020-01-05 03:00:00	109.55663222	14.72455598	33.74122563	75.66685237	77.11032909
2020-01-05 04:00:00	111.98797604	15.08873778	33.54266667	75.52222222	76.76207576
2020-01-05 05:00:00	111.50163271	14.64503264	42.57533333	68.03722222	65.52838033
2020-01-05 06:00:00	112.00701903	13.69759670	42.79200000	66.55055556	66.83635132
2020-01-05 07:00:00	117.62623187	13.90013517	33.66133333	73.34277778	74.60626263
2020-01-05 08:00:00	140.42344041	13.92179543	35.57433333	74.06444444	72.88284878
2020-01-05 09:00:00	127.70704460	13.90512876	33.97200000	74.01666667	74.95410478
2020-01-05 10:00:00	119.30798399	13.89487238	33.56200000	73.99833333	75.48007095
2020-01-05 11:00:00	123.90992115	13.91137876	37.25416667	73.81000000	72.83393574
2020-01-05 12:00:00	195.39967872	13.90352620	37.60250000	71.14588889	72.99939468
2020-01-05 13:00:00	132.23518111	13.91746851	33.61516667	74.01944444	75.46497434
2020-01-05 14:00:00	131.07457805	13.89615440	34.29566667	77.31166667	79.08393562
2020-01-05 15:00:00	140.20942935	13.76387456	42.89114206	74.71587744	73.89124232
2020-01-05 16:00:00	125.64978273	13.33269284	35.79183333	80.80333333	82.44625439
2020-01-05 17:00:00	124.43686154	13.56234029	37.27416667	81.17833333	83.10446094
2020-01-05 18:00:00	120.52118440	12.71971205	33.53600000	81.14833333	82.07608532
2020-01-05 19:00:00	118.39573047	13.02580180	35.86333333	81.33277778	82.90060453
2020-01-05 20:00:00	130.44582229	13.29375053	33.61383333	83.87944444	85.18478091
2020-01-05 21:00:00	130.34673693	13.50977618	34.52600000	83.07666667	84.58695483
2020-01-05 22:00:00	56.72966066	13.72724414	34.62700000	82.30055556	83.77837997
2020-01-05 23:00:00	127.69663286	13.31073771	40.62500000	78.60000000	76.98731730
2020-01-06 00:00:00	134.67207824	13.29871848	36.85616667	71.95277778	71.75603721
2020-01-06 01:00:00	136.60912828	12.79359026	35.68416667	83.13333333	84.82306594
2020-01-06 02:00:00	122.41526570	12.31009665	41.40100000	80.06111111	80.55736554
2020-01-06 03:00:00	113.67694834	11.51371377	34.50350975	81.45515320	82.58386658
2020-01-06 04:00:00	123.85114916	12.03333381	39.02933333	82.14000000	83.98610943
2020-01-06 05:00:00	208.10896488	10.84519274	40.62100000	82.36500000	82.50301767
2020-01-06 06:00:00	123.03705302	11.13876250	36.70183333	78.65044444	80.27498839
2020-01-06 07:00:00	29.59857375	11.94743638	39.01833333	75.19111111	74.11654441
2020-01-06 08:00:00	116.21568439	13.34166720	34.68100000	81.95944444	83.95893552
2020-01-06 09:00:00	123.02481205	13.90705184	33.54000000	83.18055556	84.91968429
2020-01-06 10:00:00	118.97369991	13.89679543	43.89566667	74.12777778	71.89311480

Figura 4.3 Reporte detallado de variables de la peletizadora (3).

Fecha Registro	Amperaje	Frecuencia	Presión	TemperaturaDADO	TemperaturaDDC
2020-01-06 11:00:00	127.12041381	13.58958388	38.15733333	74.93333333	75.27958787
2020-01-06 12:00:00	129.76785745	13.02147488	43.53000000	78.06388889	76.75784871
2020-01-06 13:00:00	127.39426382	12.12387869	33.64383333	78.18055556	80.12137521
2020-01-06 14:00:00	144.38287610	13.70833388	34.29897436	82.24273504	84.39427553
2020-01-07 17:00:00	111.65851585	13.23160588	33.32944099	72.63478261	74.67188617
2020-01-07 18:00:00	122.44470135	14.22932749	33.33700000	71.71666667	73.91847678
2020-01-07 19:00:00	123.29862904	14.42355827	33.43250000	70.82277778	73.01207583
2020-01-07 20:00:00	119.75005770	14.43060955	33.20266667	70.91000000	73.00482946
2020-01-07 21:00:00	119.48863667	14.42516083	33.22733333	70.05388889	71.97040919
2020-01-07 22:00:00	132.77439059	14.42740442	33.22383333	69.94388889	71.95108552
2020-01-07 23:00:00	122.29564322	13.53910311	33.22783333	70.94000000	73.21437052
2020-01-08 00:00:00	124.97194200	13.83603012	33.17816156	71.28189415	73.22324240
2020-01-08 01:00:00	109.47174479	14.38109032	33.23866667	72.14277778	74.42330769
2020-01-08 02:00:00	112.38740335	13.97339799	37.36083333	72.28555556	73.64311447
2020-01-08 03:00:00	133.72598939	13.94471210	34.40016667	73.21611111	75.71557820
2020-01-08 04:00:00	131.70339795	14.37740442	33.33033333	72.34944444	74.76871831
2020-01-08 05:00:00	130.16761397	14.38157109	33.34800000	71.78111111	74.19142364
2020-01-08 06:00:00	117.42542644	14.01025697	38.40066667	69.93166667	69.09903243
2020-01-08 07:00:00	125.87780930	13.17772489	33.72200000	72.74722222	75.00784874
2020-01-08 08:00:00	122.63612088	13.20112232	33.35250000	71.07944444	73.41787293
2020-01-08 09:00:00	127.44572149	13.18685950	34.66500000	71.10888889	73.26630288
2020-01-08 10:00:00	120.43257188	13.18012873	33.27850000	68.21333333	70.63345269
2020-01-08 11:00:00	125.39483850	13.82980285	33.79666667	73.02000000	75.45591637
2020-01-08 12:00:00	129.50994350	13.77548132	33.31633333	72.58333333	75.21980526
2020-01-08 13:00:00	132.51580185	13.62004554	33.94905011	73.56267409	76.32493489
2020-01-08 14:00:00	118.04150529	13.74679542	41.03133333	80.88444444	81.55676165
2020-01-08 15:00:00	129.87373767	11.70208380	37.26883333	74.63055556	76.17511925
2020-01-08 16:00:00	116.75878759	10.84679531	34.11166667	82.72166667	85.23369395
2020-01-08 17:00:00	132.50516570	10.85576967	39.58783333	80.68055556	81.59963605
2020-01-08 18:00:00	109.83624906	12.00352612	36.36883333	76.43055556	76.78019170
2020-01-08 19:00:00	136.22629479	13.18461591	35.03350000	83.37944444	85.61171236
2020-01-08 20:00:00	124.49797665	13.19423130	34.04933333	82.37777778	84.28502247
2020-01-08 21:00:00	122.13428966	13.18541719	33.84483333	81.14666667	83.22705148
2020-01-08 22:00:00	132.07887734	12.87051334	33.23766667	79.16777778	81.48188243
2020-01-08 23:00:00	128.99279713	11.78269278	40.87433333	75.30666667	74.26086808

Figura 4.4 Reporte detallado de variables de la peletizadora (4).

Fecha Registro	Amperaje	Frecuencia	Presión	TemperaturaDADO	TemperaturaDDC
2020-01-09 00:00:00	135.18393714	10.05272476	37.88433333	77.61222222	78.86473272
2020-01-09 01:00:00	122.54319373	11.85649287	33.33977716	80.10696379	82.50151222
2020-01-09 02:00:00	118.95647076	12.04631458	33.59133333	80.76222222	83.96014325
2020-01-09 03:00:00	124.13360151	11.91842996	38.19483333	77.39666667	79.35386315
2020-01-09 04:00:00	126.27260193	11.48221200	42.24633333	69.09833333	66.25562186
2020-01-09 05:00:00	110.59659125	10.18221195	48.37200000	56.94833333	51.28019221
2020-01-09 06:00:00	117.26570512	10.03333373	46.26850000	59.67666667	59.9396015
2020-01-09 07:00:00	121.28823315	11.09967993	40.63616667	71.73111111	71.91002272
2020-01-09 08:00:00	133.44396969	11.83076970	33.24766667	77.72444444	79.73429792
2020-01-09 09:00:00	126.50877327	13.81826978	33.73083333	82.06055556	84.23127851
2020-01-09 10:00:00	124.20423185	14.47069397	36.20535211	81.68000000	82.30802040
2020-01-09 11:00:00	111.14374712	15.71746858	34.01183333	84.88222222	87.47705139
2020-01-09 12:00:00	110.44112174	15.37115446	35.21500000	85.30055556	87.33091613
2020-01-09 13:00:00	116.93060961	14.87240257	38.74612813	82.17830008	81.62529600
2020-01-09 14:00:00	125.96915616	15.14858330	34.53433333	83.37555556	85.89087900
2020-01-09 15:00:00	118.73960681	11.95625048	42.18883333	72.80111111	71.35688263
2020-01-09 16:00:00	119.19138456	11.25653891	35.94600000	74.96833333	76.48248639
2020-01-09 17:00:00	124.69777601	11.73237226	35.50316667	81.66944444	83.03139931
2020-01-09 18:00:00	113.01014210	12.24102613	33.68233333	85.10388889	87.46438703
2020-01-09 19:00:00	115.39930593	12.23397485	33.68916667	85.03166667	87.08031227
2020-01-09 20:00:00	109.86225930	12.23814152	42.27683333	79.63944444	80.11171337
2020-01-09 21:00:00	121.18506533	12.03509664	43.51966667	71.83666667	73.06219661
2020-01-09 22:00:00	114.74670404	11.47227610	36.33100000	83.30222222	87.29589197
2020-01-09 23:00:00	116.71901505	11.04423121	38.84683333	81.64166667	80.31823511

Figura 4.5 Reporte detallado de variables de la peletizadora (5).

Según la revisión detallada de este rango de fechas se puede ver que los valores obtenidos durante días de producción activos, es decir mientras no haya para de planta, se encuentran dentro de valores de trabajo normales, esto significa que cada variable está trabajando dentro de los rangos normales definidos previamente y que se pueden ver en la Figura 3.40.

Con mayor precisión se puede decir que las variables que han presentado más variación por debajo del valor mínimo y por encima del valor máximo de trabajo son la Presión y los dos tipos de Temperaturas, pero su desviación ha sido mínima.

Esto según los expertos del proceso puede darse y presentarse como una situación normal en el proceso debido a las diferentes composiciones de las materias primas que en su momento pueden generar cambios en la Presión que ejerce el equipo sobre el proceso de peletización requiriendo que el equipo se exija un poco más de lo habitual.

Respecto a las temperaturas de los elementos del equipo, se atribuye de la misma forma a la composición de las materias primas que ocasionan que los dados y el rodillo de la peletizadora se exijan con mayor frecuencia y debido al exceso de fuerza en el proceso de expulsión del pellet se genera el aumento de temperatura en los equipos.

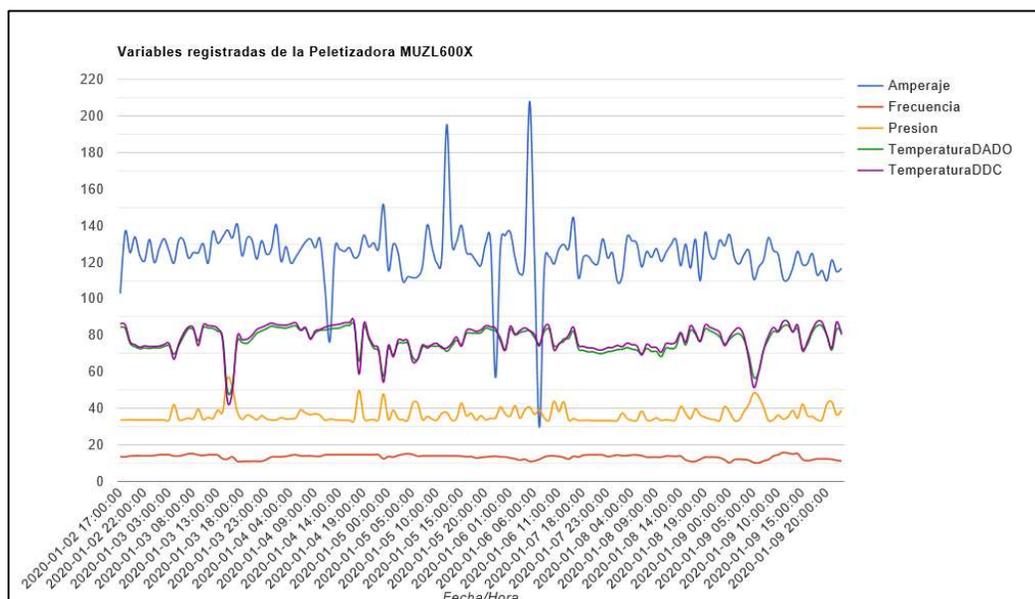


Figura 4.6 Reporte gráfico de variables de la peletizadora.

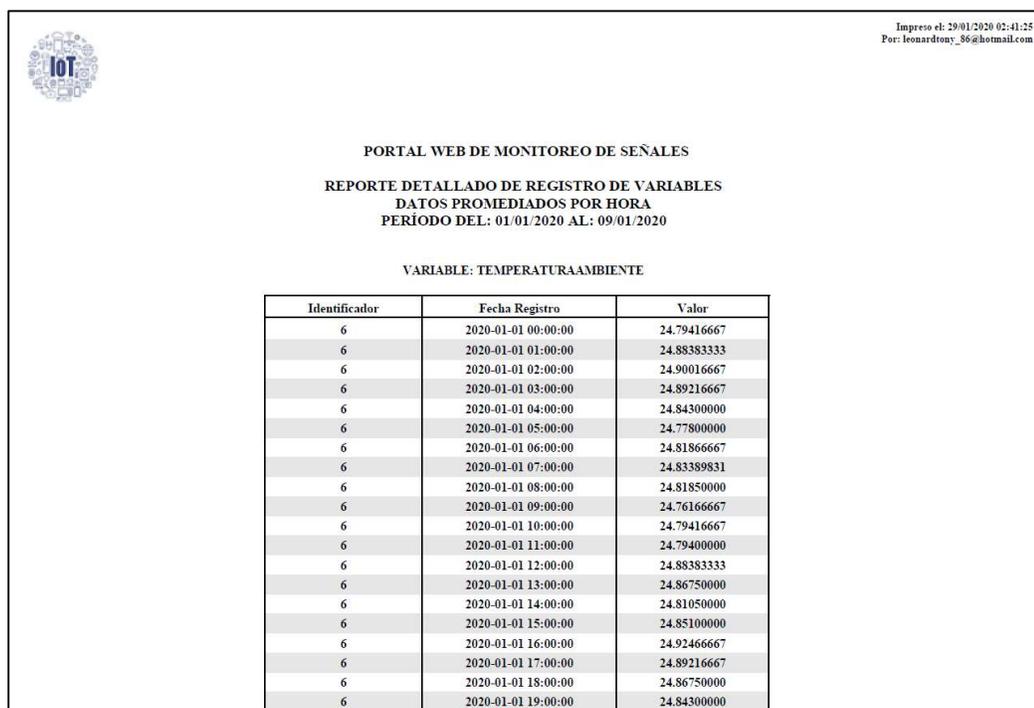
El Reporte Gráfico de variables de la peletizadora permite visualizar en forma global el comportamiento de las señales estudiadas.

Al análisis de datos del Reporte Detallado se puede complementar que el Amperaje también ha presentado variaciones en sus mediciones, pero cercanas al límite mínimo de su rango de trabajo en ciertos momentos del día. Sin embargo, se observa que la mayor parte del tiempo permanece dentro de los rangos normales de operación.

Cuando existe una caída de Amperaje en el equipo se debe a la no presencia de mezcla de materia prima para procesar, esto puede darse por demora en el proceso de dosificación y mezclado previo, ante esta situación los operadores de planta deberán revisar e informar el porqué de la situación presentada.

Las otras cuatro variables a pesar de presentar ciertos cambios en sus valores de trabajo, en la mayor parte del tiempo permanecen estables.

La variable Temperatura Ambiente también es analizada en el Reporte Detallado de Variables.



Impreso el: 29/01/2020 02:41:25
Por: leonardrony_86@hotmail.com

PORTAL WEB DE MONITOREO DE SEÑALES
REPORTE DETALLADO DE REGISTRO DE VARIABLES
DATOS PROMEDIADOS POR HORA
PERÍODO DEL: 01/01/2020 AL: 09/01/2020

VARIABLE: TEMPERATURA AMBIENTE

Identificador	Fecha Registro	Valor
6	2020-01-01 00:00:00	24.79416667
6	2020-01-01 01:00:00	24.88383333
6	2020-01-01 02:00:00	24.90016667
6	2020-01-01 03:00:00	24.89216667
6	2020-01-01 04:00:00	24.84300000
6	2020-01-01 05:00:00	24.77800000
6	2020-01-01 06:00:00	24.81866667
6	2020-01-01 07:00:00	24.83389831
6	2020-01-01 08:00:00	24.81850000
6	2020-01-01 09:00:00	24.76166667
6	2020-01-01 10:00:00	24.79416667
6	2020-01-01 11:00:00	24.79400000
6	2020-01-01 12:00:00	24.88383333
6	2020-01-01 13:00:00	24.86750000
6	2020-01-01 14:00:00	24.81050000
6	2020-01-01 15:00:00	24.85100000
6	2020-01-01 16:00:00	24.92466667
6	2020-01-01 17:00:00	24.89216667
6	2020-01-01 18:00:00	24.86750000
6	2020-01-01 19:00:00	24.84300000

Figura 4.7 Reporte detallado de la Temperatura Ambiente (1).

Identificador	Fecha Registro	Valor
6	2020-01-01 20:00:00	24.77783333
6	2020-01-01 21:00:00	24.85950000
6	2020-01-01 22:00:00	24.81033333
6	2020-01-01 23:00:00	24.94883333
6	2020-01-02 00:00:00	24.79416667
6	2020-01-02 01:00:00	24.90033333
6	2020-01-02 02:00:00	24.88366667
6	2020-01-02 03:00:00	24.87550000
6	2020-01-02 04:00:00	24.73733333
6	2020-01-02 05:00:00	24.84300000
6	2020-01-02 06:00:00	24.80233333
6	2020-01-02 07:00:00	24.73783333
6	2020-01-02 08:00:00	24.08800000
6	2020-01-02 09:00:00	24.67333333
6	2020-01-02 10:00:00	24.36516667
6	2020-01-02 11:00:00	23.70169492
6	2020-01-02 12:00:00	24.73150000
6	2020-01-02 13:00:00	25.22000000
6	2020-01-02 14:00:00	24.69916667
6	2020-01-02 15:00:00	26.18833333
6	2020-01-02 16:00:00	25.68433333
6	2020-01-02 17:00:00	24.69428571
6	2020-01-08 16:00:00	25.33411765

Figura 4.8 Reporte detallado de la Temperatura Ambiente (2).

La Temperatura Ambiente al interior de la caja de control del IOT sin la presencia del controlador difuso de temperatura permanece durante las jornadas de trabajo en los días de consulta en valores entre 22 °C y 27 °C.

Se puede considerar al rango de valores receptado por el sensor de temperatura como valor aceptable para los equipos del IOT, sin embargo, el objetivo del proyecto es mantener la temperatura estable en 23 °C, este es el valor del Set Point recomendado por los expertos en control y monitoreo de Centros de Datos y que se ha considera aplicar al proyecto por considerarse la temperatura ideal de trabajo de los dispositivos de monitoreo de señales.

La misma conclusión se puede obtener si se revisa el Reporte Grafico de la Temperatura en la Figura 4.9. Esto nos permite confirmar el comportamiento de la

variable de respuesta lenta en el tiempo y fácil determinación de posibles cambios drásticos en sus valores en el tiempo.

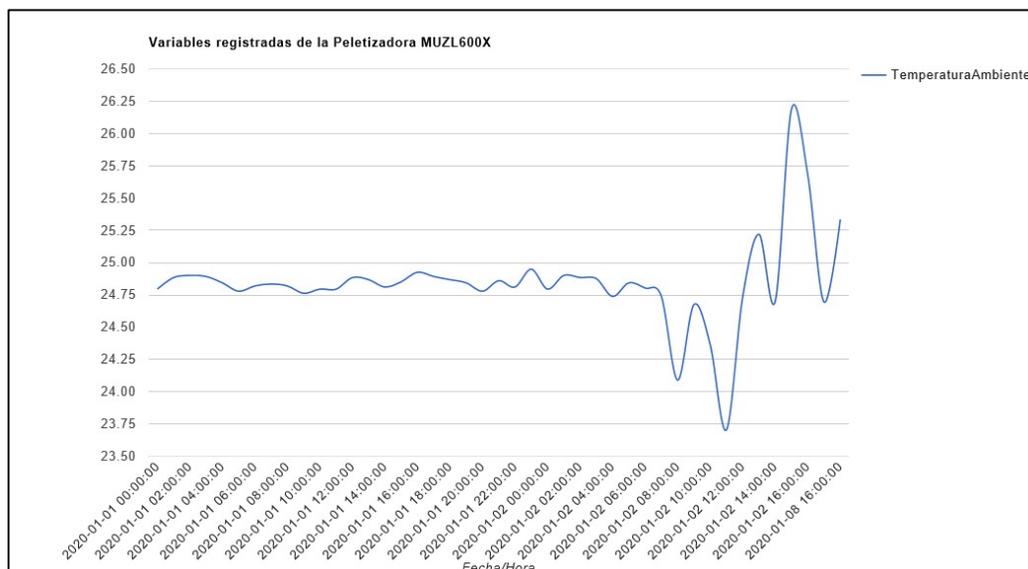


Figura 4.9 Reporte gráfico de la Temperatura Ambiente.

Adicional a las conclusiones ya reportadas en este documento relacionadas al comportamiento de las variables monitoreadas de la peletizado se puede también indicar que los valores de la escala de cada variable y que fueron calibrados por los técnicos en el proceso y en la automatización de la planta han sido parametrizados correctamente. Esto se puede confirmar considerando que los valores obtenidos en el proceso de lectura de datos a través de los dispositivos IOT al ser evaluados entre los rangos de trabajo permiten visualizar valores lógicos en su desempeño incluso en las desviaciones ocasionadas por efectos de dificultades en el proceso de producción.

4.2 Análisis del desempeño del controlador difuso de temperatura

El objetivo del controlador difuso de este proyecto es mantener la temperatura dentro de la caja de control IOT estable en 23 °C, este valor es recomendado para trabajar en Centros de Datos para garantizar el correcto funcionamiento de los equipos.

La caja de control estará ubicada en el interior de la planta de producción junto al panel eléctrico, en su interior se ubicarán los dispositivos IOT para la lectura de datos de la peletizadora.

Los dispositivos IOT utilizados en el proceso son una placa Arduino UNO y una microcomputadora Raspberry PI 3 B+, al Arduino se encuentran conectados cinco convertidores de 4-20 mA a 0-5 V, estos pequeños dispositivos permiten leer la señal que envía el módulo de comunicaciones del PLC y cuya medida de trabajo oscila justamente entre 4-20 mA.

Adicional a estos elementos IOT se debe considerar la inclusión en la misma caja de control de los elementos que implementan el controlador difuso de temperatura y que son una placa Arduino UNO y una microcomputadora Raspberry PI, estos dispositivos utilizan un sensor de temperatura LM35 para la lectura de dicha variable del ambiente de trabajo y una Celda de Peltier preparada con los disipadores y los coolers para el manejo del enfriamiento del entorno, este circuito de enfriamiento utiliza un transistor tipo Mosfet IRLB8748PBF para la regulación de la corriente de alimentación del Peltier según las necesidades que el controlador indique.

Considerando este antecedente que confirma los elementos que trabajan en la caja de control IOT se realizó el diseño del controlador difuso de temperatura a partir del Set Point indicado y los valores de muestras tomados de la implementación en campo.

Salida PWM Arduino UNO	% Ciclo de Trabajo	Voltaje Suministrado
25	0	0
64	25	1.25
127	50	2.5
191	75	3.75
255	100	5

Tabla 6: Relación del Ciclo de Trabajo PWM y el voltaje de salida del Arduino UNO.

Los valores de salida para el pin PWM utilizado por el controlador difuso de temperatura se muestran en la Tabla 6. A partir de estos valores se realizó el experimento para determinar la potencia que requiere suministrar al Peltier controlada desde el Arduino para enfriar o no el ambiente al interior de la caja de control IOT según la necesidad del momento.

La Tabla 4 hace referencia a los valores exactos parametrizados a partir del experimento de enfriamiento con la Celda de Peltier. De esta forma se indica que a más calor dentro de la caja y que está representado por el término “Mucho Calor” o “NG” se requiere de mayor amplitud en la salida PWM desde el Arduino y su valor esta entre 178.5 y 255 lo que representa un ciclo de trabajo de entre el 70% y 100%. Aquí el valor del error oscila entre los -12 y -20.

En segunda instancia al momento de estar trabajando en el rango “Poco Calor” o “NP” se requiere un valor medio en la salida del PWM y este valor oscila entre 76.5 y 178.5, con esta potencia de trabajo se puede regular el error cuando este se presenta en el rango entre -2 y -12.

En el momento en que la temperatura se encuentra trabajando cerca del Set Point y el valor del error esta estable entre -2 y +2 se requiere que el PWM utilice baja potencia, el experimento permitió definir este valor entre 25.5 y 76.5.

El rango de trabajo para el “PP” o “Poco Frio” cuyo valor del error oscila entre +2 y +12 requiere de potencia nula de parte del PWM, esta potencia se traduce en el rango de valores entre 0 y 25.5, de esta forma se establece que el Peltier deje de enfriar para regular la temperatura y acercarla al Set Point.

Finalmente, el “PG” o “Mucho Frio” cuyo valor de error permanece entre +12 y +20 requiere al igual que el rango de trabajo anterior (PP) de potencia nula en el PWM para dejar de enfriar a través del Peltier y acercar la temperatura ambiente al Set Point deseado.

El diseño del controlador difuso de temperatura para este proyecto no es complejo debido a que el ambiente de trabajo de los equipos IOT es cómodo para el mantenimiento de estos dispositivos, por tal motivo se observa en los reportes de sistema que la temperatura ambiente oscila entre 22 °C y 27 °C y a este rango de trabajo se lo obliga a mantenerse en 23 °C de acuerdo con el requerimiento que nace de la recomendación de los expertos en mantenimiento del clima de confort de equipos informáticos.

La variable Temperatura Ambiente debido a su naturaleza tiene tiempo de respuesta lento sin embargo se pudo determinar mediante el experimento en campo que cada minuto se puede establecer el efecto del controlador de temperatura para que el ambiente se mantenga en la zona del Set Point.

La sala de paneles de Automatización posee acondicionador de aire y su temperatura ambiente esta configurada en 27 °C, aplicando los diferentes cambios de temperatura para activar el controlador difuso en la caja de control IOT obtenemos los siguientes resultados:

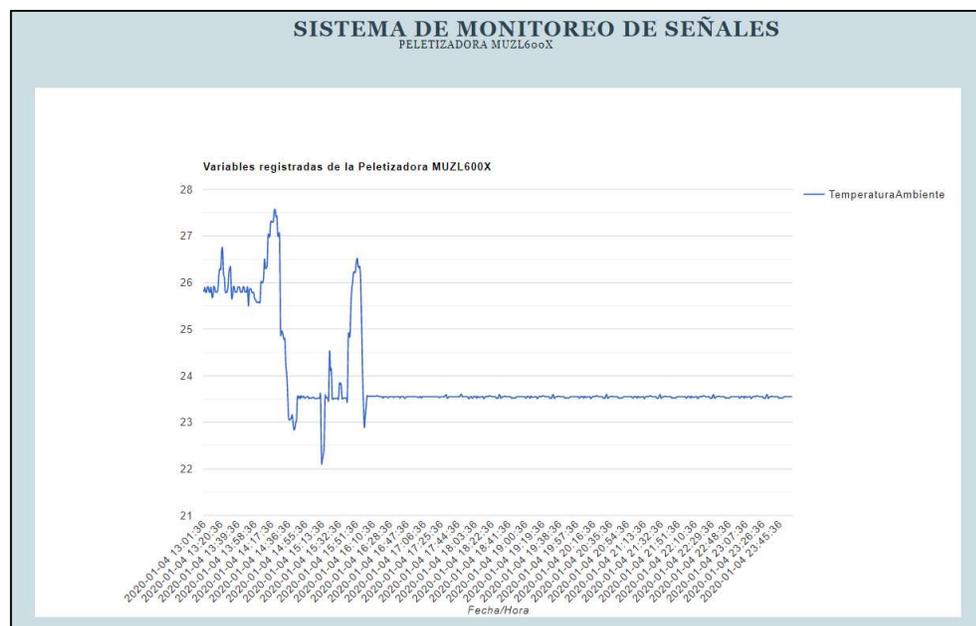


Figura 4.10 Rendimiento del controlador difuso de temperatura.

En la Figura 4.10 se puede observar el trabajo del acondicionador de aire en primera instancia manteniendo la temperatura en 27 °C, luego se apago el acondicionador para comenzar el experimento generando una subida de temperatura ambiente que supera los 27 °C, aquí se activa el controlador en el interior de la caja de control y la temperatura comienza a bajar hasta estabilizar en 23.50 °C. Luego se enciende el acondicionador de aire para comenzar a trabajar a 22 °C con el objetivo de que enfríe rápidamente, después de 8 minutos aproximadamente se configura nuevamente el equipo en 27 °C que es la temperatura ambiente recomendada por el técnico de

automatización para esta sala. Cuando se vuelve a estabilizar la temperatura, se retira la tapa de la caja de control para probar nuevamente el controlador, aquí la temperatura nuevamente comienza a aumentar hasta que finalmente se vuelve a tapar la caja y el controlador actúa sobre el ambiente estabilizando la temperatura en 23.55 °C aproximadamente.

Este experimento permite determinar que para conseguir la estabilización cuándo existe aumento de temperatura en la caja se requiere que la potencia de trabajo del Peltier sea media o alta.

La aplicación del controlador permite que se entre en el rango de trabajo ideal de la temperatura en aproximadamente un minuto, es importante recordar que el Set Point del controlador esta en 23 °C y el área de trabajo ideal de la temperatura esta entre 21 °C y 25 °C.

La gráfica de promedios de la temperatura se muestra en la Figura 4.11, aquí en resumen se puede ver dos cambios claramente diferenciados donde se ve el proceso de estabilización de la temperatura dentro de la caja de control cuando fue sometida a los experimentos indicados previamente.

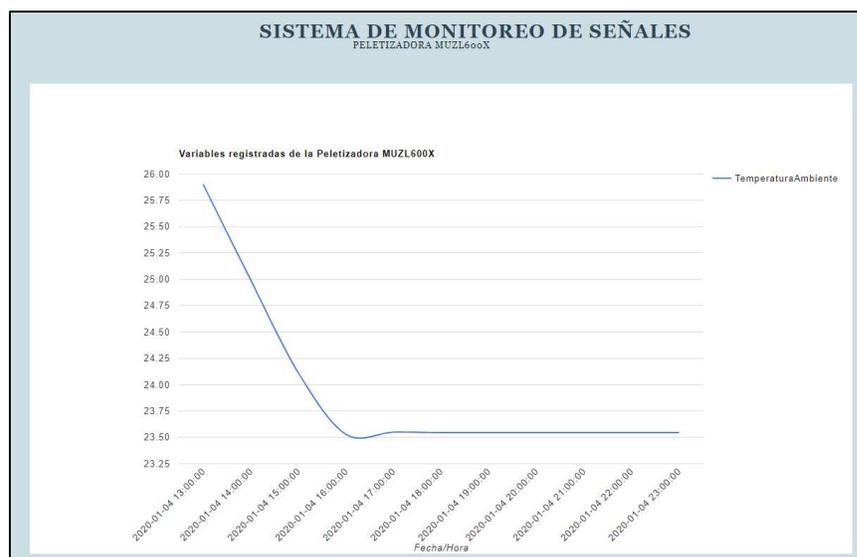


Figura 4.11 Rendimiento del controlador difuso de temperatura (Promedio).

CONCLUSIONES Y RECOMENDACIONES

El presente proyecto que consta de dos partes bien determinadas en el desarrollo de este permite confirmar muchas ideas relacionadas al uso de dispositivos electrónicos sencillos en combinación con herramientas industriales de gran escala.

Utilizar elementos como la placa Arduino UNO o la Raspberry PI 3B+ para monitorear señales de una peletizadora desde un PLC conectado a un panel eléctrico y de automatización que a su vez controla todos los equipos de una planta de producción alimento balanceado para camarón permite concluir que si es posible combinar el uso de dispositivos electrónicos en ambientes industriales siempre que se utilicen los transductores adecuados para hacer que ambos tipos de recursos trabajen al mismo nivel de comunicaciones sin mayor inconveniente.

Los dispositivos electrónicos pueden trabajar de manera eficiente todo el tiempo en un ambiente de confort pero siempre es recomendable realizar paralizaciones del proceso en paralelo con las paras de producción para permitir el descanso de los equipos, debido a que en ciertas ocasiones y a pesar de excelente rendimiento pueden inhibirse en la jornada y esto puede causar la detención del proceso de monitoreo y perdida de datos muy valiosos para el administrador de la planta cuyo objetivo es tener la mayor cantidad de información todo el tiempo que sea posible.

Las plataformas de desarrollo que proveen tanto Arduino como Raspberry son muy sencillas de manejar, estas permiten el diseño rápido de programas sencillos pero muy eficientes al momento de leer información proveniente de herramientas de automatización externas como un PLC. La ejecución y el mantenimiento de estos programas es bastante práctica y los tiempos de respuesta para ambas actividades son bastante cortos y están sujetos a la experticia del diseñador de los aplicativos.

Maniobrar con dispositivos electrónicos pequeños dentro de una planta de producción industrial bastante grande puede resultar complicado debido a la actividad existente en el entorno de trabajo por parte del personal técnico eléctrico y de producción que circula por el sitio y en ocasiones puede ocasionar accidentes involuntarios que

afecten la operación de los dispositivos IOT dejando sin efecto la toma de datos y traspaso de información a la nube.

Si no se tiene una excelente conexión Wifi dentro del lugar de trabajo en este caso en la planta de producción sería muy complicado realizar la transmisión de datos hacia la nube debido a que es la manera más fácil de realizar esta tarea, además el objetivo del uso de la Raspberry justamente es emplear sus bondades como microcomputadora poseedora de varios recursos entre ellos la tecnología inalámbrica para simplificar procesos de traslado de datos hacia la nube.

Los controladores difusos son prácticos de diseñar, para esto solo se requiere sobre todo de la experiencia del usuario experto o el diseñador tomada en campo a través de un muestreo de datos en un experimento en combinación con el uso adecuado de las herramientas de diseño.

La placa Arduino UNO resulta ser muy efectiva al momento de implementar un controlador difuso sencillo como el que se realizó en este proyecto para manejo de la temperatura, sus librerías para manejo de controles difusos lo hacen muy práctico a la hora de plasmar en su programación el análisis realizado previamente en campo y en Matlab.

Las Celdas de Peltier son dispositivos muy sencillos, pero con alto nivel de efectividad a la hora de generar frío o calor según la necesidad, en el caso de este proyecto se requería generar frío para mantener la temperatura ambiente de la caja de control IOT trabajando correctamente. Con esta celda se puede diseñar circuitos muy sencillos para control sin embargo debido al uso de un flujo de corriente considerable y a la interacción con un microcontrolador se requiere del uso de transistores Mosfet para su correcto manejo.

El uso de herramientas web para diseño de aplicaciones del mismo tipo es bastante práctico a la hora de complementar las actividades de monitoreo de señales como las realizadas en el proyecto. Se pueden realizar aplicativos pequeños, pero bastante explicativos y de fácil acceso a todos los usuarios interesados en obtener la información en cualquier momento desde cualquier lugar en el mundo donde se encuentren debido a las facilidades y seguridad que brinda la plataforma web.

Siempre será necesario que se realice un buen diseño de comunicaciones a partir del análisis de campo en conjunto con el técnico especializado en automatización de la planta industrial donde se desea trabajar, esto debido a que los técnicos de planta son aquellos que conocen en gran medida y con detalles el funcionamiento de cada elemento de automatización de su planta de producción.

Se debe preparar con mucha antelación y bastante prevención en ambiente de prueba el esquema de comunicaciones a implementar para el monitoreo de variables esto debido a que al momento de trabajar en planta no existen todas las comodidades y el tiempo necesario para instalar, configurar y monitorear el buen manejo de los equipos y en muchas ocasiones puede que los equipos instalados en planta no reciban el tratamiento adecuado de cuidado por parte del personal de planta lo que implica detenciones en el proceso de monitoreo de señales.

Se deberá considerar el uso de adecuación del circuito del controlador, pues si bien es cierto se puede trabajar con conexiones sencillas con cables tipo UTP y otros elementos no tan sofisticados, siempre será mejor adecuar correctamente el circuito para más comodidad el uso y a su vez garantizar el rendimiento de la solución.

Las Celdas Peltier son excelentes al momento de gestionar las temperaturas de los ambientes de trabajo pequeños como el utilizado en este proyecto sin embargo puede que resulte bastante útil utilizar más de una celda para ser más efectivo y rápido a la hora de cumplir con el objetivo principalmente de enfriar el entorno.

Para complementar el trabajo del Peltier se recomienda siempre utilizar para caja de control elementos con capacidad de retener el frío de esta forma se consigue la estabilización deseada por medio del controlador de manera más eficiente pues se atenúa en la medida de lo posible la pérdida de frío dentro del entorno.

Es necesario apoyarse de herramientas de monitoreo remotas como cámaras de seguridad del circuito de la planta de producción o cámaras de seguridad instaladas por cuenta propia para hacer seguimiento visual al estado de los equipos y de esta forma evitar inconvenientes por posibles accidentes involuntarios que puedan suceder con los dispositivos IOT instalados en planta.

BIBLIOGRAFÍA

- [1] J. E. C. Bajaña, “Estudio de Factibilidad para la Introducción y Comercialización de Alimento Balanceado para Camarones marca AquaFeed en Ecuador,” Tesis de Grado, Gestión de Agronegocios, Zamorano, Honduras, 2006.
- [2] P. D. P. F. T. S. Eugenio Bortone, (2007, Septiembre 5), “Interacción de Ingredientes y Procesos en la Producción de Alimentos Hidroestables para Camarones,” Ed. 2007. [Online]. Available: <https://www.engormix.com/balanceados/articulos/interaccion-ingredientes-procesos-produccion-t27301.htm>.
- [3] A. A. Pajares, “Estudio de la implantación de Internet de las Cosas, en las redes Logísticas de la Cadena de Suministro,” Tesis de Fin de Máster, Escuela Técnica Superior de Ingenieros Industriales, Universitat Politècnica de Valencia, España, 2015.
- [4] M. Talavera, (2019, Marzo 06), “Raspberry Pi,” Ed. 2019. [Online]. Available: <https://es.scribd.com/document/401247473/Raspberry-Pi>.
- [5] J. Pastor, (2018, Abril 25), “Raspberry Pi 3 Model B+,” Ed. 2018. [Online]. Available: <https://www.xataka.com/ordenadores/raspberry-pi-3-model-b-analisis-mas-potencia-y-mejor-wifi-para-un-minipc-que-sigue-asombrando>.
- [6] C. H. T.-H. M. M. Y. Ayala, “Evaluación de la Plataforma Arduino e Implementación de un Sistema de Control de Posición Horizontal,” Tesis de Grado, Facultad de Ingenierías, Universidad Politécnica Salesiana, Guayaquil, Ecuador, 2013.
- [7] I. J. V. G.-F. H. Hernández, “Caracterización de Sistemas de Refrigeración ‘Absorción y Peltier’ y Optimización de Geometrías en Transmisores de Calor,” Tesis de Postgrado, Centro de Investigación en Materiales Avanzados, Chihuahua, México, 2012.
- [9] F. F. F. Morelo, “Construcción del Motor de Inferencia Difuso del Prototipo Software de Simulación Para el Estudio de la Polución del Aire MID-CASS,” Tesis de Grado, Facultad de Ingeniería Físico-Mecánicas, Universidad Industrial de Santander, Bucaramanga, Colombia, 2007.

[10] R. J. B. M.-F. G. F. Piña, "Diseño y Construcción de un Módulo Didáctico de Control de Temperatura," Tesis de Grado, Facultad de Ingeniería Eléctrica y Electrónica, Escuela Politécnica Nacional, Quito, Ecuador, 2009.

[11] I. Rectifier, (2019, Marzo), "International Rectifier - MOSFET IRLB8748PBF," Ed. 2008. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/311650/IRF/IRLB8748PBF.html>.

ANEXOS

ESCALAS DE CALIBRACIÓN VARIABLE AMPERAJE (PLC-HOSTING)

Factor de Conversión (mA - Muestra)	
220	4
1	0.01818182
Factor de Conversión (Muestra - Amperaje)	
220	62.5
1	0.28409091

Amperaje	mA	Muestras										Promedio Muestra
0	4	34	30	32	36	31	36	40	37	31	34	40
62.5	8	266	265	252	246	248	264	253	252	255	260	260
125	12	495	499	480	463	480	474	490	491	491	464	480
187.5	16	687	686	687	703	693	726	691	695	717	720	700
250	20	914	908	902	904	909	942	915	911	952	947	920

#	mA Planta	Promedio Muestra Tomada	Valor Real	
			Amperaje	Id_variable
1	4	0	0	1
2	4	1	0	1
3	4	2	0	1
4	4	3	0	1
5	4	4	0	1
6	4	5	0	1
7	4	6	0	1
8	4	7	0	1
9	4	8	0	1
10	4	9	0	1
11	4	10	0	1
12	4	11	0	1
13	4	12	0	1
14	4	13	0	1
15	4	14	0	1
16	4	15	0	1
17	4	16	0	1
18	4	17	0	1
19	4	18	0	1
20	4	19	0	1
21	4	20	0	1
22	4	21	0	1
23	4	22	0	1
24	4	23	0	1
25	4	24	0	1
26	4	25	0	1
27	4	26	0	1
28	4	27	0	1
29	4	28	0	1
30	4	29	0	1
31	4	30	0	1
32	4	31	0	1
33	4	32	0	1
34	4	33	0	1
35	4	34	0	1
36	4	35	0	1
37	4	36	0	1
38	4	37	0	1
39	4	38	0	1
40	4	39	0	1
41	4	40	0	1

42	4.0181818	41	0.2840909	1
43	4.0363628	42	0.5681818	1
44	4.0545438	43	0.8522727	1
45	4.0727248	44	1.1363636	1
46	4.0909058	45	1.4204546	1
47	4.1090868	46	1.7045455	1
48	4.1272678	47	1.9886364	1
49	4.1454488	48	2.2727273	1
50	4.1636298	49	2.5568182	1
51	4.1818108	50	2.8409091	1
52	4.1999918	51	3.125	1
53	4.2181728	52	3.4090909	1
54	4.2363538	53	3.6931818	1
55	4.2545348	54	3.9772727	1
56	4.2727158	55	4.2613637	1
57	4.2908968	56	4.5454546	1
58	4.3090778	57	4.8295455	1
59	4.3272588	58	5.1136364	1
60	4.3454398	59	5.3977273	1
61	4.3636208	60	5.6818182	1
62	4.3818018	61	5.9659091	1
63	4.3999828	62	6.25	1
64	4.4181638	63	6.5340909	1
65	4.4363448	64	6.8181818	1
66	4.4545258	65	7.1022728	1
67	4.4727068	66	7.3863637	1
68	4.4908878	67	7.6704546	1
69	4.5090688	68	7.9545455	1
70	4.5272498	69	8.2386364	1
71	4.5454308	70	8.5227273	1
72	4.5636118	71	8.8068182	1
73	4.5817928	72	9.0909091	1
74	4.5999738	73	9.375	1
75	4.6181548	74	9.6590909	1
76	4.6363358	75	9.9431818	1
77	4.6545168	76	10.227273	1
78	4.6726978	77	10.511364	1
79	4.6908788	78	10.795455	1
80	4.7090598	79	11.079545	1
81	4.7272408	80	11.363636	1
82	4.7454218	81	11.647727	1
83	4.7636028	82	11.931818	1
84	4.7817838	83	12.215909	1

ESCALAS DE CALIBRACIÓN VARIABLE FRECUENCIA (PLC-HOSTING)

Factor de Conversión (mA - Muestra)	
260	4
1	0.01538462
Factor de Conversión (Muestra - Frecuencia)	
260	15
1	0.05769231

Frecuencia	mA	Muestras											Promedio Muestra
0	4	71	73	73	70	81	71	81	78	73	68	70	
15	8	321	329	336	328	341	339	338	329	327	329	330	
30	12	606	603	574	612	606	604	606	577	577	598	590	
45	16	860	853	820	866	830	848	870	836	869	874	850	
60	20	1023	1023	1023	1023	1023	1023	1023	1023	1023	1023	1110	

#	mA Planta	Promedio Muestra Tomada	Valor Real	
			Frecuencia	Id_variable
1	4	0	0	2
2	4	1	0	2
3	4	2	0	2
4	4	3	0	2
5	4	4	0	2
6	4	5	0	2
7	4	6	0	2
8	4	7	0	2
9	4	8	0	2
10	4	9	0	2
11	4	10	0	2
12	4	11	0	2
13	4	12	0	2
14	4	13	0	2
15	4	14	0	2
16	4	15	0	2
17	4	16	0	2
18	4	17	0	2
19	4	18	0	2
20	4	19	0	2
21	4	20	0	2
22	4	21	0	2
23	4	22	0	2
24	4	23	0	2
25	4	24	0	2
26	4	25	0	2
27	4	26	0	2
28	4	27	0	2
29	4	28	0	2
30	4	29	0	2
31	4	30	0	2
32	4	31	0	2
33	4	32	0	2
34	4	33	0	2
35	4	34	0	2
36	4	35	0	2
37	4	36	0	2
38	4	37	0	2
39	4	38	0	2
40	4	39	0	2
41	4	40	0	2

42	4	41	0	2
43	4	42	0	2
44	4	43	0	2
45	4	44	0	2
46	4	45	0	2
47	4	46	0	2
48	4	47	0	2
49	4	48	0	2
50	4	49	0	2
51	4	50	0	2
52	4	51	0	2
53	4	52	0	2
54	4	53	0	2
55	4	54	0	2
56	4	55	0	2
57	4	56	0	2
58	4	57	0	2
59	4	58	0	2
60	4	59	0	2
61	4	60	0	2
62	4	61	0	2
63	4	62	0	2
64	4	63	0	2
65	4	64	0	2
66	4	65	0	2
67	4	66	0	2
68	4	67	0	2
69	4	68	0	2
70	4	69	0	2
71	4	70	0	2
72	4.0153846	71	0.0576923	2
73	4.0307692	72	0.1153846	2
74	4.0461539	73	0.1730769	2
75	4.0615385	74	0.2307692	2
76	4.0769231	75	0.2884616	2
77	4.0923077	76	0.3461539	2
78	4.1076923	77	0.4038462	2
79	4.123077	78	0.4615385	2
80	4.1384616	79	0.5192308	2
81	4.1538462	80	0.5769231	2
82	4.1692308	81	0.6346154	2
83	4.1846154	82	0.6923077	2
84	4.2000001	83	0.75	2

ESCALAS DE CALIBRACIÓN VARIABLE PRESIÓN (PLC-HOSTING)

Factor de Conversión (mA - Muestra)	
250	4
1	0.01600000
Factor de Conversión (Muestra - Presión)	
250	15
1	0.06000000

Presión	mA	Muestras										Promedio Muestra
0	4	88	82	77	87	88	83	82	83	87	86	80
15	8	326	330	324	344	328	342	327	344	326	324	330
30	12	569	600	567	597	574	593	590	567	587	567	580
45	16	818	838	817	816	853	853	813	856	834	840	830
60	20	1023	1023	1023	1023	1023	1023	1023	1023	1023	1023	1080

#	mA Planta	Promedio Muestra Tomada	Valor Real		Id_variable
			Presión		
1	4	0	0		3
2	4	1	0		3
3	4	2	0		3
4	4	3	0		3
5	4	4	0		3
6	4	5	0		3
7	4	6	0		3
8	4	7	0		3
9	4	8	0		3
10	4	9	0		3
11	4	10	0		3
12	4	11	0		3
13	4	12	0		3
14	4	13	0		3
15	4	14	0		3
16	4	15	0		3
17	4	16	0		3
18	4	17	0		3
19	4	18	0		3
20	4	19	0		3
21	4	20	0		3
22	4	21	0		3
23	4	22	0		3
24	4	23	0		3
25	4	24	0		3
26	4	25	0		3
27	4	26	0		3
28	4	27	0		3
29	4	28	0		3
30	4	29	0		3
31	4	30	0		3
32	4	31	0		3
33	4	32	0		3
34	4	33	0		3
35	4	34	0		3
36	4	35	0		3
37	4	36	0		3
38	4	37	0		3
39	4	38	0		3
40	4	39	0		3
41	4	40	0		3
42	4	41	0		3
43	4	42	0		3
44	4	43	0		3
45	4	44	0		3
46	4	45	0		3
47	4	46	0		3
48	4	47	0		3
49	4	48	0		3
50	4	49	0		3
51	4	50	0		3
52	4	51	0		3
53	4	52	0		3
54	4	53	0		3
55	4	54	0		3
56	4	55	0		3
57	4	56	0		3
58	4	57	0		3
59	4	58	0		3
60	4	59	0		3
61	4	60	0		3
62	4	61	0		3
63	4	62	0		3
64	4	63	0		3
65	4	64	0		3
66	4	65	0		3
67	4	66	0		3
68	4	67	0		3
69	4	68	0		3
70	4	69	0		3
71	4	70	0		3
72	4	71	0		3
73	4	72	0		3
74	4	73	0		3
75	4	74	0		3
76	4	75	0		3
77	4	76	0		3
78	4	77	0		3
79	4	78	0		3
80	4	79	0		3
81	4	80	0		3
82	4.016	81	0.06		3
83	4.032	82	0.12		3
84	4.048	83	0.18		3

ESCALAS DE CALIBRACIÓN VARIABLE TEMPERATURA DADO (PLC-HOSTING)

Factor de Conversión (mA - Muestra)	
250	4
1	0.01600000
Factor de Conversión (Muestra - TemperaturaDADO)	
250	50
1	0.20000000

TemperaturaDADO	mA	Muestras											Promedio Muestra
-50	4	53	62	56	57	57	52	64	61	49	66	55	
0	8	316	304	315	316	301	308	296	301	292	302	305	
50	12	546	577	556	573	545	560	547	551	570	552	555	
100	16	776	780	789	824	785	806	818	787	831	825	805	
150	20	1023	1023	1023	1021	1023	1023	1023	1023	1023	1023	1055	

#	mA Planta	Promedio Muestra Tomada	Valor Real	
			TemperaturaDADO	Id_variable
1	4	0	-50	4
2	4	1	-50	4
3	4	2	-50	4
4	4	3	-50	4
5	4	4	-50	4
6	4	5	-50	4
7	4	6	-50	4
8	4	7	-50	4
9	4	8	-50	4
10	4	9	-50	4
11	4	10	-50	4
12	4	11	-50	4
13	4	12	-50	4
14	4	13	-50	4
15	4	14	-50	4
16	4	15	-50	4
17	4	16	-50	4
18	4	17	-50	4
19	4	18	-50	4
20	4	19	-50	4
21	4	20	-50	4
22	4	21	-50	4
23	4	22	-50	4
24	4	23	-50	4
25	4	24	-50	4
26	4	25	-50	4
27	4	26	-50	4
28	4	27	-50	4
29	4	28	-50	4
30	4	29	-50	4
31	4	30	-50	4
32	4	31	-50	4
33	4	32	-50	4
34	4	33	-50	4
35	4	34	-50	4
36	4	35	-50	4
37	4	36	-50	4
38	4	37	-50	4
39	4	38	-50	4
40	4	39	-50	4
41	4	40	-50	4

42	4	41	-50	4
43	4	42	-50	4
44	4	43	-50	4
45	4	44	-50	4
46	4	45	-50	4
47	4	46	-50	4
48	4	47	-50	4
49	4	48	-50	4
50	4	49	-50	4
51	4	50	-50	4
52	4	51	-50	4
53	4	52	-50	4
54	4	53	-50	4
55	4	54	-50	4
56	4	55	-50	4
57	4.016	56	-49.8	4
58	4.032	57	-49.6	4
59	4.048	58	-49.4	4
60	4.064	59	-49.2	4
61	4.08	60	-49	4
62	4.096	61	-48.8	4
63	4.112	62	-48.6	4
64	4.128	63	-48.4	4
65	4.144	64	-48.2	4
66	4.16	65	-48	4
67	4.176	66	-47.8	4
68	4.192	67	-47.6	4
69	4.208	68	-47.4	4
70	4.224	69	-47.2	4
71	4.24	70	-47	4
72	4.256	71	-46.8	4
73	4.272	72	-46.6	4
74	4.288	73	-46.4	4
75	4.304	74	-46.2	4
76	4.32	75	-46	4
77	4.336	76	-45.8	4
78	4.352	77	-45.6	4
79	4.368	78	-45.4	4
80	4.384	79	-45.2	4
81	4.4	80	-45	4
82	4.416	81	-44.8	4
83	4.432	82	-44.6	4
84	4.448	83	-44.4	4

