

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIERO ELECTRÓNICO**

**TEMA:
DESARROLLO DE UN SISTEMA AUTOMÁTICO CON CONTROL DIFUSO
PARA EL BATIDO DE GELATINA EN LA FABRICACIÓN DE DULCES
ARTESANALES.**

**AUTOR:
GORDILLO CHINCHERO LUIS STEVEN**

**TUTOR:
RUMIPAMBA LÓPEZ JOSÉ ALCIDES**

Quito, marzo del 2019

CESIÓN DE DERECHOS DE AUTOR

Yo Luis Steven Gordillo Chinchero con documento de identificación N° 172216670-7 manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy el autor del trabajo de titulación intitulado: “DESARROLLO DE UN SISTEMA AUTOMÁTICO CON CONTROL DIFUSO PARA EL BATIDO DE GELATINA EN LA FABRICACIÓN DE DULCES ARTESANALES”, mismo que ha sido desarrollado para optar por el título de Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autor me reservo los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Luis Steven Gordillo Chinchero
C.I. 172216670-7

Quito, marzo del 2019

CARTA DE DECLARATORIA DE COAUTORÍA DEL DOCENTE

TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Trabajo de Titulación, “DESARROLLO DE UN SISTEMA AUTOMÁTICO CON CONTROL DIFUSO PARA EL BATIDO DE GELATINA EN LA FABRICACIÓN DE DULCES ARTESANALES”, realizado por Luis Steven Gordillo Chinchero, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

Quito, marzo del 2019



José Alcides Rumipamba López

C.I: 1803854353

DEDICATORIA

Dedico este logro a mi padre Alfredo Gordillo y a mi madre Sylvia Chinchero, quienes siempre me brindaron su apoyo, supieron educarme y motivarme a ser mejor cada día.

Mi padre, un hombre muy trabajador, humilde y extremadamente inteligente, tuviste mi admiración desde pequeño y quería ser como tú al crecer, pero siempre me decías *“No quiero que me iguales, quiero que me superes”*. Esas palabras siempre las tengo presentes, como cada frase y enseñanza que salía de tu boca... ¡Gracias papá!

Mi madre, la mujer más luchadora del mundo, inteligente y amorosa, supiste educarme con paciencia y cariño, estas en cada uno de los recuerdos de mi infancia y de mi adolescencia, jamás terminaré de agradecerte por tanto afecto recibido de ti mamá.

Gracias a ustedes he cumplido muchos de los objetivos que me he propuesto y sé que con esfuerzo y sacrificio terminaré de cumplirlos todos, por eso este logro se lo dedico a mis héroes: mis padres.

Luis Steven Gordillo Chinchero

ÍNDICE

CESIÓN DE DERECHOS DE AUTOR.....	i
CARTA DE DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR	ii
DEDICATORIA.....	iii
ÍNDICE	iv
ÍNDICE DE ILUSTRACIONES:.....	vii
ÍNDICE DE TABLAS:	ix
RESUMEN.....	x
ABSTRACT	xi
INTRODUCCIÓN	xii
CAPÍTULO 1	1
ANTECEDENTES	1
1.1. Planteamiento del problema	1
1.2. Justificación.....	2
1.3. Objetivos	2
1.3.1. Objetivo General	2
1.3.2. Objetivos Específicos	2
CAPÍTULO 2	3
MARCO TEÓRICO	3
2.1. La Grenetina	3
2.2. Sistemas de control	4
2.3. Tipos de sistemas de control	4
2.3.1. Sistema de control a lazo abierto.....	4
2.3.2. Sistema de control a lazo cerrado	5
2.4. Fundamentos de la lógica difusa	5
2.5. Función de membresía	6
2.6. Conjunto difuso.	6

2.7.	Tipos de conjuntos difusos.....	7
2.7.1.	Función triangular.....	7
2.7.2.	Función trapezoidal	7
2.7.3.	Función sigmoïdal	8
2.8.	Operaciones con conjuntos.....	9
2.8.1.	Operación unión	9
2.8.2.	Operación intersección.....	9
2.8.3.	Operación complemento	10
2.9.	Control difuso.....	11
2.9.1.	Fusificación	11
2.9.2.	Inferencia difusa	11
2.9.3.	Defusificación	11
2.10.	Software de aplicación con lógica difusa	12
2.10.1.	“Fuzzy Logic Toolbox”.....	12
2.11.	Tarjeta de control “Arduino Nano”.....	13
2.12.	Sensor ultrasónico de nivel “HC-SR04”	14
2.13.	Termopar tipo “K” y módulo amplificador “MAX6675”.....	15
2.14.	Caracterización del variador de frecuencia	15
CAPÍTULO 3		19
DISEÑO E IMPLEMENTACIÓN		19
3.1.	Caracterización de la fabricación de dulces en la micro empresa “Anly” ..	19
3.2.	Proceso de batido	20
3.3.	Selección de las variables.....	21
3.4.	Automatización del proceso	22
3.5.	Diseño de la estructura metálica.....	25
3.6.	Cálculo del nivel mediante el sensor “HC- SR04”.....	27
3.7.	Lectura y ubicación del sensor de temperatura	30

3.8.	Diseño del control difuso	30
3.8.1.	Diseño de los conjuntos difusos	32
3.8.2.	Diseño de las reglas lógicas	39
3.9.	Simulación del control difuso con “Fuzzy Logic Toolbox”.	40
3.9.1.	Creación del controlador difuso con “Fuzzy Logic Toolbox”.	40
3.10.	Implementación del control difuso en el sketch de “Arduino”.....	45
3.10.1.	Implementación de la salida difusa en el variador de frecuencia.	49
CAPÍTULO 4		55
PRUEBAS Y RESULTADOS		55
4.1.	Descripción de la máquina automatizada.....	55
4.2.	Funcionamiento de la máquina automatizada	56
4.3.	Pruebas de funcionamiento del controlador difuso	58
4.4.	Pruebas de medición del tiempo de batido.....	59
4.5.	Pruebas de medición del desperdicio de gelatina	60
4.6.	Comparación entre el volumen de batido y el tiempo de batido.	61
CAPÍTULO 5		63
CONCLUSIONES Y RECOMENDACIONES		63
BIBLIOGRAFÍA		66
ANEXO		

ÍNDICE DE ILUSTRACIONES:

Figura 2.1. Presentación comercial y física de la grenetina.	3
Figura 2.2. Diagrama de un sistema de control simple.....	4
Figura 2.3. Diagrama de un sistema de control en lazo abierto.	5
Figura 2.4. Diagrama de un sistema de control en lazo cerrado.....	5
Figura 2.5. Ejemplo de función de membresía y su representación gráfica.....	6
Figura 2.6. Función de membresía triangular y su representación gráfica.	7
Figura 2.7. Función de membresía trapezoidal y su representación gráfica.	8
Figura 2.8. Función de membresía tipo” S” y su representación gráfica.	8
Figura 2.9. Operación unión entre conjuntos difusos.	9
Figura 2.10. Operación intersección entre conjuntos difusos.	10
Figura 2.11. Operación complemento.....	10
Figura 2.12. Diagrama de un sistema de control difuso.....	11
Figura 2.13. Herramientas de lógica difusa de “Matlab”.....	12
Figura 2.14. Tarjeta de control “Arduino Nano”.....	14
Figura 2.15. Sensor de nivel.	14
Figura 2.16. Sensor de temperatura.	15
Figura 2.17. Sistema numérico representativo del inversor marca “LG”.	16
Figura 2.18. Diagrama de conexiones del inversor.	17
Figura 3.1. Flujograma de la fabricación de dulces artesanales en la microempresa.	19
Figura 3.2. Motores utilizados para el batido de gelatina.	21
Figura 3.3. Diagrama eléctrico de la batidora no automatizada.	22
Figura 3.4. Sistema de control lazo abierto usado por la batidora no automatizada. .	23
Figura 3.5. Sistema de control de lazo cerrado implementado por la nueva batidora	24
Figura 3.6. Diagrama eléctrico de la nueva batidora para la micro empresa.	25
Figura 3.7. Elementos principales de la máquina.	25
Figura 3.8. Diseño de la máquina asistido por computadora.	26
Figura 3.9. Diseño de la máquina en “Auto CAD”.	27
Figura 3.10. Funcionamiento del sensor de nivel.	28
Figura 3.11. Ubicación del sensor de nivel.	29
Figura 3.12. Ubicación del sensor de temperatura.	30
Figura 3.13. Tarjeta de control principal basada en “Arduino Nano”.	31
Figura 3.14. Primer diseño de los conjuntos difusos para el <i>nivel</i>	33

Figura 3.15. Diseño final de los conjuntos difusos para el <i>nivel</i>	34
Figura 3.16. Primer diseño de los conjuntos difusos para la <i>temperatura</i>	35
Figura 3.17. Diseño final de los conjuntos difusos para la <i>temperatura</i>	36
Figura 3.18. Primer diseño de los conjuntos difusos para la <i>frecuencia</i>	37
Figura 3.19. Diseño final de los conjuntos difusos para la <i>frecuencia</i>	38
Figura 3.20. Ventana de diseño de lógica difusa.	40
Figura 3.21. Incorporación de una segunda variable de entrada.	41
Figura 3.22. Renombrar variables del controlador difuso.	42
Figura 3.23. Editor de funciones de membresía.	43
Figura 3.24. Editor de reglas lógicas.	44
Figura 3.25. Visualizador de reglas lógicas.	45
Figura 3.26. Creación de los conjuntos difusos en “Arduino”.	46
Figura 3.27. Creación de las reglas lógicas en “Arduino”.	47
Figura 3.28. Fusificación y defusificación en “Arduino”.	48
Figura 3.29. Relación entre los parámetros voltaje y frecuencia del inversor.	49
Figura 3.30. Amplificador con ganancia igual a dos.	51
Figura 3.31. Gráfica de los valores de frecuencia de control y “PWM”.	53
Figura 4.1. Máquina automatizada	56
Figura 4.2. Flujograma del batido con la máquina automatizada.	57
Figura 4.3. Gráfica comparativa del tiempo de batido.	59
Figura 4.4. Gráfica comparativa del desperdicio de gelatina.	61

ÍNDICE DE TABLAS:

Tabla 2.1. Comportamiento empírico de la gelatina bajo determinadas condiciones.	4
Tabla 2.2. Terminales para el control de frecuencia por voltaje “V1” y corriente “I”	18
Tabla 3.1. Datos obtenidos de forma empírica por los obreros de la micro empresa.	21
Tabla 3.2. Reglas lógicas para el control difuso.....	39
Tabla 3.3. Valores de voltaje de entrada al variador en función de “PWM”.	50
Tabla 3.4. Voltaje y “PWM” de “Arduino” tras implementar el amplificador por dos.....	52
Tabla 4.1. Valores obtenidos de la primera prueba.	58
Tabla 4.2. Medición de los tiempos de batido.....	59
Tabla 4.3. Medición del desperdicio de gelatina.	60
Tabla 4.4. Comparación del volumen de producción y el tiempo de batido.....	61

RESUMEN

En el sistema propuesto se implementó la lógica difusa para el control de velocidad del motor que realiza el batido en la fabricación de dulces artesanales (gomitas). La implementación del controlador difuso aporta una velocidad óptima al motor en función del nivel y la temperatura de la gelatina, con el fin de automatizar el proceso para reducir el desperdicio de materia prima y mejorar el volumen de producción de la micro empresa “Anly”.

El nivel de la gelatina en el recipiente y la temperatura de la misma se tomaron como variables de entrada del controlador difuso, mientras que, como salida se tomó a la frecuencia del inversor, quien tiene una relación directamente proporcional con la velocidad del eje del motor.

El cerebro del controlador difuso es un “Arduino Nano”, una placa de desarrollo pequeña y completa basada en el microcontrolador “ATmega328P”. Los sensores que entregan la información sobre las condiciones de la gelatina de acuerdo a la magnitud a medir son: un sensor ultrasónico para conocer el nivel de crecimiento de la misma dentro del recipiente y una termocupla tipo K conectada al amplificador “MAX6675” para saber la temperatura de la sustancia. La salida del controlador difuso entregada por “Arduino” se acopló a un variador de frecuencia “SV008iG5-2” a través de un hardware diseñado que otorga una variación de voltaje acorde a la entrada de control analógica del inversor.

Una comparación de la producción antes y después de la automatización mostró un índice de reducción de desperdicio del 97.5 %, se disminuyó el tiempo del proceso en 27 segundos y se aumentó el volumen de gelatina batida en 1.35 L. Los resultados obtenidos por la máquina fueron satisfactorios para la micro empresa.

ABSTRACT

In the proposed system, the fuzzy logic for the speed control of the motor that makes the shake in the manufacture of artisan sweets (gummies) was implemented. The implementation of the fuzzy controller provides an optimal speed to the engine depending on the level and temperature of the gelatin, in order to automate the process to reduce raw material waste and improve the volume of production of the micro company "Anly".

The level of the gelatin in the container and the temperature thereof were taken as input variables of the fuzzy controller, while, as an output, it was taken at the frequency of the inverter, which has a relationship directly proportional to the speed of the motor shaft.

The brain of the fuzzy controller is an "Arduino Nano", a small and complete development board based on the "ATmega328P" microcontroller. The sensors that provide information on the conditions of the gelatin according to the magnitude to be measured are: An ultrasonic sensor to know the level of growth of the same inside the container and a thermocouple type K connected to the amplifier "MAX6675" to know the temperature of the substance. The output of the fuzzy controller delivered by "Arduino" was coupled to a frequency inverter "SV008iG5-2" through a hardware designed that gives a variation of voltage according to the analog control input of the inverter.

A comparison of the production before and after the automation showed a waste reduction index of 97.5%, the process time was reduced in 27 seconds and the volume of whipped gelatin was increased by 1.35 L. The results obtained by the machine were satisfactory for the micro company.

INTRODUCCIÓN

El proyecto se dividió en cinco capítulos que detallarán lo siguiente:

Capítulo 1: Incluye la descripción del problema encontrado en la micro empresa, la justificación del trabajo y los objetivos propuestos para el mismo.

Capítulo 2: Muestra toda la información teórica utilizada en el progreso del proyecto, se incluye el diagrama eléctrico con los datos técnicos del inversor utilizado y se detalla el uso de la herramienta de lógica difusa que posee Matlab.

Capítulo 3: En esta sección se puede observar la caracterización del proceso, el diseño del controlador difuso y también de la estructura metálica de la máquina, los cálculos y ecuaciones realizados para escalar la salida difusa y la forma de cómo se implementó en el inversor para obtener una variación de velocidad del eje del motor. Por último, se indica cómo se simuló el control difuso en Matlab.

Capítulo 4: Muestra una breve descripción sobre la máquina automatizada, se observan los resultados de las pruebas de funcionamiento del controlador difuso respecto a la simulación en Matlab y también el de las pruebas de desperdicio de gelatina y tiempo del proceso. Además, se realizó un análisis comparativo entre el volumen de producción y el tiempo de batido.

Capítulo 5: En esta sección se muestran las conclusiones obtenidas al finalizar el proyecto con las respectivas recomendaciones para mejorarlo y para garantizar el correcto funcionamiento de la máquina automatizada.

CAPÍTULO 1

ANTECEDENTES

En este capítulo se detalla el problema de estudio con la respectiva justificación del trabajo, el objetivo general y los objetivos específicos propuestos para el desarrollo del proyecto técnico de titulación.

1.1. Planteamiento del problema

La micro empresa “Anly”, ubicada al Sur de Quito en el barrio Jardín del Valle, lleva doce años en el mercado nacional produciendo dulces hechos con gelatina (gomitas) de forma artesanal. Se evidencia un problema de desperdicio de producto en el batido, debido a que para este proceso se cuenta con dos motores, cada uno con una paleta acoplada de forma mecánica a su eje y un soporte metálico donde van ubicados los recipientes que contienen la sustancia caliente.

El encendido de los motores para el batido es de forma directa, es decir, con un interruptor. Razón por la cual, el motor arranca con toda su potencia y salpica la gelatina caliente generando desperdicio, ésta se mezcla, aumentando su volumen y produciendo una mayor carga al motor.

El batido continúa hasta que la gelatina alcance el límite del recipiente; luego, se procede a apagar el motor de forma manual a través del interruptor, en caso de no hacerlo a tiempo ésta empezará a desbordarse produciendo más desperdicio.

Los motores no tienen un control de velocidad que les permita arrancar de forma suave y tener un rango de variación de acuerdo al nivel de gelatina y temperatura de la misma. Empíricamente se observó que la velocidad del motor influye en el tiempo que demora el batido, por esta razón existe un tiempo de improductividad durante este proceso ocasionando jornadas de trabajo extensas o poco productivas.

1.2. Justificación

La lógica difusa es una ciencia que utiliza la matemática convencional para brindar soluciones. Es por eso, que los controladores difusos son una excelente opción al momento de automatizar si no se conoce el modelo matemático de la planta o proceso.

Si se implementa la lógica difusa para el control de velocidad de un motor se lograría automatizar un proceso. Al disponer del elemento principal del batido (motor) se plantea el desarrollo de un controlador difuso para variar la velocidad del mismo en función del nivel y la temperatura de la gelatina, con el fin de mejorar el proceso para reducir el desperdicio de materia prima y mejorar el volumen de producción. Esto ayudará a cubrir la creciente demanda de pedidos que tiene la micro empresa.

Además, debido a la falta de un control de velocidad para el motor, la producción de gomas en la micro empresa genera un tiempo de improductividad durante el batido de gelatina, con la implementación del controlador difuso se reducirá este tiempo para aumentar la cantidad de gelatina batida.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema con control difuso para automatizar el batido de gelatina en la fabricación de dulces artesanales.

1.3.2. Objetivos Específicos

- Caracterizar el proceso de batido de gelatina para identificar las variables de control que intervienen en el mismo.
- Diseñar el software y hardware de un controlador difuso para el proceso de batido de gelatina.
- Implementar el controlador difuso para la automatización del batido de gelatina.
- Comparar la producción antes y después de la automatización para obtener índices que permitan el análisis del sistema automatizado.

CAPÍTULO 2

MARCO TEÓRICO

Este capítulo describe los fundamentos teóricos, incluye diagramas de control e información sobre los datos técnicos de cada uno de los elementos utilizados para el desarrollo del proyecto.

2.1. La Gernetina

La gernetina es una proteína obtenida de la piel, tejidos y cartílagos de vacas y cerdos, tras ser sometidos a un proceso denominado hidrólisis del colágeno. (Zinet Media Group, 2001). En la Figura 2.1 se muestra como es la gernetina, físicamente, y su presentación comercial.

Figura 2.1. Presentación comercial y física de la gernetina.



Aquí se muestra físicamente la gernetina, el principal elemento de las gomitas. Fuente: (Socialhy.com)

La gernetina, más conocida como gelatina a nivel comercial, debe ser hidratada con agua fría para posteriormente ser mezclada con agua azucarada a alta temperatura y conseguir un compuesto en forma de jarabe dulce, que será la base de las gomitas.

Luego de conocer el origen del elemento principal que compone la gelatina, es necesario conocer cómo se comporta la misma bajo ciertas condiciones. Para ello, se ha tomado en cuenta el conocimiento empírico de quienes trabajan diariamente en la fabricación de estos deliciosos dulces artesanales, dando como resultado la Tabla 2.1.

Tabla 2.1. Comportamiento empírico de la gelatina bajo determinadas condiciones.

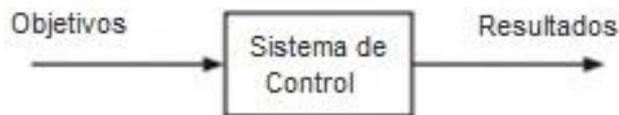
PARÁMETRO	CONDICIÓN	COMPORTAMIENTO
AGUA	Caliente	Soluble
	Fria	Absorbente
TEMPERATURA	Superior a 28° C	Acuosa
	Temp. Ambiente	Estable
	Inferior a 18° C	Sólida
CANTIDAD DE AGUA	Demasiada	Diluida
	Insuficiente	Densa

La gelatina presenta diversos comportamientos empíricos de acuerdo a las condiciones a las que esté sometida. Elaborado por: Steven Gordillo

2.2. Sistemas de control

Hace referencia a los elementos que trabajan conjuntamente para obtener una salida deseada ante la presencia de una entrada específica. (Carrillo, 2011). En la Figura 2.2 se muestra el diagrama de un sistema de control, sus componentes y la relación entre los mismos.

Figura 2.2. Diagrama de un sistema de control simple



Representación de un sistema de control. Fuente: (Carrillo, 2011, pág. 20). Elaborado por: Steven Gordillo.

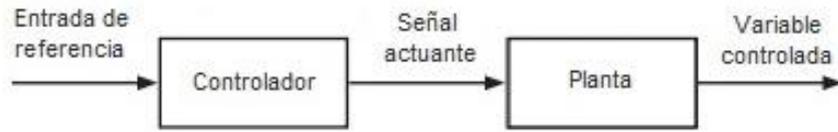
2.3. Tipos de sistemas de control

Entre los más habituales tenemos:

2.3.1. Sistema de control a lazo abierto

Se refiere a un sistema cuya variable controlada no es comparada con la entrada de referencia, así como se muestra en la Figura 2.3. (Carrillo, 2011)

Figura 2.3. Diagrama de un sistema de control en lazo abierto.

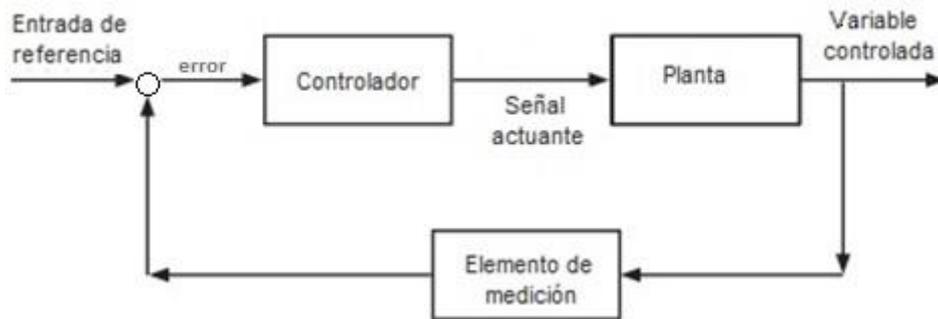


Representación de un sistema de control en lazo abierto. Fuente: (Carrillo, 2011, pág. 24). Elaborado por: Steven Gordillo.

2.3.2. Sistema de control a lazo cerrado

Se refiere a un sistema que utiliza elementos de medición (sensores) para realimentarse, determinando un error de desempeño entre las señales de entrada y salida para corregir la variable controlada de acuerdo al punto de ajuste establecido, tal como se muestra en el diagrama de control de la Figura 2.4. (Carrillo, 2011)

Figura 2.4. Diagrama de un sistema de control en lazo cerrado.



Representación de un sistema de control en lazo cerrado. Fuente: (Carrillo, 2011, pág. 25). Elaborado por: Steven Gordillo.

Al comparar la señal realimentada con la entrada de referencia se tiene como resultado el error de desempeño, el cual es reducido mediante la acción de control. (Carrillo, 2011).

2.4. Fundamentos de la lógica difusa

La lógica difusa es una ciencia desarrollada en base a la matemática convencional y la lógica para dar solución a problemas no establecidos. Fue creada para emular el razonamiento humano y decidir acertadamente a pesar de una vaga información, tal como lo haría una persona. (Ponce, 2010)

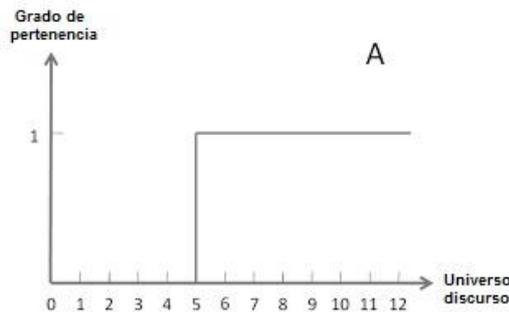
2.5. Función de membresía

Es la regla semántica que representa la ecuación matemática y define la pertenencia de un elemento en un conjunto difuso. (Ponce, 2010)

Por ejemplo, en la Figura 2.5, se observa gráficamente una función escalón con su respectiva ecuación matemática (Ec. 2.1). Donde la función tiene un universo discurso que va desde 0 hasta 12. El grado de pertenencia varía entre 0 y 1 dependiendo del valor del eje X. Si X presenta valores en el rango de 0 a 5 el grado de pertenencia se representa con un valor de 0, y si X es mayor o igual a 5 el grado de pertenencia se representa con un valor de 1.

Figura 2.5. Ejemplo de función de membresía y su representación gráfica.

$$\mu_A(x) = \begin{cases} 0, & 0 \leq x < 5 \\ 1, & x \geq 5 \end{cases} \quad \text{Ec. (2.1)}$$



Representación de una función de membresía. Fuente: (Gómez, 2011, pág. 5). Elaborado por: Steven Gordillo.

Donde $\mu_A(x)$ se refiere a la función de membresía que representa la pertenencia (eje y) de cada uno de los elementos del universo discurso (eje x) del conjunto difuso A. (Gómez, 2011)

2.6. Conjunto difuso.

Es la agrupación de elementos similares representados por una función de membresía. (Ponce, 2010)

2.7. Tipos de conjuntos difusos

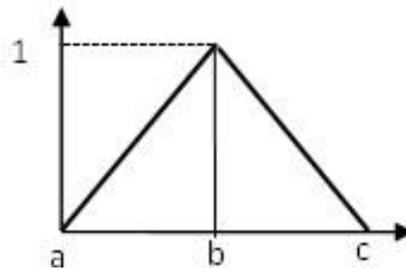
Existen muchas formas de representar los conjuntos difusos, pero entre los más comunes y utilizados encontraremos a las funciones de membresía de tipo triangular, trapezoidal y sigmoïdal o de tipo “S”.

2.7.1. Función triangular

Esta función tiene una pendiente ascendente, un valor unitario central y una pendiente descendente, tal como se observa en gráfica de la Figura 2.6. (Ponce, 2010)

Figura 2.6. Función de membresía triangular y su representación gráfica.

$$\mu(x) = \begin{cases} 0, & x < a \text{ ó } x > c \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b < x \leq c \end{cases} \quad \text{Ec. (2.2)}$$



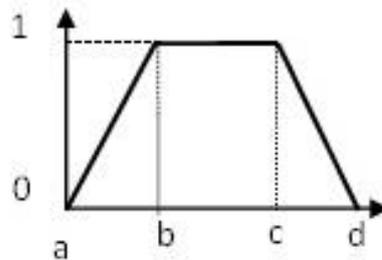
Representación de una función de membresía triangular. Fuente: (Gómez, 2011, pág. 8). Elaborado por: Steven Gordillo.

2.7.2. Función trapezoidal

Se trata de una función generalización de la función triangular que tiene un valor constante de pertenencia unitaria, con una pendiente que crece de forma ascendente “positiva” y otra que crece de forma descendente “negativa”, tal como se muestra en la gráfica de la Figura 2.7.

Figura 2.7. Función de membresía trapezoidal y su representación gráfica.

$$\mu(x) = \begin{cases} 0, & x < a \text{ ó } x > d \\ \frac{x-a}{b-a}, & a \leq x < b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c < x \leq d \end{cases} \quad \text{Ec. (2.3)}$$



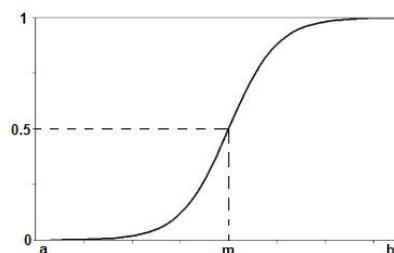
Representación de una función de membresía trapezoidal. Fuente: (Gómez, 2011, pág. 11). Elaborado por: Steven Gordillo.

2.7.3. Función sigmoideal

Conocida como función “S”, presenta una curva ascendente de segundo orden y termina con un estado de pertenencia unitario continuo, tal como se muestra en la gráfica de la Figura 2.8.

Figura 2.8. Función de membresía tipo “S” y su representación gráfica.

$$\mu(x) = \begin{cases} 0, & x < a \\ 2 \left(\frac{x-a}{b-a} \right)^2, & a \leq x \leq m \\ 1 - 2 \left(\frac{x-b}{b-a} \right)^2, & m < x \leq b \\ 1, & x > b \end{cases} \quad \text{Ec. (2.4)}$$



Representación de la función de membresía tipo “S”. Fuente: (Gómez, 2011, pág. 10). Elaborado por: Steven Gordillo.

2.8. Operaciones con conjuntos

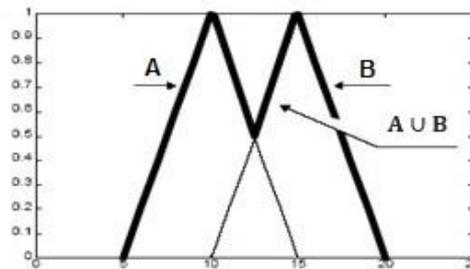
Los conjuntos difusos pueden realizar varias operaciones entre ellos, de modo que el resultado será otro conjunto. A continuación, se mostrará las tres operaciones básicas más implementadas en los conjuntos difusos, para ello se propone los conjuntos difusos A y B, determinados sobre un mismo universo de discurso (x). (Sanz & del Brio, 2007)

2.8.1. Operación unión

La operación unión entre conjuntos difusos viene a ser un equivalente de la operación lógica OR y radica en obtener el valor máximo entre los elementos de los conjuntos involucrados. La Figura 2.9, muestra el resultado de esta operación, que entrega un nuevo conjunto difuso establecido bajo el mismo universo de discurso y con la función de membresía dada por la Ec. 2.5. (Sanz & del Brio, 2007)

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad \text{Ec. (2.5)}$$

Figura 2.9. Operación unión entre conjuntos difusos.



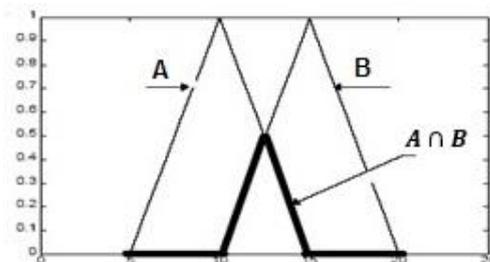
Representación gráfica del nuevo conjunto difuso $A \cup B$, Fuente: (Castillo, 2015). Elaborado por: Steven Gordillo.

2.8.2. Operación intersección

La operación intersección entre conjuntos difusos viene a ser un equivalente de la operación lógica AND y consiste en encontrar el valor mínimo entre los elementos de los conjuntos involucrados. La Figura 2.10, muestra el resultado de esta operación, que entrega un nuevo conjunto difuso determinado bajo el mismo universo de discurso y con la función de membresía entregada por la Ec. 2.6. (Sanz & del Brio, 2007)

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)] \quad \text{Ec. (2.6)}$$

Figura 2.10. Operación intersección entre conjuntos difusos.



Representación gráfica del nuevo conjunto difuso $A \cap B$, Fuente: (Castillo, 2015). Elaborado por: Steven Gordillo.

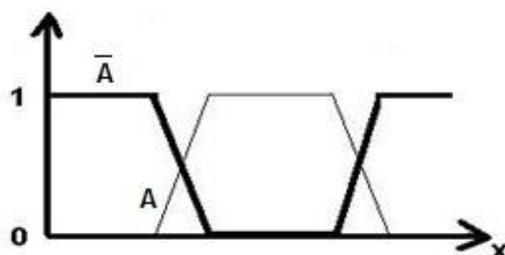
2.8.3. Operación complemento

La operación complemento en conjuntos difusos viene a ser un equivalente de la operación lógica NOT y consiste en calcular la diferencia entre uno y el conjunto al que se le aplica el complemento. La Figura 2.11, muestra el resultado de esta operación, que es un nuevo conjunto difuso determinado bajo el mismo universo de discurso y con la función de membresía dada por la Ec. 2.7 (Sanz & del Brio, 2007) .

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \text{Ec. (2.7)}$$

Se entiende que el complemento viene a ser el resto, la parte sobrante o lo contrario al conjunto que se aplica la operación.

Figura 2.11. Operación complemento.



Representación gráfica del nuevo conjunto difuso \bar{A} , Fuente: (Castillo, 2015). Elaborado por: Steven Gordillo.

2.9. Control difuso

La lógica difusa no requiere de un modelado matemático ya que se basa en los grados de pertenencia de los conjuntos. El controlador toma decisiones de acción sobre el sistema basándose en el conocimiento empírico de una o más personas. (Sanz & del Brio, 2007). La estructura generalmente empleada en un controlador difuso se muestra en la Figura 2.12.

Figura 2.12. Diagrama de un sistema de control difuso.



Representación de un sistema de control difuso, Fuente: (Sanz & del Brio, 2007, pág. 40). Elaborado por: Steven Gordillo.

2.9.1. Fusificación

La primera etapa denominada “Fusificación” es donde son procesados los valores de entrada para deducir el grado de pertenencia que tendrán dentro del controlador. (Sanz & del Brio, 2007)

2.9.2. Inferencia difusa

El motor de inferencia va de la mano con la base de conocimientos y efectúan la toma de decisiones que determinará el comportamiento del sistema. Las reglas lógicas son del tipo antecedente-consecuente (IF *< antecedente >* , THEN *< consecuente >*). (Sanz & del Brio, 2007)

2.9.3. Defusificación

La última etapa que se tiene dentro del controlador es el defusificador, quien a través de métodos matemáticos convierte los valores difusos conseguidos de la inferencia en valores utilizables para el proceso a controlar. (Sanz & del Brio, 2007).

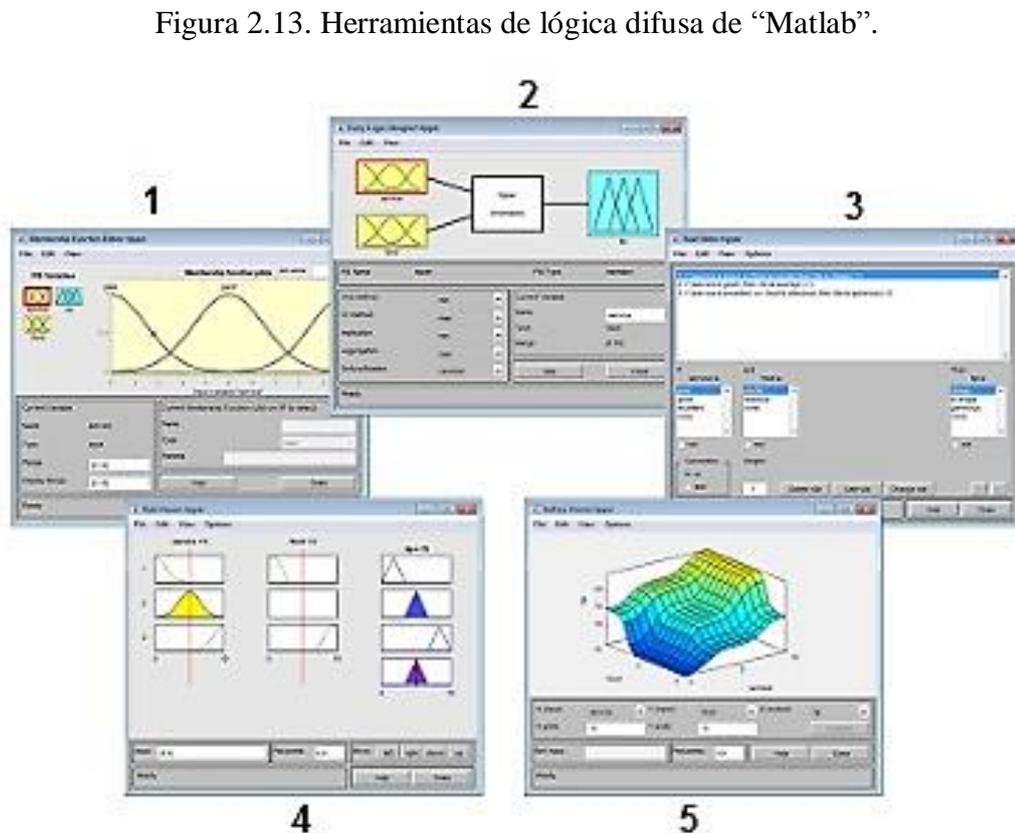
2.10. Software de aplicación con lógica difusa

Existen varios softwares de aplicación, pero el que más destaca es Matlab por ser un lenguaje de programación matemático de alto nivel que integra un entorno gráfico amigable perfecto para realizar aplicaciones en el campo de la mecatrónica, robótica, control y automatización. (Reyes, 2012)

Un sistema de lógica difusa puede ser simulado e implementado por este software a través de una herramienta denominada “Fuzzy Logic Toolbox”.

2.10.1. “Fuzzy Logic Toolbox”

Al usar la herramienta de lógica difusa, es posible crear el conjunto de reglas, definir las funciones de membresía y analizar el comportamiento de un sistema de inferencia difusa (FIS). “Fuzzy Logic Toolbox” proporciona los siguientes editores y espectadores: (MathWorks, 2018)



1. Editor de funciones de membresía, 2. Editor FIS (Sistema de inferencia difuso), 3. Editor de reglas lógicas, 4. Visualizador de reglas lógicas y 5. Visualizador de superficie. Fuente: (MathWorks, 2018).

En la Figura 2.13 se observan las herramientas que proporciona el “Fuzzy Logic Toolbox” de Matlab. Las funciones que cumplen cada una de ellas son:

- **Editor de funciones de membresía:** le permite visualizar y editar las funciones de membresía asociadas con las variables de entrada y salida del FIS.
- **Editor FIS (Sistemas de Inferencia Difusa):** muestra información general sobre un sistema de inferencia difuso.
- **Editor de reglas lógicas:** le permite ver y editar reglas difusas utilizando uno de los tres formatos: sintaxis similar a la del inglés completo, notación simbólica concisa o notación indexada.
- **Visualizador de reglas lógicas:** le permite ver el comportamiento detallado de un FIS para ayudar a diagnosticar el comportamiento de reglas específicas o estudiar el efecto de cambiar las variables de entrada.
- **Visualizador de superficie:** genera una superficie 3D a partir de dos variables de entrada y la salida de un FIS. (MathWorks, 2018)

2.11. Tarjeta de control “Arduino Nano”.

Es una tarjeta de prototipado electrónico basada en el microcontrolador “ATmega328”. Tiene el mismo número de pines y la misma funcionalidad que el “Arduino UNO”, solo que en un tamaño reducido e incorpora un conector USB mini-B en vez del conector USB tipo B. (Torrente, 2013)

La placa “Arduino Nano” se muestra en la Figura 2.14, (Para más información visite el link: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>)

Figura 2.14. Tarjeta de control “Arduino Nano”.



“Arduino Nano” con cable USB 2.0 para programación. Fuente: (Amazon.com).

2.12. Sensor ultrasónico de nivel “HC-SR04”

Es un sensor capaz de medir distancias desde los 3 cm hasta los 300 cm. El emisor (uno de los cilindros que se aprecia en la Figura 2.15) envía una señal de ultrasonido y espera a que este rebote sobre un objeto y sea detectado por el receptor. El sensor entrega el tiempo transcurrido entre el envío y la recepción de la onda. (Torrente, 2013).

(Para más información visite el link: <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/sensor-ultrasonico-hc-sr04-y-arduino/>)

Figura 2.15. Sensor de nivel.



Sensor ultrasónico “HC-SR04” para medir el nivel de la gelatina. Fuente: (BestElectronics, 2018)

El sensor de la Figura 2.15 es muy utilizado en proyectos de electrónica, robótica y automatización de sistemas de nivel o distancia debido a su bajo costo y buenos resultados.

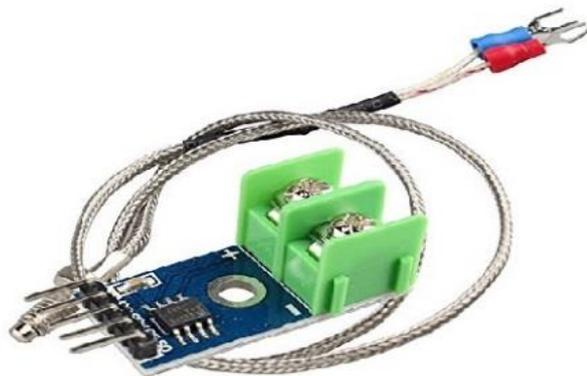
2.13. Termopar tipo “K” y módulo amplificador “MAX6675”.

Un termopar está formado por la unión de dos metales distintos que generan un pequeño voltaje debido a la diferencia de temperaturas entre sus extremos. El termopar no se puede conectar fácilmente al microcontrolador, debido a que genera una señal muy pequeña.

El módulo “Max6675” que se muestra en la Figura 2.16, soluciona este inconveniente ya que internamente tiene la electrónica analógica necesaria para amplificar, compensar y convertir a digital la señal generada por el termopar, lo que facilitará la lectura de la temperatura. (GeekFactory.mx, 2018).

(Para más información visite el link: <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/termopar-con-arduino-y-max6675-medicion-de-temperatura/>).

Figura 2.16. Sensor de temperatura.

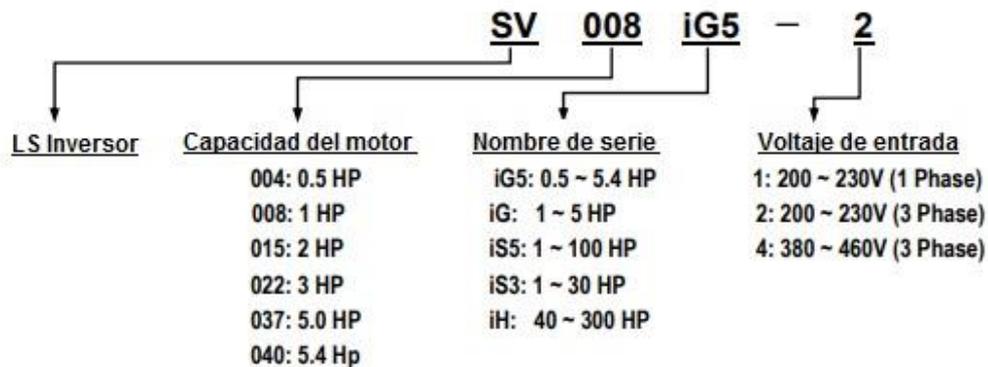


Termopar tipo K y módulo “Max6675”. Fuente: (Wantitall, 2018).

2.14. Caracterización del variador de frecuencia

El variador de frecuencia o inversor “SV008iG5-2” de la marca “LG”, es un pre-actuador, que será implementado para controlar la velocidad de un motor, de acuerdo a ciertas condiciones de temperatura y nivel de gelatina, la variable de control que gobernará al inversor vendrá de un controlador difuso.

Figura 2.17. Sistema numérico representativo del inversor marca “LG”.



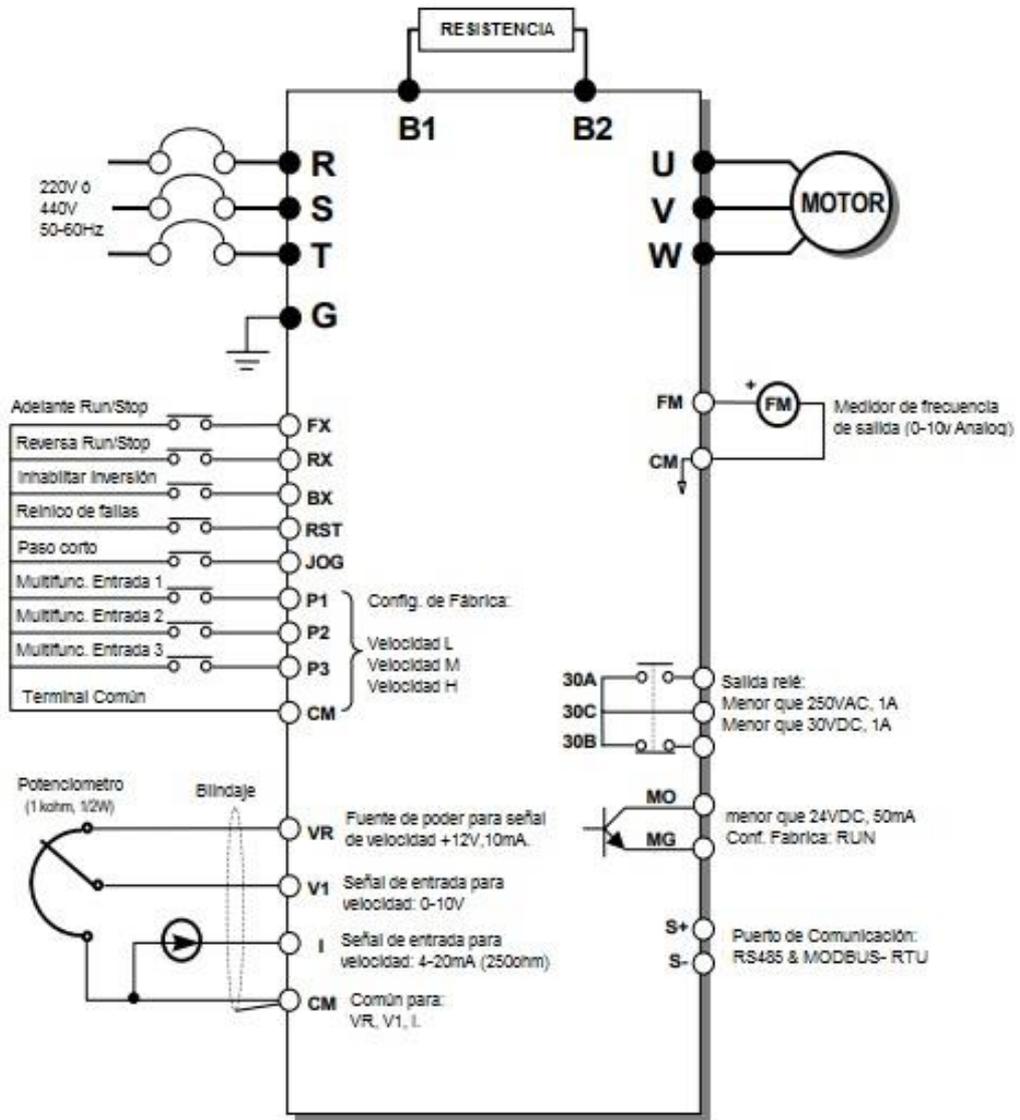
El código describe un inversor para un motor trifásico de 1HP que funciona con 220V. Fuente: (Isis, 2018). Elaborado por: Steven Gordillo.

Según la Figura 2.17, el inversor “SV008iG5-2” fue diseñado para un motor de hasta 1 HP de potencia y funciona con un voltaje máximo de hasta 230 VAC.

En la Figura 2.18 se muestra un diagrama de conexiones del variador de frecuencia, donde se evidencia que tiene terminales de entrada “R”, “S”, “T” para su alimentación monofásica o trifásica, terminales de salida “U”, “V”, “W” para energizar al motor trifásico, terminales “B1”, “B2” para colocar una resistencia de frenado dinámico, terminales “FM”, “CM” para conectar un medidor de frecuencia, terminales “30A”, “30C” y “30B” que son salidas tipo relé, los pines “S+”, “S-“ hacen referencia al puerto de comunicación “RS585” o “MODBUS-RTU”.

El inversor posee entradas analógicas “VR”, “V1”, “T”, “CM” para el control de frecuencia por voltaje y por corriente, además ofrece terminales digitales de control externo “FX”, “RX”, “P1”, “P2”, “P3” para invertir el giro del motor o poner velocidades establecidas por defecto que aumentan 10 Hz respecto a la anterior.

Figura 2.18. Diagrama de conexiones del inversor.



Descripción de los terminales que ofrece el inversor y su diagrama de conexiones. Fuente: (Isis, 2018).
Elaborado por: Steven Gordillo.

La Tabla 2.2, muestra las dos entradas analógicas del inversor, que sirven para cambiar el valor de la frecuencia mediante la variación del voltaje en “V1” con un rango de 0 a 10 V o a través de la modificación de corriente en “I” en un rango de 4 a 20 mA. El pin “VR” corresponde a la alimentación de 12 V del variador y “CM” es el terminal común para ajuste de frecuencia por voltaje o corriente.

Tabla 2.2. Terminales para el control de frecuencia por voltaje “V1” y corriente “I”.

Nombre	Tipo	Descripción
VR	Alimentación de ajuste de Frecuencia (+10V)	Usado para alimentar el ajuste de frecuencia, maxima salida de +12V, 10mA.
V1	Frecuencia de referencia (por voltaje)	Usado como entrada de referencia de frecuencia de 0-10V. Usar resistencia de 20K Ω .
I	Frecuencia de referencia (por corriente)	Usado como entrada de referencia de frecuencia de 4-20mA. Usar resistencia de 250 Ω .
CM	Terminal común	Terminal común para el ajuste de control externo por voltaje o corriente.

Descripción de las entradas analógicas para control de frecuencia por voltaje o corriente. Fuente: (Isis, 2018). Elaborado por: Steven Gordillo.

CAPÍTULO 3

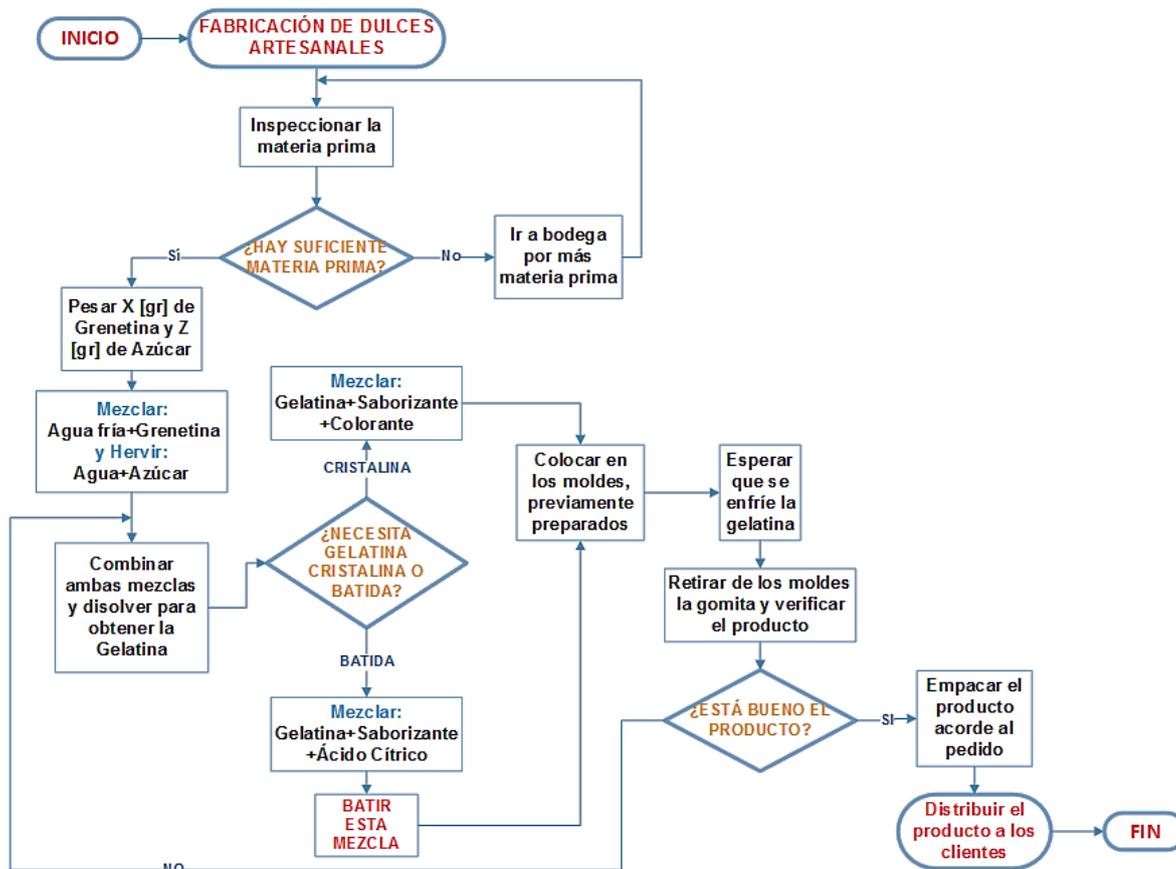
DISEÑO E IMPLEMENTACIÓN

En este capítulo se muestra la caracterización del proceso, el diseño del controlador difuso y también de la estructura metálica de la máquina, los cálculos y ecuaciones realizados para escalar la salida difusa y la forma de cómo se implementó en el inversor para obtener una variación de velocidad del eje del motor. Al final del capítulo, se indica cómo se simuló el control difuso en Matlab.

3.1. Caracterización de la fabricación de dulces en la micro empresa “Anly”.

La micro empresa “Anly”, ubicada al Sur de Quito en el barrio Jardín del Valle, lleva doce años en el mercado nacional produciendo dulces hechos con gelatina (gomitas) de forma artesanal. El proceso de fabricación se muestra en la Figura 3.1.

Figura 3.1. Flujograma de la fabricación de dulces artesanales en la micro empresa.



Procesos detallados para la fabricación de gomitas. Elaborado por: Steven Gordillo.

Como se puede observar en el flujograma de la Figura 3.1, el proceso comienza con la medición de la materia prima (Grenetina y azúcar), a continuación, se hidrata la grenetina en agua fría y se hierve el agua con azúcar. Luego, se combinan ambas y se disuelve obteniendo la gelatina compuesta que es una mezcla ligera similar a la miel caliente, se prepara por separado dos tipos de combinaciones de acuerdo a la clase de dulce que se desea obtener.

- **Gelatina cristalina:** Se obtiene combinando la gelatina con colorantes artificiales, esencias frutales y un poquito de ácido cítrico para lograr varias combinaciones de colores, sabores y nivel de acidez. Cabe mencionar que esta gelatina no necesita batirse.
- **Gelatina batida:** Es el resultado de mezclar la gelatina con un poquito de ácido cítrico y esencia frutal. Esto se consigue después de pasar por un proceso de batido que la hará crecer, es decir, aumentar su volumen. Además, adquirirá una tonalidad blanquecina, similar a lo que sucede cuando se realiza la tradicional y conocida espumilla.

En base a estas combinaciones se podrían realizar muchos dulces, así que lo siguiente es preparar los moldes esparciendo un ligero rocío de aceite en spray y harina de maíz para evitar que las gomitas se queden pegadas en el molde al endurecerse la mezcla.

Después de haberse endurecido por completo las gomitas, se procede a retirarlas de los moldes para empacarlas y distribuir las de acuerdo al pedido del cliente.

3.2. Proceso de batido

El batido es el único proceso ayudado por un motor eléctrico, ya que el resto de ellos son totalmente manuales. Al batir, se invierte menor cantidad de gelatina para cubrir un mayor rango de producción a comparación de cuando solo se utiliza la mezcla cristalina, esto se debe a que la sustancia caliente al ser batida empieza a aumentar su volumen considerablemente. En el proceso sin automatizar intervinieron dos motores (motor 1 y motor 2) de los cuales se obtuvo información empírica sobre la influencia

que tiene la velocidad del eje del motor con respecto al estado del producto batido y el tiempo que se demoró en terminarlo de batir. Con estos datos se realizó la Tabla 3.1. Es preciso mencionar que el motor 1 se encontraba en buenas condiciones y el motor 2 estaba deteriorado.

Tabla 3.1. Datos obtenidos de forma empírica por los obreros de la micro empresa.

Motor	Estado del motor	Velocidad del motor [%]	Tiempo del proceso [min]	Estado de la gelatina
1	Buen Estado	100	8	Acuosa
2	Deteriorado	85	6	Espesa

Se puede observar que la velocidad del motor influye en el tiempo que demora el proceso y el estado de la gelatina. Elaborado por: Steven Gordillo.

Los resultados de la Tabla 3.1 reflejan que el motor deteriorado, brinda menos velocidad que su homólogo, pero termina de batir la gelatina antes que el que se encuentra en buen estado, por lo tanto, reduciendo la velocidad la mezcla crece más rápido y queda más espesa.

Figura 3.2. Motores utilizados para el batido de gelatina.



Motor 1 en buen estado de conservación. Motor 2 deteriorado. Elaborado por: Steven Gordillo.

La Figura 3.2 muestra los dos motores que aún conserva la micro empresa para el batido de gelatina en la elaboración de los dulces de forma artesanal.

3.3. Selección de las variables

En el proceso de batido se pudo haber tomado en cuenta como variables de entrada a ciertas condiciones físicas que están directamente relacionadas con la gelatina como son:

- La temperatura
- El nivel de crecimiento
- La viscosidad.

Por otro lado, como salida del sistema se tiene únicamente a la velocidad del eje del motor que realiza el batido.

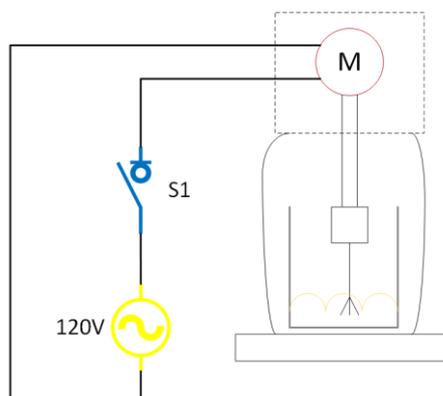
Se optó por realizar un sistema de control realimentado, donde se utilizó la temperatura y el nivel de crecimiento de la gelatina como variables de entrada para diseñar un controlador difuso bidimensional, descartando la viscosidad como una tercera variable de entrada por ser una condición compleja de medir.

La variable de salida del sistema es la frecuencia, cuya relación con la velocidad del motor es directamente proporcional. Dicho en otras palabras, se realizará el control de la velocidad del motor en función del nivel y la temperatura de la gelatina.

3.4. Automatización del proceso

En el proceso sin automatizar, el batido requería la presencia de un obrero que dé inicio al proceso y lo termine cuando la gelatina haya llegado al límite del recipiente. Esto, debido al arranque y paro que se realiza de forma manual desde un interruptor situado cerca del motor. La Figura 3.3 muestra un diagrama eléctrico de la batidora no automatizada.

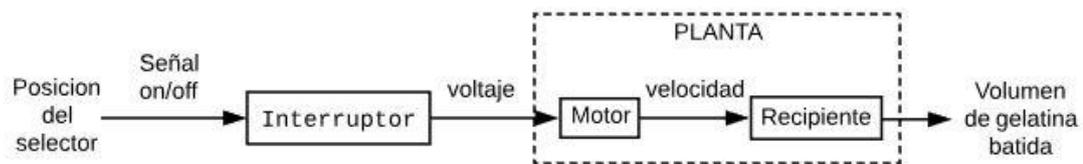
Figura 3.3. Diagrama eléctrico de la batidora no automatizada.



Aquí se muestra el esquema eléctrico de la batidora sin automatizar. Elaborado por: Steven Gordillo.

Como se puede observar en la Figura 3.3, la batidora con la que se trabajaba, era un simple motor monofásico energizado con control ON/OFF a través de un interruptor. Este tipo de control es conocido como lazo abierto, ya que no posee una realimentación que permita condicionar y corregir la salida, en la Figura 3.4 se muestra el diagrama de control de lazo abierto que posee la batidora no automatizada.

Figura 3.4. Sistema de control lazo abierto usado por la batidora no automatizada.



Control de lazo abierto (ON/OFF) de la máquina sin automatización. Elaborado por: Steven Gordillo.

Debido a este tipo de control que energiza al motor de forma directa, éste arranca con toda su potencia salpicando la gelatina caliente y generando desperdicio.

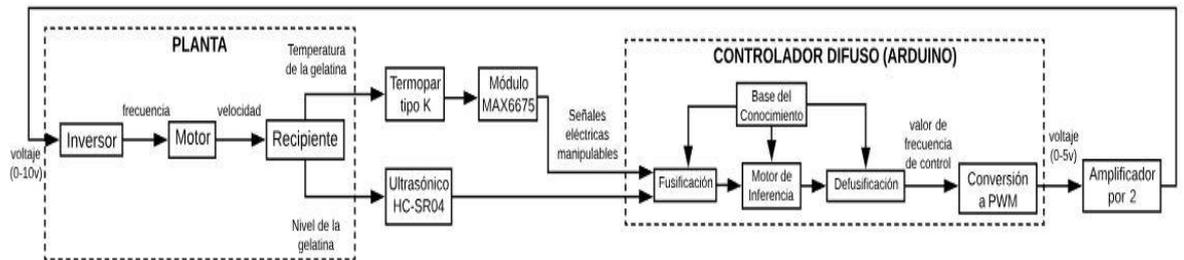
Según la Tabla 3.1 se observó que la velocidad del motor influye tanto en el tiempo que se demora la mezcla hasta llegar al límite del recipiente, como en la consistencia de la misma. Por estas razones, es recomendable realizar un control de lazo cerrado en el sistema para hacer correcciones en la salida.

Se optó por diseñar un sistema realimentado implementando un controlador difuso, debido a que se cuenta con una base de conocimiento empírico sobre el proceso y la fabricación de los dulces proporcionado por los obreros de la micro empresa, quienes llevan muchos años en esta actividad. Por lo tanto, se definieron reglas lógicas de acuerdo a la información compartida por los trabajadores para conseguir el funcionamiento adecuado de la máquina.

Otra de las opciones que se tomó en cuenta para diseñar el controlador difuso fue que no era necesario realizar un modelado matemático de la planta, ni depender de cálculos avanzados para realizar el control, bastó con utilizar la lógica y la matemática convencional basada en conjuntos.

Para la automatización de este proceso se pretende implementar la lógica difusa, emulando la lógica humana para una toma de decisiones acertada en base a una vaga información entregada por el sistema.

Figura 3.5. Sistema de control de lazo cerrado implementado por la nueva batidora.



Sistema realimentado con control difuso. Elaborado por: Steven Gordillo.

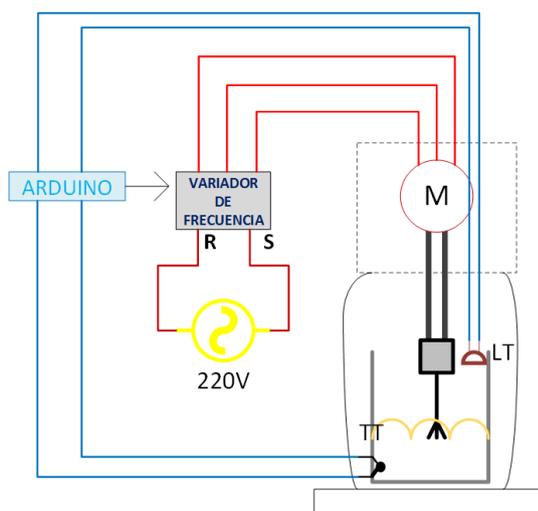
Como se puede observar en el diagrama de control de la Figura 3.5, se tiene una realimentación del sistema desde el controlador difuso hacia la planta a través de un amplificador por dos.

Los sensores toman las condiciones de la gelatina almacenada en el recipiente y generan una señal eléctrica manipulable para el “Arduino”, quien utilizando la lógica difusa entrega un valor de frecuencia al finalizar la defusificación y lo convierte a “PWM” para obtener una variación de voltaje de 0 a 5v a la salida de la placa. El amplificador con ganancia de dos, permite la conexión entre el “Arduino” y el inversor tras haber aumentado el voltaje a valores entre 0 y 10v.

En la Figura 3.6, se muestra el esquema eléctrico para la automatización de la batidora. El sensor de nivel y el de temperatura entregan respectivamente una señal eléctrica que “Arduino” utiliza para realizar la lógica difusa.

El controlador emite una señal de voltaje “PWM” que se debe amplificar para ser utilizada en el inversor y lograr la variación de velocidad en el eje del motor.

Figura 3.6. Diagrama eléctrico de la nueva batidora para la micro empresa.



Aquí se muestra el esquema eléctrico de la batidora automatizada. Elaborado por: Steven Gordillo.

3.5. Diseño de la estructura metálica

La estructura de la máquina, se diseñó en un software especializado en dibujo asistido por computadora, conocido como “Auto CAD”. Para dimensionar la máquina se tomó como punto de partida las medidas del recipiente de acero inoxidable, de la caja para el tablero de control y del motor trifásico adquirido con antelación.

En la parte izquierda de la Figura 3.7, se observa el motor trifásico, en el centro está el recipiente de acero inoxidable que tiene una capacidad de 6 L y en la parte derecha de dicha figura se muestra el tablero eléctrico.

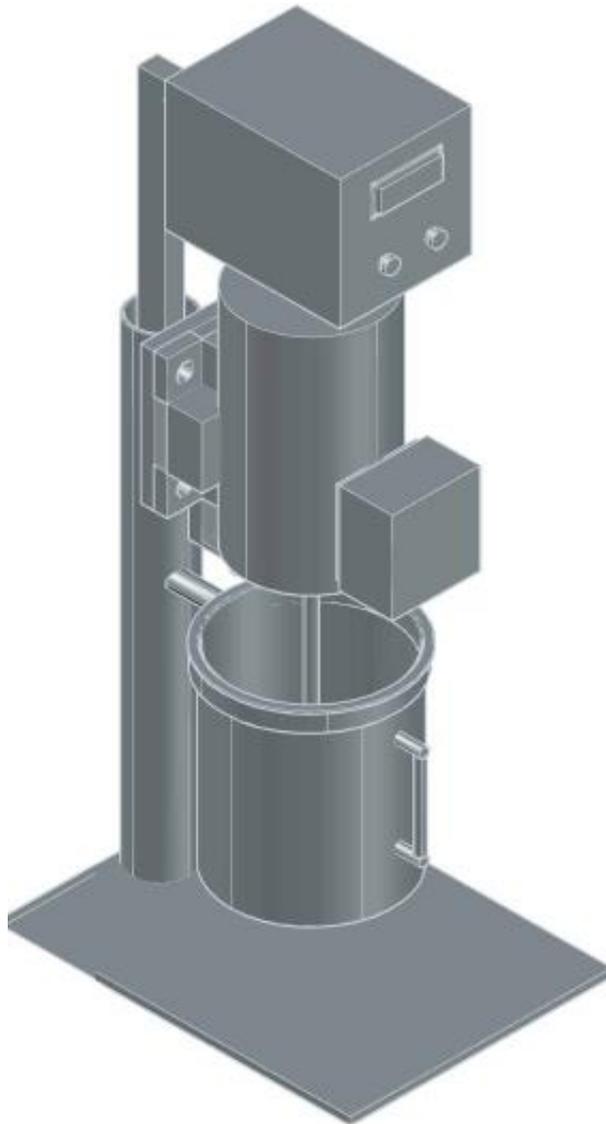
Figura 3.7. Elementos principales de la máquina.



Motor, recipiente y caja para tablero de control. Elaborado por: Steven Gordillo.

El esqueleto de la máquina será de acero fundido, mientras que las partes que tienen contacto directo con la gelatina, como el aspa para batir y el soporte del recipiente serán cuidadosamente realizados en acero inoxidable. En la Figura 3.8, se observa el diseño 3D en perspectiva isométrica de la batidora automatizada realizado en “CAD”.

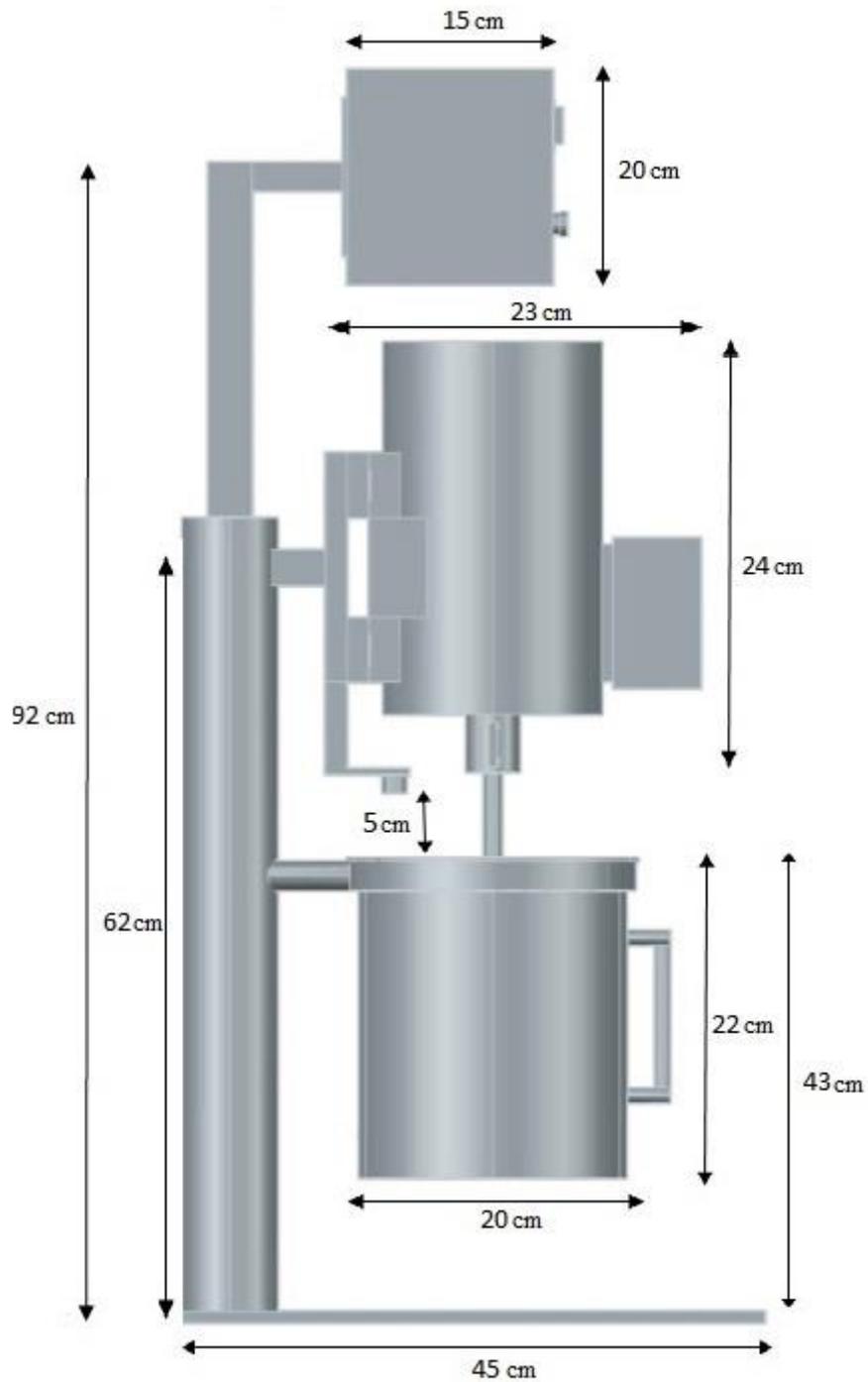
Figura 3.8. Diseño de la máquina asistido por computadora.



Dibujo 3D de la máquina con perspectiva isométrica. Elaborado por: Steven Gordillo.

La máquina quedará con un grosor de 45 cm, una altura de 100 cm y un ancho de 35 cm. En la Figura 3.9, se muestra una vista lateral con acotaciones de dimensionamiento del diseño 3D observado en la Figura 3.8.

Figura 3.9. Diseño de la máquina en “Auto CAD”.



Diseño a computadora de la máquina y sus dimensiones en centímetros, Elaborado por: Steven Gordillo.

3.6. Cálculo del nivel mediante el sensor “HC- SR04”

Para determinar la distancia que existe entre el sensor ultrasónico y la gelatina almacenada en el recipiente, es necesario saber que la velocidad del sonido es de $343 \frac{m}{s}$, bajo condiciones normales como temperatura ambiente, presión atmosférica a

nivel del mar y humedad relativa de 50%. Este valor es calculable en metros sobre segundos, pero fue necesario reducirlo a unidades más pequeñas y mostrarlo en centímetros sobre microsegundos, por lo tanto:

$$velocidad\ del\ sonido = 343.2 \frac{m}{s} \times 100 \frac{cm}{m} \times \frac{1}{1000000} \frac{m}{\mu s} = \frac{1}{29.1} \frac{cm}{\mu s}$$

Del cálculo anterior, se puede decir que el sonido tarda 29.2 μs en recorrer 1 cm. A partir de esto se determinó la distancia utilizando la siguiente fórmula:

$$d = v \times t$$

$$d = \left(\frac{1}{29.1} \frac{cm}{\mu s} \times \frac{tiempo [\mu s]}{2} \right) - 0.5$$

$$d = \left(\frac{tiempo [\mu s]}{58.2} \right) - 0.5 \quad \text{Ec. (3.1)}$$

En la Ecuación 3.1, se divide el tiempo para dos, ya que, como se puede apreciar en la Figura 3.10, el sensor envía una onda de ultrasonido que luego de rebotar con un objeto o sustancia es detectado por el receptor, emitiendo el tiempo transcurrido por la señal en ir y regresar, por esta razón, se debe tomar únicamente el tiempo de ida.

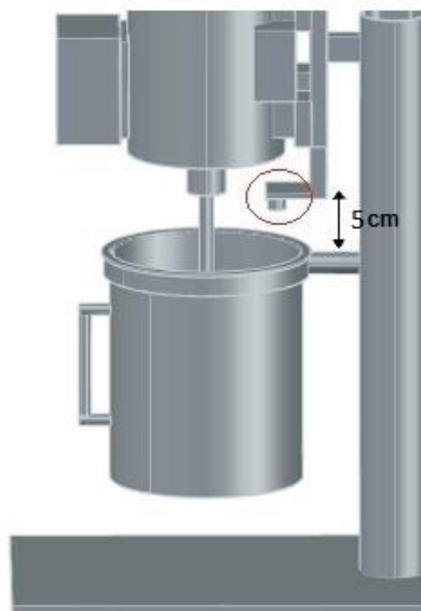
Figura 3.10. Funcionamiento del sensor de nivel.



Sensor ultrasónico enviando y recibiendo las ondas sonoras, luego de rebotar en un objeto. Fuente: (Llamas, 2018). Elaborado por: Steven Gordillo.

El valor de 0.5 que se resta en la Ec.3.1, es un valor obtenido empíricamente, que brinda mayor precisión en el cálculo de la distancia. Esto, debido a que en los cálculos se utiliza la constante de velocidad del sonido bajo ciertas condiciones que para este proyecto no se cumplen. Por ejemplo, en el caso de la presión atmosférica el valor difiere debido a los 2850 m de altura sobre el nivel del mar que se encuentra ubicada la Ciudad de Quito, donde se desarrolló el presente trabajo. Además, la temperatura también es diferente ya que el sensor estará expuesto al calor que emite la gelatina caliente.

Figura 3.11. Ubicación del sensor de nivel.



Sensor ultrasónico ubicado a 5 cm del recipiente. Elaborado por: Steven Gordillo.

El sensor de nivel estará situado a 5 cm de la parte superior del recipiente como se puede observar en la Figura 3.11, por lo tanto, se tendrá una lectura de 27 cm cuando el recipiente este vacío. Se enviará una señal al controlador para terminar el proceso de batido cuando el sensor dicte un valor de 9 cm, es decir, 4 cm por debajo del límite superior del recipiente.

3.7. Lectura y ubicación del sensor de temperatura

La implementación del módulo “MAX6675” y el termopar con “Arduino” se logró por medio de una librería, cuyo código fue modificado a conveniencia. En la Figura 3.12, se muestra la ubicación del sensor de temperatura.

Figura 3.12. Ubicación del sensor de temperatura.



Termopar tipo K ubicado al interior del recipiente. Elaborado por: Steven Gordillo.

El sensor de temperatura fue situado tangente al cuerpo del recipiente donde tenía contacto directo con la gelatina. Se observó una lectura de 40 °C cuando la mezcla alcanzó una consistencia espesa, es decir, empezó a pasar de estado líquido a sólido. Por el contrario, el sensor mostró una lectura de 95°C cuando la mezcla era completamente líquida y estaba cerca de llegar a su punto de ebullición.

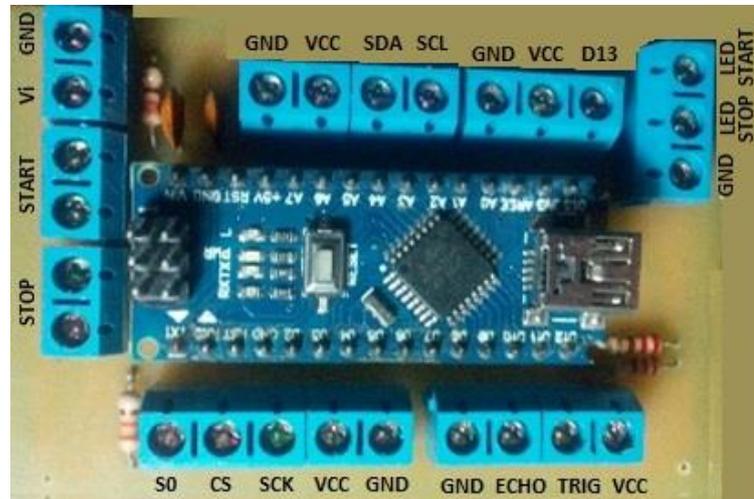
3.8. Diseño del control difuso

Para el desarrollo del control difuso se utilizó “Arduino”, que es una plataforma para prototipado electrónico de código abierto basada en hardware y software flexibles y fáciles de usar.

“Arduino” es relativamente barato a comparación de otras tarjetas de control, es multiplataforma ya que funciona con Windows, Macintosh y Linux, dispone de interrupciones, comunicación serial, pines con salida digital PWM y otras cualidades que hacen de la tarjeta una excelente opción para este proyecto.

Se utilizó el “Arduino Nano” que tiene un tamaño reducido ideal para colocarlo en un circuito PCB. Se colocó en el tablero de control como tarjeta principal.

Figura 3.13. Tarjeta de control principal basada en “Arduino Nano”.



Circuito PCB electrónico encargado del control difuso. Elaborado por: Steven Gordillo.

En la Figura 3.13 se observa la distribución de los pines de “Arduino Nano” en la tarjeta de control. A continuación, se detallará cada uno de ellos:

- **“GND”, “VCC”, “SDA” y “SCL”**: Son pines destinados para la conexión de la pantalla “LCD” a través del módulo de comunicación “I2C”.
- **“GND”, “VCC” y “D13”**: Son pines de salida a relé, precisamente en el pin “D13” se colocará una pequeña bocina.
- **“STOP” y “START”**: Son pines destinados para los pulsadores de inicio y paro del proceso respectivamente.
- **“GND” y “Vi”**: Son pines de alimentación principal de la tarjeta de control.
- **“S0”, “CS”, “SCK”, “VCC” y “GND”**: Son pines destinados para la conexión del termopar mediante el módulo amplificador “Max6675”.

- **“GND”, “ECHO”, “TRIG” y “VCC”**: Son pines destinados para la conexión del sensor ultrasónico.
- **“GND”, “LED START” y “LED STOP”**: Son pines destinados para conectar leds indicadores de inicio y paro de proceso respectivamente.

3.8.1. Diseño de los conjuntos difusos

Para empezar a diseñar los conjuntos difusos hay que tener en cuenta cuatro reglas muy importantes:

- Un conjunto difuso siempre se dibuja empezando en 1 y se termina dibujando en 1.
- Los conjuntos difusos deben solaparse obligatoriamente uno con su consecutivo al menos en un 50%, es decir deben coincidir en 0,5.
- Los conjuntos difusos pueden ser simétricos o asimétricos.
- Se debe dibujar conjuntos difusos en números impares, como mínimo 3 conjuntos y como máximo 9 conjuntos. (Sanz & del Brio, 2007)

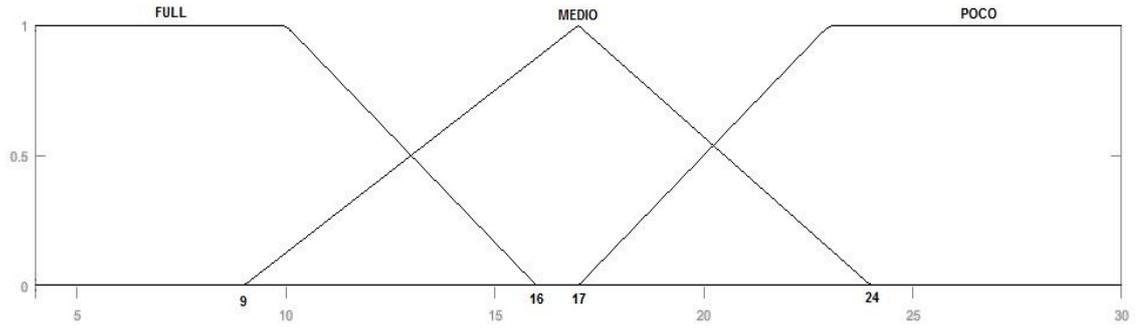
Para el proyecto se realizó varios diseños de conjuntos, en algunos se modificó el tipo de conjunto, los rangos e incluso los nombres, pero siempre respetando las reglas nombradas anteriormente.

A continuación, se detallan los diseños de los conjuntos difusos realizados para las entradas del sistema *“temperatura y nivel”* y para la salida del sistema *“frecuencia”*. Se expone tanto el primer diseño como el diseño final que fue implementado en la máquina.

- **Conjuntos difusos para la entrada *“Nivel”***

En la Figura 3.14, se observa que el nivel de la gelatina tubo un universo de discurso [0-30] cm para este primer diseño realizado sin tener claro a que distancia del recipiente iba a ser colocado el sensor ultrasónico. Este detectaba valores superiores a los 2 cm, así que el rango de [0-1] cm solo fueron valores teóricos.

Figura 3.14. Primer diseño de los conjuntos difusos para el *nivel*.



Conjuntos difusos de la entrada *nivel*. Elaborado por: Steven Gordillo.

$$\mu_{FULL}(x) = \begin{cases} 1, & 0 \leq x \leq 10 \\ \frac{16-x}{6}, & 10 < x \leq 16 \end{cases} \quad \text{Ec. (3.2)}$$

$$\mu_{MEDIO}(x) = \begin{cases} \frac{x-9}{8}, & 9 \leq x \leq 17 \\ \frac{24-x}{7}, & 17 < x \leq 24 \end{cases} \quad \text{Ec. (3.3)}$$

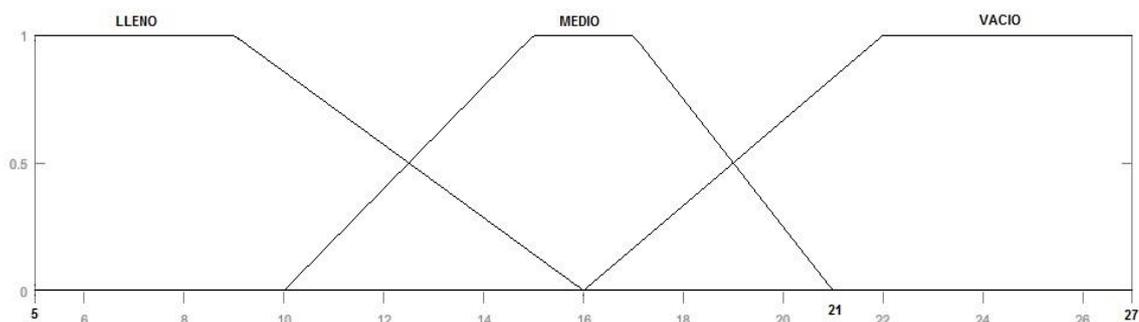
$$\mu_{POCO}(x) = \begin{cases} \frac{x-17}{6}, & 17 \leq x \leq 23 \\ 1, & 23 < x \leq 30 \end{cases} \quad \text{Ec. (3.4)}$$

Las Ecuaciones 3.2, 3.3 y 3.4 representan las funciones de membresía de cada uno de los conjuntos de la variable de entrada *nivel* para este primer diseño.

En la Figura 3.15, se observa que el nivel de la gelatina tiene un universo de discurso [5-27] cm para este diseño realizado conociendo la ubicación exacta del sensor en la máquina. Se cambió el conjunto *MEDIO* de la Figura 3.14 de triangular a trapezoidal para dar un grado de pertenencia total superior al diseño inicial, es decir, antes se tenía que el grado de pertenencia “1” se daba exclusivamente para un valor de 17, ahora se dará para valores entre 15 y 17.

Se cambiaron también los nombres de los conjuntos *FULL* a *LLENO* y *POCO* por *VACÍO*. A demás se modificó el rango de *LLENO* y *VACIO* para cumplir con la regla del solapamiento de los conjuntos.

Figura 3.15. Diseño final de los conjuntos difusos para el *nivel*.



Conjuntos difusos de la entrada *nivel*. Elaborado por: Steven Gordillo.

$$\mu_{LLENO}(x) = \begin{cases} 1, & 5 \leq x \leq 9 \\ \frac{16-x}{7}, & 9 < x \leq 16 \end{cases} \quad \text{Ec. (3.5)}$$

$$\mu_{MEDIO}(x) = \begin{cases} \frac{x-10}{5}, & 10 \leq x < 15 \\ 1, & 15 \leq x \leq 17 \\ \frac{21-x}{4}, & 17 < x \leq 21 \end{cases} \quad \text{Ec. (3.6)}$$

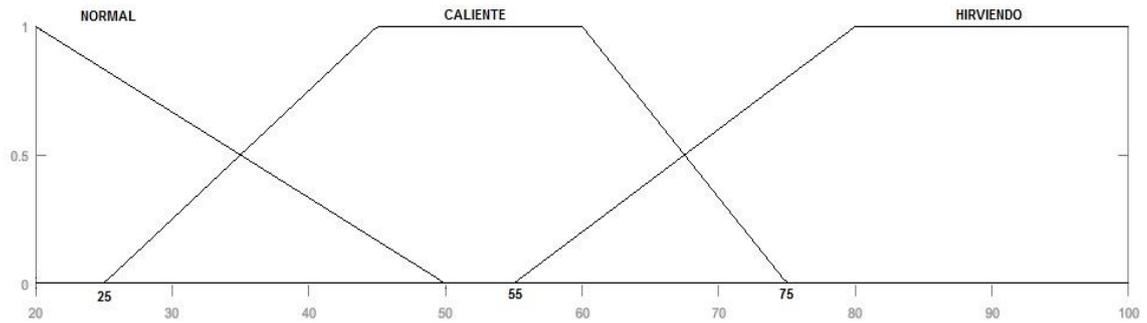
$$\mu_{VACIO}(x) = \begin{cases} \frac{x-16}{6}, & 16 \leq x < 22 \\ 1, & 22 \leq x \leq 27 \end{cases} \quad \text{Ec. (3.7)}$$

Las Ecuaciones 3.5, 3.6 y 3.7 representan las funciones de membresía de cada uno de los conjuntos de la variable de entrada *nivel* para el diseño final.

- **Conjuntos difusos para la entrada “*Temperatura*”**

En la Figura 3.16, se muestra que la temperatura de la gelatina tenía un universo de discurso [20-100] °C para este primer diseño, poniendo a la temperatura ambiente como la más baja, a pesar de que a esa temperatura la gelatina prácticamente ya estuviera endurecida, pero se la incluyó como un valor teórico. La temperatura más alta está referenciada teóricamente al punto de ebullición de una sustancia que tiene un valor entre 90 y 100 °C.

Figura 3.16. Primer diseño de los conjuntos difusos para la *temperatura*.



Conjuntos difusos de la entrada *temperatura*. Elaborado por: Steven Gordillo.

$$\mu_{NORMAL}(x) = \begin{cases} \frac{50 - x}{30}, & 20 \leq x \leq 50 \end{cases} \quad \text{Ec. (3.8)}$$

$$\mu_{CALIENTE}(x) = \begin{cases} \frac{x - 25}{20}, & 25 \leq x < 45 \\ 1, & 45 \leq x \leq 60 \\ \frac{75 - x}{15}, & 60 < x \leq 75 \end{cases} \quad \text{Ec. (3.9)}$$

$$\mu_{HIRVIENDO}(x) = \begin{cases} \frac{x - 55}{25}, & 55 \leq x < 80 \\ 1, & 80 \leq x \leq 100 \end{cases} \quad \text{Ec. (3.10)}$$

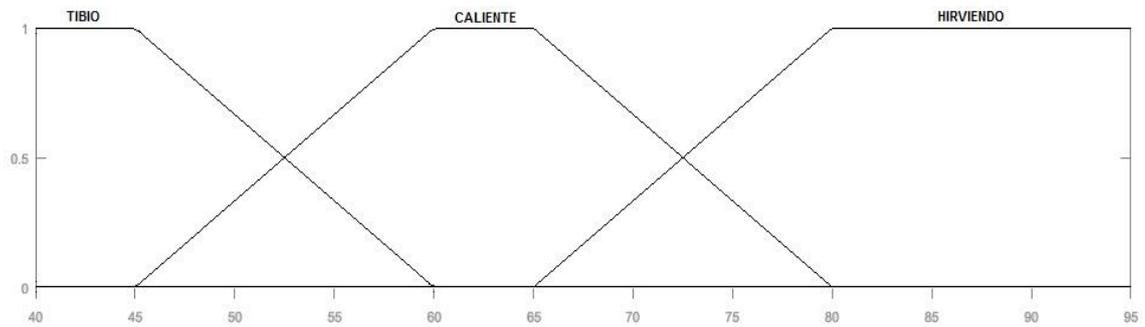
Las Ecuaciones 3.8, 3.9 y 3.10 representan las funciones de membresía de cada uno de los conjuntos de la variable de entrada *temperatura* para este primer diseño.

En la Figura 3.17, se muestra que la temperatura de la gelatina tiene un universo de discurso [40-95] °C para este diseño, ya que se tomó los 40°C como la temperatura más baja a la que el compuesto aún no se ha endurecido y se puede batir, mientras que el límite superior lleva un valor de 95°C que es la temperatura máxima a la que llega a estar sometida la gelatina.

Se cambió el nombre del conjunto *NORMAL* de la Figura 3.16 por *TIBIO*, también se cambió el tipo de conjunto de triangular a trapezoidal para dar una pertenencia de “1” a valores de temperatura entre los 40 y 45°C. Por otro lado, el conjunto *NORMAL* tenía

valores teóricos que fueron descartados en este diseño, como por ejemplo el valor de la temperatura ambiente para la gelatina.

Figura 3.17. Diseño final de los conjuntos difusos para la *temperatura*.



Conjuntos difusos de la entrada *temperatura*. Elaborado por: Steven Gordillo.

$$\mu_{TIBIO}(x) = \begin{cases} 1, & 40 \leq x \leq 45 \\ \frac{60 - x}{15}, & 45 < x \leq 60 \end{cases} \quad \text{Ec. (3.11)}$$

$$\mu_{CALIENTE}(x) = \begin{cases} \frac{x - 45}{15}, & 45 \leq x < 60 \\ 1, & 60 \leq x \leq 65 \\ \frac{80 - x}{15}, & 65 < x \leq 80 \end{cases} \quad \text{Ec. (3.12)}$$

$$\mu_{HIRVIENDO}(x) = \begin{cases} \frac{x - 65}{15}, & 65 \leq x < 80 \\ 1, & 80 \leq x \leq 95 \end{cases} \quad \text{Ec. (3.13)}$$

Las Ecuaciones 3.11, 3.12 y 3.13 representan las funciones de membresía de cada uno de los conjuntos de la variable de entrada *temperatura*.

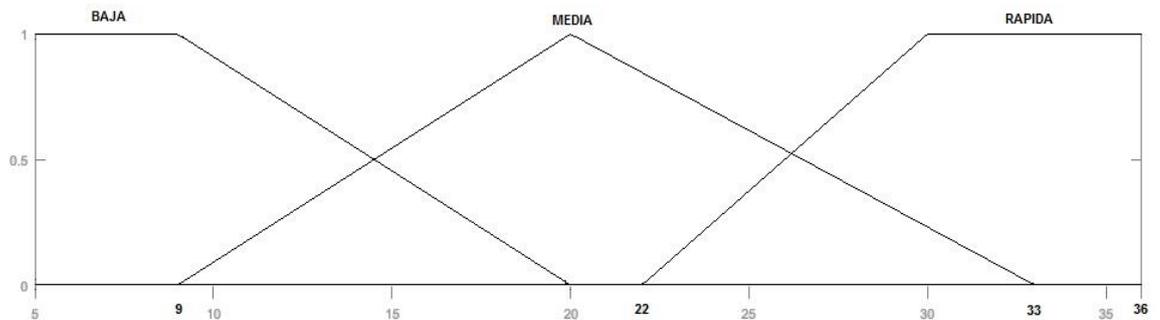
- **Conjuntos difusos para la salida “Frecuencia”**

Inicialmente se realizó el diseño de los conjuntos difusos para la salida del sistema en función de la velocidad en porcentaje, es decir, se tendría una variación de 0 al 100%. Hubo muchos inconvenientes con este diseño en la implementación del controlador difuso y el variador de frecuencia, así que se lo descartó inmediatamente.

Entonces se tomó como variable de salida del sistema a la frecuencia, por ser una magnitud directamente proporcional a la velocidad. Es decir, si se aumenta la frecuencia, aumentará la velocidad del motor y si se la disminuye de igual manera se reducirá la velocidad del eje del motor.

Como se puede observar en la Figura 3.18, para este primer diseño, la frecuencia tenía un universo de discurso [5-36] Hz, donde se estableció un valor mínimo de frecuencia de 5 Hz para un arranque suave del motor y un valor máximo de 36 Hz para las condiciones que necesiten una mayor velocidad, por tanto, solo se ocupó un 60% de toda la velocidad del motor.

Figura 3.18. Primer diseño de los conjuntos difusos para la *frecuencia*.



Conjuntos difusos de la salida *frecuencia*. Elaborado por: Steven Gordillo.

$$\mu_{BAJA}(x) = \begin{cases} 1, & 5 \leq x \leq 9 \\ \frac{20-x}{11}, & 9 < x \leq 20 \end{cases} \quad \text{Ec. (3.14)}$$

$$\mu_{MEDIA}(x) = \begin{cases} \frac{x-9}{11}, & 9 < x \leq 20 \\ \frac{33-x}{13}, & 20 < x \leq 33 \end{cases} \quad \text{Ec. (3.15)}$$

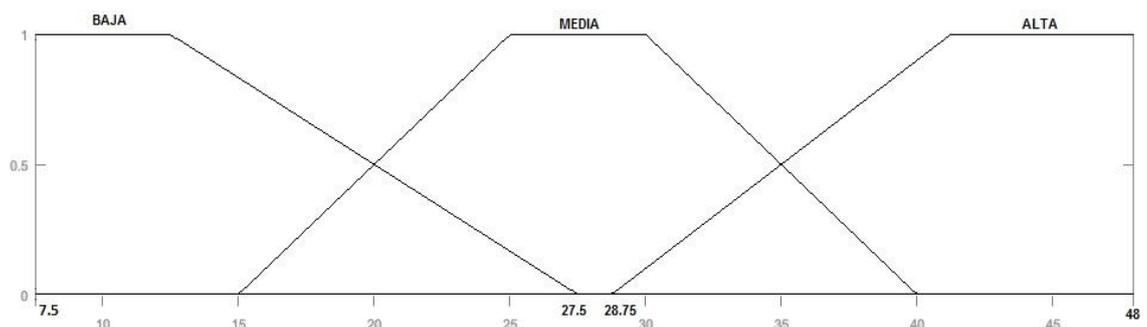
$$\mu_{RAPIDA}(x) = \begin{cases} \frac{x-22}{8}, & 22 \leq x < 30 \\ 1, & 30 \leq x \leq 36 \end{cases} \quad \text{Ec. (3.16)}$$

Las Ecuaciones 3.14, 3.15 y 3.16 representan las funciones de membresía de cada uno de los conjuntos de la variable de salida *frecuencia* para este primer diseño.

Para el diseño final, mostrado en la Figura 3.19, la frecuencia tiene un universo de discurso [7.5-48] Hz. Se tomó como la frecuencia más baja el valor de 7.5 Hz que evita un arranque brusco del motor sin regar la gelatina del recipiente, la frecuencia más elevada fue de 48 Hz y proporcionó una velocidad adecuada para mantener estable el compuesto y espesarlo, es decir, se ocupó un 80% de la velocidad total del motor.

Se ajustaron los rangos de los conjuntos difusos de la Figura 3.18 de [5-36] Hz a valores entre [7.5-48] Hz. debido a la expansión del universo de discurso mostrado en la Figura 3.19, manteniendo el solapamiento del 50% entre conjuntos y se cambió el conjunto *MEDIA* de triangular a trapezoidal para ampliar el rango de valores que puedan tener un grado de pertenencia de “1”.

Figura 3.19. Diseño final de los conjuntos difusos para la *frecuencia*.



Conjuntos difusos de la salida *frecuencia*. Elaborado por: Steven Gordillo.

$$\mu_{BAJA}(x) = \begin{cases} 1, & 7.5 \leq x \leq 12.5 \\ \frac{27.5 - x}{15}, & 12.5 < x \leq 27.5 \end{cases} \quad \text{Ec. (3.17)}$$

$$\mu_{MEDIA}(x) = \begin{cases} \frac{x - 15}{10}, & 15 \leq x < 25 \\ 1, & 25 \leq x \leq 30 \\ \frac{40 - x}{10}, & 30 < x \leq 40 \end{cases} \quad \text{Ec. (3.18)}$$

$$\mu_{ALTA}(x) = \begin{cases} \frac{x - 28.75}{12.5}, & 28.75 \leq x < 41.25 \\ 1, & 41.25 \leq x \leq 48 \end{cases} \quad \text{Ec. (3.19)}$$

Las Ecuaciones 3.17, 3.18 y 3.19 representan las funciones de membresía de cada uno de los conjuntos de la variable de salida *frecuencia*.

3.8.2. Diseño de las reglas lógicas

Las reglas lógicas determinaron que valor de frecuencia entregar bajo ciertas condiciones de temperatura y nivel de gelatina, para ello se basó en el conocimiento empírico de los trabajadores de la microempresa. Tras muchas pruebas de ensayo, errores y correcciones se plantearon las reglas a continuación mostradas.

Tabla 3.2. Reglas lógicas para el control difuso.

TEMP \ NIVEL	VACIO	MEDIO	LLENO
TIBIO	ALTA	ALTA	MEDIA
CALIENTE	ALTA	MEDIA	BAJA
HIRVIENDO	MEDIA	ALTA	BAJA

La tabla muestra la frecuencia que se debe entregar al motor bajo determinadas combinaciones de temperatura y nivel de gelatina. Elaborado por: Steven Gordillo.

La Tabla 3.2 muestra en la parte superior los conjuntos de la variable *Nivel*, mientras que de su lado izquierdo se tiene los conjuntos de la variable *Temperatura*. Cada cuadro formado por las combinaciones de estas dos variables indica que conjunto de frecuencia se activará en el controlador difuso. Por ejemplo, si se tiene una temperatura de la gelatina de 60 °C, una lectura del sensor de nivel de 17 cm y se toma en cuenta los diseños de los conjuntos finales para cada variable, se observa que:

El valor de temperatura es parte del conjunto *CALIENTE*, mientras que el valor del nivel forma parte del conjunto *MEDIO*. Con estos datos, $Nivel = MEDIO$ y $TEMPERATURA = CALIENTE$, se activará el conjunto de la frecuencia *MEDIA* que después del respectivo cálculo tomará valores entre 25 y 30 Hz. Para determinar el valor que será entregado por el controlador difuso es necesario realizar la fusificación, inferencia por mínimos y la defusificación, para esto se utilizó el “Fuzzy Logic Toolbox” de Matlab que entregó el valor exacto, y evitó todos los cálculos y gráficos que se deberían analizar en estos procesos.

3.9. Simulación del control difuso con “Fuzzy Logic Toolbox”.

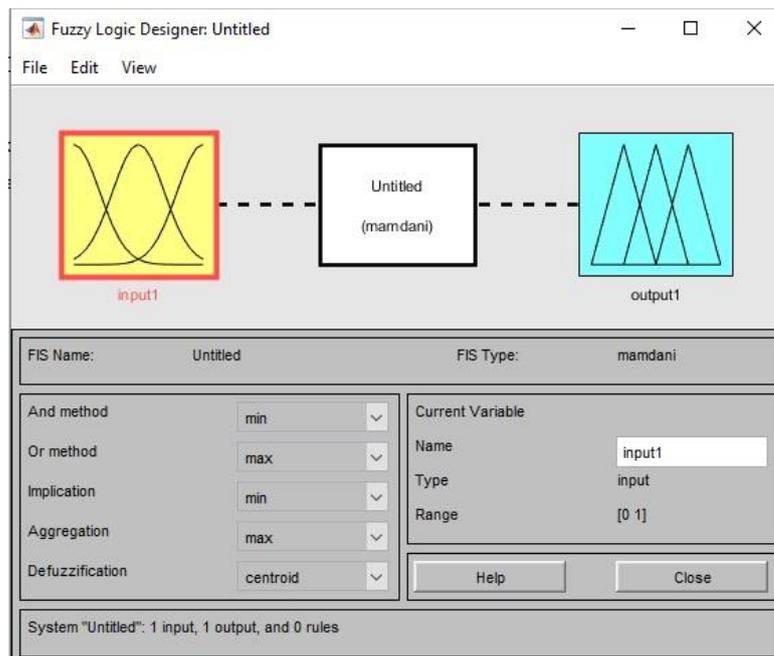
Con la ayuda de “MATLAB” se simuló todo el proceso del control difuso para entregar el valor de la salida que luego fue interpretado en función de “PWM” en el microcontrolador “Arduino Nano” a través del sketch de programación del mismo.

Con esta herramienta se evitó muchísimos cálculos, análisis de gráficas y métodos matemáticos que fueron necesarios para la fusificación, la inferencia por el mínimo y la defusificación. Realizar los cálculos de forma manual es extenso y todo el análisis sirve únicamente para los valores que se toman para una prueba en concreto, y para otras condiciones se debería volver a repetir todo el proceso. Eso no sucede con esta herramienta, ya que entrega el valor de la salida difusa para cualquier condición que se necesite evaluar.

3.9.1. Creación del controlador difuso con “Fuzzy Logic Toolbox”.

Para utilizar esta herramienta se realizó lo siguientes pasos: Primero se ingresó a “Matlab”. Segundo, se escribió la palabra “Fuzzy” en la ventana de comandos. Como tercer y último paso se presionó “Enter”. Al completar los pasos se mostró una ventana que se representa en la Figura 3.20.

Figura 3.20. Ventana de diseño de lógica difusa.

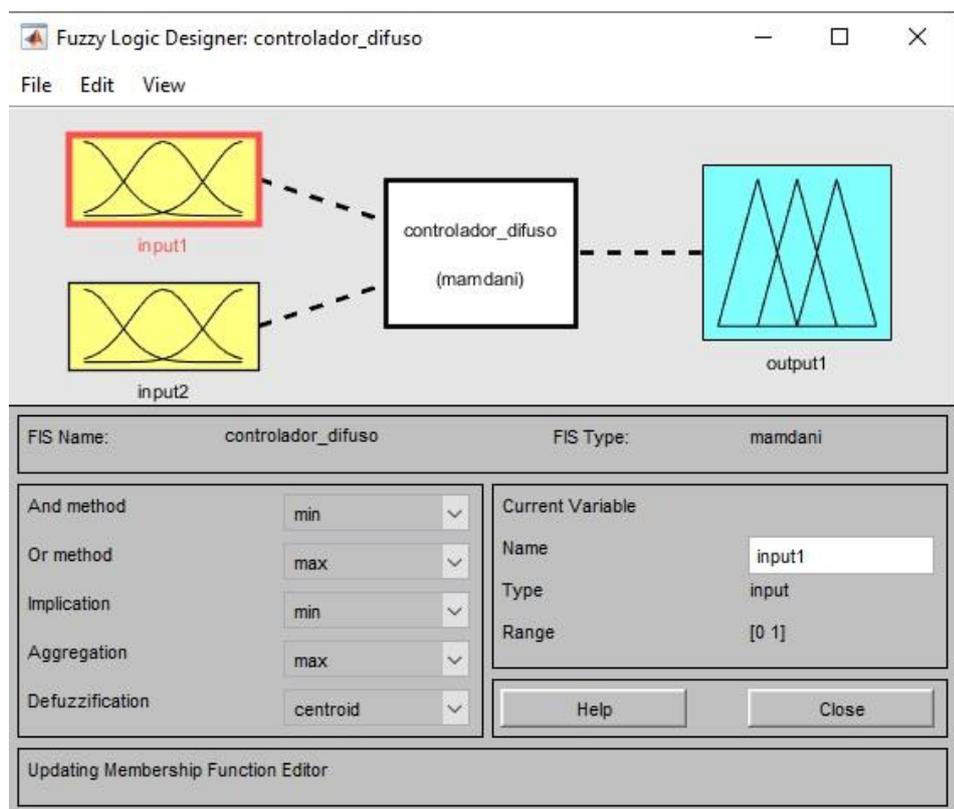


Entorno gráfico de la herramienta de diseño de lógica difusa. Elaborado por: Steven Gordillo.

En la ventana de diseño de lógica difusa se puede observar en la parte superior la estructura del controlador, con sus entradas y salidas, a la izquierda está el método de inferencia para la toma de decisiones, a la derecha se tiene la descripción de los conjuntos de cada variable y en la parte inferior se observan las reglas lógicas.

Se procedió a guardar el diseño con el nombre de *controlador_difuso*, presionando las teclas “Control” + “S”, el nombre se cambió automáticamente. A continuación, se seleccionó la opción “Edit”, situada en la barra de herramientas, se escogió “Add Variable”, para concluir se dio clic en “Input”, lo que agregó otra variable de entrada, tal como se muestra en la Figura 3.21. Realizar este procedimiento fue necesario debido a que la interfaz de lógica difusa por defecto pone una variable de entrada y otra de salida mientras que este diseño requiere de dos entradas y una salida.

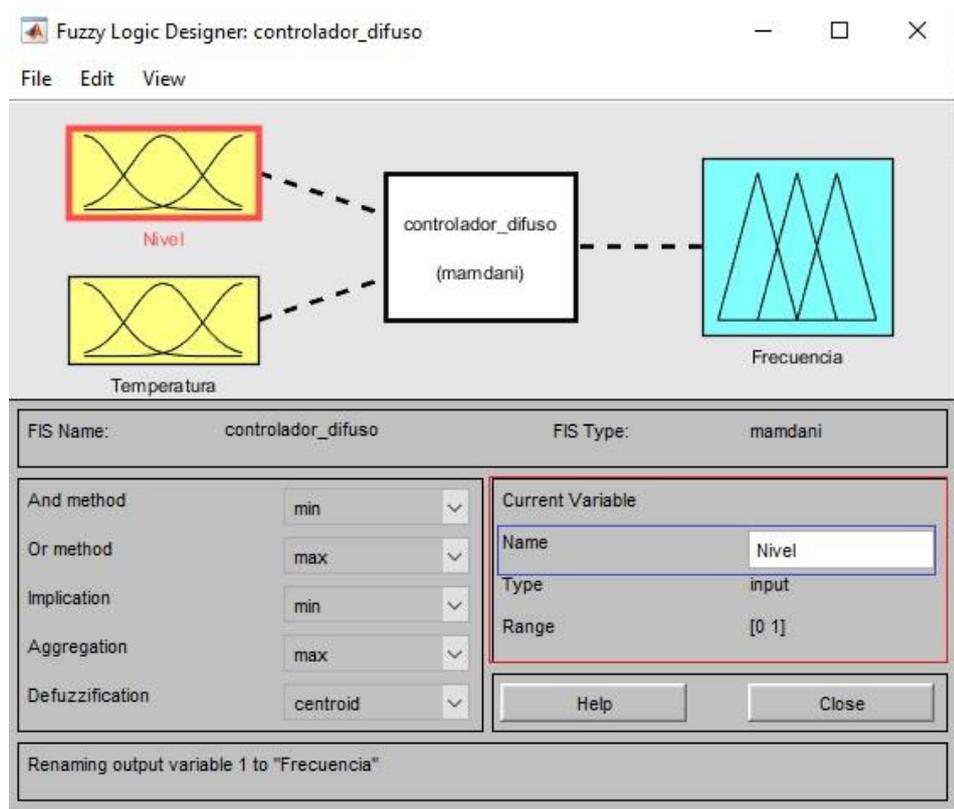
Figura 3.21. Incorporación de una segunda variable de entrada.



Diseño del controlador difuso con dos variables de entrada y una variable de salida. Elaborado por: Steven Gordillo.

En la parte derecha de la Figura 3.22 se observa un recuadro llamado “Current Variable” donde se procedió a cambiar el nombre de cada una de las variables, luego se dio doble clic en una de ellas para ingresar al editor de funciones de membresía que permitió crear y modificar las propiedades de los conjuntos difusos de cada variable.

Figura 3.22. Renombrar variables del controlador difuso.

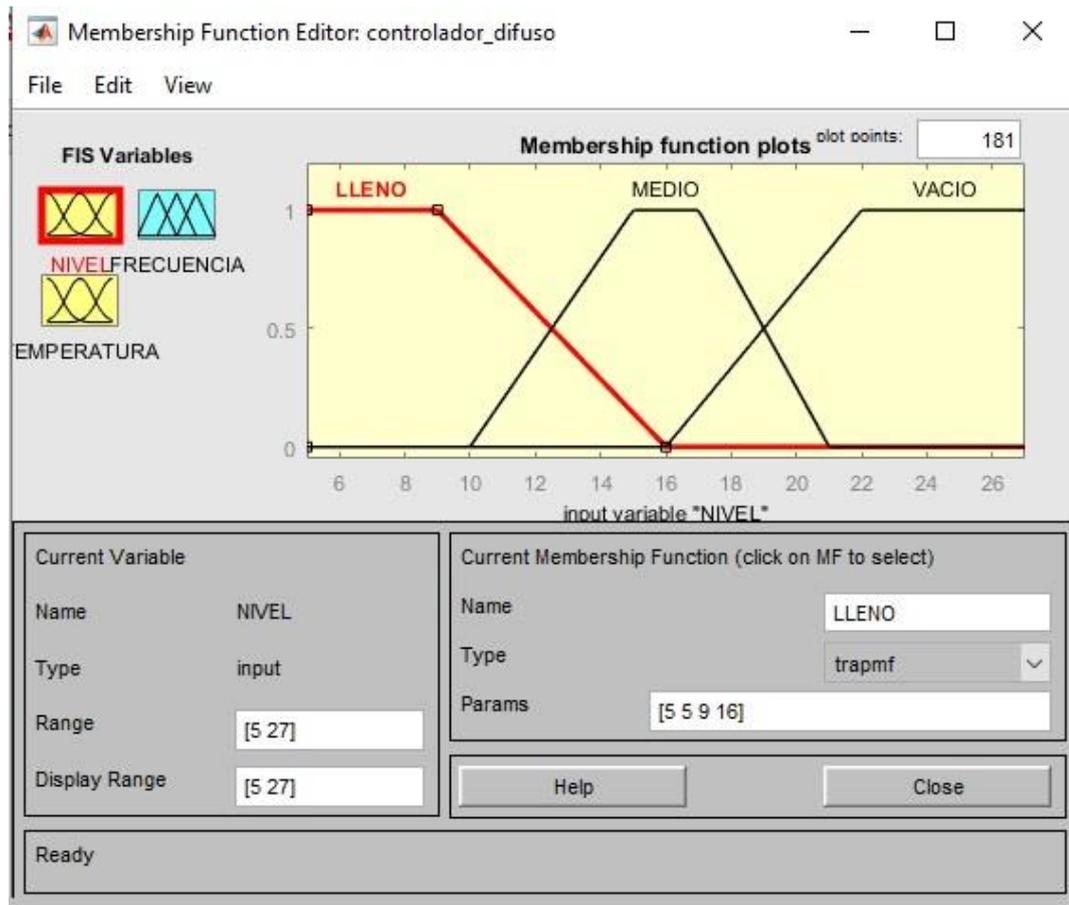


Cambio de nombre de las variables de entrada y salida. Elaborado por: Steven Gordillo.

En la ventana del editor de funciones de membresía en la Figura 3.23 se empezó a diseñar los conjuntos difusos. La parte superior izquierda del editor muestra las variables de entrada y salida del controlador, en la parte superior derecha se observan gráficamente los conjuntos difusos correspondientes a la variable seleccionada, el sector inferior izquierdo indica el nombre, tipo y rango de la variable.

Por último, en la parte inferior derecha se tiene la información de los conjuntos difusos, como por ejemplo el nombre del conjunto, el tipo de función de membresía y sus dimensiones.

Figura 3.23. Editor de funciones de membresía.

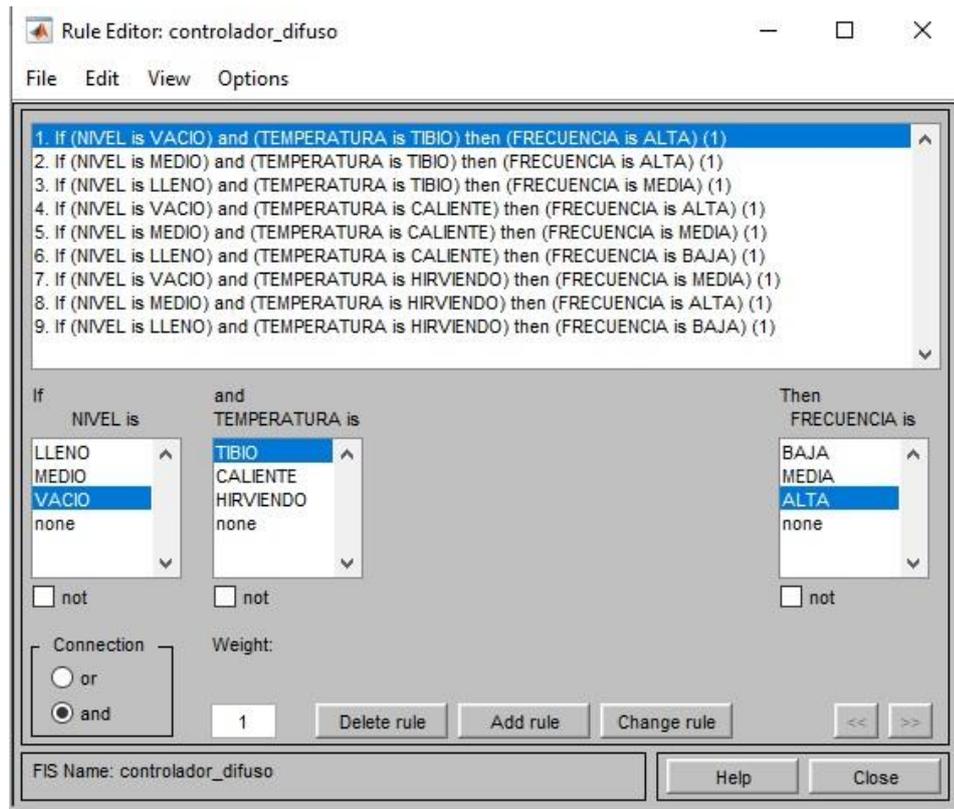


Conjuntos difusos de la variable de entrada *Nivel*, Elaborado por: Steven Gordillo.

Después de haber creado los conjuntos difusos para cada una de las variables del sistema como se observa en la Figura 3.23, se escogió la opción “Edit” de la barra de herramientas y se dio clic en “Rules”. Se desplegó una ventana denominada “Editor de reglas lógicas” que se muestra en la Figura 3.24. Esta ventana permitió crear las condiciones del sistema. Para ello fue necesario ubicarse en la parte inferior de la ventana y seleccionar el conjunto de cada variable que interviene en la regla y su respectivo método de inferencia (“OR” o “AND”).

Por último, se dio clic en el botón “Add rule” para agregar la regla al diseño. Las reglas agregadas se muestran en la parte superior de la ventana, en caso de querer eliminar una regla duplicada o errónea, se tiene la opción “Delete rule” y para modificarla está la opción “Change rule”.

Figura 3.24. Editor de reglas lógicas.

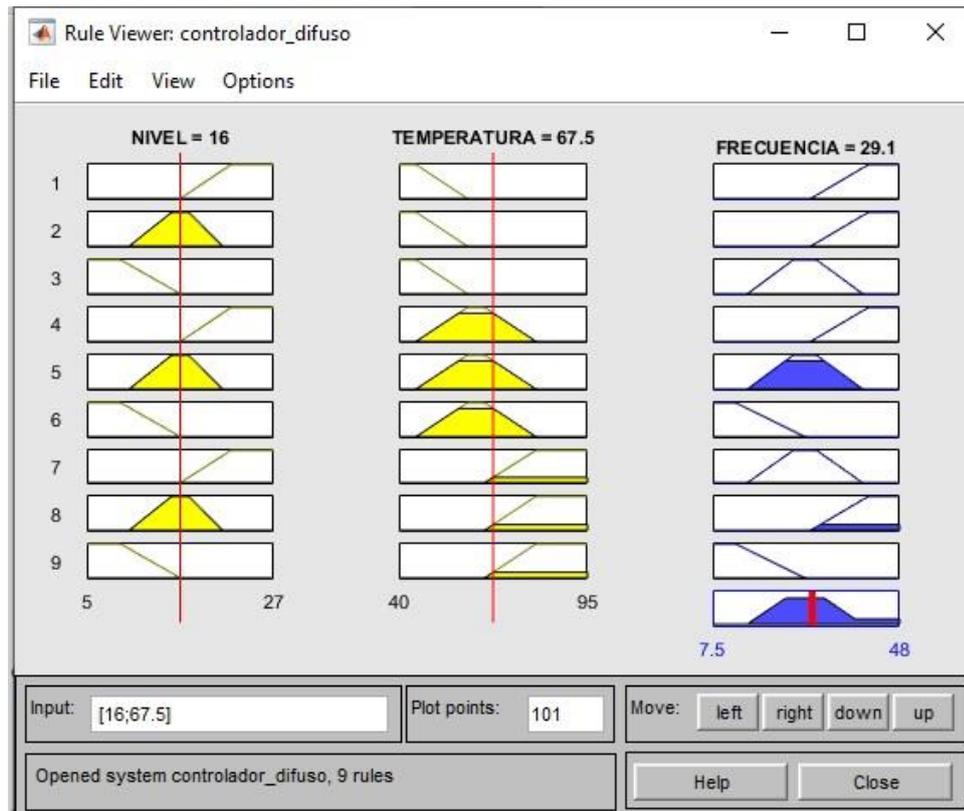


Reglas lógicas del controlador difuso. Fuente: Matlab 2017. Elaborado por: Steven Gordillo.

Habiendo creado los conjuntos difusos y las reglas lógicas se verificó el funcionamiento del controlador. Para ello, se seleccionó la opción “View” de la barra de herramientas que desplegó una ventana denominada “Visualizador de reglas lógicas”.

Esta ventana se muestra en la Figura 3.25 y simula el controlador difuso permitiendo modificar el valor de las entradas posicionándose en la línea roja que esta sobre los conjuntos y arrastrando el mouse hacia la izquierda o hacia la derecha para disminuir o incrementar el valor de la entrada respectivamente. También es posible modificar los valores de las entradas ubicándose en la parte inferior izquierda de la ventana, cambiando los números que se encuentran dentro de los corchetes de la opción “input”. El valor de la salida difusa se encuentra situado en la parte superior derecha de la ventana.

Figura 3.25. Visualizador de reglas lógicas.



Simulación del controlador difuso, que me permite modificar los valores de entrada y observar la respuesta de la salida. Fuente: Matlab 2017. Elaborado por: Steven Gordillo.

3.10. Implementación del control difuso en el sketch de “Arduino”.

El sketch es una plataforma de programación para el microcontrolador incorporado en “Arduino” que facilita la implementación de módulos externos compatibles con las tarjetas de control. Las librerías creadas por la comunidad permiten reutilizar el código en diferentes aplicaciones según las necesidades del usuario.

Las librerías ocupadas en este proyecto son:

- **LiquidCrystal_I2C:** Permite utilizar la pantalla “LCD” mediante el módulo “I2C”.
- **Fuzzy:** Permite implementar la lógica difusa en el microcontrolador.
- **Max6675:** Facilita el uso de la termocupla tipo k mediante el módulo amplificador y acondicionador “Max6675”.

La librería “Fuzzy” es la más importante, ya que en ella se programó la lógica del controlador que permitió realizar la toma de decisiones acertadas sobre las condiciones de las variables de entrada. Para crear los conjuntos difusos se utilizó el código que se muestra en la Figura 3.26.

Figura 3.26. Creación de los conjuntos difusos en “Arduino”.

```

//Creo un nuevo Objeto Fuzzy
Fuzzy*CONTROL = new Fuzzy();

////////////////////////////////////
////////// IN 1 //////////
////////////////////////////////////

1 FuzzyInput* NIVEL = new FuzzyInput(1);
2 FuzzySet* LLENO = new FuzzySet(5, 5, 9, 16);
3 NIVEL->addFuzzySet(LLENO);

4 FuzzySet* MEDIO = new FuzzySet(10, 15, 17, 21);
5 NIVEL->addFuzzySet(MEDIO);

6 FuzzySet* VACIO = new FuzzySet(16, 22, 27, 27);
7 NIVEL->addFuzzySet(VACIO);

8 CONTROL->addFuzzyInput(NIVEL);

////////////////////////////////////
////////// IN 2 //////////
////////////////////////////////////

1 FuzzyInput* TEMPERATURA = new FuzzyInput(2);
2 FuzzySet* TIBIO = new FuzzySet(40, 40, 45, 60);
3 TEMPERATURA->addFuzzySet(TIBIO);

4 FuzzySet* CALIENTE = new FuzzySet(45, 60, 65, 80);
5 TEMPERATURA->addFuzzySet(CALIENTE);

6 FuzzySet* HIRVIENDO = new FuzzySet(65, 80, 95, 95);
7 TEMPERATURA->addFuzzySet(HIRVIENDO);

8 CONTROL->addFuzzyInput(TEMPERATURA);

////////////////////////////////////
////////// OUT 1 //////////
////////////////////////////////////

1 FuzzyOutput* FRECUENCIA = new FuzzyOutput(1);
2 FuzzySet* BAJA = new FuzzySet(7.5, 7.5, 12.5, 27.5);
3 FRECUENCIA->addFuzzySet(BAJA);

4 FuzzySet* MEDIA = new FuzzySet(15, 25, 30, 40);
5 FRECUENCIA->addFuzzySet(MEDIA);

6 FuzzySet* ALTA = new FuzzySet(28.75, 41.25, 48, 48);
7 FRECUENCIA->addFuzzySet(ALTA);

8 CONTROL->addFuzzyOutput(FRECUENCIA);

```

Captura de pantalla del código de programación de los conjuntos difusos para cada variable del controlador con sus respectivos parámetros. Fuente: sketch de “Arduino” 1.6.5. Elaborado por: Steven Gordillo.

Los conjuntos de cada una de las variables tienen la misma estructura, así que bastará con explicar uno solo de ellos para entender la programación de los demás.

Como se pueda observar en la Figura 3.26 se empezó creando un objeto de tipo “Fuzzy” denominado *CONTROL*; La primera línea de código creó una variable llamada *NIVEL* de tipo entrada difusa 1, a continuación, se creó el conjunto *LLENO* con los parámetros correspondientes al tipo de función de membresía que representa,

la tercera línea agregó este conjunto a la variable *NIVEL*, de la misma manera la cuarta línea creó el conjunto *MEDIO* y la quinta línea lo agregó. La sexta línea creó el conjunto *VACIO* y la séptima lo agregó, la última línea agregó la variable *NIVEL* con los conjuntos *LLENO*, *MEDIO* y *VACIO* al objeto tipo “Fuzzy” llamado *CONTROL*.

Ahora, es necesario crear las reglas lógicas. Para ello se implementó el código mostrado en la Figura 3.27. La programación para cada una de las reglas tiene la misma estructura, por lo tanto, se explicará cómo crear una regla para entender el resto.

Figura 3.27. Creación de las reglas lógicas en “Arduino”.

```

////////////////////////////////////
////////////////// REGLAS //////////////////
////////////////////////////////////

//1////////////////////////////////////
// SI (IN1=VACIO) and (IN2=TIBIO) then (OUT=ALTA)////////
////////////////////////////////////
1 FuzzyRuleAntecedent* Si_NIVEL_es_VACIO_Y_TEMPERATURA_es_TIBIO = new FuzzyRuleAntecedent();
2 Si_NIVEL_es_VACIO_Y_TEMPERATURA_es_TIBIO->joinWithAND (VACIO, TIBIO);
3 FuzzyRuleConsequent* Entonces_FRECUENCIA_es_ALTA = new FuzzyRuleConsequent();
4 Entonces_FRECUENCIA_es_ALTA->addOutput (ALTA);

5 FuzzyRule* REGLA1 = new FuzzyRule(1, Si_NIVEL_es_VACIO_Y_TEMPERATURA_es_TIBIO, Entonces_FRECUENCIA_es_ALTA);
6 CONTROL->addFuzzyRule (REGLA1);

//2////////////////////////////////////
//// SI (IN1=MEDIO) and (IN2=TIBIO) then (OUT=ALTA)/////
////////////////////////////////////
1 FuzzyRuleAntecedent* Si_NIVEL_es_MEDIO_Y_TEMPERATURA_es_TIBIO = new FuzzyRuleAntecedent();
2 Si_NIVEL_es_MEDIO_Y_TEMPERATURA_es_TIBIO->joinWithAND (MEDIO, TIBIO);

5 FuzzyRule* REGLA2 = new FuzzyRule(2, Si_NIVEL_es_MEDIO_Y_TEMPERATURA_es_TIBIO, Entonces_FRECUENCIA_es_ALTA);
6 CONTROL->addFuzzyRule (REGLA2);

//3////////////////////////////////////
//// SI (IN1=LLENO) and (IN2=TIBIO) then (OUT=MEDIA)/////
////////////////////////////////////
1 FuzzyRuleAntecedent* Si_NIVEL_es_LLENO_Y_TEMPERATURA_es_TIBIO = new FuzzyRuleAntecedent();
2 Si_NIVEL_es_LLENO_Y_TEMPERATURA_es_TIBIO->joinWithAND (LLENO, TIBIO);
3 FuzzyRuleConsequent* Entonces_FRECUENCIA_es_MEDIA = new FuzzyRuleConsequent();
4 Entonces_FRECUENCIA_es_MEDIA->addOutput (MEDIA);

5 FuzzyRule* REGLA3 = new FuzzyRule(3, Si_NIVEL_es_LLENO_Y_TEMPERATURA_es_TIBIO, Entonces_FRECUENCIA_es_MEDIA);
6 CONTROL->addFuzzyRule (REGLA3);

```

Captura de pantalla del código de programación de las reglas lógicas. Fuente: sketch de “Arduino” 1.6.5.
Elaborado por: Steven Gordillo.

En la Figura 3.27, la primera línea creó una variable de tipo antecedente con un nombre que describe las condiciones de las variables de entrada, a continuación, se le asignó los conjuntos difusos *VACIO* y *TIBIO* que intervienen en esta primera regla mostrada en la Figura 3.27. La tercera línea de código creó una variable de tipo consecuencia

con un nombre que indica la condición que tendrá la variable de salida, la cuarta línea le asignó el conjunto *ALTA*.

La quinta línea creó un objeto llamado *REGLA1* de tipo “Fuzzy rule” y como argumentos entre los paréntesis se colocó el número 1 por ser la primera regla, y también se puso las variables antecedente y consecuente. Por último, en la sexta línea se agregó *REGLA1* al objeto *CONTROL* de tipo “Fuzzy”.

En la segunda regla, se omitió la tercera y cuarta línea debido a que la variable de tipo consecuencia que involucra al conjunto *ALTA* ya fue creada anteriormente. Se siguió el mismo procedimiento para crear las ocho reglas restantes.

Figura 3.28. Fusificación y defusificación en “Arduino”.

```
CONTROL->setInput(1, distancia); //LOGICA DIFUSA IN1
CONTROL->setInput(2, temp); //LOGICA DIFUSA IN2

//FUSIFICACION
CONTROL->fuzzify();

//DEFUSIFICACION
output = CONTROL->defuzzify(1);
```

Captura de pantalla del código de programación para la fusificación y defusificación. Fuente: sketch de “Arduino” 1.6.5. Elaborado por: Steven Gordillo.

Como se puede ver en la Figura 3.28, para terminar la implementación de la lógica difusa en “Arduino”, la variable *distancia* que proporciona una lectura en tiempo real del sensor de nivel fue asignada a la variable *NIVEL* que ocupa la posición 1 del objeto *CONTROL*. Posteriormente, se replicó el procedimiento anterior, pero esta vez se asignó la variable de lectura de la termocupla a la variable *TEMPERATURA* que ocupa la posición 2 del objeto *CONTROL*.

En la Figura 3.28 se observa que se llamó a la función “fuzzify()” que es parte de la librería “Fuzzy” para realizar el proceso de fusificación. La última línea de código asignó el valor entregado por la función “defuzzify()” a la variable global de tipo

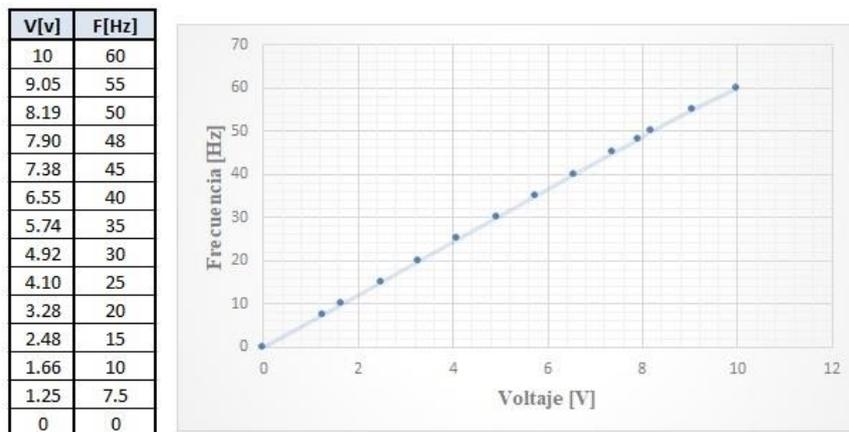
flotante llamada *output* que debió ser creada con anterioridad en el “void setup” del sketch de “Arduino”.

3.10.1. Implementación de la salida difusa en el variador de frecuencia.

El valor almacenado en la variable *output* de tipo flotante mostrado en la Figura 3.28 obtenido del controlador difuso, fue convertido en valor “PWM” y se utilizó una de las salidas digitales del “Arduino” para proveer un voltaje que varía entre 0 y 5 v. El “PWM” acepta valores que van desde 0 hasta 255, entregando un voltaje de 0v en el pin de salida digital para un valor de 0, mientras que para un valor de 255 entregará el máximo voltaje que puede suministrar el microcontrolador a sus pines, es decir 5v.

El Inversor tiene un terminal de entrada para control de frecuencia mediante variación de voltaje, este maneja un rango entre 0 y 10 Vdc. Es primordial conocer la frecuencia entregada por el variador bajo cierto valor de voltaje colocado en el pin “V1” mostrado en el diagrama de la Figura 2.19, para ello se tomó mediciones de ambos parámetros, usando un potenciómetro de 20 KΩ y un multímetro.

Figura 3.29. Relación entre los parámetros voltaje y frecuencia del inversor.



Grafica de la relación entre el voltaje y la frecuencia del inversor. Fuente: Excel 2019. Elaborado por: Steven Gordillo.

Como se puede observar en la gráfica de la Figura 3.29, la frecuencia de salida del inversor y el voltaje de entrada en “V1” tienen una relación lineal. También, es necesario conocer el equivalente de valor “PWM” como voltaje de salida de la placa

ya que la modulación por ancho de pulso es quien determina la tensión en los puertos de salida digital del “Arduino”, para ello se utilizó la siguiente formula:

$$Voltaje = \frac{Valor "PWM"}{51} \quad Ec. (3.20)$$

La Ec. 3.20 se obtuvo realizando una simple regla de tres, de modo que 255 corresponde a 5v, por lo tanto, 1v equivaldrá a 51. En la Tabla 3.3 se muestran los valores de voltaje de entrada al terminal “V1” del inversor en función de “PWM”.

La modulación por ancho de pulso del “Arduino” tiene como valor máximo al 255 correspondiente a 5v, por lo tanto, es posible conectar directamente la salida del controlador difuso con la entrada “V1” del inversor, obteniendo una frecuencia máxima de 30 Hz. Sin embargo, el diseño de los conjuntos de la variable de salida mostrados en la Figura 3.19 tiene un rango que va desde los 7.5 Hz hasta los 48 Hz.

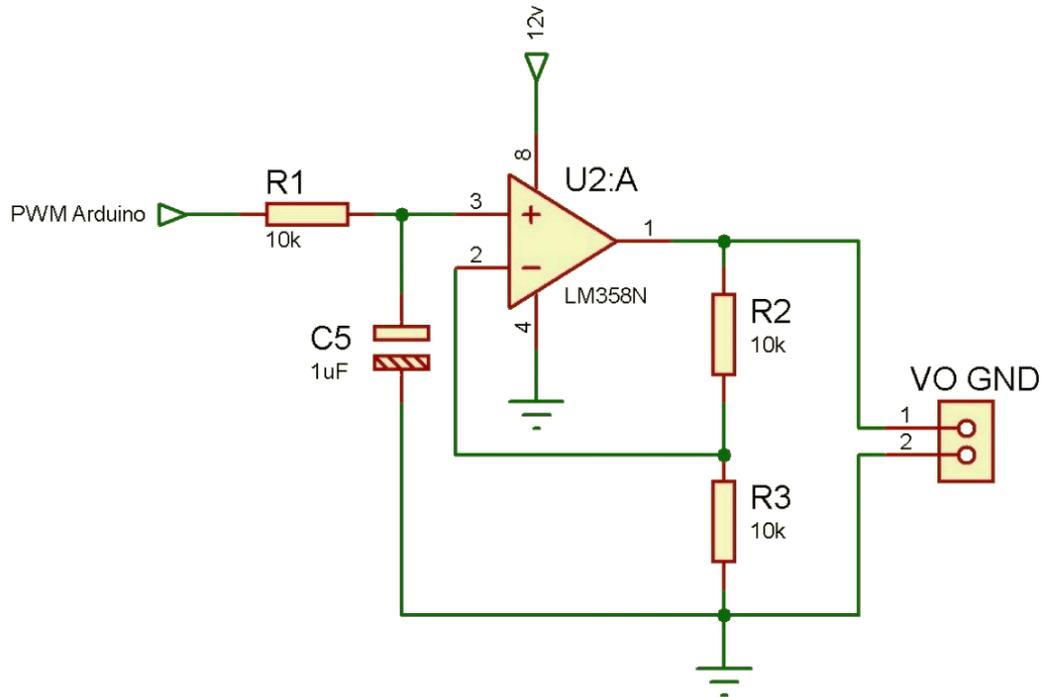
Tabla 3.3. Valores de voltaje de entrada al variador en función de “PWM”.

F[Hz]	V[v]	PWM
60	10	
55	9.05	
50	8.19	
48	7.90	
45	7.38	
40	6.55	
35	5.74	
30	4.92	
25	4.10	209.10
20	3.28	167.28
15	2.48	126.48
10	1.66	84.66
7.5	1.25	63.75
0	0	0

La tabla muestra en celeste el rango de valores del diseño difuso y en verde la frecuencia máxima de trabajo alcanzable. Elaborado por: Steven Gordillo.

Para solucionar el problema anteriormente expuesto y conseguir una frecuencia de 48 Hz, se implementó externamente el amplificador de la Figura 3.30 para multiplicar el voltaje por dos.

Figura 3.30. Amplificador con ganancia igual a dos.



Circuito amplificador de señal por dos usando el integrado “LM358N”. Fuente: Proteus 8.1. Elaborado por: Steven Gordillo.

El amplificador de la Figura 3.30 recibe una señal de voltaje entre 0 y 5v desde el “Arduino” para duplicarlo. Así se obtiene un rango entre 0 y 10v necesarios para controlar la frecuencia del inversor mediante la variación de voltaje en el terminal “V1”.

Se compensó el uso del amplificador reduciendo los valores de tensión de la Tabla 3.3 por la mitad y se obtuvieron sus nuevos valores de “PWM” correspondientes, esta información se muestra en la Tabla 3.4.

Tabla 3.4. Voltaje y “PWM” de “Arduino” tras implementar el amplificador por dos

F _{control} [Hz]	V _{placa} [v]	PWM
60	5	255
55	4.53	230.78
50	4.10	208.85
48	3.95	201.45
45	3.69	188.19
40	3.28	167.03
35	2.87	146.37
30	2.46	125.46
25	2.05	104.55
20	1.64	83.64
15	1.24	63.24
10	0.83	42.33
7.50	0.63	31.88
0	0	0

Valores que maneja “Arduino”. Elaborado por: Steven Gordillo

A partir de los valores mostrados en la Tabla 3.4 se empezó a desarrollar una ecuación que permita la conversión de la frecuencia de control entregada por el “Arduino” al finalizar la defusificación (almacenada en la variable *output*) en un valor “PWM”. La estructura de la ecuación lineal que se utilizó se muestra a continuación:

$$PWM(output) = m (output) + b \quad \text{Ec. (3.21)}$$

Esta es la ecuación de la recta que interpreta al valor “PWM” como la variable dependiente y a la frecuencia de control del “Arduino” como variable independiente.

Donde:

m = Pendiente

b = Coeficiente de posición.

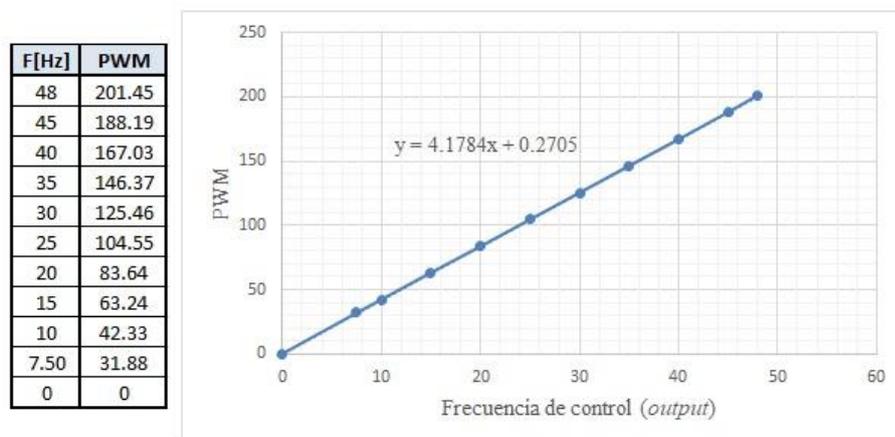
Para determinar el valor de la pendiente se tomó los datos de frecuencia de control ($f_{max} = 48 \text{ Hz}$ y $f_{min} = 7.5 \text{ Hz}$) y “PWM” correspondientes, mostrados en la Tabla 3.4 y se utilizó la Ec. 3.22 obteniendo:

$$m = \frac{PWM_{max} - PWM_{min}}{f_{Control\ max} - f_{Control\ min}} \quad \text{Ec. (3.22)}$$

$$m = \frac{201.45 - 31.88}{48 - 7.5} = 4.19$$

Después del cálculo de la pendiente, se determinó el valor del coeficiente de posición con ayuda del programa “Excel 2019”. La Figura 3.31 muestra la gráfica realizada con los valores de frecuencia de control y “PWM” de la Tabla 3.4 y la ecuación lineal obtenida tras haber activado la opción de “colocar línea de tendencia” en el gráfico.

Figura 3.31. Gráfica de los valores de frecuencia de control y “PWM”.



Se observa la tabla de valores, la gráfica y la ecuación lineal. Elaborado por: Steven Gordillo.

La ecuación lineal $y(x) = 4.18 x + 0.27$ que se muestra en la Figura 3.31, demuestra que el valor calculado de la pendiente está correcto y además da a conocer el coeficiente de posición (b) que se requería, por lo tanto, la Ec. 3.21 se reescribió tal como se muestra en la Ec. 3.23.

$$PWM(output) = 4.19 \times (output) + 0.27 \quad \text{Ec. (3.23)}$$

La Ec. 3.23 se implementó en el controlador difuso para convertir el valor defusificado (*frecuencia de control*) en "PWM". Así se logró una variación de voltaje entre 0 y 3.95v en el pin de salida digital #6 de la placa "Arduino". Esta tensión ingresa al amplificador de la Figura 3.30 para ser duplicada y posteriormente conectada al terminal "V1" del inversor para realizar el cambio de frecuencia.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

El proyecto ha sido sometido a pruebas que permitieron obtener un índice de comparación de la producción, antes y después de la automatización. Estas pruebas son:

- Pruebas de funcionamiento del controlador difuso
- Pruebas de medición del tiempo de batido
- Pruebas de medición de desperdicio de gelatina.

Se finaliza el presente capítulo mostrando unas tablas que comparan el volumen de producción con el tiempo de batido. Para empezar este capítulo se ha dado una breve descripción y funcionamiento de la máquina que se muestran a continuación.

4.1. Descripción de la máquina automatizada

La máquina consta de un tablero de control con botones de inicio y paro general, una pantalla LCD que muestra en tiempo real la información del proceso, como lo son: temperatura y nivel de la gelatina, la frecuencia que entrega el inversor y las “RPM” (Revoluciones por minuto) del eje del motor.

El controlador difuso está a cargo de la placa “Arduino Nano”. Para modificar la velocidad del motor se utiliza un inversor de la marca “LG” situado dentro del tablero de control.

El recipiente de acero inoxidable tiene capacidad para 6 L, la batidora posee un anillo de acero inoxidable con seguro en forma de rosca para sujetar el recipiente. El aspa para batir la gelatina es de acero inoxidable y el motor trifásico alcanza una velocidad de hasta 3600 RPM.

Figura 4.1. Máquina automatizada



Batidora automatizada con controlador difuso. Elaborado por: Steven Gordillo.

4.2. Funcionamiento de la máquina automatizada

La batidora debe ser conectada a una tensión de 220Vac, una vez encendida se debe verificar que el recipiente este limpio y disponible para colocar la gelatina. Luego de colocar la mezcla, se debe asegurar el recipiente para dar inicio al proceso con el botón verde ubicado en el tablero de control.

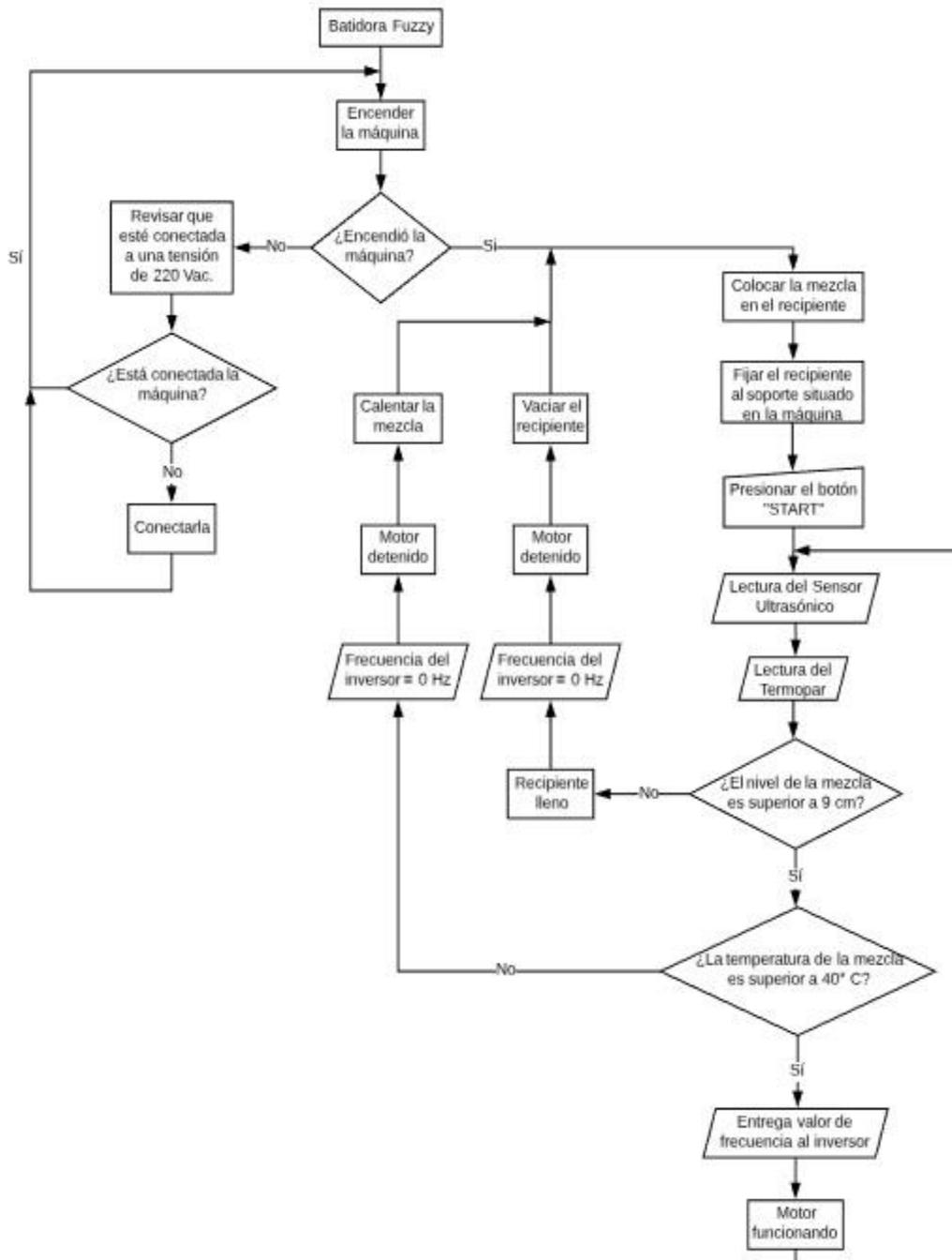
Cuando la máquina empieza a batir, arranca de forma suave para evitar regar la mezcla y poco a poco va cambiando la velocidad del motor según la información que entregan los sensores de nivel y temperatura sobre las condiciones de la gelatina.

La máquina tiene dos formas para detener el proceso:

- **Manualmente.** - Cuando se presione el botón rojo situado en el tablero de control.
- **Automáticamente.** - Cuando el sensor de nivel determine un valor inferior a los 9 cm, es decir, cuando la gelatina este 4 cm por debajo del límite superior del recipiente.

La batidora no funcionará a pesar de que se presione el botón verde de “START” si la temperatura de la gelatina es inferior a los 40 °C. La Figura 4.2 resume el funcionamiento de la máquina automatizada a través de un flujograma.

Figura 4.2. Flujograma del batido con la máquina automatizada.



Descripción del proceso de batido con la máquina automatizada. Elaborado por: Steven Gordillo.

4.3. Pruebas de funcionamiento del controlador difuso

Esta prueba se realizó con la finalidad de comparar el funcionamiento del controlador difuso real implementado en el “Arduino” y el simulado con “Matlab”.

Se procedió a colocar diez valores de temperatura y distancia desde el simulador de lógica difusa de “Matlab” para comparar el valor de la salida obtenida desde esta interfaz con el valor que entrega el controlador difuso real bajo las mismas condiciones de temperatura y distancia simuladas.

Para colocar los valores correspondientes de temperatura y nivel de gelatina en el controlador se implementó potenciómetros para facilitar la manipulación de estos parámetros y no replicar las condiciones físicamente.

Tabla 4.1. Valores obtenidos de la primera prueba.

N° Pruebas	Valores de prueba		Frecuencia del Controlador	Frecuencia de Matlab	error[%]
	Nivel	Temperatura			
1	25	42	41.1	41.1	0.0
2	27	63	41.1	41.1	0.0
3	22	85	27.2	27.5	1.1
4	16	40	41.1	41.1	0.0
5	13	55	26.6	26.8	0.7
6	19	90	32.7	32.8	0.3
7	9	47	26.1	26.3	0.8
8	12	60	21.1	21.3	0.9
9	10	75	14.9	15.2	2.0
10	20	58	35.8	36	0.6

Resultados de la variable de salida bajo determinadas condiciones y el error porcentual calculado.

Elaborado por: Steven Gordillo.

En la Tabla 4.1 se evidencia la similitud entre los valores de frecuencia entregados por el controlador difuso y la simulación realizada en “Matlab”, donde se observa un error de funcionamiento menor al 2%. Con este error obtenido se puede interpretar que el controlador físico presenta un comportamiento equivalente al teórico.

4.4. Pruebas de medición del tiempo de batido

Se realizó diez pruebas de batido antes de la automatización y diez más después de haber implementado el controlador difuso. En esta prueba se cronometró el tiempo que se demoró la máquina automatizada en terminar el proceso de batido de gelatina para determinar si se mejoró o no dicho tiempo. Los resultados se muestran en la Tabla 4.2.

Tabla 4.2. Medición de los tiempos de batido.

N° Pruebas	Tiempo de batido	
	Sin automatización	Con automatización
1	7min, 55 s	7min, 20s
2	8 min, 05 s	7min, 12s
3	7 min, 58 s	7 min, 08s
4	7 min, 56 s	7min, 25s
5	8min, 01 s	7 min, 10s
6	7 min, 52 s	7 min, 15s
7	8min, 0 s	7min, 12s
8	8 min, 08 s	7min, 18s
9	7 min, 57 s	7min, 22s
10	7min, 53 s	7min, 16s
Promedio:	7min, 45 s	7 min, 18 s

Tiempos de producción antes y después de la automatización. Elaborado por: Steven Gordillo.

De las diez mediciones tomadas se obtuvo un promedio del tiempo de producción, tanto para el batido manual como para el proceso automatizado. Los datos mostrados en la Tabla 4.2 indican que se disminuyó de 7 minutos con 45 segundos a 7 minutos con 18 segundos, es decir, se redujo casi medio minuto del tiempo de batido.

Figura 4.3. Gráfica comparativa del tiempo de batido

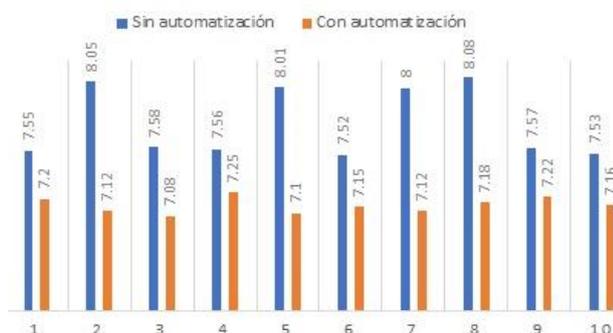


Gráfico de barras sobre el tiempo de batido. Elaborado por: Steven Gordillo.

En la Figura 4.3 se observa que la automatización disminuyó el tiempo de batido en cada una de las diez pruebas realizadas.

En líneas generales el tiempo, no se redujo mucho, pero si se utiliza las dos batidoras que la micro empresa dispone y la máquina automatizada, se llevaría a cabo el batido con tres máquinas a la vez, reduciendo el tiempo de inproductividad y permitiendo que un trabajador este libre para realizar otra actividad ya que la máquina no necesita de una persona que esté pendiente del proceso hasta terminarlo.

4.5. Pruebas de medición del desperdicio de gelatina

En esta prueba se midió la cantidad de mezcla regada fuera del recipiente con la ayuda de una balanza digital para determinar si se redujo o no el desperdicio de gelatina con la automatización. Los resultados obtenidos fueron se muestran en la Tabla 4.3.

Tabla 4.3. Medición del desperdicio de gelatina.

N° Pruebas	Desperdicio de gelatina [gr]	
	Sin automatización	Con automatización
1	12.1	0
2	14.3	1
3	13.4	0
4	10.8	0.2
5	15.1	0
6	11.7	0
7	15.6	1.2
8	11	0
9	12.4	0.8
10	12.2	0
Promedio:	12.86	0.32

Cantidad de gelatina desperdiciada antes y después de la automatización. Elaborado por: Steven Gordillo.

De las diez mediciones tomadas se obtuvo un promedio de la cantidad de gelatina desperdiciada, tanto para el batido manual como para el proceso automatizado. Los datos mostrados en la Tabla 4.3 indican que se disminuyó el desperdicio de gelatina de un promedio de 12.86 gr a 0.32 gr, es decir, se redujo un 97.5 % del desperdicio.

Figura 4.4. Gráfica comparativa del desperdicio de gelatina



Gráfico de barras del desperdicio de gelatina antes y después de la automatización. Elaborado por: Steven Gordillo.

En la Figura 4.4 se observa que la automatización disminuyó el desperdicio de gelatina en cada una de las diez pruebas realizadas.

4.6. Comparación entre el volumen de batido y el tiempo de batido.

En la Tabla 4.3 se muestra el tiempo promedio del proceso y la capacidad de los recipientes. Se realizó un análisis comparativo, para determinar la cantidad de volumen de gelatina batida aumentada luego de la automatización.

Tabla 4.4. Comparación del volumen de producción y el tiempo de batido.

Sin automatización		Con automatización	
Capacidad del recipiente [L]	Tiempo promedio de batido	Capacidad del recipiente [L]	Tiempo promedio de batido
3.5	7 min, 45 s	4.85	7 min, 18 s

Valores de volumen de gelatina batida y tiempo que demora. Elaborado por: Steven Gordillo.

Se puede observar en la Tabla 4.4, se logró aumentar la cantidad de gelatina batida gracias a la capacidad superior del nuevo recipiente. Cabe mencionar que el recipiente anterior tenía una capacidad máxima de 4 L, mientras que el recipiente utilizado en la automatización tiene una capacidad máxima de 6 L.

El tiempo de batido promedio anterior es casi el mismo que el tiempo promedio de la máquina automatizada, por lo que se podría asumir que la máquina automatizada bate

1.35 L más que las batidoras anteriores. Con estos resultados también se justifica la reducción del tiempo en el proceso de batido, ya que con las máquinas sin automatizar se conseguía menor cantidad de gelatina batida en el mismo tiempo.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

En el presente capítulo se muestran las conclusiones obtenidas al finalizar el proyecto con las respectivas recomendaciones para mejorarlo y para garantizar el correcto funcionamiento de la máquina automatizada.

5.1 Conclusiones

Se desarrolló un sistema con control difuso bajo las condiciones mencionadas en los objetivos del capítulo 1 y se demostró que la automatización del proceso fue exitosa porque el proyecto técnico arrojó resultados positivos, entre los cuales destacan: la reducción de desperdicio de gelatina en un 97.5%, la disminución del tiempo de batido en 27 segundos y el incremento del volumen de producción en 1.35 L más respecto al proceso no automatizado.

Al finalizar el trabajo se observó que las variables (temperatura, nivel y frecuencia) fueron suficientes para automatizar el proceso porque la temperatura y el nivel de gelatina en el recipiente son condiciones físicas que describen en su totalidad a esta sustancia utilizada en el proceso. Mientras que, la frecuencia tiene una relación directamente proporcional con la velocidad del eje del motor.

Se verificó que el controlador difuso programado en base a la Ec. 3.23 convirtió la frecuencia de control (obtenida de la lógica difusa) en valores “PWM” porque el “Arduino” proporcionó una variación de voltaje entre 0 y 3.95 V a uno de los pines de salida digital de la placa. El mismo que fue duplicado mediante un amplificador con ganancia igual a dos y posteriormente utilizado por el inversor para modificar la frecuencia desde 0 hasta 48 Hz.

Se verificó el funcionamiento del controlador difuso real implementado en el “Arduino” y el simulado con “MATLAB” y se obtuvo un índice de error menor al 2%, por lo que se interpretó que el controlador físico presenta un comportamiento equivalente al teórico.

Se cronometró y se comparó el tiempo de batido antes y después de la automatización, y se obtuvo un índice de reducción de 27 segundos por lo que se comprobó que se redujo casi medio minuto respecto al proceso anterior. A pesar de no haber reducido significativamente el tiempo de batido, se verificó que, utilizando las dos batidoras de la micro empresa junto con la máquina automatizada a la vez, se eliminó el tiempo de improductividad porque se consiguió mayor cantidad de gelatina batida en casi el mismo tiempo.

Se comparó el desperdicio de gelatina antes y después de la automatización, y se observó que se redujo un 97.5%. Por lo tanto, se eliminó casi por completo el desperdicio de gelatina durante el proceso.

Se comparó el volumen de gelatina batida antes y después de la automatización, y se obtuvo un índice de aumento de 1.35 L. Por lo tanto, se incrementó el volumen de gelatina batida, gracias a la capacidad superior del recipiente utilizado en este proyecto. Además, con estos resultados también se justificó la reducción del tiempo en el proceso de batido, ya que con las máquinas sin automatizar se conseguía menor cantidad de gelatina batida en el mismo tiempo.

Por último, se evidenció una mejora estética y funcional de la máquina automatizada con respecto a las batidoras que aún conserva la micro empresa, esta comparación es evidenciable si se observa la Figura 3.2 (batidoras simples) y la Figura 4.1 (máquina automatizada) ya que la batidora actual presenta un mejor diseño en su estructura metálica, mejores acabados y un tablero de control con pulsadores “Start”, “Stop” y una pantalla “LCD” que permite visualizar la información del proceso.

5.2 Recomendaciones

Este proyecto se podría mejorar, realizando un estudio sobre la forma y las dimensiones de la paleta para batir. Sin duda, este fue el inconveniente más grande durante el desarrollo de este proyecto. Se probó algunas variaciones de paletas sin lograr los resultados deseados. Los mejores resultados se obtuvieron cuando se replicó la paleta que utilizaba la micro empresa para batir. Al parecer influye bastante su forma y dimensiones.

Acerca del voltaje de alimentación, la batidora funciona con una tensión eléctrica de 220 Vac. Asegurarse de la máquina esté conectada al voltaje recomendado, caso contrario, un valor de tensión inferior no encenderá la batidora y una tensión superior podría dañar la tarjeta principal ubicada dentro del tablero de control.

Se recomienda no cambiar el recipiente de la máquina, ya que el anillo de acero inoxidable que la sujeta fue construido en base a sus dimensiones, al colocar otro recipiente diferente no se garantiza que la máquina funcione correctamente.

BIBLIOGRAFÍA

- Arduino. (2018). *Manual de Arduino Nano*. Obtenido de <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>
- BestElectronics. (2018). *Imagen de sensor ultrasonico HC-SR04*. Obtenido de <http://www.bestelectronics.com.mx/index.php/sensor-ultrasonico.html>
- Carrillo, J. (2011). *Sistemas Automáticos de Control*. Venezuela: UNERMB.
- Castillo, I. (2015). *Logica Difusa*. Paíta: UNP PROEDUNP. Obtenido de <https://sites.google.com/site/logicadifusaingindustrialpaíta/logica-difusa/conjuntos-difusos>
- GeekFactory.mx. (14 de Julio de 2018). *Termopar con Arduino y MAX6675 medición de temperatura*. Obtenido de <https://www.geekfactory.mx/tutoriales/tutoriales-arduino/termopar-con-arduino-y-max6675-medicion-de-temperatura/>
- Gómez, J. G. (2011). *Logica Difusa*. Obtenido de <http://www.lcc.uma.es/~ppgg/FSS/FSS1.pdf>
- Isis. (2018). *Manual de Variadores de frecuencia "LG"*. Obtenido de [www.lsis.com/usa/common/handler/download.ashx?path=/usa/download/no-check/download/SV-iG5\(Standard_V.5.50\)_0511.pdf](http://www.lsis.com/usa/common/handler/download.ashx?path=/usa/download/no-check/download/SV-iG5(Standard_V.5.50)_0511.pdf)
- Llamas, L. (2018). *Imagen del sensor ultrasónico*. Obtenido de <https://www.luisllamas.es/medir-distancia-con-arduino-y-sensor-de-ultrasonidos-hc-sr04/>
- MathWorks. (2018). *Fuzzy Logic Toolbox*. Obtenido de <https://www.mathworks.com/products/fuzzy-logic/features.html>
- Ponce, P. (2010). *Inteligencia artificial con aplicaciones a la ingeniería*. México D.F.: Alfaomega Grupo Editor, S.A. de C.V., México.
- Reyes, F. (2012). *Matlab aplicado a robótica y mecatrónica*. México D.F.: Alfaomega Grupo Editor, S.A. de C.V., México .
- Sanz, A., & del Brio, M. (02 de 12 de 2007). *Redes Neuronales y Sistemas Difusos*. México D.F.: Alfaomega Grupo Editor, S.A. de C.V., México. Obtenido de

http://catarina.udlap.mx/u_dl_a/tales/documentos/lep/hernandez_b_ii/capitulo_4.pdf

Torrente, O. (2013). *Arduino Curso práctico de formación*. México D.F: Alfaomega Grupo Editor, S.A. de C.V., México. Obtenido de <https://saber.patagoniatec.com/2014/12/arduino-nano-328-arduino-atmega-clon-compatible-arduino-argentina-ptec/>

Wantitall. (2018). *Imagen del modulo MAX6675*. Obtenido de https://www.wantitall.co.za/toys/for-arduino-arduino-kits-ldtr-wg0119-max6675-type-k-thermocouple-temperature-sensor-module-0-1024-c-__b07fgf4grc

Zinet Media Grup. (1 de Febrero de 2001). *Como se fabrica la gelatina*. Obtenido de <https://www.muyinteresante.es/curiosidades/preguntas-respuestas/icomose-fabrica-la-gelatina>

ANEXO

Lista de los elementos utilizados en el proyecto y sus respectivos precios

Elementos	Descripción	Costo [S]
Motor trifásico "SIEMENS"	Potencia de 1Hp, 3600 RPM	125
Variador de Frecuencia "LG"	Para motor de 1 Hp, tensión de 220v	150
Tablero eléctrico	Caja con dimensiones (20x15x25)cm	22
Recipiente de acero inoxidable	Capacidad para 6 L, dimensiones (22x20)cm	18
Material para la estructura metálica	Planchas de acero y tubos	110
Mano de obra (mecánicos)	Soldadura, trabajo en torno, pintura, entre otros	180
Material para instalación eléctrica	Rollo de cable, tomacorriente (220v), taípe, entre otros	45
Trasporte	Movilización ida y vuelta a la micro-empresa	50
Material para armar el tablero de control	Cable, Arduino, sensores, LCD, pulsadores, entre otros	50
Mano de obra (electrónico)	Diseño e implementación del controlador difuso	100
PRECIO TOTAL =		850

Elaborado por: Steven Gordillo