



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

INSTITUTO TECNOLÓGICO DE DURANGO
DIVISIÓN DE ESTUDIOS DE POSGRADO E
INVESTIGACIÓN

**INSTITUTO TECNOLÓGICO DEL VALLE DEL
GUADIANA**



Controlador de riego basado en lógica difusa para el cultivo de manzanas

TESIS

Que como parte de los requisitos para obtener el grado de

Maestría en Ingeniería

Presenta:

Ing. Manuel de Jesús Miranda Luna

Director de tesis:

M.C. José Antonio Martínez Rivera

Co-Director

M.I. Eduardo Gamero Inda

Durango, Dgo. México, Marzo, 2022.





EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Controlador de riego basado en lógica difusa para el cultivo de manzanas

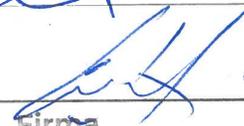
Presenta:

Ing. Manuel de Jesús Miranda Luna

Dirigido Por

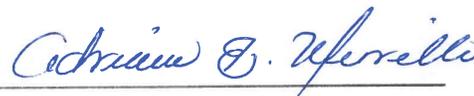
M.C. José Antonio Martínez Rivera

COMITÉ TUTORIAL

<p>Director M.C. José Antonio Martínez Rivera Nombre</p>	 <hr/> <p>Firma</p>
<p>Codirector M.I. Eduardo Gamero Inda Nombre</p>	 <hr/> <p>Firma</p>
<p>Asesor Dr. Francisco Javier Godínez García Nombre</p>	 <hr/> <p>Firma</p>



M.C. José Antonio Martínez Rivera
Coordinador de la Maestría en Ingeniería



M.C. Adriana Eréndira Murillo
Jefa de la División de Estudios de Posgrado
e Investigación

Durango, Dgo. México, Marzo, 2022



Autorización de impresión de tesis.



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Durango
División de Estudios de Posgrado e Investigación

Victoria de Durango, Dgo., a 16 / Marzo / 2022.

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
DEPI / C / 060 / 22.

ASUNTO: Autorización de Impresión de Tesis de Maestría.

C. MANUEL DE JESÚS MIRANDA LUNA
No. DE CONTROL G06040761
PRESENTE.

De acuerdo al reglamento en vigor y tomando en cuenta el dictamen emitido por el jurado que le fue asignado para la revisión de su trabajo de tesis para obtener el **Grado de Maestro en Ingeniería**, esta División de Estudios de Posgrado e Investigación le autoriza la impresión del mismo, cuyo título es:

“Controlador de riego basado en lógica difusa para el cultivo de manzanas”

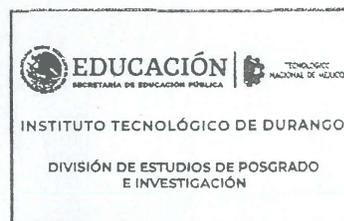
Sin otro particular de momento, quedo de Usted.

ATENTAMENTE.

Excelencia en Educación Tecnológica
“La Técnica al Servicio de la Patria”

Cochran B. Ferrallo

C. ADRIANA ERÉNDIRA MURILLO
JEFA DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



AEMammc.



Av. Felipe Pescador #1830 Ote. Col. Nueva Vizcaya C.P.34060 Durango, Durango.
Tel. (518) 8290900 e-mail: dir_itdurango@tecnm.mx | tecnm.mx | itdurango.edu.mx



2022 Flores
Año de Magón
PRESENCIA DE LA REVOLUCIÓN MEXICANA

Autorización de tema de tesis de maestría.



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Durango
División de Estudios de Posgrado e Investigación

Victoria de Durango, Dgo., a **16 / Marzo / 2022.**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
DEPI / C / 059 / 22.

ASUNTO: Autorización de Tema de Tesis de Maestría.

C. MANUEL DE JESÚS MIRANDA LUNA
No. DE CONTROL G06040761
PRESENTE.

Con base en el Reglamento en vigor y teniendo en cuenta el dictamen emitido por el Jurado que le fue asignado, se le autoriza a desarrollar el tema de tesis para obtener el **Grado de Maestro en Ingeniería** cuyo título es:

“Controlador de riego basado en lógica difusa para el cultivo de manzanas”

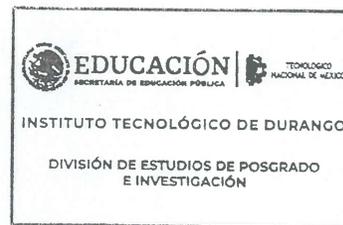
CONTENIDO:

	RESUMEN
CAPÍTULO I	INTRODUCCIÓN
CAPÍTULO II	MARCO TEÓRICO
CAPÍTULO III	MATERIALES Y MÉTODOS
CAPÍTULO IV	RESULTADOS Y DISCUSIÓN
CAPÍTULO V	CONCLUSIONES Y RECOMENDACIONES
	REFERENCIAS
	ANEXOS

ATENTAMENTE.

Excelencia en Educación Tecnológica
"La Técnica al Servicio de la Patria"

C. ADRIANA ERÉNDIRA MURILLO
JEFA DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



AEM'ammc.



Av. Felipe Pescador #1838 Ote. Col. Nueva Vizcaya C.P. 24080 Durango, Durango.
Tel. (618) 8290900 e-mail: dar_itdurango@tecnm.mx | tecnm.mx | itdurango.edu.mx



2022 Flores
Año de Magón
POR EL LOGRO DE LA MEJOR CALIDAD EDUCATIVA

Índice General.

Índice de Ilustraciones	IX
Índice de tablas	X
Índice de ecuaciones	XI
Resumen	XII
1 CAPITULO 1 Introducción.....	1
1.1 Antecedentes.....	1
1.2 Justificación.....	4
1.3 Objetivos.....	5
1.3.1 Objetivo general.....	5
1.3.2 Objetivos específicos.....	5
1.4 Delimitación.....	6
2 CAPITULO 2 Marco teórico.....	7
2.1 La manzana una fruta dulce.....	7
2.1.1 Tipos de manzanas.....	7
2.1.2 Condiciones de cosecha de la manzana.....	8
2.1.3 Métodos y tipos de riego.....	8
2.2 Lógica difusa.....	10
2.2.1 Conjunto difuso.....	10
2.2.1.1 Operaciones entre conjuntos difusos.....	11
2.2.2 Variable lingüística.....	12
2.2.3 Funciones de membresía.....	13
2.2.4 Características de los conjuntos difusos.....	15
2.3 Controlador Difuso.....	15
2.3.1 Fuzzificador.....	17
2.3.1.1 Técnicas de Fuzzificacion.....	17
2.3.2 Mecanismo de inferencia.....	18
2.3.2.1 Diferentes tipos de métodos de inferencia.....	18
2.3.3 Base de reglas.....	20
2.3.4 Defusificador.....	20
2.3.4.1 Método del centro de área.....	21

2.3.4.2	Método del promedio de centros	21
2.4	Sistema Embebido.....	22
2.4.1	Microcontroladores.....	23
2.4.2	Sensores.	24
2.5	Programación en C.	24
2.6	Simulación.	25
2.6.1	PROTEUS.....	25
2.6.2	MATLAB.....	26
3	CAPITULO 3 Materiales y métodos.	28
3.1	Microcontrolador PIC18F4550.	28
3.2	Xbee.....	29
3.3	Sensor AM2302/x.....	29
3.4	Sensor DS18B20.	30
3.5	Sensor capacitivo de humedad del suelo.....	31
3.6	Pantalla LCD.....	31
3.7	Servomotor MG995.....	32
3.8	Otros materiales.....	33
3.9	Lista de materiales propuesta.	33
3.10	Métodos.	35
3.10.1	Definir entradas y salidas del controlador difuso.	35
3.10.2	Recolectar conocimiento para crear la base de datos, definir variables lingüísticas y funciones de membresía.....	35
3.10.3	Obtener variables físicas y fusificar variables de entrada.....	35
3.10.4	Determinar cuáles reglas usar.....	36
3.10.5	Método de inferencia. Determinar conclusiones.	36
3.10.6	Convertir decisiones en acciones.	37
3.10.7	Creación y control de la electroválvula.	38
3.10.8	Simulación en proteus.....	39
3.10.9	Simulación en MATLAB.	39
3.10.10	Pruebas en protoboard.	42
3.10.11	Creación del sistema de alimentación.	42
4	CAPITULO 4 Resultados y discusión.	43

5	CAPITULO 5 Conclusiones y recomendaciones	55
6	REFERENCIAS.....	57
7	ANEXOS	61
7.1	Constancia de registro de obra provisional.....	61
7.2	Curriculum vitae único. Conacyt	62
7.3	Código en CCS Compiler de la estación meteorológica.	63
7.4	Controlador Difuso	69
7.5	Electrovalvula	91
7.6	Cuestionario dirigido al humano experto.	92

Índice de Ilustraciones.

Figura 1 Función de membresía	13
Figura 2 Diferentes tipos de formas de una función de membresía.....	14
Figura. 3 Características de un conjunto difuso	15
Figura 4 Esquema general de un controlador difuso	16
Figura. 5 Función de pertenencia para Frio, Tibio y Caliente.....	17
Figura. 6 Grado de pertenencia para $T=29^{\circ}\text{C}$	18
Figura. 7 Sistema Mamdani.....	19
Figura. 8 Sistema Sugeno	20
Figura. 9 Método del centro de área	21
Figura. 10 Método del promedio de centros	22
Figura. 11 Elementos de un microcontrolador.	24
Figura 12 PINOUT PIC18F4550	28
Figura 13 Tipos de antenas Xbee.	29
Figura 14 Diagrama de conexión eléctrica del DHT22	30
Figura 15 Vista física del sensor DS18B20.....	30
Figura 16 Sensor Capacitivo de Humedad del Suelo.....	31
Figura 17 Pantalla LCD 16X02.....	32
Figura 18 Servomotor MG995.....	32
Figura 19 Conjuntos difusos de salida implicados	37
Figura 20 Simulación en PROTEUS de la estación meteorológica, controlador difuso y servomotor	39
Figura 21 Sistema difuso en Fuzzy Logic Designer.	40
Figura 22 Entrada TempAire con sus respectivos conjuntos difusos.....	41
Figura 23 Simulación en MATLAB de una condición aleatoria.....	41
Figura 24 Prueba en Prototboard	42
Figura 25. Inicialización del sistema de riego.....	44
Figura 26. Servicio de Interrupción.	45
Figura 27. Fusificación de la variable T_a	46
Figura 28. Método de inferencia y fusificación.....	47
Figura 29. Sistema embebido controlador difuso.....	48
Figura 30. Etapa de sensado y etapa de actuación	49
Figura 31. Electroválvula	49
Figura 32. Histograma de distribución de los porcentajes de humedad del suelo.....	50
Figura 33. Histogramas de distribución de las mediciones de las variables independientes: a) Humedad relativa del aire, b) Temperatura del aire, c) Temperatura del suelo.	52

Índice de tablas.

Tabla 1 Métodos de riego y los principales factores y parámetros para su diseño.....	9
Tabla 2 Propiedades de los conjuntos difusos	12
Tabla 3 Representación matemática de una función de membresía triangular.....	14
Tabla 4 Representación matemática de una función de membresía trapezoidal	15
Tabla 5 Lista completa de materiales.....	33
Tabla 6. Variables de Entrada.....	43
Tabla 7. Estadísticos descriptivos de las variables medidas según acción de control.	53
Tabla 8. Estadísticos descriptivos por grupos según acción de control.	53
Tabla 9. Prueba de hipótesis de igualdad de los promedios de las variables.	54
Tabla 10. Coeficientes estandarizados de la función canónica discriminante.....	55
Tabla 11. Clasificación de los resultados usando la función discriminante (validación cruzada).	55

Índice de Ecuaciones

Ecuación 1 Operación Intersección	11
Ecuación 2 Operación Unión	12
Ecuación 3 Operación Complemento.....	12
Ecuación 4 Uso del operador min ejemplo 1	36
Ecuación 5 Uso del operador min ejemplo 2	37
Ecuación 6 Formula para calcula el centro de gravedad.....	38
Ecuación 7 Fórmula para calcular $\mu(i)$ si las funciones de membresía de salida son triangulos simetricos.	38

Resumen.

El agua es el factor más importante en la producción agrícola, el tiempo de irrigación, así como su distribución deben ser ajustados de acuerdo con el tipo de cultivo y condiciones ambientales para obtener los mejores resultados, estos son los objetivos de un sistema de irrigación. Los cultivos de manzanas requieren de una irrigación especial bajo ciertas condiciones atmosféricas y terrestres. El presente artículo aborda el diseño, construcción e implementación de un sistema de irrigación utilizando lógica difusa para un huerto de manzanos ubicado en Canatlán, Durango. En el diseño del controlador difuso, se definieron las variables de proceso del sistema de control y la fusificación de estas, el método de inferencia seleccionado fue tipo Mamdani y la defusificación por centro de gravedad. Se generó un conjunto de reglas IF-THEN que representan el conocimiento del humano experto. El controlador difuso implantado en un sistema embebido reemplaza al experto en irrigación.

1 CAPITULO 1 Introducción.

1.1 Antecedentes.

Se ha encontrado que el agua es considerada el factor más importante para una buena producción agrícola en regiones semiáridas (Hiessil & Plate, 1990), considerando que este recurso natural es limitado y tiene una distribución desfavorable con respecto al tiempo ya que su distribución y cantidad es variante año con año. Los sistemas de irrigación están diseñados para mitigar estas desventajas naturales y tener un consumo eficiente del agua, considerando que el tiempo de irrigación, así como su distribución deben ser ajustados de acuerdo con el tipo de cultivo con la finalidad de optimizar este recurso.

En algunos trabajos ha sido posible observar que los controladores difusos elevan la constante de proporcionalidad del sistema, pero desmejoran el factor de amortiguación. Es recomendable proporcionar un número mayor de reglas lingüísticas, como también implementar el algoritmo de control difuso en microcontroladores (Marcano, 2004).

Carpena et al. (2008), plantean el uso de componentes económicos para construir un controlador electrónico capaz de adaptarse al riego comercial existente. El nuevo dispositivo fue probado en un campo de tomate al sur de la Florida contra otros métodos de programación automática riego por golpeo y plástico acolchonado. Según sus resultados su controlador ahorro hasta un 74% de agua al tiempo que mantuvo los rendimientos con respecto a las tasas de riego fijas. Este trabajo muestra una fórmula que corresponde a la eficiencia del agua.

Fernández et al. (2008), diseñaron y probaron un sistema de irrigación automático para árboles frutales. Al terminar cada día, el dispositivo calculaba la dosis de riego a partir de las lecturas del flujo de savia en el tronco de los árboles. Luego si hay necesidad la bomba y la electroválvula se activan para irrigar el cultivo. Diseñaron un control remoto del sistema que es posible mediante el uso de una computadora o un teléfono inteligente conectado a internet. El dispositivo fue probado en un olivar al sur de España, donde demostró ser robusto y capaz de calcular datos confiables de flujo de savia, cálculo y suministro cantidades de riego diarias al huerto según el protocolo especificado. La función del control remoto demostró ser útil para

obtener información en tiempo real tanto del comportamiento del controlador como de los sensores aplicados y para cambiar los parámetros del protocolo del riego.

La interconexión de dispositivos por medio de una red es llamada IoT (Internet of Things), estos dispositivos pudieran ser desde sensores hasta objetos comunes. Según Deloitte (2020), cualquier objeto pudiera ser conectado a la red e interactuar sin necesidad humana, dando lugar únicamente a la relación entre dispositivos. El proyecto de Teja et al. (2018), toma la definición de IoT, este proyecto consiste en registrar variables en un campo agrícola como nivel del agua, humedad del aire y humedad del suelo permitiéndole al agricultor tomar decisiones a distancia para el cuidado de su huerto, se leyeron las variables por medio de diferentes sensores y se procesaron en Arduino, este proyecto utiliza la nube de código abierto "ThinkSpeak" para almacenar la información actual del campo de cultivo.

Cuatodio y Granados (2016), publicaron un libro del uso de un controlador de lógica difusa con el microcontrolador PIC18F4550 para el control de fluidos. Su trabajo se desarrolla en dos grandes aspectos, el diseño y la construcción de una maqueta. En el aspecto del diseño se aborda el desarrollo del software necesario mientras que en la implementación se llevan a cabo diferentes pruebas para garantizar el buen funcionamiento del controlador difuso.

Hu et al. (2013), crearon un controlador de riego inalámbrico basado en el microcontrolador CC1100 que está diseñado para ahorrar agua de riego y mejorar la eficiencia de los recursos hídricos. El sistema utiliza el sensor de humedad de suelo FDS-100 y los sensores digitales de temperatura y humedad AM2302, se ha realizado un monitoreo remoto de las diferentes variables. Las pruebas demuestran que obtuvieron un controlador confiable fácil de operar y que puede cumplir con los requisitos del riego con alta precisión. Con la recolección de datos como la humedad y la temperatura del aire y del suelo, el controlador puede ajustar automáticamente el tiempo de riego, el intervalo y algunos otros parámetros.

En el desarrollo de su trabajo Bahat (2000) determina que el controlador de un sistema de irrigación es el cerebro de todo el proceso, cuyo objetivo es supervisar el flujo de agua para obtener los mejores resultados de una cosecha utilizando la mínima cantidad de agua. En su estudio establece que los irrigadores se dividen en controladores de lazo abierto y de lazo cerrado. Por otra parte asume que los controladores de lazo abierto se identifican por un funcionamiento basado en una configuración del tiempo en el controlador, sin importar si el cultivo tiene la

cantidad de agua necesaria. En lo que respecta a un controlador de lazo cerrado considera la necesidad de realizar diferentes mediciones de los parámetros que influyen en la decisión de la cantidad de agua que es necesaria para el cultivo. Algunos de estos parámetros son fijos y no es necesario medirlos, como: el tipo de plantas, tipo de tierra, cobertura foliar, estado de crecimiento, pero otros parámetros varían y es necesario realizar su medición, tal es el caso de la temperatura, humedad de aire, radiación en la tierra, humedad en la tierra. Entonces cuando estos parámetros cambian, la cantidad de agua suministrada también debe de cambiar. El parámetro de salida sería abrir y cerrar las válvulas de agua. Todas las variables de entrada son definidas como variables difusas. La razón de definir estas variables como difusas es que no se requiere de una medición precisa.

Delgado et al. (2013) presentan el diseño de un controlador difuso proporcional CPD, que no utiliza un modelo matemático en cambio usa un conjunto de reglas difusas para lograr el desacoplamiento del sistema, se hace énfasis en el cuidado del agua y los resultados se presentan por medio de simulaciones.

Poyena et al. (2015), plantean la simulación en MATLAB de una red neuronal para el control de un sistema de riego donde expone que se obtienen mejores resultados que en un sistema de prendido/apagado. Sin embargo, también propone la necesidad de utilizar sensores para la toma de decisiones, por lo que en otros estudios muestra que la irrigación no controlada provoca un gasto innecesario del vital líquido, así como un decremento en la producción de los campos agrícolas, ya que se ha sido demostrado que en la India tuvieron una producción de 0.87Kg/m³ a diferencia de la aplicación de la técnica en países desarrollados donde la producción se eleva hasta 2Kg/m³. En este trabajo se propone la existencia de parámetros fijos y dinámicos, en los parámetros fijos son considerados el tipo de suelo, el estado de crecimiento de las plantas, la salinidad del suelo, cobertura foliar. En los parámetros dinámicos son considerados la humedad de la tierra, temperatura ambiente, velocidad del viento y radiación PAR. Su trabajo propone cuatro etapas para su sistema de red neuronal con sensores de entrada para la medición de temperatura, humedad relativa del aire, humedad de la tierra, velocidad del aire y radiación. Se considera un modelo de evapotranspiración, que convierte los parámetros de entrada en una medición de humedad de la tierra, así como, la humedad requerida por el usuario. Y la red neuronal compara la humedad de la tierra requerida por el usuario con la humedad actual de la tierra para después tomar una decisión.

1.2 Justificación.

La manzana es una de las especies de fruta dulce más conocida en todo el mundo y una de las de mayor valor económico, su demanda se acrecienta continuamente y con ello su cultivo, producción y comercio. México ocupa el decimotercer lugar en producción de manzana a nivel mundial según la Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO, 2017). De acuerdo con los datos de 2017, esta fruta presentó una producción total de 714,149 toneladas, con un valor de 4.66 mil millones de pesos, en una superficie cosechada de 54,248 hectáreas en México. Del total de la producción nacional de manzana Chihuahua aportó 82.51% del volumen nacional recolectado, seguido a distancia por Puebla y Durango con 5.15% y 4.57%, respectivamente (SAGARPA, 2018).

La fruticultura en el estado de Durango juega un papel importante para su desarrollo económico, de ahí el interés para aumentar los niveles de producción de la manzana. Durango es un estado que tiene las características necesarias para aumentar la producción de dicha fruta, ya que cuenta con varias regiones que son propicias para el crecimiento adecuado del fruto. Para una buena y eficiente producción de manzanas es necesario el conjunto de varias variables tanto atmosféricas como del suelo, pero principalmente de la cantidad de agua en el cultivo que afecta directamente varias variables que benefician el crecimiento de la manzana.

Los sistemas de irrigación están diseñados para tener un consumo eficiente del agua, considerando que el tiempo de irrigación, así como su distribución deben ser ajustados de acuerdo al tipo de cultivo con la finalidad de optimizar este recurso. La cosecha en muchos campos de cultivo de manzana en el estado de Durango se basa en la experiencia y el conocimiento de sus productores. El control difuso que se basa en la experiencia, la heurística y las reglas para controlar un proceso hacen que estas características sean una ventaja para implementar un controlador difuso para sistemas de riego en campos agrícolas del estado de Durango. El control difuso provee las técnicas y procedimientos necesarios para simular el comportamiento humano.

1.3 Objetivos.

1.3.1 Objetivo general.

Este proyecto pretende apoyar la producción frutícola de las zonas productoras de manzana del Estado de Durango. Así, el objetivo general del proyecto es:

Diseñar un controlador difuso multivariable en un sistema embebido para dosificar el riego aplicado al suelo del terreno agrícola basado en la medición de variables con incertidumbre.

1.3.2 Objetivos específicos.

1. Diseñar e implementar una estación meteorológica para la medición de temperatura y humedad del aire y del suelo.
2. Obtener el modelo general del sistema por medio de reglas lingüísticas para determinar el número de entradas y salidas necesarias.
3. Simular en MatLAB el comportamiento del controlador difuso tipo MISO
4. Verificar la estabilidad del controlador difuso
5. Obtener el pseudo código del controlador difuso
6. Simulación del controlador difuso en Proteus
7. Desarrollo del sistema embebido
8. Puesta en marcha del sistema de control

1.4 Delimitación.

Un controlador difuso es capaz de manejar N número de entradas, sin embargo, el número de reglas se incrementa exponencialmente con un incremento en el número de entradas del controlador difuso o del número de las funciones de membresía. Por ejemplo, si se tienen dos entradas y cada una de ellas posee 11 funciones de membresía en el universo de discurso, entonces hay 121 reglas posibles. La incorporación de más entradas al sistema con sus respectivas funciones de membresía provocara un mayor número de reglas posibles por lo que la memoria del microcontrolador podría saturarse. Sin embargo, es posible que rediseñando el software del microcontrolador sea posible incrementar sustancialmente el número de entradas permitidas. Las opciones más viables para incrementar el número de entradas serian utilizar lenguaje ensamblador en combinación con lenguaje C o reutilizar funciones para no saturar la memoria del microcontrolador. Para este proyecto se tiene contemplado utilizar cuatro entradas para el controlador difuso: temperatura aire, temperatura de la tierra, humedad relativa del aire y humedad de la tierra, sin embargo, el humano experto podría determinar que son necesarias otras entradas como podrían ser: radiación PAR, velocidad del aire, salinidad, horas frio, etc. Así mismo para cada entrada se tiene contemplado utilizar cinco funciones de membresía como máximo en el universo discurso por lo tanto el número de reglas máximas posibles asciende a 625.

El campo de cultivo no cuenta con una señal de Internet, por lo que es un inconveniente para transferir la medición de las variables de proceso captadas por la estación meteorológica. Se considera el desarrollo de una interfaz Hombre-Máquina que permita la interacción con el sistema de control a un nivel reducido.

El campo de cultivo en el que se va a trabajar posee un sistema de riego por inundación lo ideal para el proyecto sería un conjunto de tuberías delimitadas por un conjunto de válvulas, de esta manera el controlador difuso podría manejar de mejor manera áreas específicas del campo de cultivo. El tener una red de tubería en un campo de 3.5 hectáreas representa una inversión considerable. El sistema de control tiene acción sobre el sistema de irrigación sobre un sector de prueba, el cual permite la validación experimental del sistema desarrollado.

2 CAPITULO 2 Marco teórico.

2.1 La manzana una fruta dulce.

En el año 2009, el mayor consumo de fruta fresca se produce en Italia con 103Kg por persona y año (Viñas, Recansens, Usall, & Graell, 2013). Desde hace mucho tiempo el consumo de frutas ha sido valorado por el hombre, las frutas y las verduras son parte de una dieta saludable y su consumo diario puede ayudar a prevenir la aparición de enfermedades cardiovasculares o la aparición de cáncer. Se calcula que cada año podría salvarse 1.7 millones de vidas si se aumentara lo suficiente el consumo de frutas y verduras (Viñas, Recansens, Usall, & Graell, 2013). En lo que respecta a las manzanas, estas siempre han sido comprendidas entre las frutas más completas ya que aportan hidratos de carbono, un alto contenido en vitaminas y minerales, son ricas en fibra y además contienen un considerable contenido de antioxidantes.

La superficie dedicada al cultivo de manzanos en México es de 66,400 Ha, lo cual representa el 6% de la superficie total ocupada por todas las especies de árboles frutales presentes en México. Principalmente se produce en la zona norte, en los estados de Chihuahua, Coahuila y Durango. En Durango la superficie dedicada a la manzana es de 12,300 Ha de las cuales el 70% está ocupada por variedades de red delicious (González-Portillo, y otros, 2008).

2.1.1 Tipos de manzanas.

La familia de las manzanas es amplia y diversa, podemos encontrar diferentes variedades como: Fuji, Gala, Golden, Red Delicious, Pink Lady y Granny Smith, (PENELO, 2018). La manzana Fuji puede variar entre rojo y rosa, es una de las más dulces. Gala, este tipo de manzana tiene una cascara brillante a rayas o estrías rojas anaranjadas sobre una piel amarillenta. Manzana Golden tiene piel brillante y dorada, es crujiente y bastante dulce. Red delicious, tiene una piel roja oscura con un dulzor medio, es aromática y tiene una piel jugosa. Manzana Pink Lady, tiene una piel rosada con un amarillo verdoso. Granny Smith, su piel es de color verdoso y es una de las manzanas más ácidas que hay, su pulpa es verdosa, muy densa y crujiente, con un punto efervescente.

2.1.2 Condiciones de cosecha de la manzana.

Los campos de manzano requieren de 1200 unidades frío durante el invierno para producir frutos de buena calidad. La zona manzanera de Durango tiene una acumulación promedio de frío inferior a 400 unidades frío, por ello el rendimiento y calidad de la manzana pueden verse afectados, sin embargo, estas condiciones pueden producir adaptación al medio ambiente generando mutaciones (González-Portillo, y otros, 2008).

La calidad de la fruta está íntimamente relacionada con la composición mineral de la misma (Caracterización mineral de manzana Red Delicious y Golden Delicious de dos países productores, 2007). La calidad de un fruto fresco está generalmente relacionada con sus características físicas como: color, firmeza, tamaño, su composición química y bioquímica la cual determina el gusto y sabor característico. (WILSON, Boyette, & E.A, 1999) postularon que la máxima calidad del fruto es alcanzada al momento de la cosecha, después de ese momento no es posible incrementar la calidad de los parámetros decisivos.

2.1.3 Métodos y tipos de riego.

El método de riego se refiere a la forma en la que se aplica el agua a la parcela y los mecanismos que intervienen en ello y por tipo de riego se refiere a la modalidad de aplicación del agua, asociada a un momento y finalidad específica dentro del ciclo de un cultivo o parcela (Delgadillo Iriarte , 2000).

Los métodos de riego se ubican en tres categorías: riego por gravedad (superficial), riego a presión (mecánico) y riego subsuperficial. En el riego por gravedad el agua fluye gracias a una pendiente, en este método el agua se aplica al terreno en la parte más alta, por consiguiente el agua fluye a los puntos más bajos, disminuyendo en cantidad y volumen a medida en que se filtra en el suelo (GUROVICH, 1985). Por otra parte, el riego a presión requiere de dispositivos mecánicos que se necesitan para la aplicación del agua, tuberías a presión, aspersores, goteros. En el caso de riego subsuperficial se aprovecha la capilaridad de las raíces a la cercanía de una fuente natural de agua. La elección del método de irrigación debe basarse en función de lograr la mayor eficiencia en el uso del agua con el fin de obtener la máxima producción.

Tabla 1 Métodos de riego y los principales factores y parámetros para su diseño.

METODO	VARIABLES
RIEGO POR SUPERFICIE	
Cajetas	<ul style="list-style-type: none"> - Tiempo de oportunidad - Tiempo de avance - Longitud de cajeta - Tiempo de aplicación - Profundidad máxima de aplicación
Melgas	<ul style="list-style-type: none"> - Caudal unitario - Tiempo de receso - Tiempo de aplicación - Eficiencia de aplicación Condiciones de diseño: <ul style="list-style-type: none"> - Max-min. caudal unitario - Pendiente máxima - Longitud máxima de melga - Extensión de la melga
Surcos	<ul style="list-style-type: none"> - Ajuste del perímetro mojado - Tiempo de avance - Tiempo de oportunidad - Aplicación bruta de riego - Infiltración acumulada - Tiempo de oportunidad - Escurrimiento superficial - Percolación profunda - Eficiencia de aplicación
RIEGO PRESURIZADO	
Aspersión	<ul style="list-style-type: none"> - Pérdidas por fricción selección de aspersores: <ul style="list-style-type: none"> - Cálculo de la frecuencia de riego - Precipitación requerida - Capacidad del sistema - Número de cambios por día - Cálculo del caudal para diferentes espaciamientos - Efecto del viento
Goteo	<ul style="list-style-type: none"> - Pérdidas por fricción - Distribución de presiones Parámetros de diseño: <ul style="list-style-type: none"> - Descarga requerida - Profundidad pico de aplicación - Intervalo y ciclo de riego- cálculo del diámetro mojado - Aplicación mínima y máxima

Ahora bien como se mencionó anteriormente, se entenderá como tipo de riego a las distintas modalidades de riego que más bien se asocian a los momentos específicos dentro del

ciclo de cultivo y que están orientados a cumplir objetivos diferentes. Siguiendo un ciclo agrícola completo, se puede diferenciar los siguientes tipos de riego, los que corresponderán a distintos cultivos: riego de remojo, empato, de trasplante, de siembra, al cultivo, de cosecha (Delgadillo Iriarte, 2000).

2.2 Lógica difusa.

El conjunto de lógica difusa fue introducido a mediados de los años sesenta por el ingeniero eléctrico iraní Lofti Zadeh, quien en 1965 publica el artículo de lógica difusa llamado "fuzzy sets". Más tarde en 1974 Ebrahim Mamdani aplica los conceptos de lógica difusa para desarrollar un control difuso para la regulación de un motor de vapor. En 1985 Takani y Sugeno aportan un nuevo método, como alternativa al método Mamdani. En los últimos años la lógica difusa se ha extendido a campos como: la industria, la medicina, la aeronáutica y la electrónica. (Guzmán & Castaño, 2006). Las aplicaciones de lógica difusa se han consolidado y ampliando en los últimos años, quizás por su sencillez conceptual, su habilidad para adaptarse a casos particulares, su habilidad para relacionar expresiones lingüísticas con datos numéricos y el no requerir algoritmos complejos. (Duarte, 1999). La lógica difusa es una metodología que proporciona una manera simple de obtener una conclusión a partir de datos de entrada vagos, ambiguos, imprecisos, con ruido o incompletos. La lógica difusa trata de imitar la toma de decisiones de una persona con datos de entrada que son difusos. La lógica difusa se caracteriza por trabajar con valores de alto grado de imprecisión a diferencia de la lógica clásica que trabaja con datos precisos. Zadeh dice: "La lógica difusa trata de copiar la forma en que los humanos toman decisiones", Zadeh menciona que la gente no necesita información numérica precisa de su entorno para desarrollar tareas de control, como por ejemplo manejar un carro o caminar por la banqueta sin chocar. Si los controladores convencionales, aceptaran entradas con ruido o imprecisas podrían trabajar de una manera más eficiente. El hombre trato de ajustar el mundo real a modelos matemáticos rígidos, como la lógica clásica, como por ejemplo una manzana es roja o no lo es, claramente podríamos obtener más variaciones del color rojo de la manzana. (D'Negri & De Vito, 2006)

2.2.1 Conjunto difuso.

Un conjunto es una colección de elementos bien definidos, en el que es posible determinar si un elemento pertenece o no al conjunto (Morales-Luna, 2002). Muchas clases de

objetos que se encuentran en el mundo real no se encuentran agrupados de manera clara en un conjunto, por ejemplo, en el conjunto de animales tendremos, perros, gatos, pericos, peces, etc. y se puede excluir muy fácilmente de este conjunto a, minerales, agua, plantas, etc. sin embargo no es tan fácil incluir o desechar dentro del conjunto los siguientes ejemplos, estrellas de mar, bacterias, etc. y este problema se puede encontrar en varios conjuntos que no forman parte del pensamiento matemático y que más bien tienen su centro en el pensamiento humano. Como por ejemplo la estatura de una persona de 1.70m en los que la mayoría de un país determinado como México clasificaría esta altura como alta, sin embargo, en otros países como por ejemplo los países europeos clasificarían esta altura como el promedio de la población o incluso baja. Los conjuntos difusos, es una clase de conjuntos que asignan un grado de membresía de acuerdo con el elemento que se esté analizando (Zadeh, 1965).

A diferencia de la teoría clásica de conjuntos que se basa en el principio básico de la lógica de forma que un individuo pertenece o no pertenece a un conjunto, la idea básica de un conjunto difuso es que un elemento forma parte de un conjunto con un determinado grado de pertenencia. Así, un conjunto difuso proporciona una transición suave entre los límites de lo que sería un conjunto clásico. El Universo del discurso se define como todos los posibles valores que puede tomar una determinada variable (González Morcillo).

Conjunto difuso: Expresa el grado de pertenencia al conjunto que tiene cada uno de los elementos (Cuatodio & Granados, 2016).

2.2.1.1 Operaciones entre conjuntos difusos.

En la teoría del control difuso, se trabaja con colecciones de conjuntos difusos por lo que se necesitan métodos útiles para hacer estas combinaciones. Estos métodos coinciden con los métodos usados para los conjuntos clásicos. Los diferentes operadores usados para combinar conjuntos difusos son llamados conectivos o de agregación.

Mínimo, máximo y complemento. Estas operaciones están determinadas por sus funciones características, las cuales son:

$$(A \cap B)(x) = \begin{cases} 1 & \text{Si } x \in A \text{ and } x \in B \\ 0 & \text{Si } x \notin A \text{ or } x \notin B \end{cases}$$

$$(A \cup B)(x) = \begin{cases} 1 & \text{Si } x \in A \text{ or } x \in B \\ 0 & \text{Si } x \notin A \text{ and } x \notin B \end{cases}$$

Ecuación 2 Operación Unión

$$(X - A)(x) = \begin{cases} 1 & \text{Si } x \notin A \\ 0 & \text{Si } x \in A \end{cases}$$

Ecuación 3 Operación Complemento

Los conjuntos clásicos y los difusos tienen las mismas propiedades, que se pueden ver en la Tabla 2.

Tabla 2 Propiedades de los conjuntos difusos

Propiedad	
Conmutativa	$A \cap B = B \cap A$
Asociativa	$A \cup (B \cap C) = (A \cup B) \cap C$
Distributiva	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Idempotencia	$A \cup A = A$ y $A \cap A = A$
Involución	$\neg(\neg A) = A$
Transitiva	If $(A \subset B) \cap (B \subset C)$ then $A \subset C$
Leyes de Morgan	$\neg(A \cap B) = \neg A \cup \neg B$ y $\neg(A \cup B) = \neg A \cap \neg B$

Se puede ver fácilmente, que las operaciones difusas normales se realizan como las operaciones correspondientes para los conjuntos clásicos o tradicionales (Reina, 2008).

2.2.2 Variable lingüística.

Una variable lingüística es aquella cuyos valores son palabras o sentencias en un lenguaje natural o artificial (Zadeh, 1975). Una variable lingüística sirve para representar cualquier elemento que sea demasiado complejo o del cual no tengamos una definición concreta.

Una variable lingüística se caracteriza totalmente por un quintuplo $(v, T(x), X, g, m)$ en donde, "v" es el nombre de la variable lingüística, "X" es el conjunto universo, "T" es el conjunto de términos lingüísticos de "v" que se refiere a una variable base de quien el rango de valores es un conjunto X, "g" es una regla sintáctica por lo general los términos lingüísticos en "T", m es una regla semántica que asigna a cada termino lingüístico $t \in T$ su significado $m(t)$ que es un conjunto difuso en X.

2.2.3 Funciones de membresía.

El objetivo de una función de membresía (μ) es ser la referencia para cuantificar los valores lingüísticos. En la Figura.1, se muestra una gráfica que representa una función de membresía donde el eje de las "x" es el error que existe en un péndulo y el eje de la "y" es el valor de pertenencia a la función de membresía. Por ejemplo si $e(t)=\pi/4$ entonces $\mu(t)=1$, lo que indicaría que el valor pertenece en un 100% a este conjunto, otro ejemplo sería considerar un valor de $e(t)=\pi/8$ entonces $\mu(t)=0.5$ indicando que el valor pertenece en un 50% a este conjunto, también se podría observar que para valores de $e(t)$ menores de 0 o mayores $\pi/2$, $\mu(t)$ valdría cero por lo que estos valores no serían parte del conjunto. Entonces podemos decir que las funciones de membresía cuantifican de una manera continua los que puede tomar el eje "x".

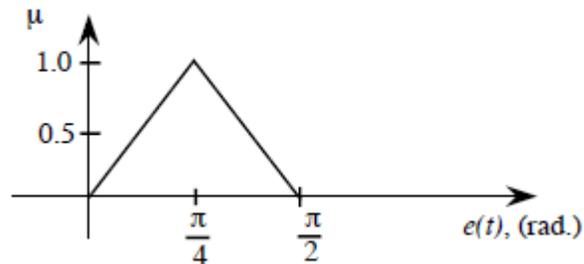


Figura 1 Función de membresía

Otras funciones de membresía que se pudieran utilizar dependiendo de la aplicación y del diseñador se muestran en la Figura 2. La definición de la forma de la función de membresía es más subjetiva que objetiva (Passino & Yurkovich, 1998).

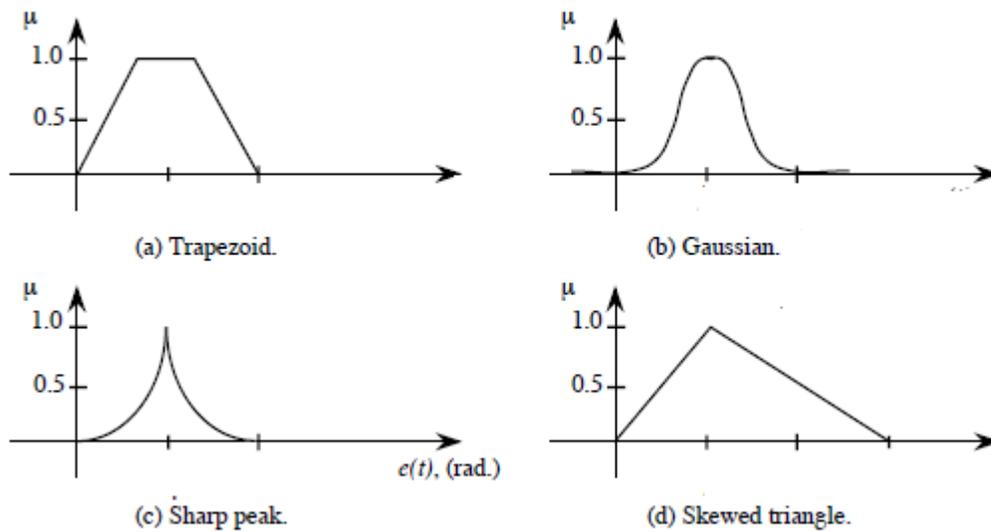


Figura 2 Diferentes tipos de formas de una función de membresía

Las funciones de membresía más usadas en la teoría de control son la triangular, trapezoidal, gaussiana y sigmoideal.

Triangulares y trapezoidales, son muy usadas en las aplicaciones, la representación gráfica y las operaciones con estos conjuntos difusos son muy simples, además pueden ser construidas fácilmente con poca información.

La función triangular A comprendida entre los puntos $(a, 0)$ y $(b, 0)$ y con una altura en la coordenada (c, α) está definida por:

Tabla 3 Representación matemática de una función de membresía triangular

A(x)	$\alpha \left(\frac{x-a}{c-a} \right)$	Si $a \leq x \leq c$
	$\alpha \left(\frac{x-b}{c-b} \right)$	Si $c \leq x \leq b$
	0	Cualquier otro valor

La función trapezoidal B comprendida entre los puntos $(a, 0)$ y $(b, 0)$ y con alturas en los puntos (c, α) y (d, α) está definida por:

Tabla 4 Representación matemática de una función de membresía trapezoidal

B(x)	$\alpha \left(\frac{x-a}{c-a} \right)$	Si $a \leq x \leq c$
	α	Si $c \leq x \leq d$
	$\alpha \left(\frac{x-b}{c-b} \right)$	Si $d \leq x \leq b$
	0	Cualquier otro valor

2.2.4 Características de los conjuntos difusos.

Apoyo, en un conjunto difuso A es el conjunto de todos los puntos x para los cuales la función de pertenencia ($\mu_A(x)$) es mayor que cero. Centro, en un conjunto difuso A es el conjunto de todos los puntos para los cuales la función de pertenencia ($\mu_A(x)$) es igual a 1. Normalidad, un conjunto difuso es normal si siempre existe un punto para el cual la función de pertenencia es 1. Puntos de cruce, son los puntos del conjunto difuso para los cuales $\mu_A(x)=0.5$.

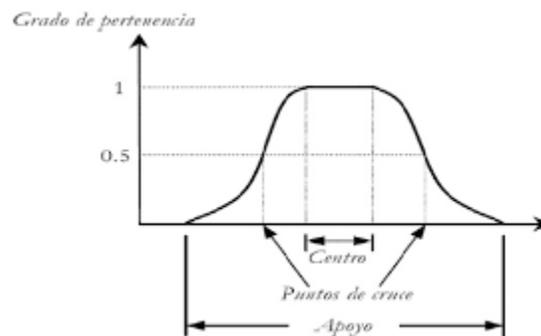


Figura. 3 Características de un conjunto difuso

2.3 Controlador Difuso.

La principal característica de estos controladores es que no es necesario disponer de un modelo matemático del sistema, sino que el proceso de diseño consiste en la definición de reglas cualitativas. Con estos controladores es posible aplicar en una amplia gama de problemas no lineales y de dinámica complicada. El principal problema de la teoría de control es encontrar la manera de modificar la respuesta de un sistema para que su salida $y(t)$ sea lo más parecido a su referencia $r(t)$.

Llamamos planta a un sistema físico real cuyo comportamiento se desea controlar mediante alguna cantidad de parámetros. El modelo clásico para atacar el problema de control es utilizar un controlador que opere sobre las entradas de la planta para que esta responda satisfactoriamente. Para ello necesitamos tener un modelo matemático de la planta mediante ecuaciones diferenciales.

El diseño de un controlador clásico utilizando la teoría de control lineal necesita que la planta cumpla lo siguiente: un modelo matemático basado en ecuaciones diferenciales y la planta debe ser LTI (Linear Time Invariant) es decir que al mismo tiempo sea lineal e invariante en el tiempo. Con la idea de contrarrestar la falta de un modelo matemático, aparecieron los controladores difusos.

Un controlador difuso decide la mejor opción de control $u(t)$ en función de los datos disponibles en el vector $m(t)$ según una base de conocimientos que están presentes en forma de reglas lingüísticas. Un controlador difuso se divide en cuatro partes.

Una interfaz real-difusa que convierte la entrada en n variables lingüísticas, una base de reglas lingüísticas que cuantifica el conocimiento necesario para controlar la planta utilizando lógica difusa, un mecanismo de inferencia que decide cual es la mejor acción en función del estado actual y de la información almacenada en la base de reglas, una interfaz difusa-real que convierte las conclusiones lingüísticas del mecanismo de inferencia en una acción de control (Theler, 2007).

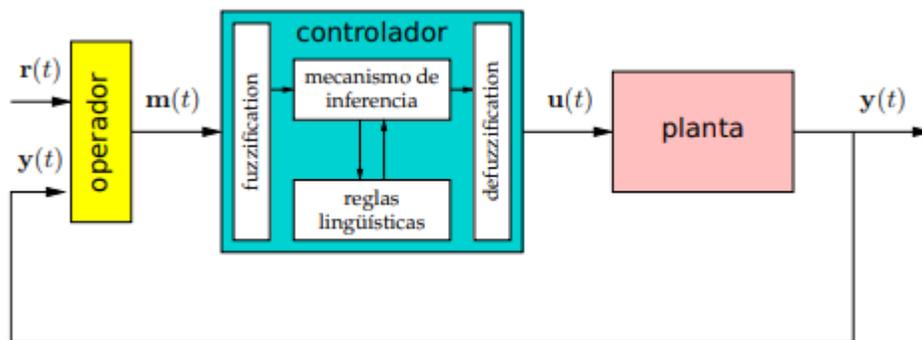


Figura 4 Esquema general de un controlador difuso

2.3.1 Fuzzificador.

Será la primera parte del controlador difuso, donde las diferentes mediciones de los sensores serán convertidas a valores difusos de acuerdo a las funciones de membresía previamente establecidas en el controlador, este proceso es llamado fuzzificación.

En esta etapa se toman los datos de entrada y se determina el grado en el que pertenecen a cada uno de los conjuntos borrosos a través de las funciones de pertenencia.

2.3.1.1 Técnicas de Fuzzificación.

Es un proceso que permite asociar a un valor numérico un conjunto difuso, asignándole un grado de pertenencia según un término lingüístico a partir de la función de pertenencia. Este proceso responde a un conjunto de normas preestablecidas, conceptualizadas a partir del conocimiento que brinda el razonamiento humano a través de un sistema implementado vía software. También se conoce como “borrosificación” (Sanahuja, 2017).

Por ejemplo si se tienen tres conjuntos difusos Frío, Tibio y Caliente podemos representar estos conjuntos como en la figura 5.

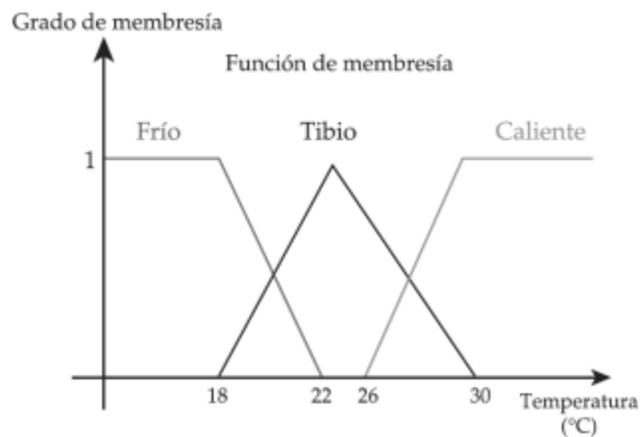


Figura. 5 Función de pertenencia para Frío, Tibio y Caliente

Para saber el grado de pertenencia de determinada temperatura, por ejemplo 29°C bastaría con trazar una perpendicular en ese punto y precisar el valor de Y (Grado de membresía) del punto de la intersección de la perpendicular con la función o funciones de membresía activadas, como se muestra en la figura 6. Así pues la variable $T=29^{\circ}\text{C}$ representa los siguientes grados de pertenencia, $\mu_{\text{frío}}(29) = 0$, $\mu_{\text{tibio}}(29) = 0.22$ y $\mu_{\text{caliente}}(29) = 0.82$.

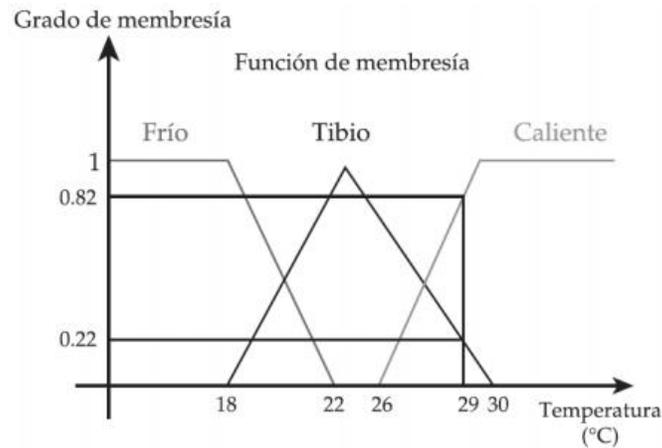


Figura. 6 Grado de pertenencia para T=29°C

2.3.2 Mecanismo de inferencia.

Se entiende por inferencia borrosa la interpretación de las reglas del tipo SI-ENTONCES con el fin de obtener las conclusiones de las variables lingüísticas de salida, a partir de los actuales valores de las variables de entrada. La inferencia se basa en el paradigma Modus Ponens Generalizado, el cual se puede interpretar como la transformación de los grados de cumplimiento del lado izquierdo de las reglas a grados de posibilidad de los lados derechos de las mismas (Maguiña Pérez, 2010).

El mecanismo de inferencia representa el núcleo del controlador difuso pues agrupa toda la lógica de inferencia borrosa del sistema, del barrido de las reglas durante ésta, la determinación de las reglas activadas. Cada regla tiene un nivel de disparo que depende del grado de similitud que guarda, del estado actual de las variables de entrada del sistema y la estructura de los antecedentes de cada regla (Aja Fernández., 2003).

2.3.2.1 Diferentes tipos de métodos de inferencia.

Inferencia en sistemas Mamdani. Comprende dos fases básicas. Cálculo de la parte SI (IF) de las reglas, evalúa el grado de certeza o activación del lado izquierdo de cada regla para los valores actuales de las variables de entrada, si la regla que se evalúa es la "n", el grado de certeza o activación se representa por $\mu(n)$, en esta fase se usan generalmente los operadores mínimo y producto para evaluar el and que conecta las preposiciones del lado izquierdo. Cálculo de la parte Entonces de las reglas. A partir del grado de activación se determina la conclusión de la regla, asigna a cada variable de salida del consecuente el conjunto borroso correspondiente modificado.

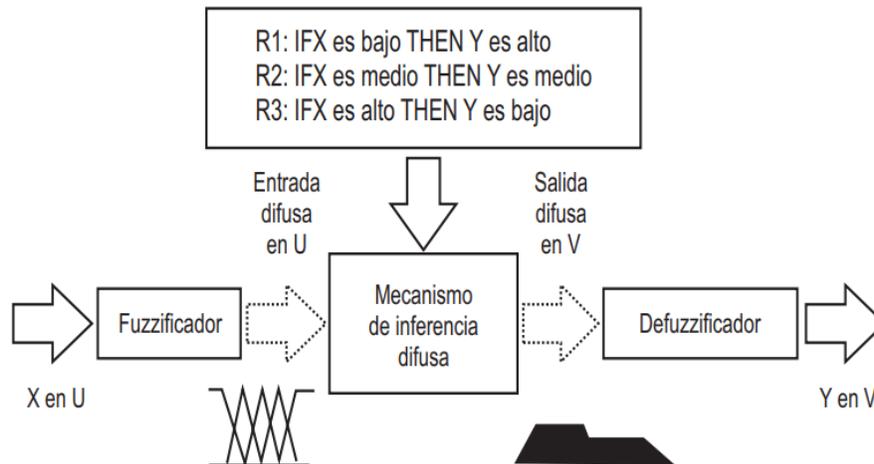


Figura. 7 Sistema Mamdani

Inferencia en sistemas Sugeno. En lugar de trabajar con una salida borrosa, Takagi, Sugeno y Kang propusieron un nuevo modelo basado en reglas donde el antecedente está compuesto de variables lingüísticas y el consecuente se representa como una función lineal de las variables de entrada (Sugeno & Takagi, 1985). Cálculo de la parte SI(IF) de las reglas, el cálculo del lado izquierdo de las reglas borrosas en estos sistemas es el mismo que en los sistemas Mamdani; al aplicar el operador de implicación escogido se obtiene un grado de pertenencia o activación α_j para cada una de las reglas disparadas. Cálculo de la parte ENTONCES de las reglas, en el lado derecho de estas reglas se obtiene el respectivo valor de salida mediante la combinación lineal de las entradas: $v_j = f(u_1, u_2, \dots, u_n)$ donde el subíndice en la variable de salida v_j se refiere al número de la regla disparada.

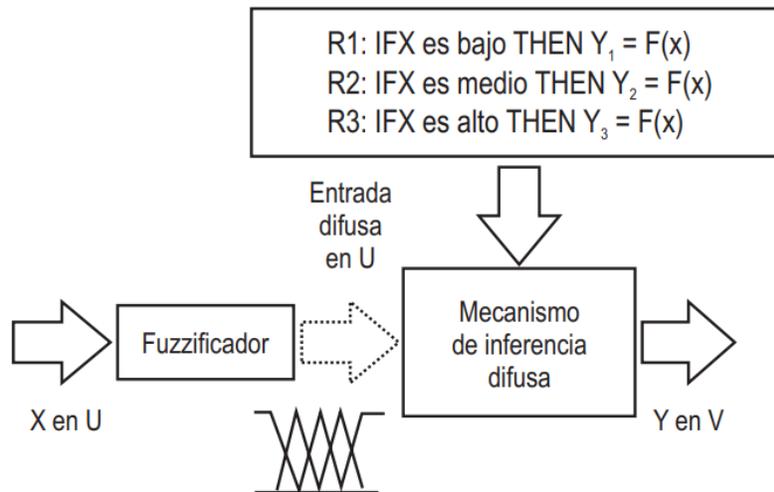


Figura. 8 Sistema Sugeno

2.3.3 Base de reglas.

La principal diferencia y ventaja entre los controladores clásicos y difusos radica en la forma de expresar la información necesaria para poder controlar la planta. Esta información está almacenada en una base de conocimiento que contiene un conjunto de reglas lingüísticas causa y efecto IF-THEN.

Las reglas borrosas se pueden definir como la combinación de uno o más conjuntos borrosos de entrada llamados antecedentes y a los que se les asocia un conjunto borroso de salida llamado consecuente (Farfan & Hoyos, 2008).

2.3.4 Defusificador.

Se trata del proceso inverso a la fuzzificación. El proceso de defuzzificación permite asociar a un conjunto difuso un valor numérico y se lleva a cabo para calcular el valor de salida de los modelos difusos. De hecho, el sistema de inferencia difusa obtiene una conclusión a partir de la información de la entrada, pero se expresa en términos difusos. Esta conclusión o salida difusa es obtenida por la etapa de inferencia borrosa, pero el dato de salida del sistema debe ser un número real y debe ser representativo de todo el conjunto; es por eso que existen diferentes métodos de defuzzificación y arrojan resultados distintos. El más común y ampliamente usado es el método del centroide (Aja Fernández., 2003).

2.3.4.1 Método del centro de área.

En este método se calcula el centro de gravedad del área limitada por la curva de la función de membresía, para ser el valor de salida representativo de la cantidad difusa. Entre las propiedades positivas de este método hay que destacar que es continuo, es decir que un pequeño cambio en las entradas no implica un cambio brusco en las salidas, y no es ambiguo, es decir que solamente se obtiene un solo valor como resultado del proceso. Sin embargo también este método cuenta con algunos inconvenientes como, no es plausible, ya que la salida puede corresponder a un punto del soporte del conjunto con bajo grado de activación, la principal limitación de este método es la necesidad de recorrer todo el universo de discurso para evaluar el centroide, esto hace que el cálculo del valor representativo sea muy costoso en número de operaciones (Montufar Villafuerte, 2012).

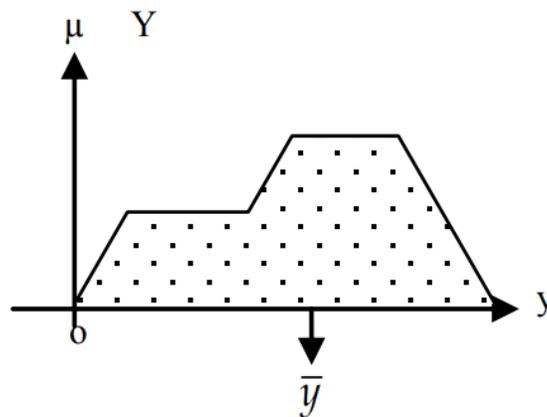


Figura. 9 Método del centro de área

2.3.4.2 Método del promedio de centros.

Es un método frecuentemente utilizado debido a su reducida complejidad comparada con el método anterior ya que es independiente del soporte del conjunto difuso inferido por cada regla. Este método toma en consideración solo el punto en que las funciones de pertenencia individuales de la variable de salida tienen valor igual a la unidad (Montufar Villafuerte, 2012).

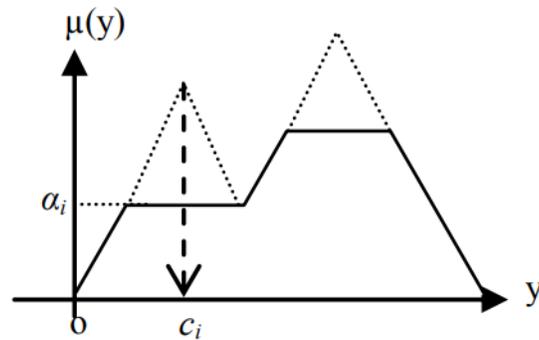


Figura. 10 Método del promedio de centros

2.4 Sistema Embebido.

Existen numerosas definiciones de sistemas embebidos algunas son:

Un sistema embebido es cualquier dispositivo que incluye un computador programable, pero en sí mismo no es un computador de propósito general (Wolf, 2008).

Un sistema embebido es un sistema electrónico que contiene un microprocesador o microcontrolador; sin embargo no pensamos en ellos como un computador (Morton, 2000).

Las personas usan el término de sistema embebido para referirse a cualquier sistema de cómputo escondido en algún producto o dispositivo (Simon, 1999).

Un sistema embebido es un sistema cuya función principal no es computacional, pero es controlado por un computador integrado. Este computador puede ser un microcontrolador o un microprocesador. La palabra embebido implica que se encuentra dentro del sistema general oculto a la vista y forma parte de un todo de mayores dimensiones (Wilmshurst, 2003).

Un sistema embebido debe tener hardware junto con software. Un sistema embebido tiene tres principales componentes: Hardware, Software y un Sistema Operativo.

Los sistemas embebidos poseen ciertas características que los distinguen de otros sistemas de cómputo a continuación se enlistan los más importantes (Vahid):

Funcionamiento específico. Un sistema embebido usualmente ejecuta un programa específico de forma repetitiva.

Fuertes limitaciones. Todos los sistemas de computación poseen limitaciones en sus métricas de diseño, pero en los sistemas embebidos son muy fuertes. Una métrica de diseño es una medida de algunas características de implementación, como: costo, tamaño, desempeño, y consumo de energía. Los sistemas embebidos generalmente deben ser poco costosos, poseer un tamaño reducido, tener un buen desempeño para procesar datos en tiempo real, y además consumir un mínimo de energía para extender el tiempo de vida de las baterías o prevenir la necesidad de elementos adicionales de enfriamiento.

Reactivos y tiempo real. Muchos sistemas embebidos deben ser reactivos o reaccionar ante cambios en el ambiente, además de realizar algunos cálculos en tiempo real sin ningún retraso, es decir, se deben tener resultados en tiempos fijos ante cualquier eventualidad. Por ejemplo, el módulo de control de viaje de un automóvil continuamente monitorea la velocidad y los sensores de frenos, reaccionando ante cualquier eventualidad. Ante un estímulo anormal, el módulo de control debe realizar los cálculos de forma precisa y acelerada para garantizar la entrega de los resultados dentro de un tiempo límite, una violación en este tiempo podría ocasionar la pérdida del control del automóvil. En contraste, un sistema de escritorio se enfoca en realizar cálculos con una frecuencia no determinada y la demora de los mismos no producen fallas en el sistema.

2.4.1 Microcontroladores.

Un microcontrolador es un circuito integrado que consiste en muchas de las mismas cualidades que una computadora de escritorio, tales como CPU (Central Process Unit), la memoria, etc., pero no incluye ningún dispositivo de comunicación con humanos (monitor, teclado, etc.).

Para resolver el problema de la complejidad y el alto costo de los sistemas basados en microprocesadores, se crean los microcontroladores, que no son otra cosa que un sistema de microcomputadora completo. Es decir, un microcontrolador contiene en un solo circuito integrado, el microprocesador, la memoria de datos, la memoria de programa y las unidades de entrada y salida, lo cual lo hace muy pequeño, barato, fácil de manejar, por lo que es ideal para muchas aplicaciones de propósito específico (Rossano).

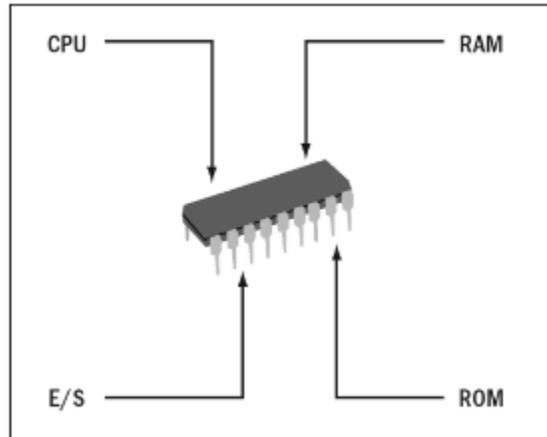


Figura. 11 Elementos de un microcontrolador.

Los microcontroladores son cada vez más utilizados en muchos campos debido a sus ventajas y la facilidad de diseñar circuitos con ellos, desde los más sencillos hasta las aplicaciones más complejas. Desde aficionados hasta profesionales, incluso a nivel industrial, cada vez son más los sistemas que se gobiernan por algún microcontrolador. Algunos ejemplos son, horno de microondas, lavadoras, juguetes, automóviles, etc. Actualmente puede resultar más fácil, rápido y más barato implementar un circuito electrónico con un microcontrolador que hacerlo con componentes discretos. Existen una gran gama de microcontroladores de muchos fabricantes con características variadas. En la actualidad los fabricantes más importantes son: Microchip, Freescale, Intel, Atmel.

2.4.2 Sensores.

Los sensores imitan la capacidad de percepción de los seres humanos, debido a esta característica podemos encontrar sensores relacionados con los diferentes sentidos, así como nuestro cerebro reacciona a la información que percibe de nuestros sentidos, los dispositivos que incorporan sensores reaccionan a la información que reciben de ellos. Los sensores son dispositivos electrónicos que nos permiten interactuar con el entorno, de forma que nos proporcionan información de ciertas variables que nos rodean para poder procesarlas y así generar ordenes o activar procesos (Serna Ruiz, Ros Garcia , & Rico Noruega, 2010) .

2.5 Programación en C.

El lenguaje C fue desarrollado en 1972 en los laboratorios Bell Telephone, su autor es un experto en software llamado Dennis Ritchie. Entre las ventajas de este lenguaje tenemos: lenguaje

de propósito general, no está diseñado para ningún tipo de software en particular, sino que puede ser empleado para realizar casi cualquier tipo de programas. Juego de operadores variado. Concisión de los programas, con pocas instrucciones es posible realizar muchas acciones. Los compiladores de C son rápidos y compactos. Portabilidad, Gracias el estándar ANSI en C, los programas escritos en C son muy fáciles de trasladar de un sistema a otro. Potencia de un ensamblador, aunque el lenguaje C es considerado un lenguaje de alto nivel, muchos autores consideran que es de un nivel medio, pues es posible trabajar con el como si se tratara del lenguaje ensamblador (Badenas Carpio, Llopis, & Coltell Simon, 2001).

Un programa en C está constituido por una sucesión de funciones independientes. De entre todas las funciones destaca la función main, que debe aparecer siempre en todos los programas y que es la primera función en ejecutarse. La primera función de la instrucción main es la primera instrucción del programa en ejecutarse.

2.6 Simulación.

Simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos periodos de tiempo (Coss Bu, 2003).

Simulación es el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso y conducir experimentos con este modelo con el propósito de entender el comportamiento del sistema o evaluar varias estrategias con las cuales se puede operar el sistema (Coss Bu, 2003).

Los pasos necesarios para llevar a cabo un experimento de simulación son: definición del sistema, formulación del modelo, colección de datos, implementación del modelo en computadora, validación, experimentación, interpretación y documentación.

2.6.1 PROTEUS.

El entorno de diseño electrónico PROTEUS VSM de LABCENTER ELECTRONICS (unvw.labcenter.co.uk) ofrece la posibilidad de simular código para el microcontrolador de alto y bajo nivel y, simultáneamente, con la simulación en modo mixto de SPICE, permite el diseño tanto a nivel hardware como software y realizar la simulación en un mismo y único entorno. Para

ello, se suministran tres potentes subentornos como son el ISIS para el diseño gráfico, VSM (Virtual System Modelling) para la simulación y el ARES para el diseño de placas (García Breijo, 2008).

ISIS es un potente programa de diseño electrónico que permite realizar esquemas que pueden ser simulados en el entorno VSM o pasados a un circuito impreso ya en el entorno ARES posee una muy buena colección de librerías de modelos tanto para dibujar, simular o para hacer PCBs. Además, permite la creación de nuevos componentes, su modelización para su simulación e incluso, la posibilidad de solicitar al fabricante que cree un nuevo modelo.

La característica más importante del PROTEUS VSM es la capacidad de depurar programas fuente de distintos lenguajes de programación. Además de aceptar el archivo de programación Intel Hex (HEX), también admite ficheros MRUBROF (D39), ByteCraft COD (COD), Microchip Compatible COF (COF) y Crownhill Proton Plus (BAS). Al utilizar estos archivos se puede abrir una ventana de código fuente llamada SOURCE CODE mediante la cual se puede seguir el programa fuente línea a línea de código.

Además, permite visualizar elementos internos del PIC como son la memoria de programa, la memoria de datos RAM o la EEPROM, los registros especiales (FSR) y la pila (Stack).

2.6.2 MATLAB.

El uso de herramientas para simular y entender un controlador difuso es esencial para trabajar con su diseño. (AKCAYOL, ELMAS, ERDEM, & KURT, 2004) presentan una herramienta para enseñar y comparar controladores de lógica difusa y controladores clásicos a estudiantes mediante el control de la velocidad de un motor. El objetivo de cualquier herramienta centrada en lógica difusa es desarrollar una escena real que relacione un pensamiento de ingeniería con un problema del mundo real. La mejor solución para escoger los parámetros de control es el método de prueba y error el cual puede hacerse en un ambiente de simulación.

MATLAB es una excelente herramienta para la simulación de controladores difusos, (ERENTURK, 2005) realiza dos ejemplos (Motor de corriente directa y regulador de voltaje automático) para mostrar la utilidad de la herramienta y poder visualizar los efectos de cambiar los parámetros del controlador difuso. Centra su atención en los efectos del número de reglas, el tipo de funciones de membresía y el número de salidas del controlador difuso. Menciona que la

relación entre la entrada y la salida es la parte más importante de un controlador de lógica difusa, esta relación deberá ser obtenida correctamente para manejar el funcionamiento del controlador difuso (Nguyen, Prasad, Walker, & Walker, 2003).

3 CAPITULO 3 Materiales y métodos.

3.1 Microcontrolador PIC18F4550.

Los microcontroladores de Microchip llamados PIC son los más vendidos en el mercado. Se caracterizan por tener un amplio número de familias, gran variedad de unidades funcionales embebidas (temporizadores, USART, I2C, SPI, PWM, Convertidores A/D, USB, Ethernet), precios competitivos y buen soporte por medio de diversas fuentes de información.

Los microcontroladores PIC existen en gamas de 8 bits, 16 bits y 32 bits. Dentro de la gama más simple de 8 bits se encuentra el microcontrolador PIC18F4550. Su memoria de programa, memoria RAM, número de entradas y salidas, número de canales analógicos y tipos de puertos de comunicación, han hecho de este PIC uno de los más utilizados para diversas aplicaciones.

Entre las características más importantes del PIC18F4550 se tiene: Esta basado en arquitectura RISC, uso de 75 instrucciones para programación, 40 pines de Hardware, máxima frecuencia de operación de 48Mhz, puede almacenar hasta 32,768 bytes en su programa base, cuenta con una memoria RAM de 2,048 bytes, una memoria EEPROM interna de 256 bytes, capaz de reaccionar a 21 diferentes interrupciones, 35 líneas de entrada y salida, 4 temporizadores, 1 módulo de PWM, canales de comunicación serie, 1 canal USB, 1 puerto paralelo de transmisión de datos, 13 canales de conversión A/D de 10 bits de resolución, 2 comparadores analógicos, entre otras muchas funciones.

40-Pin PDIP

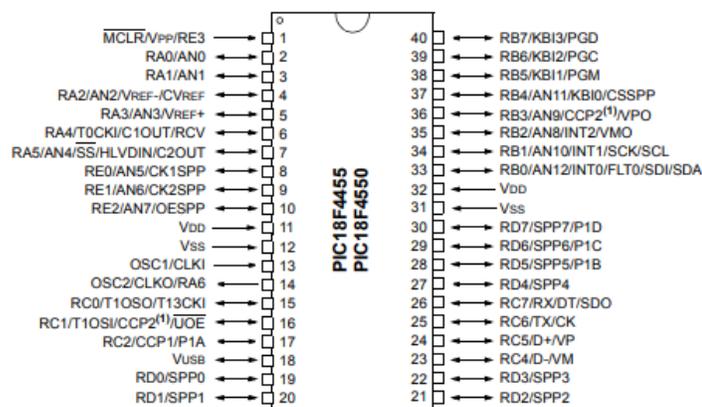


Figura 12 PINOUT PIC18F4550

3.2 Xbee.

Los Xbee son pequeños dispositivos que pueden comunicarse entre sí de manera inalámbrica. Son fabricados por Digi International, los cuales ofrecen una gran variedad de combinaciones de hardware, protocolos, antenas y potencias de transmisión.

Los XBee's son pequeñas radios que pueden comunicarse de forma inalámbrica unas con otras. Pueden hacer cosas simples, como reemplazar un par de cables en una comunicación serial. Estas radios además cuentan con entradas y salidas digitales y analógicas que pueden controlar e incluso pueden conectar sensores y leer sus mediciones en forma remota. Una gran ventaja de estas radios es su bajo consumo además de su posibilidad de realizar conexiones punto a punto, punto a multipunto y mesh. Existen dos tipos básicos de variantes de hardware: Xbee serie 1, no necesitan ser configurados, comunicación punto a punto. Xbee serie 2 comunicación punto a punto, punto multipunto, necesitan ser configurados.

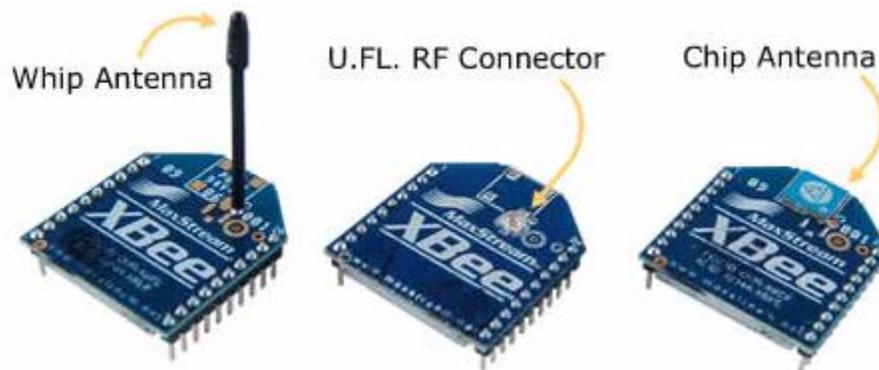


Figura 13 Tipos de antenas Xbee.

Hay pocas diferencias entre un XBee regular y un XBee PRO. La diferencia en cuanto a hardware es que el XBee PRO es un poco más largo. Con respecto a comunicación, la versión Pro tiene un mayor alcance (1,6 Km línea vista), pero para ello tiene un mayor consumo de potencia. El criterio para tomar la decisión de cual usar es la distancia que uno requiera comunicar dos XBee. Los dos modelos se pueden mezclar dentro de la misma red.

3.3 Sensor AM2302/x

Es un sensor capacitivo de humedad relativa y temperatura del aire, que posee una alta precisión, de bajo consumo y que puede transmitir información hasta 100m. Posee una señal de

salida digital calibrada, se puede energizar con un rango de voltaje de 3.3V a 5.5V, utiliza solo un pin como señal de salida, tiene un capacitor polimérico para sensor la humedad, puede trabajar en un rango de temperatura de -40°C a 80°C o 0% a 100% de humedad relativa, posee una precisión de $\pm 2\%$ de RH y $\pm 0.5^{\circ}\text{C}$.

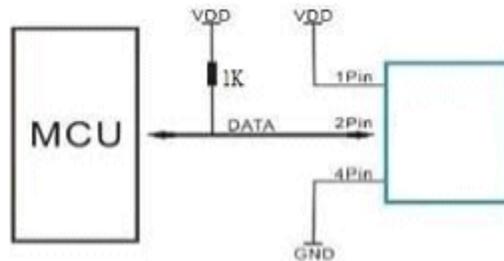


Figura 14 Diagrama de conexión eléctrica del DHT22

3.4 Sensor DS18B20.

Es un sensor programable de temperatura de un cable de Maxim Integrated. Es ampliamente usado en la medición de temperatura en ambientes hostiles como: soluciones químicas, minas o suelo. El sensor es muy resistente y también cuenta con una opción a prueba de agua. Puede medir un amplio rango de temperatura de -55°C a $+125^{\circ}\text{C}$ con una precisión $\pm 0.5^{\circ}\text{C}$. Cada sensor tiene una única dirección y requiere únicamente un pin del microcontrolador para transmitir información.



Figura 15 Vista física del sensor DS18B20

3.5 Sensor capacitivo de humedad del suelo.

Mide los niveles de humedad del suelo por medio de un sensado capacitivo. Esta hecho de materiales resistentes a la corrosión por lo que tiene un tiempo de vida útil muy aceptable. Incluye un regulador a la entrada que puede soportar voltajes desde 3.3V hasta 5.5V, lo que lo hace perfecto para trabajar con microcontroladores ya que operan al mismo nivel de voltaje. Su salida puede entregar valores de 0V hasta 3V dependiendo del valor de la humedad del suelo, este sensor debe ser calibrado antes de ser usado, la corriente de consumo máxima es de 5mA.



Figura 16 Sensor Capacitivo de Humedad del Suelo.

3.6 Pantalla LCD

Es una pantalla plana basado en el uso de una sustancia liquida atrapada entre dos placas de vidrio, haciendo pasar por este una corriente eléctrica a una zona específica, para que así esta se vuelva opaca. La pantalla LCD de 16 X 2 es ideal para utilizarse en proyectos de arduino o con microcontroladores PIC. Es de 16 caracteres y 2 líneas, el chip controlador de esta pantalla es extremadamente común y el código para controlar la pantalla es muy sencillo, se puede utilizar fácilmente con cualquier microcontrolador que tenga al menos 6 pines disponibles, su modo de operación puede ser de 4 o 8 bits y su voltaje de operación de 4.5V a 5.5V.



Figura 17 Pantalla LCD 16X02

3.7 Servomotor MG995.

Es un servomotor que es popular por su rendimiento y bajo precio. El motor es usado en muchas aplicaciones principalmente en robótica y drones. Este servomotor puede rotar aproximadamente 180°, 90° en cada dirección. El MG995 tiene tres terminales: un pin de señal, uno de energización y uno más de tierra. Es un motor a prueba de golpes, que posee una rotación de alta velocidad para una respuesta rápida, tiene un par constante en todo el rango de recorrido del servomotor, excelente poder de sujeción, un peso de 55gr, un voltaje de operación ente 4.8V a 7.2V, un torque máximo 11Kgr/cm, una velocidad máxima de 0.16s/60°, una operación de temperatura entre 0°C a 55°C, una corriente de consumo en inactivo de 10mA y una corriente a máxima cara de 1.2A.



Figura 18 Servomotor MG995

3.8 Otros materiales.

Para llevar al campo este proyecto además de los materiales anteriormente expuestos es necesario el uso de componentes secundarios que brindan una función específica y ayudan al buen funcionamiento del proyecto en condiciones hostiles que se pudieran presentar en el ambiente de trabajo, puesto que este modulo estará a la intemperie.

Así también es necesario considerar materiales y componentes para realizar las siguientes actividades o funciones:

- Energizar los módulos independientemente de la red eléctrica.
- Probar el buen funcionamiento de los módulos antes de ser llevados al campo.
- Montaje de los circuitos integrados en placas para realizar las diferentes interconexiones.
- Proteger los módulos por medio de una carcasa.
- Poner en marcha el proyecto en el entorno para el que fue creado.

Entre otros componentes podemos enlistar: tarjetas de desarrollo para el PIC18F4550, botones, resistencias, diodos, capacitores, protoboards, adaptadores para Xbee, pilas recargables, conectores, pinzas crimpadoras, cables, contenedores de agua, tubería, cargadores, paneles solares, materiales de plomería, fuentes de alimentación, etc.

3.9 Lista de materiales propuesta.

Tabla 5 Lista completa de materiales

Cantidad	Descripción	Función
1	Tarjeta de desarrollo PIC18F4550	Controlador difuso
1	Botón 12mm normalmente abierto	
1	Switch On/Off 16mm	
1	Adaptador para Xbee	
1	Display LCD 1602 Pantalla 16x2	
1	Batería Recargable 6V 4A de ácido sellada Sb-0604	
1	Conector de corriente hembra	
1	Digi Xbee S2C 3.1 mW 2.4g serie 2c antena wire	
1	PIC18f4550 Microcontrolador	
1	Celda Solar 12v 1.5w 125 mA	
1	Fuente para protoboard 5v/3.3v	
1	Tarjeta de desarrollo PIC18F4550	Electroválvula
1	Botón 12mm normalmente abierto	

1	Switch On/Off 16mm	
1	Adaptador para Xbee	
1	Batería Recargable 6V 4A de ácido sellada Sb-0604	
1	Conector de corriente hembra	
1	Digi Xbee S2C 3.1 mW 2.4g serie 2c antena wire	
1	PIC18f4550 Microcontrolador	
1	Servomotor MG995 Towerpro 15kg	
1	Panel celda solar 10w para sistema de 12v Epcom Pro1012	
1	Fuente para protoboard 5v/3.3v	
1	Tarjeta de desarrollo PIC18F4550	
1	Botón 12mm normalmente abierto	
1	Switch On/Off 16mm	
1	Adaptador para Xbee	
1	Display LCD 1602 Pantalla 16x2	
1	Batería Recargable 6V 4A de ácido sellada Sb-0604	
1	Conector de corriente hembra	
1	Digi Xbee S2C 3.1 mW 2.4g serie 2c antena wire	
1	PIC18f4550 Microcontrolador	
1	Módulo sensor temperatura humedad DHT22/AM2302	
1	Sensor de temperatura digital DS18B20	Interconexiones
1	Sensor capacitivo de humedad del Suelo	
1	Celda solar 12V 1.5W 125 mA 115x85x3 mm	
1	Fuente para protoboard 5v/3.3v	
1	SN01BM Terminal Crimpadora Herramienta Para Dupont Ph2.0 Xh2	
1	Cables Jumpers Dupont 120 Pzas (10 Cm) Hembra-Hembra	
1	Cables Jumpers Dupont 120 Pzas (10 Cm) Hembra-Macho	
1	Cables Jumpers Dupont 120 Pzas (10 Cm) Macho-Macho	
1	100 Pzas Terminales Dupont Macho	
1	100 Pzas Terminales Dupont Hembra	
1	20 Pzas Dupont Terminal Jumper Pin Hembra Conector 1	Otros
1	10 Pzas Dupont Terminal Jumper Pin Hembra Conector 3	
3	Conectores 80cm Nema 23-dupont. Arduino, Shield Cnc 3d	Pruebas
2	Cargador para pila de 6v con conector de zapatas	
1	Filamento ABS 1.75 Mm negro para impresora 3D	Puesta en marcha
1	Bote de plástico para agua con llave de 40 Litros	
1	Jardinera de plástico mediana 61cm de largo	
1	Tinaco 450 litros con accesorios	
1	Válvula de esfera (de Paso) de 1/2 marca Tiemme	
1	Base para tinaco 1100 litros	
1	20 metros de tuboplus 1/2.	
5	Niple galvanizado 1/2" X 19" Foset 47522	

5	Cople 1/2' galvanizado Foset 47550 Cg-201	
5	Codo Tubo 90 Grados Rosca Ext. 20 X 1/2 PuLG Tuboplus	
3	Codo Tubo 90 Grados Rosca Int. 32 X 3/2 PuLG Tuboplus	
2	Paquete de 10 Cinta de Teflon 3/4 Pretul 27011	

3.10 Métodos.

El objetivo principal de la ingeniería de control es aplicar un conocimiento específico acerca de cómo controlar un proceso.

3.10.1 Definir entradas y salidas del controlador difuso.

Para empezar un controlador difuso es necesario definir el número de entradas y salidas. En nuestro caso la salida es simplemente el abierto o cerrado de una electroválvula proporcional encargada de suministrar o no agua al sembradío. En el caso de las entradas un humano experto en temas agrícolas fue el encargado de definir si las entradas propuestas fueron necesarias y suficientes. El sistema fue diseñado para soportar 4 entradas, sin embargo, esto no significa que no pueda ser ampliado a un universo más grande de entradas. En base a las entradas definidas fue necesario crear una estación meteorológica capaz de medir estas variables y mandarlas al controlador difuso por medio de comunicación Xbee.

3.10.2 Recolectar conocimiento para crear la base de datos, definir variables lingüísticas y funciones de membresía.

Mediante el uso de un cuestionario (Ver Anexos), el especialista suministro la información necesaria para crear la base de datos (sentencias IF-THEN) que contiene el conocimiento del humano experto, así también con esta información se pudieron definir las variables lingüísticas y las funciones de membresía del sistema, obtenida esta información, posteriormente, fue necesario transcribir este conocimiento a lenguaje C para que el microcontrolador tuviera la capacidad de procesarla.

3.10.3 Obtener variables físicas y fusificar variables de entrada.

Aplicando los sensores expuestos en la sección de materiales y realizando el software correspondiente para su adecuado uso, fue posible tener una medición de las variables como humedad y temperatura, del suelo y del aire, posteriormente estas mediciones serán convertidas en valores difusos por medio de las funciones de membresía y del software desarrollado.

Se utilizó triángulos equiláteros como funciones de membresía de entrada/salida debido a su fácil análisis y comprensión para obtener el centro de gravedad y fusificar las variables. Para fusificar las variables de entrada fue necesario aplicar las formulas expuestas en la Tabla 3.

3.10.4 Determinar cuáles reglas usar.

Todas las reglas son comparadas con las entradas difusas (salidas del fusificador) del controlador para determinar cuáles se aplicarán en la situación actual. Este proceso de pareo involucra determinar qué "tan cierto" una regla aplica a la condicione actuales.

3.10.5 Método de inferencia. Determinar conclusiones.

Es preciso determinar que conclusión se alcanzara con las reglas que se han activado para decidir la magnitud de la señal de salida. Para hacer esto es necesario considerar las recomendaciones de cada regla de manera independiente. Luego se combinarán todas las recomendaciones de cada regla de la base que se hayan activado para determinar la señal de salida.

Se puede considerar el siguiente ejemplo:

Supóngase que la siguiente regla (1) se ha activado, con los valores difusos de HUMEDAD NORMAL igual a 0.25 y TEMPERATURA FRIO igual a 1.

(1) IF Humedad es NORMAL AND Temperatura es FRIO THEN Válvula es ABIERTO

Por lo tanto

$$\mu_{\text{premise}(1)} = \min \{0.25, 1\} = 0.25$$

Ecuación 4 Uso del operador min ejemplo 1

Esto quiere decir que se está 0.25 seguro de que la regla (1) aplica para la situación propuesta.

La justificación para usar el operador "min" es que no se puede estar "más seguro" acerca de la premisa de la regla.

Ahora considérese la conclusión alcanzada por otra regla (2) que se ha activado:

(2) IF Humedad es NORMAL AND Temperatura es MUY FRIO THEN Válvula es CASI ABIERTO

$$\mu_{\text{premise}(1)} = \min \{0.75, 1\} = 0.75$$

Ecuación 5 Uso del operador min ejemplo 2

Por lo tanto, se está 0.75 seguro que la regla 2 aplica para esta situación. Esto quiere decir que se está más seguro que la regla (2) aplica para esta situación que la regla (1).

Esto completa las operaciones del mecanismo de inferencia, mientras que la entrada del proceso de inferencia es el conjunto de reglas que están activas su salida es el conjunto de conjuntos difusos que representan las conclusiones alcanzadas.

3.10.6 Convertir decisiones en acciones.

Ahora considérese la operación de defusificación que es la salida del controlador difuso. La defusificación opera sobre los conjuntos difusos implicados en el mecanismo de inferencia y combina sus efectos para proveer la salida más adecuada (entrada de la planta).

Debido a que es uno de los más usados para la defusificación se escogió al centro de gravedad (COG) como el método de defusificación para este proyecto.

Considere el siguiente ejemplo basado en suposiciones para entender el método de defusificación:

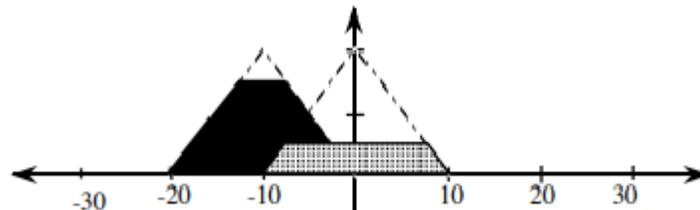


Figura 19 Conjuntos difusos de salida implicados

$b_1 = 0.0$ (Centro de la base del conjunto difuso de la derecha)

$b_2 = -10$ (Centro de la base del conjunto difuso de la izquierda)

$$\int \mu_{(i)}$$

(Expresión para denotar el área bajo la función de membresía resultante de la combinación de todas las reglas activadas)

$$u = \frac{\sum_i b_i * \int \mu_{(i)}}{\sum_i \int \mu_{(i)}}$$

Ecuación 6 Formula para calcula el centro de gravedad

Para poder utilizar prácticamente la ecuación anterior es necesario tener en consideración los siguientes puntos: No es posible tener funciones de membresía de salida que tengan un área infinita, se debe tener especial cuidado al definir las funciones de membresía de entrada y de salida para que la suma en el denominador de la Ecuación 6 no sea igual a 0, si se utilizan funciones de membresía de salida de triángulos simétricos es posible calcular la Ecuación 6 por medio de la Ecuación 7.

$$\int \mu_{(i)} = w(h - \frac{h^2}{2})$$

Ecuación 7 Fórmula para calcular $\int \mu_{(i)}$ si las funciones de membresía de salida son triangulos simetricos.

Usando la Ecuación 6 con la figura 19 se obtiene el siguiente resultado:

$$COG = \frac{(0)(4.375) + (-10)(9.375)}{4.375 + 9.375} = -6.81$$

Nótese que los valores 4.375 y 9.375 es el área sombreada de los triángulos de la figura 19 calculadas por medio de la Ecuacion7.

De acuerdo a lo expuesto en los temas 3.10.2 hasta 3.10.6 fue necesario crear el código del controlador difuso capaz de recibir datos de la estación meteorológica así mismo como enviar el resultado del procesamiento a la electroválvula.

3.10.7 Creación y control de la electroválvula.

Las electroválvulas proporcionales comerciales no se caracterizan por ser económicas, por lo que se decidió hacer una electroválvula proporcional que no sobrepasa el 10% del valor comercial de una prefabricada, además cuenta con una comunicación Xbee y una alimentación independiente a la red eléctrica.

La base física de la electroválvula fue hecha gracias a una impresora 3D la cual brinda soporte para un servomotor y una válvula de esfera, los cuales constituyen las partes físicas de la electroválvula más el pensamiento lógico proveído por un microcontrolador.

El servomotor permite un rango de acción de 180° , sin embargo, no es necesario utilizar esta característica ya que la válvula de esfera solo permite un rango de acción de 90° . El microcontrolador recibe el valor del COG del controlador difuso por medio de comunicación Xbee para después por medio de una regla de tres simple calcular el ángulo a la que se debe de abrir la válvula.

Cualquier falla en los sensores o en la alimentación de la estación meteorológica o el controlador difuso provocaran que la electroválvula se cierre.

3.10.8 Simulación en proteus.

De manera paralela a los pasos anteriormente expuestos, fue necesario hacer una simulación en Proteus para asegurarse del comportamiento adecuado de los diferentes microcontroladores con su respectivo software y componentes electrónicos relacionados.

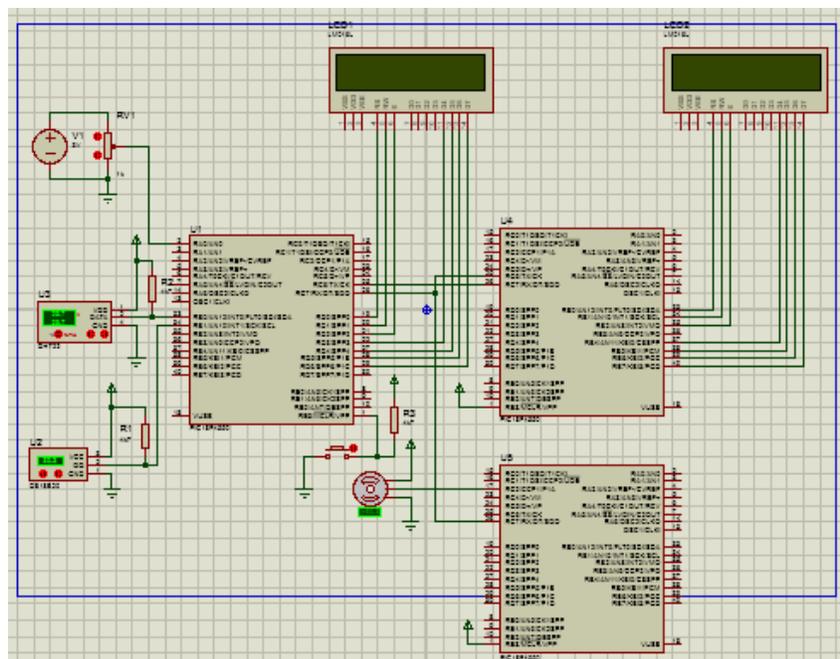


Figura 20 Simulación en PROTEUS de la estación meteorológica, controlador difuso y servomotor

3.10.9 Simulación en MATLAB.

MATLAB contiene una herramienta llamada Fuzzy Logic Designer que permite que sistemas basados en lógica difusa puedan ser simulados mediante la definición de entradas, conjuntos difusos, reglas, métodos y salidas.

Para la simulación del sistema de riego, fue necesaria la definición de 4 entradas (Temperatura aire, temperatura tierra, humedad aire y humedad tierra) cada una conformada por 3 conjuntos difusos con su respectiva etiqueta. Al tener 4 entradas y cada una de ellas 3 conjuntos difusos, el conjunto total del número de reglas posibles es 81, de estas, 54 fueron definidas por el humano experto, las restantes fueron consideradas para tener un efecto de salida de válvula cerrada. El método de inferencia y defuzzificación utilizado en la simulación es el de Mamdani y centroide respectivamente. Se considera una salida con dos conjuntos difusos y una condición de válvula cerrada.

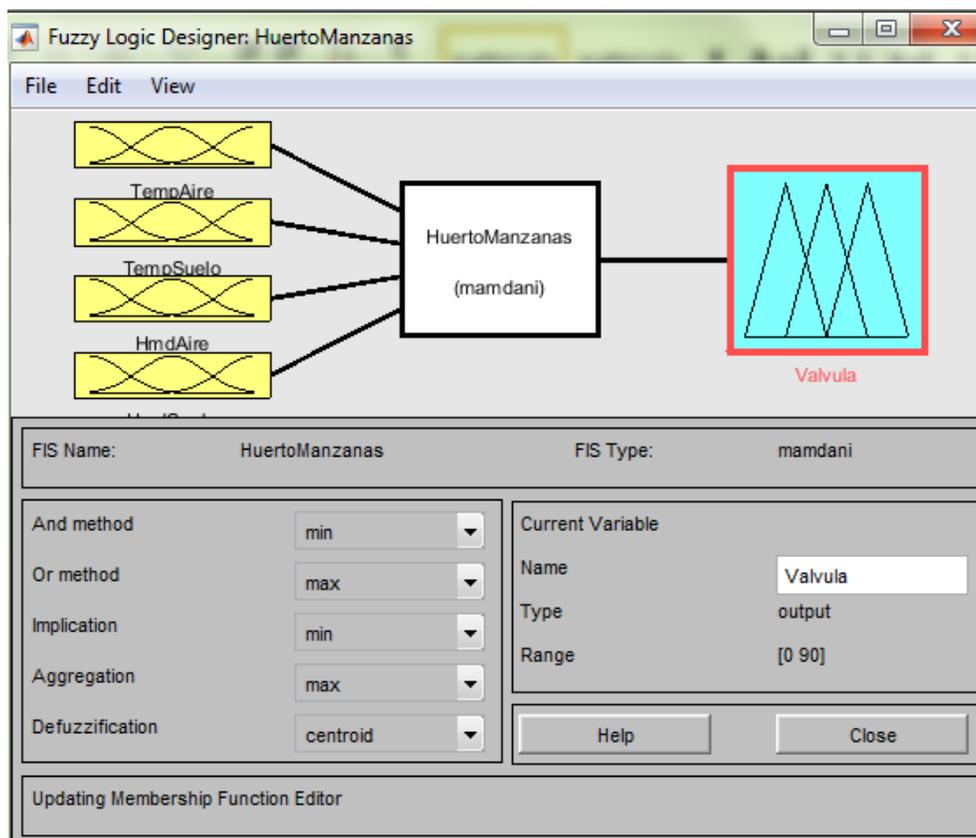


Figura 21 Sistema difuso en Fuzzy Logic Designer.

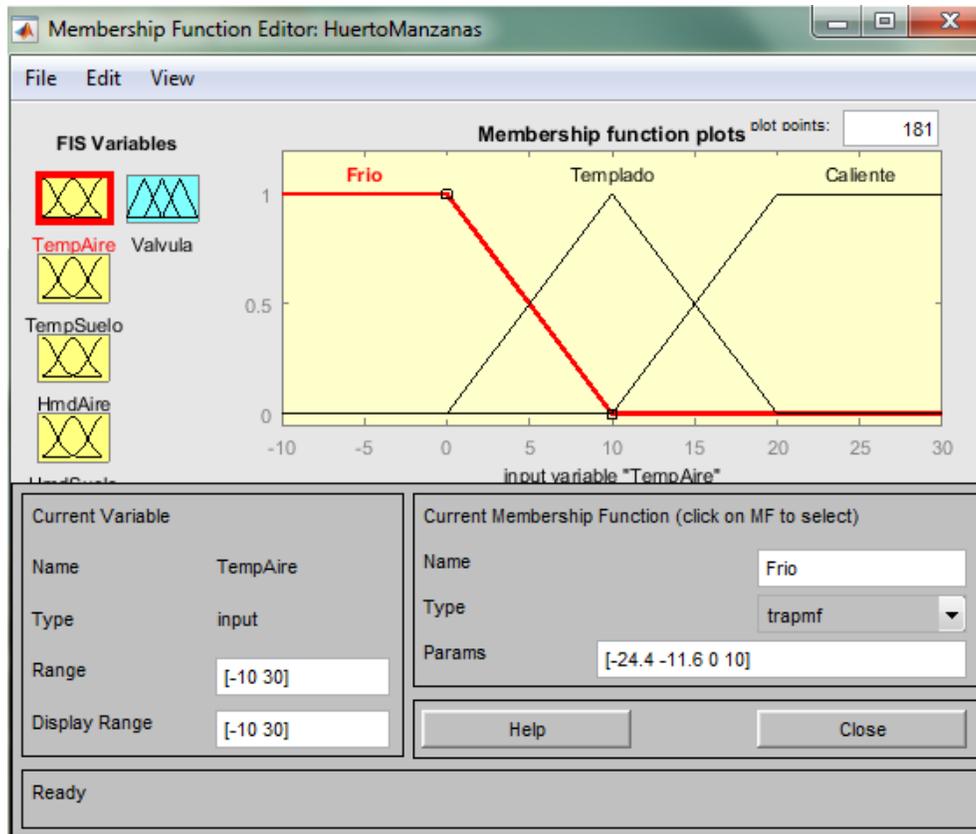


Figura 22 Entrada TempAire con sus respectivos conjuntos difusos.

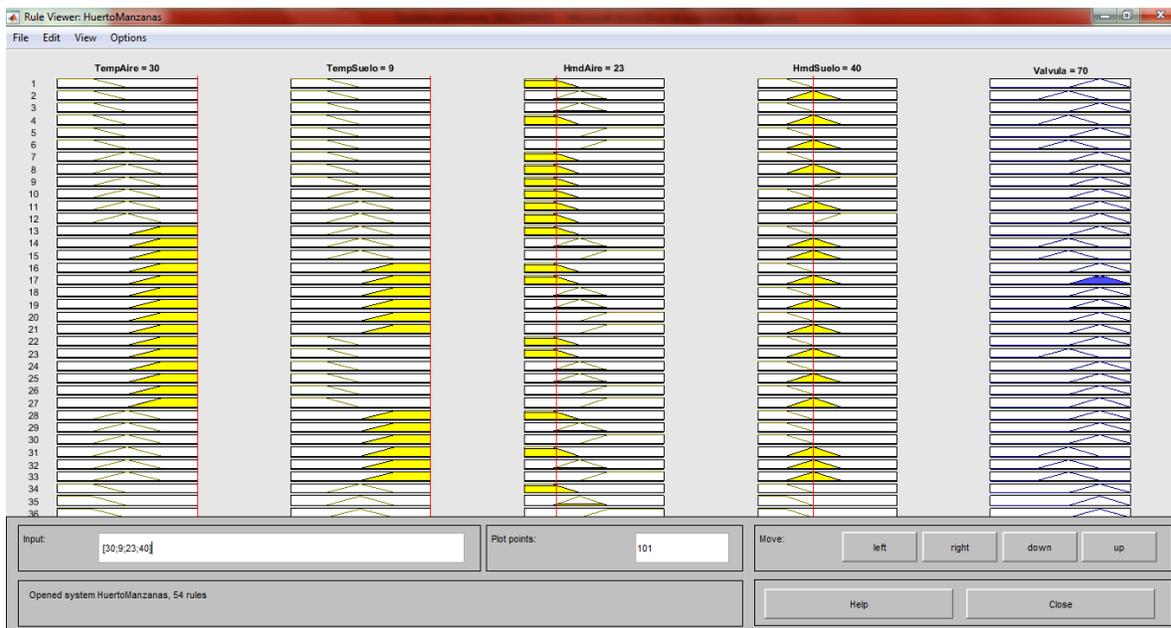


Figura 23 Simulación en MATLAB de una condición aleatoria

3.10.10 Pruebas en protoboard.

El sistema antes de ser llevado a su respectiva carcasa fue probado mediante el uso de protoboard, con la realización de este paso se buscó que el sistema fuera probado antes de armar el sistema que será instalado en el huerto de manzanas.

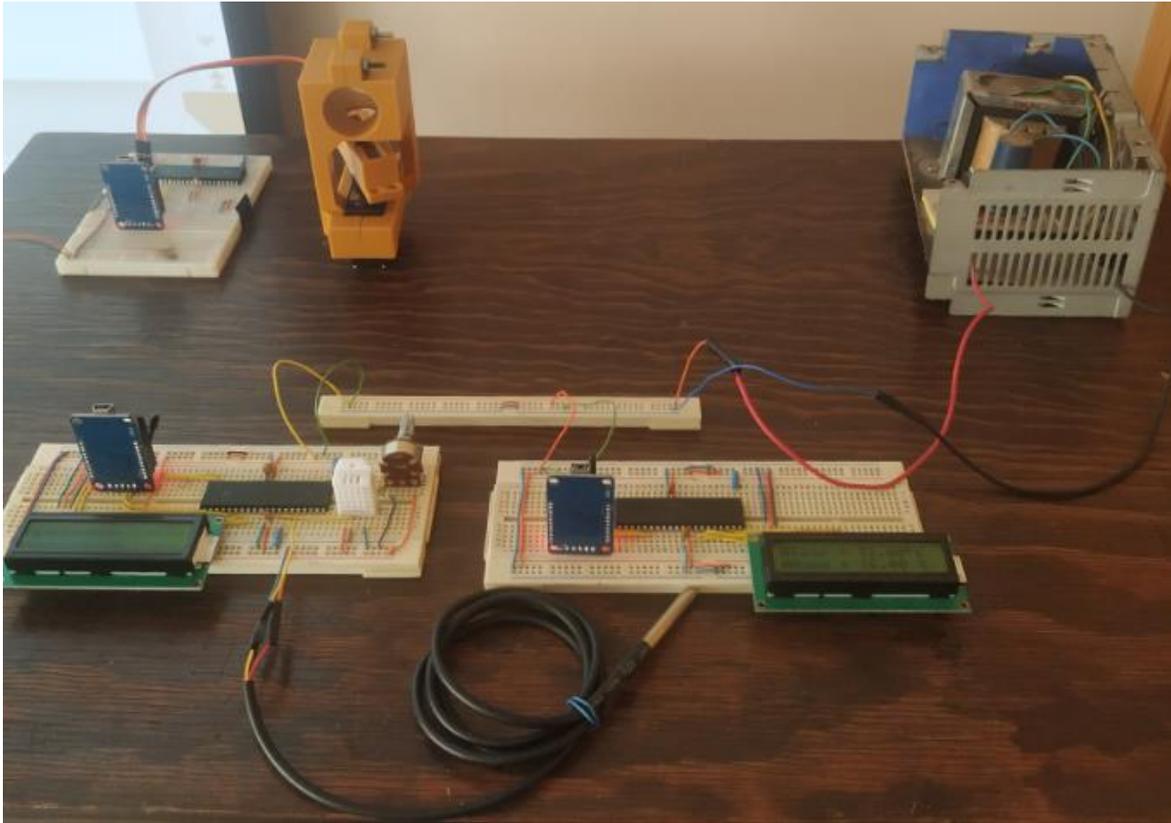


Figura 24 Prueba en Prototboard

3.10.11 Creación del sistema de alimentación.

Puesto que el sistema estará en un lugar donde el acceso al servicio eléctrico no es posible se tuvo que diseñar un sistema básico de alimentación para proveer al sistema de energía. El sistema solar de alimentación consta de una celda solar que proporciona energía al sistema embebido durante el día, por la noche una batería que fue cargada por la celda solar se activa y permite que el sistema siga funcionando durante algunas horas. El sistema es capaz de auto-reiniciarse al siguiente día si por algún motivo se quedara sin energía.

4 CAPITULO 4 Resultados y discusión.

Los resultados de la evaluación del comportamiento dinámico del controlador difuso de riego son mostrados en este capítulo. Se analizaron los efectos de la temperatura ($^{\circ}\text{C}$) y humedad relativa del aire (%), temperatura ($^{\circ}\text{C}$) y humedad del suelo (%) en el desempeño del sistema de control.

Se utilizó software de simulación para resolver el sistema de control difuso a lazo cerrado, se realizaron una serie de pruebas con entradas predeterminadas para garantizar la estabilidad del sistema de control.

La información proporcionada por el especialista en irrigación permitió definir tres conjuntos difusos y sus etiquetas lingüísticas para cada una de las cuatro variables de interés, mostradas en la siguiente tabla.

Tabla 6. Variables de Entrada.

	Variables de entrada			
	(Hrs %)	(Hra %)	(Ts $^{\circ}\text{C}$)	(Ta $^{\circ}\text{C}$)
Etiqueta Rango	PocaHumedad Hrs < 40%	PocaHumedad Hra < 40%	Frio Ts < 3 $^{\circ}\text{C}$	Frio Ta < 10 $^{\circ}\text{C}$
	Normal 20% \leq Hrs \leq 60%	Normal 20% \leq Hra \leq 60%	Normal 0 $^{\circ}\text{C}$ \leq Ts \leq 6 $^{\circ}\text{C}$	Templado 0 $^{\circ}\text{C}$ \leq Ta \leq 20 $^{\circ}\text{C}$
	MuchaHumedad Hrs > 40%	MuchaHumedad Hra > 40%	Caliente Ts > 3 $^{\circ}\text{C}$	Caliente Ta > 10 $^{\circ}\text{C}$

De un universo de 81 reglas, para la construcción de la base de datos se seleccionaron 54 como las más significativas para representar el conocimiento del especialista en irrigación.

La simulación del controlador difuso se realizó mediante Fuzzy Logic Designer de MATLAB, esta herramienta permite que sistemas basados en lógica difusa puedan ser simulados mediante la definición de entradas, conjuntos difusos, reglas, métodos de inferencia y salidas continuas.

El algoritmo de control mostrado a continuación se desarrolló en lenguaje C.



Figura 25. Inicialización del sistema de riego.

La Figura 25 muestra el diagrama de flujo para la inicialización del sistema de riego, se define tipo y frecuencia del oscilador del microcontrolador, se habilita el watchdog, la interrupción global, interrupción RS232 y la pantalla LCD. Para evitar que el microcontrolador se reinicie se restablece el watchdog después la pantalla LCD muestra el mensaje “MANDANDO ACTIVACION” y se produce un retardo de 500ms. Se envía una señal a la estación meteorológica con el objetivo de activarla e iniciar la recepción de las variables utilizadas en el sistema de control, si la estación no responde el microcontrolador se cicla hasta que reciba una respuesta positiva por parte de la estación meteorológica.

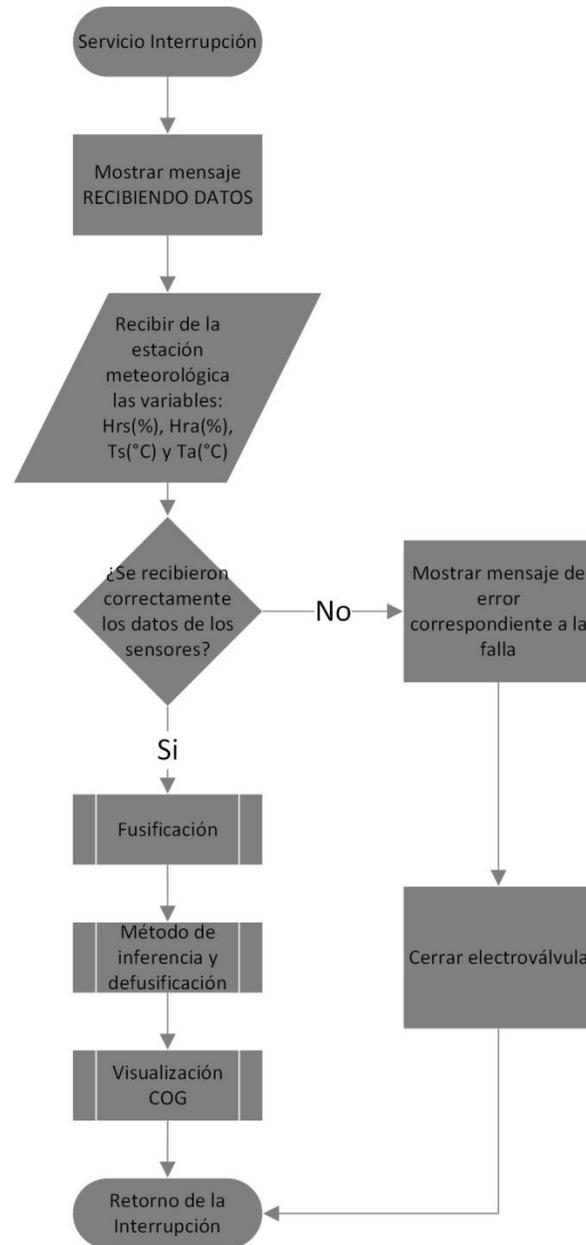


Figura 26. Servicio de Interrupción.

Una vez recibida la interrupción lo cual significa que la estación meteorológica se encuentra en operación, el controlador difuso muestra el mensaje “RECIBIENDO DATOS”, empieza a recibir datos de la estación meteorológica por medio de dispositivos Xbee de las variables Hrs(%), Hra(%), Ts(°C) y Ta (°C) estas son almacenadas en la memoria del microcontrolador, fusificadas y aplicadas en el método de inferencia Mandami, para después obtener el centro de gravedad. Estos procesos se encuentran definidos en las

subrutinas “Fusificación”, “Método de inferencia y defusificación” y “Visualización COG”, al finalizar estas subrutinas el microcontrolador regresa al ciclo principal del programa.

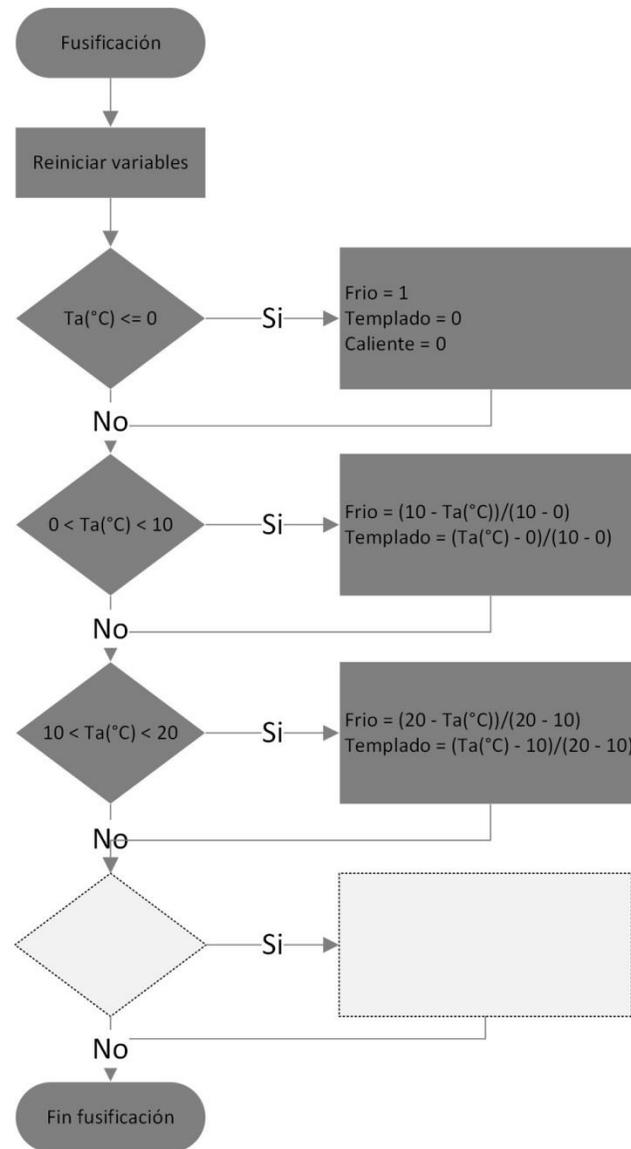


Figura 27. Fusificación de la variable Ta.

El diagrama de flujo de la Figura 27 muestra el proceso de fusificación de la variable Ta, la fusificación de las variables restantes sigue un proceso similar. Se restablecen los valores de las etiquetas lingüísticas que intervienen en la subrutina de “Fusificación”, si Ta es menor o igual que 0°C, significa que el valor difuso de la etiqueta Frio es igual a uno, de la etiqueta Templado es igual a cero y la etiqueta caliente es igual a cero, si Ta cae dentro del rango de 0°C a 10°C, se aplica la

formula indicada en el diagrama de flujo. El objetivo de esta subrutina es definir el valor difuso para las etiquetas lingüísticas de las variables Hrs(%), Hra(%), Ts(°C) y Ta (°C).

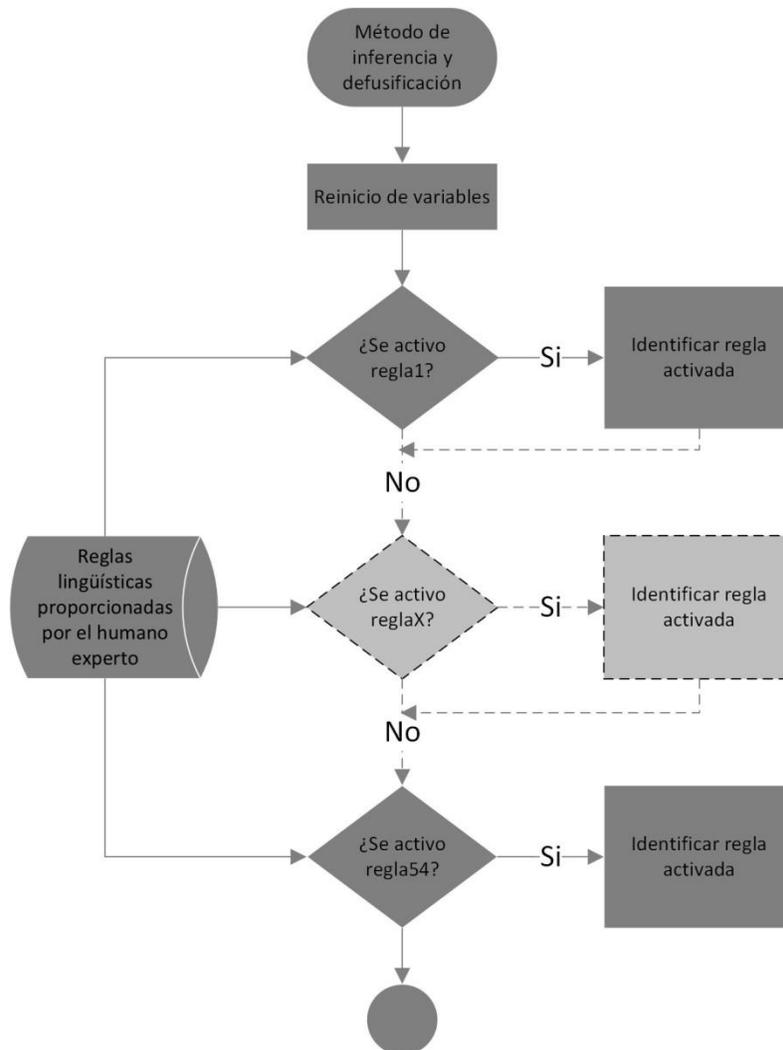


Figura 28. Método de inferencia y fusificación.

Una vez fusificadas las variables de entrada, es preciso utilizar el método de inferencia Mandami, para posteriormente defusificar la función de membresía de salida entregada por el método de inferencia. Como se muestra la Figura 28 para aplicar el método de inferencia tipo Mandami, se requiere identificar cuantas y cuales reglas lingüísticas proporcionadas por el humano experto están activadas de acuerdo a las condiciones atmosféricas actuales en el huerto de manzanos.

En la Figura 29 se muestra el prototipo del sistema del controlador difuso, se observan los diferentes dispositivos como: el microcontrolador que contiene el algoritmo de control difuso, la pantalla LCD donde se exhiben los valores de las variables de entrada así como el centro de

gravedad calculado, la placa y la batería de alimentación para energizar los diferentes componentes y un dispositivo Xbee que permite una comunicación serial con los demás subsistemas, así como las diferentes conexiones entre estos componentes.

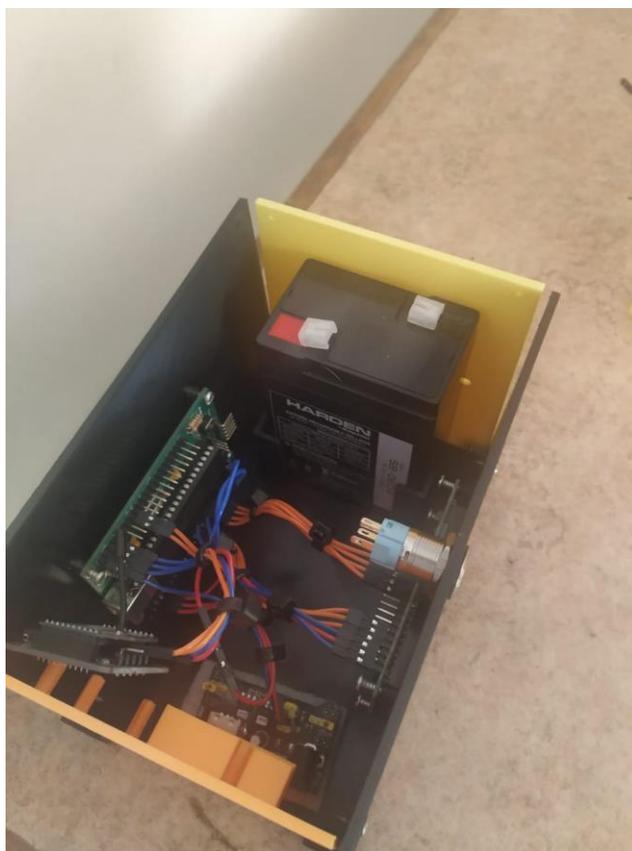


Figura 29. Sistema embebido controlador difuso.

La integración de los subsistemas, etapa de sensado y acondicionamiento de señales de interés, controlador difuso y etapa de actuación se pueden observar en la Figura 30.



Figura 30. Etapa de sensado y etapa de actuación

El actuador del sistema de control se compone de un motor a pasos y una válvula insertados en una base elaborada en un programa de diseño mecánico y fabricada en una impresora 3D la cual permite girar el vástago de la válvula de acuerdo al valor del centro de gravedad enviado por el controlador difuso, el actuador se muestra en la Figura 31.



Figura 31. Electroválvula

Para el análisis estadístico del comportamiento del controlador difuso se generó una base de datos compuesta por las mediciones de las cuatro variables de interés, donde Hrs(%)

representa la variable dependiente o respuesta, mientras que Hra(%), Ta(°C) y Ts(°C) representan variables independientes.

La Figura 32 muestra el histograma de los datos de Hrs(%) donde se distinguen dos distribuciones que indican diferentes dinámicas del comportamiento de la variable, la primera distribución corresponde a la condición previa a la aplicación de la acción de control y se caracterizó por valores menores de humedad mientras que la segunda distribución es posterior a la implementación del controlador difuso el cual incrementó los valores de humedad del suelo. Para distinguir ambas acciones de control se implementó el algoritmo de separación de Otsu, el valor de separación de Otsu para Hrs(%) es de 93% indicado por la línea vertical roja.

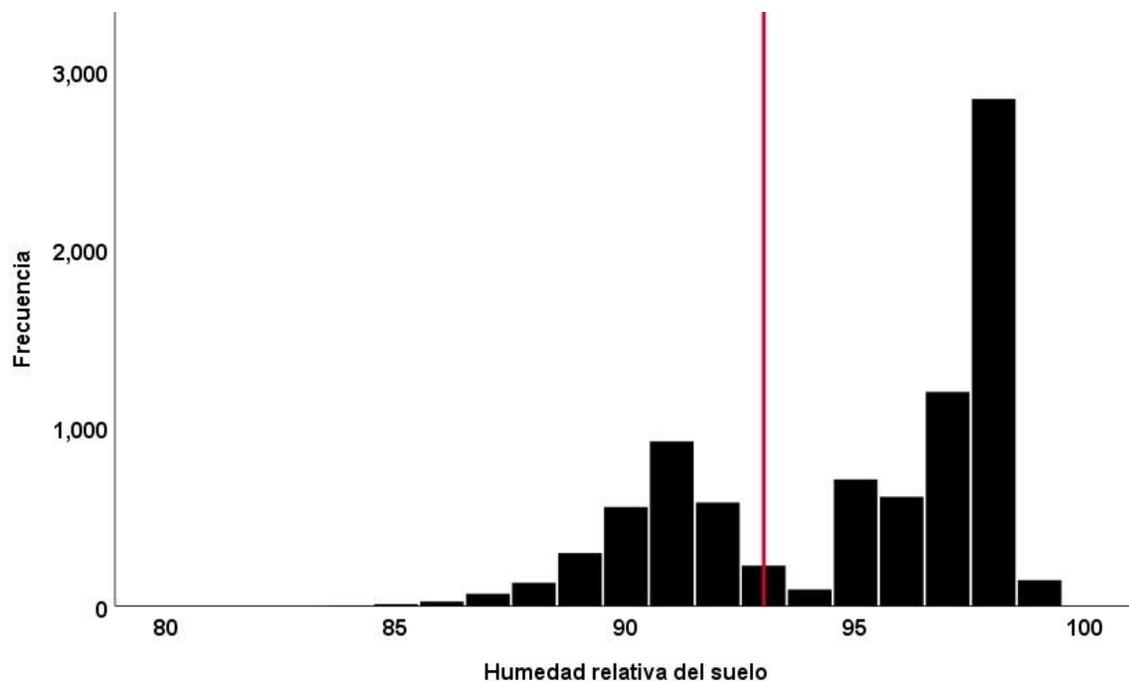
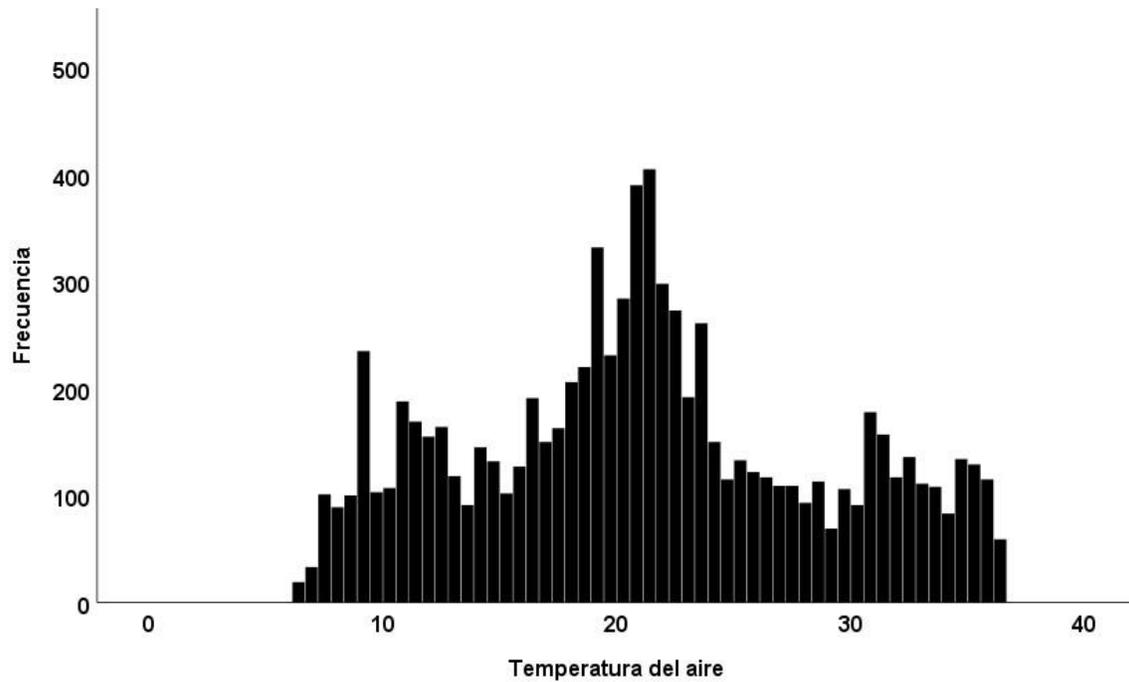
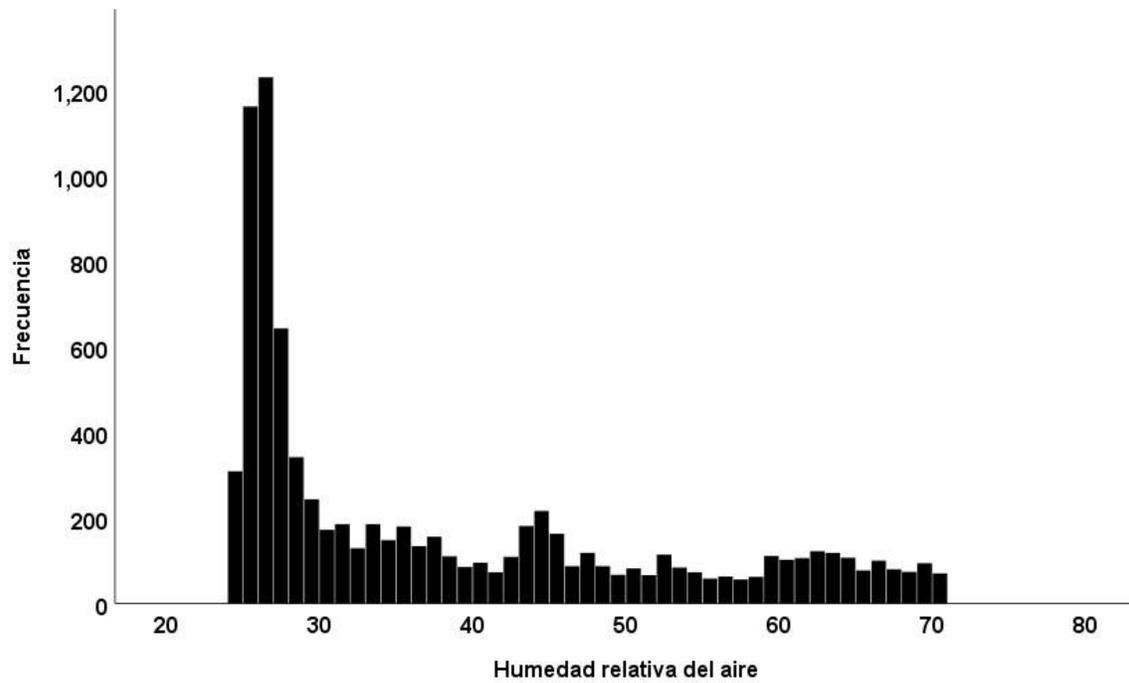


Figura 32. Histograma de distribución de los porcentajes de humedad del suelo.

En tanto las variables independientes, muestran los siguientes comportamientos, Hra(%) registró lecturas de valores menores al 40% de humedad, esto debido a la escasa presencia de lluvias en las fechas en que se realizaron las mediciones Figura 10(a). La temperatura ambiente, registró una distribución trimodal, correspondiente a las mediciones nocturnas, diurnas y vespertinas Figura 10(b). La temperatura del suelo presentó una distribución bimodal, en donde la

primera se caracterizó por valores de temperaturas de aproximadamente 18°C, en tanto que en la segunda distribución la temperatura se incrementó por encima de los 19°C Figura 10(c).



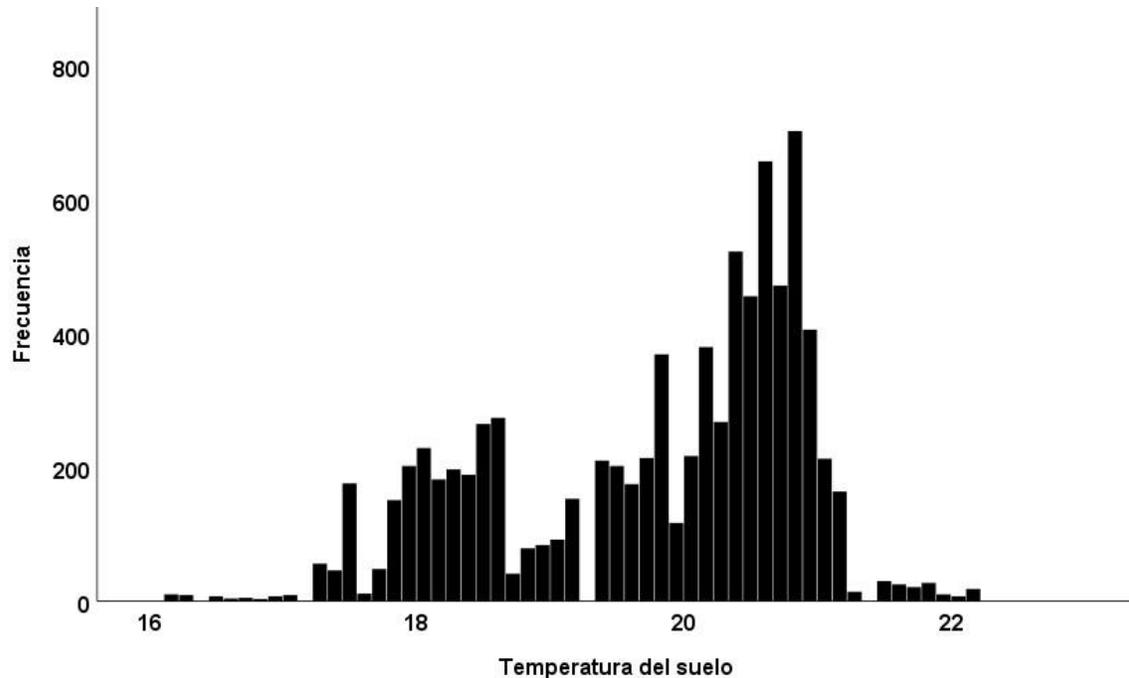


Figura 33. Histogramas de distribución de las mediciones de las variables independientes: a) Humedad relativa del aire, b) Temperatura del aire, c) Temperatura del suelo.

Las características de los dos grupos para determinar la acción de control se resumen en la Tabla 3. Destacan las variables Hrs(%) y Hra(%), cuyas diferencias en los promedios son mayores para el grupo donde la acción de control está presente 97.15% y 39.49% respectivamente en contraste con el grupo donde la acción de control no está presente 90.65% y 34.88%. La diferencia de promedios de las variables $T_a(^{\circ}\text{C})$ y $T_s(^{\circ}\text{C})$ son del orden de un punto porcentual entre ambos grupos, sin embargo, destaca $T_s(^{\circ}\text{C})$ en el grupo de acción de control presente, al tener un rango de temperaturas entre 16.2°C y 21.3°C, contra temperaturas de 19°C y 22°C cuando la acción de control no está siendo aplicada.

Tabla 7. Estadísticos descriptivos de las variables medidas según acción de control.

Acción de control	Mínimo	Máximo	Promedio	Desviación estándar
NO				
Hrs(%)	84	93	90.65	1.49
Hra(%)	24.61	70.74	34.88	14.33
Ta(°C)	6.54	36.33	20.59	7.67
Ts(°C)	18.98	22.15	20.57	0.51
SI				
Hrs(%)	94	99	97.15	1.16
Hra(%)	24.61	70.74	39.49	13.35
Ta(°C)	6.45	36.33	21.43	7.48
Ts(°C)	16.17	21.33	19.39	1.16

Como puede observarse en la Tabla 8, Hrs(%) es la variable afectada por la acción de control al incrementarse de 90.65% al 97.15% de humedad. Esto es el resultado del encendido del actuador que el sistema controla para mantener Hrs(%) dentro de los valores establecidos en la base de reglas lingüísticas definida por el experto en irrigación. Por su parte, Hra(%) registró un promedio de 39.5% durante la acción de control que previo a este fue de 34.9%, en tanto que Ts(°C) registró valores menores cuando se aplicó la acción de control.

Para determinar la existencia de un espacio de separación de las respuestas de las variables previas a la acción de control y una vez aplicado la acción de control, se realizó un análisis discriminante.

Tabla 8. Estadísticos descriptivos por grupos según acción de control.

Acción de control	Variable	Promedio	Desviación Estándar	Unweighted
NO	Hra(%)	34.88	14.33	2821
	Ta(°C)	20.59	7.67	2821
	Ts(°C)	20.57	0.51	2821
SI	Hra(%)	39.49	13.35	5607
	Ta(°C)	21.43	7.48	5607
	Ts(°C)	19.39	1.16	5607
Total	Hra(%)	37.95	13.86	8428
	Ta(°C)	21.15	7.55	8428
	Ts(°C)	19.78	1.14	8428

La prueba de hipótesis de diferencia entre los promedios de los grupos de acción de control vs no acción de control para cada una de las variables independientes se muestra en la Tabla 9. En los tres casos, los $p - valor \ll 0.0001$, por lo que existe una diferencia significativa a un nivel de significancia de $\alpha = 0.05$ entre los grupos de no acción de control y el de acción de control.

Tabla 9. Prueba de hipótesis de igualdad de los promedios de las variables.

Variable	Wilks' Lambda	F	Grados de libertad 1	Grados de libertad 2	p-valor
Hra(%)	0.975	213.678	1	8426	0
Ta(°C)	0.997	22.973	1	8426	0
Ts(°C)	0.76	2661.037	1	8426	0

Por otra parte, la fuerza de la relación entre las variables independientes (predictores) y la variable dependiente que en este caso es una variable dicotómica que representa si la acción de control está presente o no, está dada por el coeficiente de correlación canónica, que en este caso es igual a 0.520, en tanto que el valor del estadístico de Wilk's Lambda = 0.729, lo cual indica el poder de predicción de la función discriminante. En este caso, se obtiene un $p - valor < 0.0001$, que significa que el modelo hace predicciones estadísticamente significantes entre los grupos.

Analizando los coeficientes la Tabla 10 se puede distinguir que la variable de temperatura del suelo es la que mayor poder predictivo le genera al modelo de discriminante (0.924), en tanto que la Hra(%) es de un valor de (-0.466) y Ts(°C) es de un valor de (-0.310) ofrecen poder

predictivo similar. Los coeficientes de la matriz de estructura producen resultados similares en relación a la importancia de cada coeficiente, lo cual sustenta la validez del modelo discriminante.

Tabla 10. Coeficientes estandarizados de la función canónica discriminante.

Variable	Función Canónica	Matriz de estructura
Hra(%)	-0.466	-0.261
Ta(°C)	-0.31	-0.086
Ts(°C)	0.924	0.922

Las mediciones registradas por ambos eventos de interés, definidas como ausencia o presencia de la señal de control fueron utilizadas por el modelo empleando un esquema de validación cruzada. La predicción de la acción de control resulta más exitosa al alcanzar 84% de casos detectados en este grupo, en tanto que la clasificación dentro del grupo de no acción de control resultó menos efectiva 65.6%. En promedio, el porcentaje de casos clasificados correctamente fue de aproximadamente 78%, lo cual es un indicador de que existe separabilidad de las acciones de control.

Tabla 11. Clasificación de los resultados usando la función discriminante (validación cruzada).

Acción de Control	Grupo de Predicción		Total de mediciones
	NO	SI	
NO	65.6	34.4	2821
SI	16	84	5607

5 CAPITULO 5 Conclusiones y recomendaciones.

Sería muy conveniente poder observar el comportamiento del controlador difuso a distancia, así mismo como las variables reales captadas por la estación meteorológica. Sin embargo, el campo de cultivo no cuenta con una señal de internet o es demasiado débil, por lo tanto si se quisiera llevar a cabo esta parte del proyecto sería necesario contemplar routers para obtener una buena señal de internet.

Para medir de manera eficiente las diferentes variables en el campo de cultivo de 3.5Ha es conveniente la implementación de varios sensores colocados en áreas específicas propuestas por

el especialista en irrigación, estos sensores deberán estar conectados en red y transmitir en comunicación Xbee mediante dispositivos LORA la información al controlador difuso. El crear una red de sensores en el campo de cultivo presenta un tiempo considerable que no está contemplado para el desarrollo del presente proyecto.

Se diseñó y construyó un sistema embebido que contiene un algoritmo de control basado en lógica difusa para la dosificación de agua de riego aplicado a un huerto de manzanos, el cual evalúa en tiempo real el grado de apertura de una electroválvula para suministrar la cantidad de agua necesaria mediante la medición de tres variables independientes de interés humedad relativa del aire Hra (%), temperatura del suelo Ts (°C) y temperatura del aire Ts (°C), en conjunto con una base de reglas lingüísticas proporcionada por el humano experto en irrigación, para controlar una cuarta variable de intereses humedad relativa del suelo Hrs (%).

La instalación del sistema embebido permitió optimizar el recurso hídrico, al medir las diferentes variables que son importantes para la irrigación y al simular la experiencia obtenida por el especialista, esto se evidencia por el análisis estadístico de los datos que muestran que hubo un incremento de la humedad relativa de suelo Hrs (%) al pasar del 90.65% de humedad al 97.15%.

6 REFERENCIAS

- Aja Fernández., S. (2003). Un nuevo marco matricial para la implementación de inferencia borrosa. *Universidad de Valladolid, Escuela Técnica Superior de Ingenieros de Telecomunicacion.*
- AKCAYOL, M., ELMAS, C., ERDEM, O., & KURT, M. (2004). An educational tool for fuzzy logic controllers and classical controllers . *Wiley Periodicals Inc*, 129-135.
- Badenas Carpio, J., Llopis, B. J., & Coltell Simon, O. (2001). *Cursos practico en programacion en c y c++*. Universitat Jaume.
- Bin Poyena, F., Roy, S., Ghosh, A., & Bandyopadhyay, R. (2015). Automated Irrigation by an ANN Controller. *Procedia Computer Science*, 257-267.
- Coss Bu, R. (2003). *Simulacion un enfoque pratico*. Monterray: Limusa.
- Cuatodio, A., & Granados, A. (2016). *Controlador difuso de procesos industriales en un microcontrolador* . Academica Española.
- D´Negri, C., & De Vito, C. L. (2006). Introducción al razonamiento aproximado: lógica difusa. *Revista Americana de Medicina Respiratoria*, vol. 6, núm. 3, 126-136.
- Delgadillo Iriarte , O. (2000). ALGUNOS APUNTES CONCEPTUALES SOBRE LOS METODOS Y TIPOS DE RIEGO CAMPESINO Y SU RELACION CON EL DISEÑO DE SISTEMAS DE RIEGO. *Ponencia presentada en el Seminario Internacional CORA*, (págs. 1-21). Cajamarca, Peru.
- Delgado-Aguíñaga, J. A., Orozco-Mora, J. L., & Ruiz-Beltrán, E. (2013). Solución al problema de desacoplamiento mediante un controlador. *Ingeniería Investigación y Tecnología*, 523-536.
- Deloitte. (s.f.). Recuperado el 8 de Abril de 2020, de <https://www2.deloitte.com/es/es/pages/technology/articles/loT-internet-of-things.html>
- Duarte, O. (1999). Sistemas de lógica difusa. Fundamentos. *Revista Ingenieria e Investigación No. 42*, 22-30.
- ERENTURK, K. (2005). MATLAB-Based GUIs for Fuzzy Logic Controller Design and Applications to PMDC Motor and AVR Control. *Wiley Periodicals Inc.*, 10-25.
- FAO. (6 de Abril de 2017). *fao.org*. Recuperado el 6 de Abril de 2020, de <http://www.fao.org/faostat/es/?#data/QC>
- Farfan, R., & Hoyos, D. (2008). Sistema De Simulación Y Evaluación De Lógicas De Control. *Avances en Energías Renovables*.

- Fernández, J., Romero, R., Montaña, J., Díaz-Espejo, A., Muriel, J., Cuevas, M., y otros. (2008). Design and testing of an automatic irrigation controller for fruit tree. *Australian Journal of Agricultural Research*, 589–598.
- García Breijo, E. (2008). *Compilador C CCS y simulador PRO TEUS para Microcontroladores PIC*. Alfaomega Grupo Editor.
- González Morcillo, C. (s.f.). *Logica Difusa una introduccion practica*.
- González-Portillo, M., Rocha-Guzmán, N., Simpson, J., Rodríguez-Guerra, R., Gallegos-Infante, A., Delgado, E., y otros. (2008). DETERMINACIÓN DE ALGUNOS ATRIBUTOS DE CALIDAD DE MANZANA STARKING Y GOLDEN DELICIOUS EN COMPARACIÓN CON SUS MUTANTES Y SU RELACIÓN CON LAS UNIDADES FRÍO. *Ciencia y tecnología alimentaria*, 27-32.
- GUROVICH, L. (1985). Fundamentos y diseño de sistemas de riego. *Levantex*, 433.
- Guzmán, D., & Castaño, V. (2006). La lógica difusa en ingeniería: Principios, aplicaciones y futuro. *Ciencia y Tecnología*, 87-107.
- HARALD HIESSIL, E. J. (1990). A Heuristic Closed-Loop Controller for Water Distribution. *WATER RESOURCES SEAR*, 1323-1333.
- Hu, F., Chen, C., & Li, X. (2013). Design of Wireless Water-Saving Irrigation Controller. *Applied Mechanics and Materials*, 1577-1580.
- M. Bahat, G. I. (2000). A fuzzy irrigation controller system. *Engineering Applications of Artificial Intelligence* 13, 137±145.
- Maguiña Pérez, R. (2010). Sistemas de inferencia basados en Lógica Borrosa: Fundamentos y caso de estudio. *Revista de Ingeniería de Sistemas e Informática Vol. 7, N.º 1*, 91-104.
- MANCERA-LÓPEZ, M., SOTO-PARRA, J., SÁNCHEZ-CHÁVEZ, E., YÁÑEZ-MUÑOZ, R., MONTES-DOMÍNGUEZ, F., & BALANDRÁN-QUINTANA, R. (2007). Caracterización mineral de manzana Red Delicious y Golden Delicious de dos países productores. *Tecnociencia Chihuahua*, 6-17.
- Marcano, C. (2004). Diseño de un algoritmo de control difuso basado en una estructura PID.
- Montufar Villafuerte, F. (2012). Diseño de un controlador por logica difusa aplicado a un robot manipulador . *Universidad Autonoma de Queretaro. Maestria en Ciencias en Instrumentacion y Control Automatico*, 1-130.
- Morales-Luna, G. (2002). Introduccion a la logica difusa. *Centro de Investigacion y Estudios Avanzados del IPN*, 1-12.
- Morton, T. (2000). *Embedded Microcontrollers*. Prentice Hall.

- Muñoz-Carpena, R., Dukes, M., Li, Y., & Klassen, W. (2008). DESIGN AND FIELD EVALUATION OF A NEW CONTROLLER. *American Society of Agricultural and Biological Engineers*, Vol. 24(2): 183-191.
- Nguyen, H., Prasad, N., Walker, C., & Walker, E. (2003). *A first course in fuzzy and neural control*. United States of America: CHAPMAN & HALL/CRC.
- Passino, K., & Yurkovich, S. (1998). *Fuzzy Control*. Menlo Park: Addison Wesley Longman, Inc.
- PENELO, L. (11 de Junio de 2018). *La Vanguardia*. Recuperado el 11 de Abril de 2020, de <https://www.lavanguardia.com/comer/materia-prima/20180611/444212284842/fruta-manzana-propiedades.html>
- Reina, D. (10 de Junio de 2008). Fundamentos de Matemática Difusa. *Fundación Universitaria Konrad Lorenz. Facultad matematicas* , 1-69.
- Rossano, V. (s.f.). Guia Practica de Programacion. *Electronica & Microcontroladores PIC*, 98.
- SAGARPA. (2018). *Gobierno de Mexico*. Recuperado el 6 de Abril de 2020, de https://nube.siap.gob.mx/gobmx_publicaciones_siap/pag/2018/Atlas-Agroalimentario-2018
- Sanahuja, S. (2017). Sistemas de Control con Logica Difusa: Metodos de Mamdani y de Takagi-Sugeno-Kang (TSK). *Universitat Jaume*, 1-73.
- Serna Ruiz, A., Ros Garcia , F., & Rico Noruega, J. (2010). *Guia Practica de Sensores*. Creaciones Copyrigh.
- Shanmukha Sai Teja, P., Vineel, M., Manisha, G., & Satyanarayana, S. (2018). Automated irrigation system using sensors and node micro. *International Journal of Engineering & Technology*, 240-242.
- Simon, D. (1999). *And Embedded Software Primer*. Addison-Wesley Professional.
- Sugeno, M., & Takagi , T. (1985). Fuzzy identifications of systems and its applications to modeling and control. *IEEE Trans. on System, Man and Cybernetics*, vol. 1, nro. 15, 116-132.
- Theler, G. (2007). Controladores basados en losgica difusa y loops de convencion natural caoticos. *Instituto Balseiro. Universidad Nacional de Cuyo. Comision Nacional de Energia Atomica. Argentina*, 1-171.
- Vahid, F. &. (s.f.). *Embedded System Design*. 2002: John Wiley & Sons, Inc. .
- Viñas, I., Recansens, I., Usall, J., & Graell, J. (2013). *Poscosecha de pera, manzana y melocoton* . España: Mundi-Prensa.

- Wilmshurst, T. (2003). *An Introduction to the Design of Small Scale Embedded Systems with examples from PIC, 80C51 and 68HC05/08 Microcontrollers*. Palgrave Foundations.
- WILSON, L., Boyette, M., & E.A, E. (1999). Postharvest handling and cooling of fresh fruits, vegetables and flowers for a small farms. *Posharvest Handling of Horticultural Crops. North Carolina Cooperative Extension Service*, 55-86.
- Wolf, W. (2008). *Computers as Components: Principles or Embedded Computing System Design*. Second Edition. Morgan Kaufmann.
- Zadeh, L. (1965). Fuzzy Sets. *INFORMATION AND CONTROL* 8, 338-353.
- Zadeh, L. (1975). The concept of a linguistic variable and its applications to approximate reasoning. *Information Science*, 199-249.

7 ANEXOS

7.1 Constancia de registro de obra provisional.



CULTURA
SECRETARÍA DE CULTURA



INDAUTOR
INSTITUTO NACIONAL DEL DERECHO DE AUTOR

Dirección del Registro Público del Derecho de Autor

**CONSTANCIA DE
REGISTRO DE OBRA
PROVISIONAL**

Fecha: 26-11-2021

Número de Registro Provisional: 26463

Con fundamento en lo dispuesto en los artículos 163, 168 y 209 fracción III de la Ley Federal del Derecho de Autor 57, 59, 64 y 105 de su Reglamento, así como en el artículo 28, tercer párrafo, de la Ley Federal de Procedimiento Administrativo, se tiene por recibida la solicitud de registro de obra, vía electrónica, y anexos que acompaña a la misma, para su atención y desahogo, en términos de lo dispuesto en el artículo tercero del Acuerdo de fecha 3 de abril de 2020, publicado en el Diario Oficial de la Federación el 17 de los mismos meses y año, en relación con los publicados el 30 de octubre del mismo año.

Habiéndose realizado por esta autoridad el estudio de forma, en términos de la Ley Federal del Derecho de Autor y su Reglamento, se le comunica que se tiene por presentada la solicitud de referencia y por registrada, de manera provisional, bajo el folio que se indica en el presente documento, por lo que, en la fecha que para el efecto se señale, deberá concluir su trámite, debiendo presentar los documentos originales correspondientes y sus anexos, a efecto de que le sea entregado el Certificado de Registro. Es importante hacer de su conocimiento que, de no dar seguimiento al trámite hasta su conclusión en los términos que en este párrafo se señalan, esta autoridad lo tendrá por no presentado.

Se le aclara que podrá solicitar su cita en línea (<https://www.serviciosindautor.cultura.gob.mx/citautor/>) para la entrega de su certificado de inscripción una vez **transcurridos 15 días a partir de la fecha de expedición de esta constancia.**

Datos de la solicitud de registro de obra,

Autor: MARTINEZ RIVERA JOSE ANTONIO, GAMERO INDA EDUARDO, GODINEZ GARCIA FRANCISCO JAVIER, MIRANDA LUNA NUEL DE JESUS, GUERRERO RIVERA RUBEN

Título: PROGRAMA DE CÓMPUTO PARA SISTEMA DE CONTROL DIFUSO DE RIEGO EN HUERTO DE MANZANO

Rama: PROGRAMA DE COMPUTO

Titular: SECRETARÍA DE EDUCACIÓN PÚBLICA / TECNOLÓGICO NACIONAL DE MÉXICO (CON FUNDAMENTO EN EL ART 83 DE LA LFDA EN RELACION CON EL ART 46 DEL RLFDA)



Jesús Parets Gómez
Director del Registro Público del Derecho de Autor
Puebla 143 piso 3, col. Roma Norte
Alcaldía Cuauhtémoc, C.P. 06700, Ciudad de México
Teléfono: 55 3601 8200 Ext. 69362 D. 3601 8209
Dirección de Registro

Con fundamento en lo dispuesto en los artículos 28 tercer párrafo y 69-C, último párrafo, en relación con el diverso 35 último párrafo, todos ellos de la Ley Federal de Procedimiento Administrativo, el presente documento no contiene firma autógrafa debido a la emergencia sanitaria, por lo que existe impedimento para imprimirlo, suscribirlo y escanearlo, sin embargo, se reconoce y ratifica su contenido. Para verificación de la presente constancia ponerse en contacto con el Director del Registro Público del Derecho de Autor.

Si realizó el trámite desde algún estado de la República (no CDMX ni área metropolitana), podrá enviar los documentos originales y sus anexos por mensajería particular o en su caso, a través de la Oficina de Enlace Educativo de la Secretaría de Educación Pública (Delegación Federal de la SEP) perteneciente a su estado y, consecuentemente, le sea enviado el certificado de registro.

Puebla No. 143, Col. Roma Norte, CP. 06700, Cuauhtémoc, Ciudad de México.
Tel: (55) 3601 8200 www.indautor.gob.mx



7.2 Curriculum vitae único. Conacyt



CURRÍCULUM VITAE ÚNICO

1. Datos básicos

[Datos generales](#)

[Domicilio de residencia](#)

2. Formación académica

3. Trayectoria profesional

4. Producción científica, tecnológica y de innovación

4.1 Científica

4.2 Tecnológica y de innovación

[Desarrollo de software](#)

5. Formación de capital humano

5.1 Docencia

5.2 Tesis dirigidas

5.3 Diplomados

6. Comunicación pública de la ciencia, tecnológica y de innovación

6.1 Difusión

[Publicación de artículos](#)

6.3 Divulgación

7. Vinculación

8. Evaluaciones

9. Premios y distinciones

10. Lenguas e idiomas

CURRÍCULUM VITAE ÚNICO

Datos generales

CURP: MILM871213HDGRNN04	Fecha de nacimiento: 13/dic/1987	RFC: null
Nombre: MANUEL DE JESUS	Primer apellido: MIRANDA	Segundo apellido: LUNA
Sexo: Masculino	Estado conyugal: Soltero(a)	País de nacimiento: México
Entidad federativa: DURANGO		CVU: 1001689
Contacto principal: mirlu8720@hotmail.com		Nacionalidad: Mexicana

Identificadores de autor

ORC ID:
 Researcher ID Thomson:
 arXiv Author ID:
 PubMed Author ID:
 Open ID:

Medios de contacto

Medio de contacto	Categoría de contacto	Correo / Teléfono	Principal
Correo electrónico	Oficial	mirlu8720@hotmail.com	SI
Correo electrónico	Personal	manuel.miranda@mx.yazaki.com	NO
Móvil	Personal	6188159670	NO

Domicilio de residencia

Estado o distrito federal: DURANGO	Municipio o delegación: DURANGO
Localidad: DURANGO	Código postal: 34070
Asentamiento: Colonia - Silvestre Dorador	

Vialidad de domicilio

Nombre de vialidad:
 Dolores del Rio

Identificación del inmueble

Número exterior:	Parte numérica: 207	Parte alfanumérica:	Número exterior anterior: null
Número interior:	Parte numérica:	Parte alfanumérica:	

Entre que calles

Nombre de vialidad:
 null null y null null

Calle posterior

Nombre:
 null null

Descripción de la ubicación:

CURRÍCULUM VITAE ÚNICO

Desarrollo de software

Título: Controlador de riego basado en lógica difusa para el cultivo de manzanas

Tipo de desarrollo: Software de ingeniería y científico (caracterizado por algoritmos, ej.: astronomía, vulcanología, biología molecular, fabricación automática,

¿Cuenta con derechos de autor? Sí

País: México

Horas hombre: 120

Inicio: 01/ago/2019

Fin: 01/ago/2021

Costo: 24000

Beneficiario: Tecnológico Nacional de México/ Instituto Tecnológico de Durango

Objetivo:

Diseñar un controlador difuso multivariable en un sistema embebido para aplicación de dosis de riego

Resumen:

Implementación de un algoritmo de control basado en lógica difusa para riego de un huerto de manzanos utilizando variables inciertas.

Generación de valor y/o impacto para el beneficiario:

La implementación del algoritmo de control en un sistema embebido permite la reducción de consumo de

Formación de recursos humanos u otros resultados:

Formación de un alumno de maestría en ingeniería.

¿Recibió apoyo CONACYT? No

Logros:

Un algoritmo de control basado en lógica difusa con un motor de inferencia tipo Mamdani para la dosificación de agua de riego aplicada a un huerto de manzanos.

7.3 Código en CCS Compiler de la estación meteorológica.

```

//////////////////////////////////////////////////////////////////
////                               Estación Meteorológica                               ////
//////////////////////////////////////////////////////////////////

#define LCD_RS_PIN PIN_D0           // Conecciones LCD
#define LCD_RW_PIN PIN_D1
#define LCD_ENABLE_PIN PIN_D2
#define LCD_DATA4 PIN_D3
#define LCD_DATA5 PIN_D4
#define LCD_DATA6 PIN_D5
#define LCD_DATA7 PIN_D6

#include < 18F4550 . h >             // PIC a utilizar
#define ADC = 10                   // Resolucion del AD
#fuses NOWDT , NOMCLR , INTRC_IO   // Fusibles declarados
#use delay ( internal = 8M )       // Uso del reloj interno de 8MHz
#use rs232 ( baud = 9600 , xmit = pin_c6 , rcv = pin_c7 , bits = 8 , parity = N , stop = 1 )
// Definicion puerto UART

#include < lcd . c >
#include < stdlib . h >
#include < stdio . h >
#use fast_io ( B )
#define Data_Pin = 0xF81 . 0       // Mapeado PORTB 0
#define Data_Pin_Direction = 0xF93 . 0 // Mapeado TRISB 0
#define DS18B20_PIN PIN_B1        // Mapeo de RB1 como DS18B20_PIN

char message1 [] = "Temp = 00.0 C" ; // Declaracion de variables
char message2 [] = "Humd = 00.0 %" ;
short Time_out ;
unsigned int8 T_byte1 , T_byte2 , RH_byte1 , RH_byte2 , CheckSum ;
unsigned int16 Temp , RH ;
char cadena1 [ 2 ] , cadena [ 2 ] ;
signed int16 raw_temp ;
float temp1 , p , o , humedad ;
int16 q ;

//////////////////////////////////////////////////////////////////
////                               Sensor DHT22                               ////
//////////////////////////////////////////////////////////////////

void start_signal ()              // Funcion para inicializar DHT22
{
    Data_Pin_Direction = 0 ;      // Configurar pin RB0 como salida
    Data_Pin = 0 ;                // Pin RB0 en bajo
    delay_ms ( 25 ) ;
    Data_Pin = 1 ;                // Pin RB0 en alto
    delay_us ( 30 ) ;
    Data_Pin_Direction = 1 ;      // Configurar pin RB0 como entrada

```



```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
int1 ds18b20_start ()
{
    output_low ( DS18B20_PIN );           // Manda senal de Reset al sensor
    output_drive ( DS18B20_PIN );        // Configura RB1 como salida
    delay_us ( 500 );                    // Espera 500 us
    output_float ( DS18B20_PIN );        // Configura RB1 como entrada
    delay_us ( 100 );                   // Espera respuesta sensor
    if ( ! input ( DS18B20_PIN ) )
    {
        delay_us ( 400 ); // Espera 400 us
        return TRUE ; // Se encontro al sensor
    }
    return FALSE ;
}

void ds18b20_write_bit ( int1 value )
{
    output_low ( DS18B20_PIN );
    output_drive ( DS18B20_PIN );        // Configura RB1 como salida
    delay_us ( 2 );                      // Espera 2 us
    output_bit ( DS18B20_PIN , value );
    delay_us ( 80 );                    // Espera 80 us
    output_float ( DS18B20_PIN );        // Configura RB1 como entrada
    delay_us ( 2 );                     // Espera 2 us
}

void ds18b20_write_byte ( int8 value )
{
    int8 i ;
    for ( i = 0 ; i < 8 ; i ++ )
        ds18b20_write_bit ( bit_test ( value , i ) );
}

int1 ds18b20_read_bit ( void )
{
    int1 value ;
    output_low ( DS18B20_PIN );
    output_drive ( DS18B20_PIN );        // Se configura pin RB1 como salida
    delay_us ( 2 );                      // Retardo de dos microsengundos
    output_float ( DS18B20_PIN );        // Se configura pin RB1 como entrada
    delay_us ( 5 );                      // Retardo de 5 microsegundos
    value = input ( DS18B20_PIN );
    delay_us ( 100 );                    // Retardo de 100 microsegundos
    return value ;
}

int8 ds18b20_read_byte ( void )         // Funcion para leer un byte
{
    int8 i , value = 0 ;

```

```

for ( i = 0 ; i < 8 ; i ++ )
    shift_right ( & value , 1 , ds18b20_read_bit ( ) );
return value ;
}

int1 ds18b20_read ( int16 * raw_temp_value )
{
    if ( ! ds18b20_start ( ) )           // Manda el pulso de inicio
        return FALSE ;
    ds18b20_write_byte ( 0xCC ) ;       // Manda comando de salto
    ds18b20_write_byte ( 0x44 ) ;       // Manda comando inicio conversion
    while ( ds18b20_read_byte ( ) == 0 ) ; // Espera completado de conversion
    if ( ! ds18b20_start ( ) )           // Send start pulse
        return FALSE ;
    ds18b20_write_byte ( 0xCC ) ;       // Manda comando de salto
    ds18b20_write_byte ( 0xBE ) ;       // Manda comando inicio conversion
    * raw_temp_value = ds18b20_read_byte ( ) ; // Lee LSB byte, guarda en rtv LSB byte
    * raw_temp_value |= ( int16 ) ( ds18b20_read_byte ( ) << 8 ;
    return TRUE ;                         // Lee MSB byte, guarda en rtv MSB byte
}

char Activacion [ 1 ] ;                  // Valor recibido por comunicacion serial

#INT_RDA                                 // Servicio de interrupcion por RS232

void rda_isr ( )
{
    Activacion [ 0 ] = getc ( ) ;         // Lee un caracter y retorna un valor
    if ( Activacion [ 0 ] == 'M' )
    {
        ////////////////////////////////////////////////////////////////////
        /////                               Sensor DHT22                               /////
        ////////////////////////////////////////////////////////////////////

        lcd_putc ( '\f' ) ;
        Time_out = 0 ;                   // Inicializa time out
        Start_signal ( ) ;               // Subrutina para inicializar sensor
        if ( check_response ( ) )        // El sensor responde
        {
            RH_byte1 = Read_Data ( ) ;   // Lee RH byte1
            RH_byte2 = Read_Data ( ) ;   // Lee RH byte2
            T_byte1 = Read_Data ( ) ;    // Lee T byte1
            T_byte2 = Read_Data ( ) ;    // Lee T byte2
            Checksum = Read_Data ( ) ;   // Lee checksum
            if ( Time_out )              // El tiempo de lectura fue mayor
            {
                lcd_putc ( '\f' ) ;      // Borra LCD
                lcd_gotoxy ( 5 , 1 ) ;
                lcd_putc ( "Time out!" ) ; // Imprime Fuera de tiempo
            }
        }
        else // El tiempo de lectura fue correcto

```

```

{
    if ( CheckSum == (( RH_Byte1 + RH_Byte2 + T_Byte1 + T_Byte2 ) & 0xFF ))
    { // El checksum fue correcto
        RH = RH_byte1 ;
        RH = ( RH << 8 ) | RH_byte2 ;
        Temp = T_byte1 ;
        Temp = ( Temp << 8 ) | T_byte2 ;
        if ( Temp > 0X8000 )
        {
            message1 [ 6 ] = '-';
            Temp = Temp & 0X7FFF ;
        }

        else
        message1 [ 6 ] = '' ;
        message1 [ 7 ] = ( Temp / 100 ) % 10 + 48 ;
        message1 [ 8 ] = ( Temp / 10 ) % 10 + 48 ;
        message1 [ 10 ] = Temp % 10 + 48 ;
        message2 [ 7 ] = ( RH / 100 ) % 10 + 48 ;
        message2 [ 8 ] = ( RH / 10 ) % 10 + 48 ;
        message2 [ 10 ] = RH % 10 + 48 ;
        message1 [ 11 ] = 223 ; // Simbolo de grados
        lcd_putc ( '\f' ); // Borra LCD
        lcd_gotoxy ( 1 , 1 );
        printf ( lcd_putc , message1 ); // Imprime message1 en el LCD;
        lcd_gotoxy ( 1 , 2 );
        printf ( lcd_putc , message2 ); // Imprime message2
        sprintf ( cadena1 , "%c%c" , message1 [ 7 ], message1 [ 8 ] );
        printf ( cadena1 ); // Envia Humedad a UART
        sprintf ( cadena , "%c%c" , message2 [ 7 ], message2 [ 8 ] );
        humedad = atof ( cadena );
    }
    else // Incorrecto valor de checksum
    {
        lcd_putc ( '\f' );
        lcd_gotoxy ( 1 , 1 );
        lcd_putc ( "Checksum Error!" );
    }
}
}
else //El sensor no respondio
{
for (;;)
{
    lcd_putc ( '\f' );
    lcd_gotoxy ( 3 , 1 );
    lcd_putc ( "No response" );
    lcd_gotoxy ( 1 , 2 );
    lcd_putc ( "from the DHT22 " );
    delay_ms ( 1000 );
}
}
}

```

```

delay_ms ( 1000 ); // Retardo de un segundo

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////                                     Sensor DS18B20                                ////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

if ( ds18b20_read (& raw_temp ) ) // Comunicacion entre PIC y sensor
{ // fue correcta
    temp1 = ( float ) raw_temp / 16 ; // Convierte raw_tem en grados Celcius
                                        // (temp en °C = raw/16)

    lcd_putc ( '\f' );
    printf ( lcd_putc , "TTemp = %f" , temp1 ); // Imprime Temp
    lcd_putc ( 223 ); // Imprime simbolo de grados ( ° )
    cd_putc ( "C " ); // Imprime 'C'
    printf ( "%1.0f" , temp1 );
}
else // Comunicacion fallida
{
    for (;;)
    {
        lcd_putc ( '\f' );
        lcd_gotoxy ( 3 , 1 );
        lcd_putc ( "No response" );
        lcd_gotoxy ( 1 , 2 );
        lcd_putc ( "from the DS18B20 " );
        delay_ms ( 1000 );
    }
}

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//////                                     Sensor De Humedad                                ////
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

set_adc_channel ( 0 ); //Habilita el canal Analogico 0
delay_us ( 40 );
q = read_adc (); //Lee el valor en el canal Analogico
p = 5 . 0 * q / 1024 . 0 ; //Convierte tension a un valor flotante

if ( p < 3 )
{
    o = 100 * p / 3 ;
    delay_us ( 100 );
    lcd_gotoxy ( 1 , 2 );
    printf ( lcd_putc , "THumd = %1.2f%%" , o );
    printf ( "%1.0f" , o );
    printf ( "%1.0f" , humedad );
    delay_ms ( 1000 );
}

else // El voltaje sobrepasa el valor esperado

```

```

{
    for (;;)
    {
        lcd_putc ( '\f' );
        lcd_gotoxy ( 3 , 1 );
        lcd_putc ( "Wrong Value" );
        lcd_gotoxy ( 1 , 2 );
        lcd_putc ( "from SOIL SENSOR " );
        delay_ms ( 1000 );
    }
}
}
else
return;
}

/////////////////////////////////////////////////////////////////
////                                Programa Principal                                ////
/////////////////////////////////////////////////////////////////

void main ()
{
    setup_oscillator ( OSC_8Mhz | OSC_INTRC | OSC_PLL_OFF );
                               // Configura reloj interno a 8Mhz
    setup_adc_ports ( AN0 );    // Definiacion de la entrada analogica
    setup_adc ( ADC_CLOCK_INTERNAL ); // Selecciona reloj conversion A/D
    lcd_init ();                // Inicializa LCD
    enable_interrupts ( INT_RDA ); // Habilita interrupción por recepción RS232
    enable_interrupts ( GLOBAL ); // Habilita interrupcion global
    lcd_putc ( '\f' );          // Borra LCD
    for (;;) {                  // Ciclo principal programa, esperando int
        printf ( lcd_putc , "\fESPERANDO" );
        printf ( lcd_putc , "\nACTIVACION" );
        delay_ms ( 500 );
    }
}

```

7.4 Controlador Difuso

```

#include <18F4550.h> // Microcontrolador a usar
#FUSES WDT2048,MCLR,INTRC_IO // Fusibles declarados
#use delay(internal=8M)
#use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8, parity=N, stop=1)
#define LCD_RS_PIN PIN_b0 // Utilizacion de los pines a controlar LCD
#define LCD_RW_PIN PIN_b1
#define LCD_ENABLE_PIN PIN_b2
#define LCD_DATA4 PIN_b4
#define LCD_DATA5 PIN_b5
#define LCD_DATA6 PIN_b6

```

```

#define LCD_DATA7 PIN_b7
#include <LCD.C> // Libreria de LCD
#include <stdlib.h> // Librerias para usar la funcion atof
#include <stdio.h>

//////////          Variables a utilizar en el programa          //////////

float Frio, Templado, Caliente, p; // Variables para atemp
float PocaHumedad, Normal, MuchaHumedad, s; // Variables para aHumd
float FrioTierra, TempladoTierra, CalienteTierra, t, tt; // Variables para
ttemp
float PocaHumedadTierra, NormalTierra, MuchaHumedadTierra, u; // Variables
para tHumd
float VArea, TotalArea; // Variables para calcular COG
float VCOG, TotalCOG;
int contador, a, Factor, Valvula, UnaVez=0;
float Altura[8], AArea[8], ACOG[8];

//////////          Fusifica las senales analogicas          //////////

void Fusificacion()
{
VCOG =0;
Frio=0; // asociados con airetemp
Templado=0;
Caliente=0;
PocaHumedad = 0; // asociados con airehmd
Normal = 0;
MuchaHumedad = 0;
FrioTierra=0; // asociados con tierratemp
TempladoTierra=0;
CalienteTierra=0;
PocaHumedadTierra = 0; // asociados con airehmd
NormalTierra = 0;
MuchaHumedadTierra = 0;
//////////          Fusificacion Temperatura Aire
//////////          La función triangular A comprendida entre los puntos
//////////          (a, 0) y (b, 0) y con una altura en la coordenada (c, 1)
//////////          está definida por:
//////////          1((x-a)/(c-a)) Si a<x<c
//////////          1((x-b)/(c-b)) Si c<x<b
//////////          0 Cualquier otro valor
if(p <= 0)
{
Frio=1;
Templado=0;
Caliente=0;
}
if((p > 0) & (p < 10))
{
Frio = (10 - p);
Frio = Frio*1;
Frio = Frio/(10-0);
Templado = (p - 0);
Templado = Templado*1;
Templado = Templado/(10-0);
}
}

```

```

if((p > 10) & (p < 20))
{
Templado = (20 - p);
Templado = Templado*1;
Templado = Templado/(20-10);
Caliente = (p - 10);
Caliente = Caliente*1;
Caliente = Caliente/(20-10);
1
C:\Users\Miranda\Desktop\Manuel
Miranda\Maestria\C_Compiler\Programas\ControladorDifuso\Ejemplo5.CuatroVa
riablesFisico\Fusificacion2.c
}
if(p >= 20)
{
Frio=0;
Templado=0;
Caliente=1;
}
if (p == 0) Frio=1;
if (p == 10) Templado = 1;
if (p == 20) Caliente = 1;

////////// Fusificacion Humedad Aire
////////// La función triangular A comprendida entre los puntos
////////// (a, 0) y (b, 0) y con una altura en la coordenada (c, 1)
////////// está definida por:
////////// 1((x-a)/(c-a)) Si a<x<c
////////// 1((x-b)/(c-b)) Si c<x<b
////////// 0 Cualquier otro valor
//////////

if(s <= 20)
{
PocaHumedad=1;
Normal=0;
MuchaHumedad=0;
}
if((s > 20) & (s < 40))
{
PocaHumedad = (40 - s);
PocaHumedad = PocaHumedad*1;
PocaHumedad = PocaHumedad/(40-20);
Normal = (s - 20);
Normal = Normal*1;
Normal = Normal/(40-20);
}
if((s > 40) & (s < 60))
{
Normal = (60 - s);
Normal = Normal*1;
Normal = Normal/(60-40);
MuchaHumedad = (s - 40);
MuchaHumedad = MuchaHumedad*1;
MuchaHumedad = MuchaHumedad/(60-40);
}
if(s >= 60)

```

```

{
PocaHumedad=0;
Normal=0;
MuchaHumedad=1;
}
if (s == 20) PocaHumedad=1;
if (s == 40) Normal = 1;
if (s == 60) MuchaHumedad = 1;

////////// Fusificacion Temperatura Tierra
////////// La función triangular A comprendida entre los puntos
////////// (a, 0) y (b, 0) y con una altura en la coordenada (c, 1)
////////// está definida por:
////////// 1((x-a)/(c-a)) Si a<x<c
////////// 1((x-b)/(c-b)) Si c<x<b
////////// 0 Cualquier otro valor
//////////

if(t <= 0)
{
FrioTierra=1;
TempladoTierra=0;
CalienteTierra=0;
}
if((t > 0) & (t < 3))
{
FrioTierra = (3 - t);
FrioTierra = FrioTierra*1;
FrioTierra = FrioTierra/(3-0);
TempladoTierra = (t - 0);
TempladoTierra = TempladoTierra*1;
TempladoTierra = TempladoTierra/(3-0);
}
if((t > 3) & (t < 6))
{
TempladoTierra = (6 - t);
TempladoTierra = TempladoTierra*1;
TempladoTierra = TempladoTierra/(6-3);
CalienteTierra = (t - 3);
CalienteTierra = CalienteTierra*1;
CalienteTierra = CalienteTierra/(6-3);
}
if(t >= 6)
{
FrioTierra=0;
TempladoTierra=0;
CalienteTierra=1;
}
if (t == 0) FrioTierra=1;
if (t == 3) TempladoTierra = 1;
if (t == 9) CalienteTierra = 1;

////////// Fusificacion Humedad Tierra
////////// La función triangular A comprendida entre los puntos
////////// (a, 0) y (b, 0) y con una altura en la coordenada (c, 1)
////////// está definida por:
////////// 1((x-a)/(c-a)) Si a<x<c

```

```

////////// 1((x-b)/(c-b)) Si c<x<b
////////// 0 Cualquier otro valor

if(u <= 20)
{
PocaHumedadTierra=1;
NormalTierra=0;
MuchaHumedadTierra=0;
}
if((u > 20) & (u < 40))
{
PocaHumedadTierra = (40 - u);
PocaHumedadTierra = PocaHumedadTierra*1;
PocaHumedadTierra = PocaHumedadTierra/(40-20);
NormalTierra = (u - 20);
NormalTierra = NormalTierra*1;
NormalTierra = NormalTierra/(40-20);
}
if((u > 40) & (u < 60))
{
NormalTierra = (60 - u);
NormalTierra = NormalTierra*1;
NormalTierra = NormalTierra/(60-20);
MuchaHumedadTierra = (u - 20);
MuchaHumedadTierra = MuchaHumedadTierra*1;
MuchaHumedadTierra = MuchaHumedadTierra/(60-20);
}
if(u >= 60)
{
PocaHumedadTierra=0;
NormalTierra=0;
MuchaHumedadTierra=1;
}
if (u == 20) PocaHumedadTierra=1;
if (u == 40) NormalTierra = 1;
if (u == 60) MuchaHumedadTierra = 1;
return;
}

////////// Visualizacion de las variables en el LCD //////////

void VisualizacionInterrupcion()
{
printf(lcd_putc, "\fCOG = %1.2f",VCOG);
// VCOG = 0.93*VCOG +11;
printf("V"); // Caracter para activar sentido
Valvula = (int)VCOG;
if (Valvula <= 45)Valvula = 45;
printf("%2d",Valvula);
delay_ms(500);
// printf(lcd_putc, "\fFuzzy = %1.2f",Altura[6]);
// delay_ms(500);
return;
}
////////// Calcula Area de las funciones membresia
////////// Mediante la formula:
//////////  $W(h-(h^2)/2)$ 

```

```

////////// Donde:
////////// W (10): Es la base del triangulo
////////// h (Altura[contador]: Es la altura del triangulo
//////////

void Area()
{
AArea[contador] = 40*(Altura[contador] - (Altura[contador]*
Altura[contador])/2);
return;
}

////////// Calcula parte de la formula para COG
////////// Mediante la formula:
////////// Sumatoria (bi(Area debajo funcion membresia)/Sumatoria de Areas
////////// Donde:
////////// bi(Factor): Es el centro de la funcion membresia

void COG()
{
ACOG[contador] = (Factor*AArea[contador]);
return;
}

//////////          Aplica el metodo de inferencia y Reglas  //////////

void ReglaAndDefusificacion()
{
// Inicializar variables de inferencia
for(a=0; a<=8; a++)
{
Altura[a] = 0;
AArea[a] = 0;
ACOG[a] = 0;
}
contador = 0;
if (Frio*FrioTierra !=0)
{
//Regla2 Listo
if(Normal !=0)
{
if(NormalTierra !=0)
{
contador++;
if (Frio <= FrioTierra && Frio <= Normal && Frio <=NormalTierra)
Altura[contador] = Frio;
else if (FrioTierra <= Normal && FrioTierra <= NormalTierra)
Altura[contador] = FrioTierra;
else if (Normal <= NormalTierra) Altura[contador] = Normal;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}
//Regla3 Listo
if(PocaHumedadTierra !=0)
{

```

```

contador++;
if (Frio <= FrioTierra && Frio <= Normal && Frio <=PocaHumedadTierra)
Altura[contador] = Frio;
else if (FrioTierra <= Normal && FrioTierra <= PocaHumedadTierra)
Altura[contador] = FrioTierra;
else if (Normal <= PocaHumedadTierra) Altura[contador] = Normal;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla4
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Frio <= FrioTierra && Frio <= Normal && Frio <=MuchaHumedadTierra)
Altura[contador] = Frio;
// else if (FrioTierra <= Normal && FrioTierra <= MuchaHumedadTierra)
Altura[contador] = FrioTierra;
// else if (Normal <= MuchaHumedadTierra) Altura[contador] = Normal;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}
if(PocaHumedad !=0)
{
//Regla1
if(PocaHumedadTierra !=0)
{
contador++;
if (Frio <= FrioTierra && Frio <= PocaHumedad && Frio
<=PocaHumedadTierra) Altura[contador] = Frio;
else if (FrioTierra <= PocaHumedad && FrioTierra <=PocaHumedadTierra)
Altura[contador] = FrioTierra;
else if (PocaHumedad <=PocaHumedadTierra) Altura[contador] = PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla5
if(NormalTierra !=0)
{
contador++;
if (Frio <= FrioTierra && Frio <= pocahumedad && Frio <=NormalTierra)
Altura[contador] = Frio;
else if (FrioTierra <= pocahumedad && FrioTierra <= NormalTierra)
Altura[contador] = FrioTierra;
else if (PocaHumedad <= NormalTierra) Altura[contador] = PocaHumedad ;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}
//Regla6

```

```

//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Frio <= FrioTierra && Frio <= pocahumedad && Frio
<=MuchaHumedadTierra) Altura[contador] = Frio;
// else if (FrioTierra <= pocahumedad && FrioTierra <= PocaHumedadTierra)
Altura[contador] = FrioTierra;
// else if (PocaHumedad <= MuchaHumedadTierra) Altura[contador] =
PocaHumedad;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}
if(MuchaHumedad !=0)
{
//Regla7
if(PocaHumedadTierra !=0)
{
contador++;
if (Frio <= FrioTierra && Frio <= MuchaHumedad && Frio
<=PocaHumedadTierra) Altura[contador] = Frio;
else if ( FrioTierra <= MuchaHumedad && FrioTierra <= PocaHumedadTierra)
Altura[contador] = FrioTierra;
else if (MuchaHumedad <= PocaHumedadTierra) Altura[contador] =
MuchaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla8
if(NormalTierra !=0)
{
contador++;
if (Frio <= FrioTierra && Frio <= MuchaHumedad && Frio <=NormalTierra)
Altura[contador] = Frio;
else if (FrioTierra <= MuchaHumedad && FrioTierra <= NormalTierra)
Altura[contador] = FrioTierra;
else if (MuchaHumedad <= NormalTierra) Altura[contador] = MuchaHumedad;
else Altura[contador] = NormalTierra;
4
C:\Users\Miranda\Desktop\Manuel
Miranda\Maestria\C_Compiler\Programas\ControladorDifuso\Ejemplo5.CuatroVa
riablesFisico\Fusificacion2.c
Area();
Factor = 50;
COG();
}
//Regla9
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Frio <= FrioTierra && Frio <= MuchaHumedad && Frio
<=MuchaHumedadTierra) Altura[contador] = Frio;

```

```

// else if (FrioTierra <= MuchaHumedad && FrioTierra <=
MuchaHumedadTierra) Altura[contador] = FrioTierra;
// else if ( MuchaHumedad <= MuchaHumedadTierra) Altura[contador] =
MuchaHumedadTierra;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}}
if(Templado*FrioTierra !=0)
{
if(PocaHumedad !=0)
{
//Regla10
if(PocaHumedadTierra !=0)
{
contador++;
if (Templado <= FrioTierra && Templado <= PocaHumedad && Templado
<=PocaHumedadTierra) Altura[contador] = Templado;
else if (FrioTierra <= PocaHumedad && FrioTierra <= PocaHumedadTierra)
Altura[contador] = FrioTierra;
else if (PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla11
if(NormalTierra !=0)
{
contador++;
if ( Templado<= FrioTierra && Templado <= PocaHumedad && Templado
<=NormalTierra) Altura[contador] = Templado;
else if (FrioTierra <= PocaHumedad && FrioTierra <= NormalTierra)
Altura[contador] = FrioTierra;
else if (PocaHumedad <= NormalTierra) Altura[contador] = PocaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 70;
COG();
}
//Regla12
if(MuchaHumedadTierra !=0)
{
contador++;
if (Templado <= FrioTierra && Templado <= PocaHumedad && Templado
<=MuchaHumedadTierra) Altura[contador] = Templado;
else if (FrioTierra <= PocaHumedad && FrioTierra <= MuchaHumedadTierra)
Altura[contador] = FrioTierra;
else if (PocaHumedad <= MuchaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = MuchaHumedadTierra;
Area();
Factor = 70;
COG();
}
}
}

```

```

}}}}
if(Templado*TempladoTierra !=0)
{
if(PocaHumedad !=0)
{
//Regla13
if(PocaHumedadTierra !=0)
{
contador++;
if (Templado <= TempladoTierra && Templado <= PocaHumedad && Templado
<=PocaHumedadTierra) Altura[contador] = Templado;
else if (TempladoTierra <= PocaHumedad && TempladoTierra <=
PocaHumedadTierra) Altura[contador] = TempladoTierra;
else if (PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla14
if(NormalTierra !=0)
{
contador++;
if (Templado <= TempladoTierra && Templado <= PocaHumedad && Templado
<=NormalTierra) Altura[contador] = Templado;
else if (TempladoTierra <= PocaHumedad && TempladoTierra <= NormalTierra)
Altura[contador] = TempladoTierra;
else if (PocaHumedad <= NormalTierra) Altura[contador] = PocaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 70;
COG();
}
//Regla15
if(MuchaHumedadTierra !=0)
{
contador++;
if (Templado <= TempladoTierra && Templado <= PocaHumedad && Templado
<=MuchaHumedadTierra) Altura[contador] = Templado;
else if ( TempladoTierra <= PocaHumedad && TempladoTierra
<=MuchaHumedadTierra) Altura[contador] = TempladoTierra;
else if ( PocaHumedad <= MuchaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = MuchaHumedadTierra;
Area();
Factor = 70;
COG();
}}}}
if(Caliente*TempladoTierra !=0)
{
//Regla17
if(Normal*NormalTierra !=0)
5
C:\Users\Miranda\Desktop\Manuel
Miranda\Maestria\C_Compiler\Programas\ControladorDifuso\Ejemplo5.CuatroVa
riablesFisico\Fusificacion2.c

```

```

{
contador++;
if (Caliente <= TempladoTierra && Caliente <= Normal && Caliente
<=NormalTierra) Altura[contador] = Caliente;
else if (TempladoTierra <= Normal && TempladoTierra <= NormalTierra)
Altura[contador] = TempladoTierra;
else if (Normal <= NormalTierra) Altura[contador] = Normal;
else Altura[contador] = NormalTierra;
Area();
Factor = 70;
COG();
}
//Regla18
if(MuchaHumedad*NormalTierra !=0)
{
contador++;
if (Caliente <= TempladoTierra && Caliente <= MuchaHumedad && Caliente
<=NormalTierra) Altura[contador] = Caliente;
else if (TempladoTierra <= MuchaHumedad && TempladoTierra <=
NormalTierra) Altura[contador] = TempladoTierra;
else if ( MuchaHumedad <= NormalTierra) Altura[contador] = MuchaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 70;
COG();
}
if(PocaHumedad!=0)
{
//Regla16
if(PocaHumedadTierra !=0)
{
contador++;
if (Caliente <= TempladoTierra && Caliente <= PocaHumedad && Caliente
<=PocaHumedadTierra) Altura[contador] = Caliente;
else if (TempladoTierra <= PocaHumedad && TempladoTierra <=
PocaHumedadTierra) Altura[contador] = TempladoTierra;
else if ( PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}}
if(Caliente*CalienteTierra !=0)
{
if(PocaHumedad !=0)
{
//Regla20
if(NormalTierra !=0)
{
contador++;
if (Caliente <= CalienteTierra && Caliente <= PocaHumedad && Caliente
<=PocaHumedadTierra) Altura[contador] = Caliente;
else if (CalienteTierra <= PocaHumedad && CalienteTierra <=
PocaHumedadTierra) Altura[contador] = CalienteTierra;
else if ( PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;

```

```

else Altura[contador] = NormalTierra;
Area();
Factor = 70;
COG();
}
//Regla19
if(PocaHumedadTierra !=0)
{
contador++;
if (Caliente <= CalienteTierra && Caliente <= PocaHumedad && Caliente
<=PocaHumedadTierra) Altura[contador] = Caliente;
else if (CalienteTierra <= PocaHumedad && CalienteTierra <=
PocaHumedadTierra) Altura[contador] = CalienteTierra;
else if ( PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla21
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Caliente <= CalienteTierra && Caliente <= PocaHumedad && Caliente
<=MuchaHumedadTierra) Altura[contador] = Caliente;
// else if ( CalienteTierra <= PocaHumedad && CalienteTierra <=
MuchaHumedadTierra) Altura[contador] = CalienteTierra;
// else if (PocaHumedad <= MuchaHumedadTierra) Altura[contador] =
PocaHumedad;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}
if(Normal !=0)
{
//Regla22
if(PocaHumedadTierra !=0)
{
contador++;
if (Caliente <= CalienteTierra && Caliente <= Normal && Caliente
<=PocaHumedadTierra) Altura[contador] = Caliente;
else if ( CalienteTierra <= Normal && CalienteTierra <=
PocaHumedadTierra) Altura[contador] = CalienteTierra;
else if (Normal <= PocaHumedadTierra) Altura[contador] = Normal;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla23
if(NormalTierra !=0)
{
contador++;

```

```

if (Caliente <= CalienteTierra && Caliente <= Normal && Caliente
<=NormalTierra) Altura[contador] = Caliente;
else if (CalienteTierra <= Normal && CalienteTierra <= NormalTierra)
Altura[contador] = CalienteTierra;
else if ( Normal <= NormalTierra) Altura[contador] = Normal;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}
//Regla24
6
C:\Users\Miranda\Desktop\Manuel
Miranda\Maestria\C_Compiler\Programas\ControladorDifuso\Ejemplo5.CuatroVa
riablesFisico\Fusificacion2.c
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Caliente <= CalienteTierra && Caliente <= Normal && Caliente
<=MuchaHumedadTierra) Altura[contador] = Caliente;
// else if ( CalienteTierra <= Normal && CalienteTierra <=
MuchaHumedadTierra) Altura[contador] = CalienteTierra;
// else if ( Normal <= MuchaHumedadTierra) Altura[contador] = Normal;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}
if(MuchaHumedad !=0)
{
//Regla25
if(PocaHumedadTierra !=0)
{
contador++;
if (Caliente <= CalienteTierra && Caliente <= MuchaHumedad && Caliente
<=PocaHumedadTierra) Altura[contador] = Caliente;
else if ( CalienteTierra <= MuchaHumedad && CalienteTierra <=
PocaHumedadTierra) Altura[contador] = CalienteTierra;
else if (MuchaHumedad <=PocaHumedadTierra) Altura[contador] =
MuchaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla26
if(NormalTierra !=0)
{
contador++;
if (Caliente <= CalienteTierra && Caliente <= MuchaHumedad && Caliente
<=NormalTierra) Altura[contador] = Caliente;
else if (CalienteTierra <= MuchaHumedad && CalienteTierra <=
NormalTierra) Altura[contador] = CalienteTierra;
else if (MuchaHumedad <= NormalTierra) Altura[contador] = MuchaHumedad;
else Altura[contador] = NormalTierra;
Area();
}
}

```

```

Factor = 50;
COG();
}
//Regla27
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Caliente <= CalienteTierra && Caliente <= MuchaHumedad && Caliente
<= MuchaHumedadTierra) Altura[contador] = Caliente;
// else if ( CalienteTierra <= MuchaHumedad && CalienteTierra <=
MuchaHumedadTierra) Altura[contador] = CalienteTierra;
// else if (MuchaHumedad <=MuchaHumedadTierra) Altura[contador] =
MuchaHumedad;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}}
if(Caliente*FrioTierra !=0)
{
if(PocaHumedad !=0)
{
//Regla28
if(PocaHumedadTierra !=0)
{
contador++;
if (Caliente <= FrioTierra && Caliente <= PocaHumedad && Caliente
<=PocaHumedadTierra) Altura[contador] = Caliente;
else if (FrioTierra <= PocaHumedad && FrioTierra <= PocaHumedadTierra)
Altura[contador] = FrioTierra;
else if (PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla29
if(NormalTierra !=0)
{
contador++;
if (Caliente <= FrioTierra && Caliente <= PocaHumedad && Caliente
<=NormalTierra) Altura[contador] = Caliente;
else if ( FrioTierra <= PocaHumedad && FrioTierra <= NormalTierra)
Altura[contador] = FrioTierra;
else if ( PocaHumedad <= NormalTierra) Altura[contador] = PocaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}
//Regla30
//if(MuchaHumedadTierra !=0)
// {
// contador++;

```

```

// if (Caliente <= FrioTierra && Caliente <= PocaHumedad && Caliente
<=MuchaHumedadTierra) Altura[contador] = Caliente;
// else if (FrioTierra <= PocaHumedad && FrioTierra <=
MuchaHumedadTierra) Altura[contador] = FrioTierra;
// else if ( PocaHumedad <= MuchaHumedadTierra) Altura[contador] =
PocaHumedad;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}
//Regla31
if(Normal !=0)
{
if(PocaHumedadTierra !=0)
{
contador++;
if (Caliente <= FrioTierra && Caliente <= Normal && Caliente
<=PocaHumedadTierra) Altura[contador] = Caliente;
else if ( FrioTierra <= Normal && FrioTierra <= PocaHumedadTierra)
Altura[contador] = FrioTierra;
else if ( Normal <= PocaHumedadTierra) Altura[contador] = Normal;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
7
C:\Users\Miranda\Desktop\Manuel
Miranda\Maestria\C_Compiler\Programas\ControladorDifuso\Ejemplo5.CuatroVa
riablesFisico\Fusificacion2.c
}
//Regla32
if(NormalTierra !=0)
{
contador++;
if (Caliente <= FrioTierra && Caliente <= Normal && Caliente
<=NormalTierra) Altura[contador] = Caliente;
else if (FrioTierra <= Normal && FrioTierra <= NormalTierra)
Altura[contador] = FrioTierra;
else if ( Normal <= NormalTierra) Altura[contador] = Normal;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}
//Regla33
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Caliente <= FrioTierra && Caliente <= Normal && Caliente
<=MuchaHumedadTierra) Altura[contador] = Caliente;
// else if ( FrioTierra <= Normal && FrioTierra <= MuchaHumedadTierra)
Altura[contador] = FrioTierra;
// else if ( Normal <= MuchaHumedadTierra) Altura[contador] = Normal;
// else Altura[contador] = MuchaHumedadTierra;
// Area();

```

```

// Factor = 25;
// COG();
// }
}
if(MuchaHumedad !=0)
{
//Regla34
if(PocaHumedadTierra !=0)
{
contador++;
if (Caliente <= FrioTierra && Caliente <= MuchaHumedad && Caliente
<=PocaHumedadTierra) Altura[contador] = Caliente;
else if ( FrioTierra <= MuchaHumedad && FrioTierra <= PocaHumedadTierra)
Altura[contador] = FrioTierra;
else if ( MuchaHumedad <= PocaHumedadTierra) Altura[contador] =
MuchaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla35
if(NormalTierra !=0)
{
contador++;
if (Caliente <= FrioTierra && Caliente <= MuchaHumedad && Caliente
<=NormalTierra) Altura[contador] = Caliente;
else if ( FrioTierra <= MuchaHumedad && FrioTierra <= NormalTierra)
Altura[contador] = FrioTierra;
else if ( MuchaHumedad <= NormalTierra) Altura[contador] = MuchaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 70;
COG();
}
//Regla36
if(MuchaHumedadTierra !=0)
{
contador++;
if (Caliente <= FrioTierra && Caliente <= MuchaHumedad && Caliente
<=MuchaHumedadTierra) Altura[contador] = Caliente;
else if (FrioTierra <= MuchaHumedad && FrioTierra <= MuchaHumedadTierra)
Altura[contador] = FrioTierra;
else if (MuchaHumedad <= MuchaHumedadTierra) Altura[contador] =
MuchaHumedad;
else Altura[contador] = MuchaHumedadTierra;
Area();
Factor = 70;
COG();
}}
if(Templado*CalienteTierra !=0)
{
if(PocaHumedad !=0)
{
//Regla37
if(PocaHumedadTierra !=0)
{

```

```

contador++;
if (Templado <= CalienteTierra && Templado <= PocaHumedad && Templado
<=PocaHumedadTierra) Altura[contador] = Templado;
else if ( CalienteTierra <= PocaHumedad && CalienteTierra <=
PocaHumedadTierra) Altura[contador] = CalienteTierra;
else if (PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla40
if(NormalTierra !=0)
{
contador++;
if (Templado <= CalienteTierra && Templado <= PocaHumedad && Templado
<=NormalTierra) Altura[contador] = Templado;
else if (CalienteTierra <= PocaHumedad && CalienteTierra <= NormalTierra)
Altura[contador] = CalienteTierra;
else if ( PocaHumedad <= NormalTierra) Altura[contador] = PocaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}
//Regla43
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Templado <= CalienteTierra && Templado <= PocaHumedad && Templado
<=MuchaHumedadTierra) Altura[contador] = Templado;
// else if (CalienteTierra <= PocaHumedad && CalienteTierra <=
MuchaHumedadTierra) Altura[contador] = CalienteTierra;
// else if ( PocaHumedad <= MuchaHumedadTierra) Altura[contador] =
PocaHumedad;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
8
C:\Users\Miranda\Desktop\Manuel
Miranda\Maestria\C_Compiler\Programas\ControladorDifuso\Ejemplo5.CuatroVa
riablesFisico\Fusificacion2.c
}
//Regla38
if(Normal !=0)
{
if(PocaHumedadTierra !=0)
{
contador++;
if (Templado <= CalienteTierra && Templado <= Normal && Templado
<=PocaHumedadTierra) Altura[contador] = Templado;
else if ( CalienteTierra <= Normal && CalienteTierra <=
PocaHumedadTierra) Altura[contador] = CalienteTierra;
else if ( Normal <= PocaHumedadTierra) Altura[contador] = Normal;

```

```

else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla41
if(NormalTierra !=0)
{
contador++;
if (Templado <= CalienteTierra && Templado <= Normal && Templado
<=NormalTierra) Altura[contador] = Templado;
else if (CalienteTierra <= Normal && CalienteTierra <= NormalTierra)
Altura[contador] = CalienteTierra;
else if (Normal <= NormalTierra) Altura[contador] = Normal;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}
//Regla44
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Templado <= CalienteTierra && Templado <= Normal && Templado
<=MuchaHumedadTierra) Altura[contador] = Templado;
// else if ( CalienteTierra <= Normal && CalienteTierra <=
MuchaHumedadTierra) Altura[contador] = CalienteTierra;
// else if (Normal <= MuchaHumedadTierra) Altura[contador] = Normal;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}
if(MuchaHumedad !=0)
{
//Regla39
if(PocaHumedadTierra !=0)
{
contador++;
if (Templado <= CalienteTierra && Templado <= PocaHumedad && Templado
<=PocaHumedadTierra) Altura[contador] = Templado;
else if ( CalienteTierra <= PocaHumedad && CalienteTierra <=
PocaHumedadTierra) Altura[contador] = CalienteTierra;
else if (PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla42
if(NormalTierra !=0)
{
contador++;
if (Templado <= CalienteTierra && Templado <= MuchaHumedad && Templado
<=NormalTierra) Altura[contador] = Templado;

```

```

else if (CalienteTierra <= MuchaHumedad && CalienteTierra <=
NormalTierra) Altura[contador] = CalienteTierra;
else if (MuchaHumedad <= NormalTierra) Altura[contador] = MuchaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}
//Regla45
//if(MuchaHumedadTierra !=0)
// {
// contador++;
// if (Templado <= CalienteTierra && Templado <= MuchaHumedad && Templado
<=MuchaHumedadTierra) Altura[contador] = Templado;
// else if (CalienteTierra <= MuchaHumedad && CalienteTierra <=
MuchaHumedadTierra) Altura[contador] = CalienteTierra;
// else if (MuchaHumedad <= MuchaHumedadTierra) Altura[contador] =
MuchaHumedad;
// else Altura[contador] = MuchaHumedadTierra;
// Area();
// Factor = 25;
// COG();
// }
}}
if(Frio*TempladoTierra !=0)
{
if(PocaHumedad !=0)
{
//Regla46
if(PocaHumedadTierra !=0)
{
contador++;
if (Frio <= TempladoTierra && Frio <= PocaHumedad && Frio
<=PocaHumedadTierra) Altura[contador] = Frio;
else if (TempladoTierra <= PocaHumedad && TempladoTierra <=
PocaHumedadTierra) Altura[contador] = TempladoTierra;
else if (PocaHumedad <= PocaHumedadTierra) Altura[contador] =
PocaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla49
if(NormalTierra !=0)
{
contador++;
if (Frio <= TempladoTierra && Frio <= PocaHumedad && Frio <=NormalTierra)
Altura[contador] = Frio;
else if ( TempladoTierra <= PocaHumedad && TempladoTierra <=
NormalTierra) Altura[contador] = TempladoTierra;
else if ( PocaHumedad <= NormalTierra) Altura[contador] = PocaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 70;
}
}
9

```

```

C:\Users\Miranda\Desktop\Manuel
Miranda\Maestria\C_Compiler\Programas\ControladorDifuso\Ejemplo5.CuatroVa
riablesFisico\Fusificacion2.c
COG();
}}
if(Normal!=0)
{
//Regla47
if(PocaHumedadTierra !=0)
{
contador++;
if (Frio <=TempladoTierra && Frio <= Normal && Frio <=PocaHumedadTierra)
Altura[contador] = Frio;
else if (TempladoTierra <= Normal && TempladoTierra <= PocaHumedadTierra)
Altura[contador] = TempladoTierra;
else if (Normal <= PocaHumedadTierra) Altura[contador] = Normal;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla50
if(NormalTierra !=0)
{
contador++;
if (Frio <=TempladoTierra && Frio <= Normal && Frio <=NormalTierra)
Altura[contador] = Frio;
else if (TempladoTierra <= Normal && TempladoTierra <= NormalTierra)
Altura[contador] = TempladoTierra;
else if (Normal <= NormalTierra) Altura[contador] = Normal;
else Altura[contador] = NormalTierra;
Area();
Factor = 70;
COG();
}}
//Regla48
if(MuchaHumedad*PocaHumedadTierra !=0)
{
contador++;
if (Frio <=TempladoTierra && Frio <= MuchaHumedad && Frio
<=PocaHumedadTierra) Altura[contador] = Frio;
else if (TempladoTierra <= MuchaHumedad && TempladoTierra <=
PocaHumedadTierra) Altura[contador] = TempladoTierra;
else if ( MuchaHumedad <= PocaHumedadTierra) Altura[contador] =
MuchaHumedad;
else Altura[contador] = PocaHumedadTierra;
Area();
Factor = 70;
COG();
}
//Regla51
if(MuchaHumedad*NormalTierra !=0)
{
contador++;
if (Frio <=TempladoTierra && Frio <= MuchaHumedad && Frio <=
NormalTierra) Altura[contador] = Frio;

```

```

else if (TempladoTierra <= MuchaHumedad && TempladoTierra <=
NormalTierra) Altura[contador] = TempladoTierra;
else if (MuchaHumedad <= NormalTierra) Altura[contador] = MuchaHumedad;
else Altura[contador] = NormalTierra;
Area();
Factor = 50;
COG();
}}
if (Contador == 0)
{
contador++;
Altura[contador] = 1;
Area();
Factor = 0;
COG();
}
//////// Calcula parte de la formula para COG
//////// Mediante la formula:
//////// Sumatoria(bi(Area debajo funcion membresia)/Sumatoria de Areas
//////// Donde:
//////// bi(Factor): Es el centro de la funcion membresia //////////

VArea = 0;
TotalArea = 0;
for(a=1; a<= 8; a++)
{
VArea = 40*(Altura[a] - (Altura[a]* Altura[a])/2);
//VArea = AArea[a];
TotalArea = TotalArea + VArea;
}
VCOG = 0;
TotalCOG = 0;
for(a=1; a<= 8; a++)
{
VCOG = ACOG[a];
TotalCOG = TotalCOG + VCOG;
}
VCOG = TotalCOG/TotalArea;
return;
}
int cadena1[2], atemperatura[8]; // Valor recibido por comunicacion
serial
int cadena2[2], ttemperatura[8]; // Valor recibido por comunicacion
serial
int cadena3[2], thumedad[8]; // Valor recibido por comunicacion serial
int cadena4[2], ahumedad[8]; // Valor recibido por comunicacion serial
#INT_RDA // Servicio de interrupcion por RS232
void rda_isr()
{
printf(lcd_putc, "\fRECIBIENDO");
printf(lcd_putc, "\nDATOS");
10
C:\Users\Miranda\Desktop\Manuel
Miranda\Maestria\C_Compiler\Programas\ControladorDifuso\Ejemplo5.CuatroVa
riablesFisico\Fusificacion2.c
cadena1[0]= getc(); // Lee un caracter y retorna un valor entero
cadena1[0]= cadena1[0]-48; // Resta el valor entero menos 0 (En ASCII)

```

```

//printf(lcd_putc, "\fACEPTADA1 ");
while( !kbhit() ){} // Espera a que exista un evento
cadena1[1]= getc();
cadena1[1]= cadena1[1]-48;
//printf(lcd_putc, "\fACEPTADA2 ");
while( !kbhit() ){}
cadena2[0]= getc(); // Lee un caracter y retorna un valor entero
cadena2[0]= cadena2[0]-48; // Resta el valor entero menos 0 (En ASCII)
//printf(lcd_putc, "\fACEPTADA3 ");
while( !kbhit() ){} // Espera a que exista un evento
cadena2[1]= getc();
cadena2[1]= cadena2[1]-48;
//printf(lcd_putc, "\fACEPTADA4 ");
while( !kbhit() ){}
cadena3[0]= getc(); // Lee un caracter y retorna un valor entero
cadena3[0]= cadena3[0]-48; // Resta el valor entero menos 0 (En ASCII)
//printf(lcd_putc, "\fACEPTADA5 ");
while( !kbhit() ){} // Espera a que exista un evento
cadena3[1]= getc();
cadena3[1]= cadena3[1]-48;
//printf(lcd_putc, "\fACEPTADA6 ");
while( !kbhit() ){}
cadena4[0]= getc(); // Lee un caracter y retorna un valor entero
cadena4[0]= cadena4[0]-48; // Resta el valor entero menos 0 (En ASCII)
while( !kbhit() ){} // Espera a que exista un evento
cadena4[1]= getc();
cadena4[1]= cadena4[1]-48;
delay_ms(1000);
sprintf(atemperatura,"%d%d",cadena1[0],cadena1[1]); // Concatenar cadena
p = atof(atemperatura); // Convierte una cadena a su
printf(lcd_putc, "\fATemp = %1.2f",p); // valor numérico(flotante)
lcd_putc(223); // Imprime simbolo de grados ( ° )
lcd_putc("C "); // Imprime 'C '
lcd_gotoxy(1, 2);
sprintf(ahumedad,"%d%d",cadena4[0],cadena4[1]); // Concatenar cadena
s = atof(ahumedad); // Convierte una cadena a su
printf(lcd_putc, "AHumd = %1.2f %%",s); // valor numérico(flotante)
delay_ms(1000);
sprintf(ttemperatura,"%d%d",cadena2[0],cadena2[1]); // Concatenar cadena
tt = atof(ttemperatura); // Convierte una cadena a su
t = tt*0.1;
printf(lcd_putc, "\fTTemp = %1.2f",t); // valor numérico(flotante)
lcd_putc(223); // Imprime simbolo de grados ( ° )
lcd_putc("C "); // Imprime 'C '
lcd_gotoxy(1, 2);
sprintf(thumedad,"%d%d",cadena3[0],cadena3[1]); // Concatenar cadena
u = atof(thumedad); // Convierte una cadena a su
printf(lcd_putc, "THumd = %1.2f %%",u); // valor numérico(flotante)
delay_ms(1000);
Fusificacion(); // Fusifica senales analogicas
ReglaAndDefusificacion(); // Metodo de inferencia
VisualizacionInterrupcion(); // Visualiza valores
UnaVez = 0;
return;
}

```

```
//////////
```

```
Programa Principal
```

```
//////////
```

```

void main()
{
  setup_oscillator(OSC_8Mhz|OSC_INTRC|OSC_PLL_OFF);
  setup_wdt(WDT_ON);
  lcd_init(); // Inicializar LCD
  enable_interrupts(INT_RDA); // Habilita interrupción por recepción RS232
  enable_interrupts(GLOBAL); // Habilita interrupcion global
  UnaVez = 0;
  for(;;){ // Ciclo principal del programa
    restart_wdt();
    printf(lcd_putc, "\fMANDANDO");
    printf(lcd_putc, "\nACTIVACION");
    delay_ms(500);
    if (UnaVez == 0)
    {
      UnaVez = 1;
      printf("M"); // Caracter para activar sensor
    }
    else
      printf("M");
  }
}

```

7.5 Electrovalvula

```

#include <18f4550.h>
#define ADC=10
#define FUSES WDT2048,NOBROWNOUT,NOLVP,INTRC_IO
#define use delay(internal=500000)
#define use rs232(baud=9600, xmit=pin_c6, rcv=pin_c7, bits=8, parity=N, stop=1)
#include <stdlib.h> // Librerias para usar la funcion atof
#include <math.h>
int16 ang1,ang3,clave;
char Activacion [ 1 ]; // Valor recibido por comunicacion serial
char cadena1[2], vCOG[8]; // Valor recibido por comunicacion serial
float ang2;
#define INT_RDA // Servicio de interrupcion por RS232
void rda_isr()
{
  Activacion [ 0 ]= getc (); // Lee un caracter y retorna un valor
  if ( Activacion [ 0 ] == 'V' )
  {
    restart_wdt();
    cadena1[0]= getc(); // Lee un caracter y retorna un valor entero
    cadena1[0]= cadena1[0]-48; // Resta el valor entero menos 0 (En ASCII)
    while( !kbhit() ){} // Espera a que exista un evento
    cadena1[1]= getc(); // Lee un caracter y retorna un valor entero
    cadena1[1]= cadena1[1]-48; // Resta el valor entero menos 0 (En ASCII)
    sprintf(vCOG,"%d%d",cadena1[0],cadena1[1]); // Concatenar cadena
    ang2 = atof(vCOG);
    if (ang2 <= 44) ang2=45;
    ang2 = 11 + (1.0)*(ang2-42);
    ang2 = fabs(ang2);
    ang3 =(int)ang2;
    if (11 <= ang3 <= 39)
      set_pwm1_duty((int16)ang3); //Configura el PWM
  }
}

```

```

else
{
}
}
void main()
{
setup_oscillator(OSC_500KHz|OSC_INTRC);
setup_wdt(WDT_ON);
enable_interrupts(INT_RDA); // Habilita interrupción por recepción RS232
enable_interrupts(GLOBAL); // Habilita interrupcion global
setup_timer_2(t2_div_by_16,155,1); //Establecer frec. PWM a 50Hz
setup_ccp1(ccp_pwm); //Configurar modulo CCP1 en modo PWM
ang1 = 14; //int lmin=11,lmax=40;
set_pwm1_duty((int16)ang1); //Configura el PWM} //Valor anterior 20
delay_ms(2000);
ang1 = 37; //int lmin=11 Cerrado,lmax=40(37)Abierto
set_pwm1_duty((int16)ang1); //Configura el PWM}
while(TRUE)
{
delay_ms(500);
}
}

```

7.6 Cuestionario dirigido al humano experto.

Preguntas para obtener el modelo de reglas difusas para un huerto de manzanas.

¿Cree que estas cuatro variables: temperatura del aire, temperatura del suelo, humedad del aire y humedad del suelo, sean esenciales para tomar la decisión de regar un huerto de manzanas?

Las siguiente preguntas son para determinar las variables lingüísticas (Etiquetas) que nos ayudaran a determinar las reglas para saber cuándo regar el huerto. Note que los rangos pueden estar sobrepuestos.

Para entender el objetivo de este documento, supongamos que se quiere controlar un Carro, con las entradas "Velocidad del carro" y "Distancia del carro a otro carro" y la salida Frenos. Primero será necesario definir las "Etiquetas" y Rangos

Ejemplo:

Supongamos la velocidad de un carro, por lo tanto las etiquetas y rangos podrían ser:

Etiquetas	Rangos	O para otra persona podrían ser como lo muestra la	Etiquetas	Rangos
Lento	0 a 50 Km/h		MuyLento	0 a 20 Km/h
Normal	40 a 70 Km/h		Lento	10 a 30 Km/h
Rápido	60 a 100km/h		Normal	20 a 40 km/h
			Rápido	30 a 50 Km/h

	tabla de la izquierda	MuyRapido	40 a 60 Km/h
--	-----------------------	-----------	--------------

Para la otra entrada "Distancia de carro a carro podrían ser" las etiquetas y rangos podrían ser.

Etiquetas	Rangos	O para otra persona podrían ser como lo muestra la tabla de la izquierda	Etiquetas	Rangos
Cerca	0.5 a 1 metros		MuyCerca	0.5 a 1 metros
Normal	1 a 1.5 metros		Cerca	0.7 a 1.5 metros
Lejos	2 a 3 metros		Normal	1 a 2 metros
			Lejos	1.8 a 3 metros
			MuyLejos	2.5 a 10 metros

Lo mismo se tendrá que hacer para las variables de entrada al sistema de riego del huerto.

Si considera que la temperatura del aire es esencial para determinar el riego, ¿Está usted de acuerdo con un rango de medición de temperatura de -10°C a 30°C para determinar posibles decisiones de riego con las siguientes etiquetas y rangos??

EtiquetasPropuestas	RangosPropuestos
Frio	-10°C a 10°C
Templado	0°C a 20°C
Caliente	10°C a 30°C

Si no está de acuerdo con los rangos o con el número de etiquetas (Frio, Templado, Caliente), por favor especifique lo más conveniente para usted.

EtiquetasPropuestas	RangosPropuestos
Frio	
Templado	
Caliente	

EtiquetasPropuestas	RangosPropuestos
Muy Frio	
Frio	
Templado	
Caliente	
MuyCaliente	

Si considera que la temperatura del suelo es esencial para determinar el riego, ¿Está usted de acuerdo con un rango de temperatura de 5°C a 21°C para determinar posibles decisiones de riego, con las siguientes etiquetas?

EtiquetasPropuestas	RangosPropuestos
Frio	5°C a 13°C
Templado	9°C a 17°C
Caliente	13°C a 21°C

Si no está de acuerdo con los rangos o con el número de etiquetas (Frio, Templado, Caliente) se pueden cambiar, por favor especifique lo más conveniente para usted.

EtiquetasPropuestas	RangosPropuestos
Frio	
Templado	
Caliente	

EtiquetasPropuestas	RangosPropuestos
Muy Frio	
Frio	
Templado	
Caliente	
MuyCaliente	

Si considera que la humedad del aire es esencial para determinar el riego, ¿Está usted de acuerdo con un rango de medición de humedad de 0% a 100% para determinar posibles decisiones de riego, con las siguientes etiquetas??

EtiquetasPropuestas	RangosPropuestos
PocaHumedad	0% a 50%
Normal	25% a 75%
MuchaHumedad	50% a 100%

Si no está de acuerdo con los rangos o con el número de etiquetas (PocaHumedad, Normal, MuchaHumedad) por favor especifique lo más conveniente para usted.

EtiquetasPropuestas	RangosPropuestos
PocaHumedad	
Normal	
MuchaHumedad	

EtiquetasPropuestas	RangosPropuestos
Seco	
MuySeco	
Normal	
Humedo	
MuyHumedo	

¿Si considera que la humedad del suelo es esencial para determinar el riego, está usted de acuerdo con un rango de medición de humedad de 0% a 100% para determinar posibles decisiones de riego, con las siguientes etiquetas??

EtiquetasPropuestas	RangosPropuestos
PocaHumedad	0% a 50%
Normal	25% a 75%
MuchaHumedad	50% a 100%

Si no está de acuerdo con los rangos o con el número de etiquetas (PocaHumedad, Normal, MuchaHumedad), por favor especifique lo más conveniente para usted.

EtiquetasPropuestas	RangosPropuestos
PocaHumedad	
Normal	
MuchaHumedad	

EtiquetasPropuestas	RangosPropuestos
Seco	
MuySeco	
Normal	
Humedo	
MuyHumedo	

Para regar el huerto se usara una válvula proporcional, las etiquetas propuestas serán:

EtiquetasPropuestas	RangosPropuestos
TotalmenteCerrado	
MedioAbierto	
MaximoAbierto	

Si no está de acuerdo con el número de etiquetas por favor especificar

EtiquetasPropuestas	

Una vez definidos los rangos y variables lingüísticas es necesario crear las reglas para regar el huerto, por ejemplo suponiendo que se hayan aceptado todas las etiquetas y variables anteriores propuestas podríamos tener los siguientes ejemplos:

Si la temperatura del aire es **FRIO** + temperatura del suelo es **FRIO** + Humedad del aire **POCA** + Humedad del suelo **POCA** = Válvula es **MaximoAbierto**

Esto quiere decir que si la temperatura del aire fue fría y la temperatura del suelo es frío y la humedad del aire es poca y la humedad del suelo es poca entonces se ha determinado que la válvula se abra completamente.

También podríamos tener:

Si la temperatura del aire es **FRIO** + temperatura del suelo es **FRIO** + Humedad del aire **NORMAL** + Humedad del suelo **NORMAL** = Válvula es **MedioAbierto**

O También:

Si la temperatura del aire es **CALIENTE** + temperatura del suelo es **CALIENTE** + Humedad del aire **MuyHumedo** + Humedad del suelo **MuyHumedo** = Valvula es **Cerrada**

Por favor especificar todas las reglas pertinentes como Experto en la materia. Al menos serían necesarias 45 reglas pero el sistema puede soportar bastante información por lo que usted puede especificar todas las reglas que crea conveniente:

No. De Regla	Temp Aire	Temp Tierra	Humedad Aire	Humedad Tierra	Valvula
Ejemplo	FRIO	FRIO	NORMAL	NORMAL	MedioAbierto
1					
2					
3					
4					
5					
6					
7					

8					
9					
10					
11					
12					
13					
14					