



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Tecnológico Nacional de México

**Centro Nacional de Investigación
y Desarrollo Tecnológico**

Tesis de Maestría

**Diseño de un sistema de riego sustentable mediante
lógica difusa.**

Presentada por

Ing. Alexis Noé Rogel Rivera

como requisito para la obtención del grado de
Maestro en Ciencias de la Ingeniería

Directora de tesis

Dra. Gloria Lilia Osorio Gordillo

Codirector de tesis

Dr. Vitervo López Caballero

Cuernavaca, Morelos, México. Junio de 2026



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Centro Nacional de Investigación y Desarrollo Tecnológico
Coordinación de Ciencias de la Ingeniería

Cuernavaca, Mor., 28/ mayo/ 2026

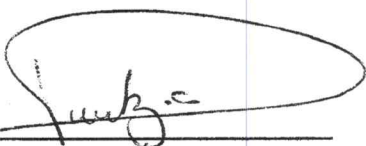
OFICIO No. MCI/033/2026
CENIDET-AC-004-M14-OFICIO

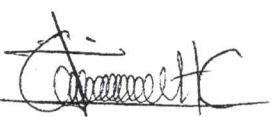
Asunto: **Aceptación de documento de tesis**

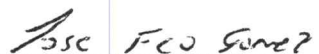
DR. CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO
PRESENTE

Por este conducto, los integrantes de Comité Tutorial del estudiante **ALEXIS NOÉ ROGEL RIVERA**, con número de control M24CE011, de la Maestría en Ciencias de la Ingeniería, le informamos que hemos revisado el trabajo de tesis de grado titulado **"DISEÑO DE UN SISTEMA DE RIEGO SUSTENTABLE MEDIANTE LÓGICA DIFUSA"** y hemos encontrado que se han atendido todas las observaciones que se le indicaron, por lo que hemos acordado aceptar el documento de tesis y le solicitamos la autorización de impresión definitiva.


DRA. GLORIA LILIA OSORIO GORDILLO
Directora de tesis


DR. VITERVO LÓPEZ CABALLERO
Codirector de Tesis


MTRA. DANIELA HERNÁNDEZ CASTAÑEDA
Revisora 1


DR. JOSÉ FRANCISCO GÓMEZ
AGUILAR
Revisor 2

C.c.p. Depto. Servicios Escolares.
Expediente / Estudiante
JAL/ang



★ 09 JUN 2026 ★
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SUBDIRECCIÓN ACADÉMICA

RECIBIDO
2026
año de
Margarita Maza



cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico



Apatzingán 212, Palmira, 62490 Cuernavaca, Mor., México
Tel. 01 (777) 3627770, ext. 4106,
e-mail: coord.ingenieria@cenidet.tecnm.mx | cenidet.tecnm.mx



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Centro Nacional de Investigación y Desarrollo Tecnológico
Subdirección Académica

Cuernavaca Mor, 03/junio/2026

Oficio No. SAC/150/2026

Asunto: Autorización de impresión de tesis

ALEXIS NOÉ ROGEL RIVERA
CANDIDATO AL GRADO DE MAESTRO EN
CIENCIAS DE LA INGENIERÍA
P R E S E N T E

Por este conducto, tengo el agrado de comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"DISEÑO DE UN SISTEMA DE RIEGO SUSTENTABLE MEDIANTE LÓGICA DIFUSA"**, ha informado a esta Subdirección Académica, que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

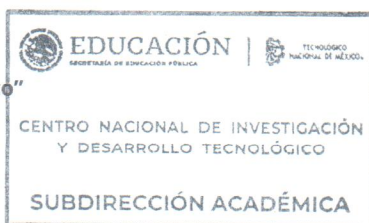
Esperando que el logro del mismo sea acorde con sus aspiraciones profesionales, reciba un cordial saludo.

ATENTAMENTE

Excelencia en Educación Tecnológica®

"Conocimiento y Tecnología al servicio de México"

CARLOS MANUEL ASTORGA ZARAGOZA
SUBDIRECTOR ACADÉMICO



c.c.p. Coordinación de Ciencias de la Ingeniería
Departamento de Servicios Escolares

CMAZ/lmz



★ 09 JUN 2026 ★

RECIBIDO
CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO
SERVICIO DE ASESORIA



2026
año de
Margarita Maza

cenidet
Centro Nacional de Investigación y Desarrollo Tecnológico



Interior Internado Palmira S/N, Col. Palmira,
C.P. 62490, Cuernavaca, Morelos Tel. 01 (777) 3627770, ext. 4106,
e-mail: acad_cenidet@tecnm.mx | cenidet.tecnm.mx

DEDICATORIA

A Dios por siempre guiarme en el camino del bien, por llenarme de bendiciones y por permitirme culminar una etapa más en mi vida.

A mis padres Noé Rogel y Blanca Rivera, para mí es un orgullo ser su hijo y les dedico este trabajo en su honor, por todo el amor, apoyo, paciencia y consejos que siempre me han brindado.

A mi hermano Daniel y mi sobrino Matías que siempre me han alegrado la vida con sus elocuencias, representan gran importancia en mi vida.

A mis abuelitos Juan Rivera y Estela Bobadilla, quiero honrar su memoria a través de este trabajo, porque a pesar de ya no estar presentes físicamente, sé perfectamente que siempre me cuidan y me protegen en alma y espíritu.

A mi novia Daniela Yaravit, en reconocimiento a todo el apoyo y amor que siempre me ha brindado, mi gran compañera de vida.

A Blacky, por todo este tiempo que siempre ha estado a mi lado en mis momentos más difíciles y de soledad.

AGRADECIMIENTOS

Deseo agradecer por la realización de este trabajo a Dios, que me permitió llegar hasta esta estancia, siempre cuidando de mí y de mis seres queridos que son lo más importante para mí, por toda la fe y fortaleza que me da para seguir adelante.

A mis padres, sin su amor y su apoyo, yo no sería la persona que soy hoy en día, y nada de lo que he logrado sería posible. Gracias por todo su amor incondicional y por siempre creer en mí.

A mi directora, la Dra. Gloria Lilia Osorio Gordillo por su tiempo, paciencia y consejos durante mi etapa en la maestría. Gracias por la confianza que me brindó para la realización de este trabajo y elaboración de la tesis.

A mi codirector, el Dr. Vitervo López Caballero por su colaboración para la realización de este trabajo, gracias por el apoyo y orientación.

A mis revisores, la Mtra, Daniela Hernández Castañeda y el Dr. José Francisco Gómez Aguilar, gracias por todos sus comentarios y observaciones las cuales fueron de gran importancia para la mejoría de este trabajo.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por abrirme las puertas a esta gran institución para seguir creando mi formación académica, junto a todos mis profesores y compañeros de la maestría, gracias por todas las convivencias y alegrías.

A la Secretaria de Ciencia, Humanidades Tecnología e Innovación (SECIHTI) por el apoyo económico recibido durante mi maestría.

A mi amigo de la infancia David Toledo, gracias por tu amistad y apoyo durante todos estos años, es algo que aprecio demasiado y de lo cual siempre estaré agradecido.

Tabla de contenido

Lista de Figuras	I
Lista de Tablas.....	II
1 Capítulo 1: Introducción.....	3
1.1 Introducción.....	3
1.2 Revisión del estado del arte	4
1.3 Antecedentes.....	6
1.4 Justificación	9
1.5 Alcances y limitaciones	10
1.6 Objetivos.....	11
1.6.1 Objetivo general.	11
1.6.2 Objetivos específicos.....	11
1.7 Organización del documento	12
2 Capítulo 2: Marco Teórico	13
2.1 El chile como emblema de México	13
2.1.1 Chile jalapeño (<i>Capsicum annuum</i> L.)	13
2.1.2 Condiciones de cosecha del chile jalapeño.....	14
2.2 Sistemas de riego	16
2.3 Lógica difusa	18
2.3.1 Conjunto difuso	18
2.3.2 Controlador difuso.....	21
Fusificación	22
Desfusificación	23
2.4 Sistema embebido.....	24
2.5 Sensores	25
2.6 Plataforma Arduino	26
2.7 MATLAB	27
3 Capítulo 3: Construcción e instrumentación del prototipo	28
3.1 Diseño y construcción del prototipo.....	28
3.2 Selección de sensores	29
3.3 Otros materiales.....	34
3.4 Montaje final del prototipo	36
4 Capítulo 4: Diseño y planteamiento del controlador difuso	40

4.1	Programación e implementación del esquema de monitoreo de las condiciones ambientales.	40
4.2	Definición de variables del sistema de control difuso	42
4.2.1	Definición de funciones de membresía del sistema de control difuso	43
4.3	Diseño y selección de reglas para el sistema de riego	44
4.4	Mecanismo de inferencia difusa	47
4.5	Desfusificación (Toma de decisiones).....	48
5	Capítulo 5: Simulación y experimentación del prototipo.	50
5.1	Simulación en Matlab	50
5.2	Pruebas en el Protoboard	51
5.3	Resultados experimentales y ajustes finales	54
5.3.1	Ajustes finales	54
5.3.2	Primer experimento: Riego manual a todas las plántulas.....	55
5.3.3	Segundo experimento: Riego manual y adición del riego difuso.....	56
5.3.4	Tercer experimento: Variación de riegos manuales	59
5.4	Evaluación del desempeño del sistema de riego difuso	61
5.4.1	Control de humedad del suelo	65
6	Capítulo 6: Conclusiones.....	69
6.1	Conclusiones Generales.....	69
6.2	Trabajos futuros.....	70
7	Referencias bibliográficas	71
8	Anexos	74
8.1	Retribuciones sociales	74
		74
8.2	Producto académico.....	75
8.3	Código para el monitoreo de variables	76
8.4	Reglas difusas del sistema	79
8.5	Código completo del sistema de riego en Arduino.....	82
8.6	Código del controlador difuso en Visual Studio	87

Lista de Figuras

Figura 1 Chile jalapeño verde (<i>Capsicum Annuum</i>)	14
Figura 2 Representación gráfica de una función de membresía triangular: En la izquierda Si $a \leq x \leq b$, en la derecha Si $b \leq x \leq c$	21
Figura 3 Diagrama de flujo del controlador difuso	22
Figura 4 Representación en funciones de membresía para $T=29^{\circ}\text{C}$	22
Figura 5 Diseño de la maceta comercial elaborada en SolidWorks	28
Figura 6 Montado y ensamblado de macetas comerciales con el arco y plástico protector.	29
Figura 7 Sensor capacitivo de humedad del suelo I2C.....	31
Figura 8 Sensor DHT11 para temperatura y humedad ambiente	33
Figura 9 Sensor BH1750 para radiación solar.....	34
Figura 10 Colocación del sensor capacitivo de humedad enterrado lo más cerca posible de la raíz de la planta	37
Figura 11 Fijación de sensores de temperatura ambiente (DHT11) y sensor de radiación solar (BH1750)	37
Figura 12 Visualización de condiciones ambientales en tiempo real a través de la pantalla LCD	38
Figura 13 Bomba de agua sumergible para suministrar agua al cultivo.....	38
Figura 14 Instalación de goteros comerciales a 1-2 centímetros del tallo de la planta	39
Figura 15 Diagrama de flujo para el sistema de monitoreo de condiciones ambientales.....	41
Figura 16 Comportamiento de condiciones ambientales a lo largo de un día con un tiempo de muestreo de 5 minutos	41
Figura 17 Comportamiento de condiciones ambientales con un tiempo de muestreo de 15 minutos	42
Figura 18 Funciones de membresía propuestas para el sistema difuso	44
Figura 19 Diagrama de flujo del método de inferencia Mamdani.....	47
Figura 20 Recorte del conjunto de salida "Medio" a 0.375 después de la etapa de defusificación	49
Figura 21 Simulaciones con las variables ambientales hechas con Matlab	49
Figura 22 Creación del controlador difuso con funciones de membresía en Matlab	50
Figura 23 Creación de reglas difusas a través de Matlab	51
Figura 24 Programación del controlador difuso en Visual Studio	52
Figura 25 Agrupación del fis_header con el archivo de Arduino	52
Figura 26 Implementación de librería en el programa principal para la lectura del fis_header	53
Figura 27 Confirmación de almacenamiento de datos por medio de la pantalla LCD.....	53
Figura 28 Montaje completo del prototipo	54
Figura 29 Sensor de flujo YF-S401.....	54
Figura 30 Plántulas de chile jalapeño	55
Figura 31 Floración del cultivo de chile jalapeño	56
Figura 32 Agua no aprovechada por el cultivo debido a un exceso de riego	57
Figura 33 Gráficas de variables durante el segundo experimento.....	58

Figura 34 Histograma de cantidad de litros suministrados por el riego controlado mediante lógica difusa.....	59
Figura 35 Visualización del riego por goteo en el sistema.....	60
Figura 36 Comparativa de crecimiento en ambos riegos, del lado izquierdo el riego manual y del lado derecho el riego controlado por lógica difusa.....	60
Figura 37 Consumo de agua de ambos riegos durante el periodo de experimentación.....	62
Figura 38 Histograma de los litros suministrados de ambos riegos durante el periodo de experimentación.....	63
Figura 39 Comportamiento de variables en un día completo y la correlación con la activación de la bomba para suministrar agua.....	64
Figura 40 Gráfica del consumo de agua de ambos riegos (línea azul representa el riego difuso, línea naranja el riego manual)	65
Figura 41 Comportamiento de la humedad del suelo durante el periodo de experimentación con el sistema de riego difuso.....	66
Figura 42 Comportamiento de la humedad del suelo en un día completo durante el periodo de experimentación.....	66
Figura 43 Ahorro acumulado de agua durante el periodo de experimentación entre ambos riegos	67
Figura 44 Frutos finales posterior al periodo de experimentación	68

Lista de Tablas

Tabla 1 Principales cultivos con agricultura protegida en Morelos 2022	9
Tabla 2 Datos aproximados sobre la siembra y cosecha del chile jalapeño	14
Tabla 3 Principales métodos de riego.....	16
Tabla 4 Representación matemática de una función de membresía triangular	21
Tabla 5 Sensores de humedad del suelo comerciales más reportados en trabajos de agricultura.....	30
Tabla 6 Sensores de temperatura ambiente más reportados en trabajos de investigación ...	32
Tabla 7 Sensores de radiación/iluminación solar más comerciales.....	33
Tabla 8 Lista de materiales completa	35
Tabla 9 Reglas difusas para la automatización del sistema de riego.....	45
Tabla 10 Valores de pertenencia de cada conjunto difuso en el sistema.....	61

Resumen

El presente trabajo describe el desarrollo de un sistema de riego automatizado basado en lógica difusa, diseñado para optimizar el uso del agua en el cultivo de chile jalapeño bajo condiciones ambientales del estado de Morelos. El sistema integra sensores de temperatura, humedad del suelo, humedad ambiental y radiación solar, cuyos datos son procesados por un controlador difuso implementado en un sistema embebido, en este caso, Arduino.

El sistema de inferencia difusa fue diseñado mediante la definición de funciones de membresía, reglas lingüísticas y un método de defusificación basado en el centroide. Además, con el objetivo de mejorar la organización y eficiencia del código, el controlador difuso fue implementado en un archivo independiente (fis_header.h), permitiendo su integración modular sin alterar o modificar componentes dentro del programa principal.

El sistema desarrollado permite la toma de decisiones en tiempo real para el control del riego, ajustando la activación de una bomba de agua sumergible de acuerdo con las condiciones ambientales detectadas. Asimismo, se implementó un sistema de monitoreo y registro de datos utilizando almacenamiento en tarjeta SD y un módulo de reloj en tiempo real (RTC).

Los resultados obtenidos demuestran que la implementación de la lógica difusa en sistemas de riego permite un control más flexible y eficiente en comparación con métodos tradicionales de riego, adaptándose a la variabilidad de las condiciones ambientales y contribuyendo a un uso más racional del recurso hídrico.

Palabras clave: Lógica difusa, riego, inferencia difusa, sensores, Arduino.

Abstract

This work describes the development of an automated irrigation system based on fuzzy logic, designed to optimize water usage in jalapeño pepper cultivation under the environmental conditions of the state of Morelos. The system integrates sensors for temperature, soil moisture, air humidity, and solar radiation, whose data are processed by a fuzzy controller implemented on an embedded system, in this case, Arduino.

The fuzzy inference system was designed through the definition of membership functions, linguistic rules, and a centroid-based defuzzification method. Additionally, in order to improve code organization and efficiency, the fuzzy controller was implemented in an independent file (`fis_header.h`), allowing its modular integration without altering or modifying components within the main program.

The developed system enables real-time decision-making for irrigation control by adjusting the activation of a submersible water pump according to the detected environmental conditions. Likewise, a monitoring and data logging system was implemented using SD card storage and a real-time clock (RTC) module.

The results obtained demonstrate that the implementation of fuzzy logic in irrigation systems allows for more flexible and efficient control compared to traditional irrigation methods, adapting to the variability of environmental conditions and contributing to a more rational use of water resources.

Keywords: Fuzzy logic, irrigation, fuzzy inference, sensors, Arduino

1 Capítulo 1: Introducción

1.1 Introducción

La agricultura se entiende como una actividad que ha existido desde que la humanidad se asentó en un solo lugar, con la finalidad de producir insumos y alimentos. Desde entonces, dicha actividad se ha convertido en una necesidad prioritaria. El conocimiento con el que se cuenta actualmente acerca de la buena práctica agrícola es resultado del conocimiento y experiencia que se comparte a través de generaciones de manera “empírica”, con el objetivo de perfeccionar la práctica de esta actividad creando una relación entre el cultivo y las condiciones del ecosistema. Aunado a lo anterior, el sector agrícola es la mayor fuente de empleo a nivel mundial, alrededor del 40% de la población mundial se sustenta gracias a este sector (Food And Agriculture Organization, 2023). Sin embargo, la producción agrícola tradicional ha tenido un desarrollo y crecimiento limitados, en numerosas ocasiones, se ha visto estancado, en mayor parte, porque los modos y prácticas de la agricultura tradicional siguen siendo hasta cierto punto “ancestrales”, de manera que aún se utilizan métodos de riego ineficientes, con mangueras, pipas, surcos o canales los cuales no priorizan las necesidades que tiene el cultivo, dando como resultado un desperdicio significativo del recurso hídrico. Afortunadamente, la agricultura tradicional ha ido creciendo de manera paralela de la mano con la ciencia/tecnología, facilitando y creando más herramientas y conocimiento para un uso correcto del recurso hídrico. Una de estas tecnologías es la lógica difusa y frente a este contexto, los sistemas de riego inteligentes han cobrado gran relevancia al permitir un control más preciso y adaptable para el suministro de agua en los cultivos. La implementación de la lógica difusa en los sistemas de riego es una de las alternativas más prometedoras para automatizar estos sistemas de riego, gracias a que esta lógica nos permite simular y generar decisiones a partir de reglas similares a las que usaría un agricultor experto basadas en la intuición y experiencia, por lo que nos permite trabajar con valores lingüísticos, en lugar de valores numéricos exactos permitiéndonos trabajar con los datos que los sensores nos proporcionen y que, en algunas ocasiones, son datos imprecisos o variables no lineales.

1.2 Revisión del estado del arte

Año de publicación	Título	Resumen
2023	Smart Irrigation Systems in Agriculture: A systematic review	Una revisión sistemática de la literatura disponible sobre sistemas de riego inteligentes. Se abordan temas en sistemas que utilizan técnicas de inteligencia artificial en la agricultura urbana y rural para optimizar este proceso con el fin de identificar aquellos que se están utilizando actualmente.
2023	A fuzzy logic-and Internet of Things-Based Smart Irrigation	Se observa una combinación del uso e implementación del Internet de las cosas y la lógica difusa, demostraron que el uso de un sistema de riego inteligente logra la conservación de agua y energía en comparación de los métodos tradicionales de riego.
2023	Design a solar system for fuzzy control of intelligent irrigation system.	El diseño de un control difuso para optimizar el consumo de agua y los insumos en Irán, que son zonas áridas y extremadamente secas por lo que el momento y la duración del riego son clave para lograr un riego sostenible.
2023	Implementation of fuzzy logic control method on chilli cultivation technology based smart drip irrigation.	Un sistema de riego por goteo basado en lógica difusa tomando como parámetros de incertidumbre la temperatura y la humedad del aire para el cultivo de chiles, lo que demostró que la implementación de la lógica difusa arrojó un éxito del 99.98%
2022	Intelligent Control of Irrigation Systems Using Fuzzy Logic Controller	Un sistema de control de riego inteligente basado en lógica difusa para el control automático de bombas de agua utilizadas en granjas e invernaderos, eliminando la intervención humana con sensores para el monitoreo de los parámetros climáticos y del suelo.

2021	An intelligent irrigation system based on fuzzy logic control: A case study for Moroccan oriental climate region	Proponen un sistema de riego basado en una estrategia inteligente y eficiente para el riego automático, adaptado a las condiciones climáticas del oriente de Marruecos. El sistema emplea lógica difusa para definir el tiempo de duración del riego de manera adaptativa y precisa, utilizando como valores de entrada del sistema difuso las condiciones climatológicas (temperatura del aire y radiación solar) y la humedad del suelo (contenido de agua). La recopilación de estos datos se realiza a través de sensores digitales y analógicos, utilizando la base de reglas difusas de Mamdani para tomar la mejor decisión en cada situación. Para examinar y evaluar el desempeño del sistema de riego seleccionaron una planta de manzana, que, según expertos, esta planta necesita un contenido de humedad del suelo por encima del 30%. Con base a los resultados, el uso del control difuso mantiene la humedad del suelo por encima del valor definido por el usuario, eliminando cualquier tipo de riesgo de que exista un sub-riego.
2020	Automated Irrigation System based on irrigation gates using fuzzy logic	diseñan un control de riego con lógica difusa bajo condiciones climáticas de Pakistán, en donde se habla de que cerca del 60% del agua se desperdicia durante el riego, esto debido a la falta de conocimiento sobre el riego y una deficiente temporización de este. En este trabajo utilizaron como variables de entrada la temperatura y la humedad del suelo, recolectando estos valores con sensores inalámbricos, ya que implementaron una base de datos y en base a ello, lograron programar y simular las reglas difusas empleadas en el sistema de riego para controlar el voltaje de salida de una electroválvula que se

		encargaba de regar un cultivo de papa obteniendo valores de un ahorro de agua de hasta 14 litros por planta.
2019	Fuzzy Logic Controller for Automation of an Autonomous Irrigation System Designed for Habanero Pepper.	Un controlador difuso basado en Sugeno con la implementación de 18 reglas del tipo si-entonces para hacer sostenible el riego en el cultivo del chile habanero, el cual es uno de los mas producidos en México, facilitando el control y monitoreo de este.
2018	Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms.	Un sistema de riego basado en una red neuronal para poder predecir la humedad del suelo mediante un marco temporal de una hora exacta basado en el aprendizaje automático.

1.3 Antecedentes

El recurso hídrico es el componente más importante para el desarrollo de las plantas en la agricultura y los sistemas de riego son importantes para el suministro de dicho recurso, además, están diseñados para aminorar las desventajas con las que cuenta un esquema de riego tradicional. Para poder cumplir estas proyecciones, en algunos trabajos de investigación ha sido posible detectar que el uso de nuevas tecnologías inteligentes tiende a ser una opción adecuada para evaluar y crear estrategias que permitan optimizar los parámetros de los sistemas de riego, sin sacrificar la calidad y productividad de la planta/cultivo. Dentro de la revisión del estado del arte 1.3 se presentan algunos artículos seleccionados que se analizaron con la finalidad de asimilar y comprender las ventajas con las que cuentan estas tecnologías aplicadas en sistemas de riego, y en particular, la lógica difusa, la cual es la tecnología empleada en este trabajo de investigación. Además, a continuación, se presentan algunos casos de estudio en trabajos anteriores sobre la implementación de la lógica difusa en algunos cultivos y los resultados obtenidos.

Caso 1: Cultivo de tomate. En la investigación de Molina (2003) se plantea un sistema de control difuso en un invernadero de aproximadamente una superficie de 2000 m^2 ubicado en el estado de Querétaro, para una cosecha de alrededor de 5,400 plantas de tomate, donde las variables de entrada para el control implementado son la temperatura y la edad de la planta, además, la variable de salida de dicho control se trata del porcentaje de aumento de riego programado el cual es la variable lingüística de salida. Con base a estas variables, diseñaron un conjunto de 13 reglas aplicadas del tipo SI-ENTONCES, aunado a esto, para poder efectuar las mediciones del riego real, realizaron un muestreo de 10 plantas escogidas aleatoriamente. Finalmente, durante el desarrollo del proyecto, se apreció que la cantidad de riego propuesto se asemeja a la cantidad de riego tradicional que de manera empírica se aplica al invernadero. Las plantas no presentaron estrés hídrico y se obtuvieron resultados donde demostraron que al realizar el riego de forma empírica se utilizaban alrededor de 10,000 litros de agua, con el sistema de riego propuesto se utilizaron 9,300 litros aproximadamente, evidenciando que el sistema propuesto sí mostraba eficacia, sin embargo, únicamente se midieron 2 variables y se crearon 13 reglas, mientras que en otros trabajos se ha demostrado que entre más reglas se diseñen, más porcentaje de efectividad tendrá el sistema de riego basado en lógica difusa.

Caso 2: Cultivo de lechugas. En Emmanuel Luna (2022) desarrollaron un controlador basado en lógica difusa para la programación del riego en un sistema por goteo, en donde los valores lingüísticos de entrada nuevamente son la humedad relativa del suelo y la temperatura con la ayuda de sensores capacitivos y un sensor DHT22, el cual es uno de los más implementados en este tipo de proyectos, debido a su bajo costo y además, proporciona buenas mediciones de humedad y temperatura en un cultivo de lechugas controlando el encendido y apagado de una electroválvula. Se les asignaron valores lingüísticos a las variables de entrada y de salida, para la humedad existen 4 valores los cuales son Seco (SC), Húmedo (HU), Saturado (SA), Sobre Saturado (SS). Para la temperatura de la misma forma se cuenta con 4 valores lingüísticos de entrada, Muy frío (MF), Frío (FR), Templado (TP) y Caliente (CT). Finalmente, para la electroválvula se cuenta con dos valores, Apagado (AP) y Encendido (EN).

Con estos valores se diseñaron 43 reglas con un sistema de inferencia Mamdani con el que controlaban el momento en el que la electroválvula era activada para el riego del cultivo. De acuerdo a los resultados de las simulaciones que realizaron se observó un ahorro de agua de un 33% en un primer periodo y de un 25% en un segundo periodo, logrando disminuir el tiempo de riego y los litros de agua que se requieren para regar el cultivo.

Caso 3: Cultivo de manzanas. En Manuel Luna (2022) implementan un controlador de riego basado en lógica difusa para un cultivo de manzanas en el estado de Durango, en el cual nuevamente se observa que las variables de entrada para el control del sistema difuso son la temperatura y la humedad del aire y del suelo, controlando una válvula de agua para el riego. Los resultados de la creación de 81 reglas difusas del tipo SI-ENTONCES con ayuda de un sistema de inferencia difusa tipo Mamdani para este sistema fueron positivos en términos de que se consiguió un buen uso del recurso hídrico, además, uno de los objetivos de esa investigación fue incrementar los valores de la humedad relativa del suelo, por lo que se evidenció un incremento en este parámetro del 90.65% al 97.15% mostrando que el sistema de control difuso implementado para un sistema de riego es una alternativa viable si se desean alcanzar valores deseados.

Caso 4: Cultivo de rosas. En el trabajo de investigación de Hidalgo Aguaisa y Tumbaco Sango (2021) desarrollan un sistema de riego automatizado mediante lógica difusa para un invernadero de rosas, para la creación de las reglas difusas utilizaron como variables de entrada la humedad relativa del suelo y la temperatura, con dichos valores crearon un conjunto de 9 reglas que permiten controlar el tiempo de riego para una electroválvula programada en Arduino. A diferencia del riego manual, el sistema de riego automatizado utilizando lógica difusa obtuvo un valor de tiempo de riego menor, esto debido a que el prototipo cerraba la electroválvula en cuanto la humedad del suelo era alcanzada. Además, lograron optimizar el consumo de agua hasta en un 33.33% lo cual, es un valor favorable para el ahorro de este recurso.

Caso 5: Cultivo de fresas. En Edgar Maya Olalla (2023) plantean la automatización de un cultivo hidropónico de fresas en un invernadero mediante un sistema de control difuso. Este controlador se implementa a través de reglas difusas con base en las variables de entrada que en este caso fueron el pH del sustrato, la conductividad eléctrica, la temperatura y la humedad del sustrato que

implementaron. El sistema de riego mejoró la calidad del cultivo, lo que evidenció un mayor crecimiento de las plantas y un tamaño más grande de los frutos en comparación con un sistema de riego automatizado con un temporizador fijo.

1.4 Justificación

El cultivo y producción del chile está directamente ligado a México y América Central, debido a que a nivel mundial es un símbolo representativo. Este cultivo pertenece a la familia de las solanáceas y al género *Capsicum*. Además, México es un país líder en la producción de chile verde, de acuerdo a la Secretaría de Agricultura y Desarrollo Rural, México se encuentra entre el segundo y cuarto lugar a nivel mundial. En las últimas décadas, la producción de chile jalapeño ha incrementado hasta en un 25%, lo que lo convierte y posiciona en uno de los cultivos más importantes hablando económicamente a nivel mundial, además, este chile necesita climas cálidos para desarrollarse, por lo que las condiciones climáticas del estado de Morelos favorecen en gran medida al crecimiento de esta hortaliza. A nivel nacional, Morelos ocupa el lugar 25 en volumen agrícola y agropecuario. Además, un sistema de producción que ha presentado crecimiento en el país es la agricultura protegida, esto debido a que, en gran medida, dentro de estos sistemas es posible implementar tecnología e infraestructura las cuales nos brindan la posibilidad de controlar factores ambientales teniendo como objetivo un mayor rendimiento. De acuerdo al censo agropecuario del 2022, en el estado de Morelos, estos sistemas de agricultura protegida abarcaron una superficie de 794 hectáreas, además, en la Tabla 1 se pueden observar los principales cultivos producidos en agricultura protegida en el estado de Morelos.

Tabla 1 Principales cultivos con agricultura protegida en Morelos 2022

Cultivo	Superficie sembrada (hectáreas)			Producción (toneladas)		
	Total	En agricultura protegida	Porcentaje	Total	En agricultura protegida	Porcentaje
Jitomate (tomate rojo)	1,397	275	19.7	67,618	40,380	59.7
Pepino	699	228	32.6	32,671	25,421	77.8
Chile	382	16	4.2	4,204	964	23.4

Fuente: INEGI, Censo Agropecuario, 2022

1.5 Alcances y limitaciones

Un controlador difuso nos permite manejar N número de entradas, no obstante, esto tiene como consecuencia también un incremento de reglas, resultando en un mayor número de entradas para nuestro controlador difuso. Por ejemplo, si se tienen dos entradas y cada una de ellas cuenta con 12 funciones de membresía, entonces, se habla de que existen 144 reglas posibles. Por ende, la implementación de más entradas al sistema con su cantidad N de funciones de membresía resultará en un mayor número de reglas posibles, lo cual puede llegar a provocar que la memoria de un microcontrolador se sature y no pueda funcionar de manera correcta. Sin embargo, es posible acotar el número de reglas utilizadas tomando en cuenta los datos recopilados en un sistema de monitoreo de las variables que tendrán efecto en el método de inferencia difusa, logrando observar el comportamiento de las variables y descartando escenarios que no se cumplen o que son poco probables que sucedan en el ecosistema en el que se trabaje. Para este trabajo de investigación, se contemplan cuatro entradas para el controlador difuso: Temperatura del aire (°C), Humedad relativa del aire (%), Humedad del suelo (%) y Radiación solar (lux), no obstante, la opinión de un experto podría ser considerar algunas otras variables necesarias para el cultivo como: Velocidad del viento, pH del suelo, conductividad eléctrica, etc. No obstante, puede resultar algo contraproducente utilizar tantas entradas por lo anteriormente expuesto, se tendrían que considerar a detalle las funciones de membresía para no saturar la memoria del microcontrolador.

El trabajo está pensado para instalarse en dimensiones reducidas, como para un uso doméstico. Sin embargo, si se habla de trasladarlo a un campo más extenso, representa una inversión considerable debido a los componentes y que en un campo de cultivo podría no contar con alguna red inalámbrica de internet o con alguna instalación eléctrica.

1.6 Objetivos

1.6.1 Objetivo general.

- Diseñar un sistema automático de riego sustentable para controlar el suministro del recurso hídrico, empleando el enfoque de lógica difusa considerando condiciones ambientales.

1.6.2 Objetivos específicos

- Determinar las condiciones ambientales necesarias para el crecimiento del tipo de cultivo elegido
- Diseñar y construir un sistema de riego a pequeña escala que permita monitorear condiciones ambientales.
- Diseñar reglas difusas considerando las condiciones ambientales para gestionar el recurso hídrico del sistema de riego utilizando lógica difusa.
- Evaluar el desempeño del sistema de riego sustentable en comparación con esquemas clásicos de riego

1.7 Organización del documento

La estructura del documento se presenta de la siguiente manera:

- **Capítulo 1:** Se aborda de manera general la problemática y la importancia frente al mundo moderno, como las investigaciones que anteceden la investigación, los objetivos a cumplir y los alcances y limitaciones con las que cuenta.
- **Capítulo 2:** Se presentan los conceptos básicos del trabajo de investigación, con el objetivo de que el lector tenga una premisa antes de abordar completamente el tema.
- **Capítulo 3:** Se aborda la parte de ensamblado e instrumentación del prototipo, los sensores que se seleccionaron y el montaje final con la maceta, la bomba de agua y el sistema de riego.
- **Capítulo 4:** Se expone la creación y programación de las reglas difusas para el cultivo seleccionado, las reglas que se seleccionaron y como fueron implementadas en el sistema de riego para la decisión del suministro de agua.
- **Capítulo 5:** Se presentan los resultados experimentales del prototipo y su simulación, así como las gráficas de los valores obtenidos durante el periodo de experimentación.
- **Capítulo 6:** Se exponen las conclusiones finales del trabajo, así como propuestas futuras que podrían ser implementadas en otros trabajos.

2 Capítulo 2: Marco Teórico

2.1 *El chile como emblema de México*

El chile como alimento y condimento es el producto agrícola más emblemático de México. Asimismo, es un producto con una versatilidad considerable, ya que este fruto puede comerse fresco, cocido o como algún condimento extra en platillos típicos del mundo. Además, en la agroindustria es posible procesarlos y crear nuevas variantes como los chiles deshidratados, encurtidos y enlatados, y por supuesto, en una infinidad de salsas. Este cultivo pertenece al género *Capsicum*, dentro de este género existen cinco especies domésticas: *C. annuum*, *C. baccatum*, *C. chinense*, *C. frutescens* y *C. pubescens*. La especie más cultivada en el mundo se trata de la *C. annuum*, en donde entra el chile jalapeño. La planta de chile cuenta con un tallo característico, de tipo leñoso y que crece como si fuera un arbusto, además, las flores que brotan de esta planta por lo general son de color blanco y en algunas ocasiones son verdosas. El tamaño, color y forma del fruto son muy variables, sobre todo en esta especie llamada *Capsicum annuum*. No obstante, a pesar de que es una planta que se puede cultivar durante todo el año, durante el segundo ciclo del año disminuye su productividad ya que en este ciclo el cultivo presenta brotaciones menos vigorosas y suelen proporcionar frutos de menor tamaño y calidad. El chile junto al tomate son las hortalizas con mayor relevancia en México, tan solo el chile aporta alrededor del 20.2% en la producción a nivel nacional de hortalizas, además, México es el segundo país más productor de chile en el mundo (INTAGRI, 2020).

2.1.1 Chile jalapeño (*Capsicum annuum* L.)

Es un cultivo hortícola de gran relevancia económica y alimentaria en México, ampliamente utilizado tanto en consumo fresco como en productos procesados bajo la industria agroalimentaria. Su desarrollo y crecimiento dependen en gran medida de las condiciones ambientales, el manejo agronómico y, de manera particular por la disponibilidad y administración del riego. Una de las ventajas con la que cuenta este chile es que se desarrolla bien en diferentes tipos de suelo. Los más recomendables son los francos arenosos, con una buena aireación, un buen drenaje y una retención de humedad considerable. Para obtener buenos resultados en la germinación, es preferible contar con una cama de siembra pareja y suelta, además, el chile cuenta con una tolerancia media a la salinidad, sin embargo, se aconseja buscar terrenos sin problemas de sal y con una profundidad

mínima de al menos 40 centímetros de profundidad para favorecer el crecimiento. Además, es conveniente que el terreno esté libre de piedras que puedan obstaculizar el crecimiento de las plántulas. En la Tabla 2 se pueden observar algunos parámetros a considerar para el crecimiento de este chile y en la Figura 1 se puede apreciar los frutos que se obtienen de esta hortaliza.



Figura 1 Chile jalapeño verde (*Capsicum Annuum*)

Tabla 2 Datos aproximados sobre la siembra y cosecha del chile jalapeño

<i>Profundidad de siembra</i>	<i>Distancia entre plantas</i>	<i>Días de germinación</i>	<i>Altura aprox.</i>	<i>Días para cosechar</i>
-------------------------------	--------------------------------	----------------------------	----------------------	---------------------------

6-10 mm	60-70 cm	8-12 días	40 cm	70-120 días
---------	----------	-----------	-------	-------------

2.1.2 Condiciones de cosecha del chile jalapeño

El clima correcto en el que se desarrolla este cultivo se trata de ambientes que oscilan lo templado a cálido. Además, es importante considerar parámetros ambientales que intervienen y afectan directamente el crecimiento de esta planta, tales como la temperatura ambiente, la humedad del suelo o sustrato y la radiación solar, estos parámetros se tienen que considerar dentro de ciertos rangos para evitar futuras afectaciones al cultivo.

Temperatura

Para germinar la semilla se toma alrededor de 9 a 12 días y se necesita una temperatura de entre 20 y 30 °C para crecer y dar frutos, sin demasiados cambios bruscos debido a que si la temperatura baja de 18 °C el crecimiento de la planta se verá afectado y lo retrasará drásticamente. Durante la etapa de fructificación, se recomienda mantener temperaturas moderadas para asegurar un

desarrollo uniforme del fruto. Sin embargo, aunque el rango térmico de esta especie es de 12 a 35 °C, en Yucatán se ha llegado a observar que tolera temperaturas de hasta 40 °C (Ruiz et al., 2013).

Humedad relativa del aire

Un aspecto clave a tener en cuenta y que juega un papel importante en el desarrollo del cultivo es la humedad relativa que esta planta requiere. El rango óptimo de humedad relativa se encuentra entre el 50% y 70%, con esto se provocar una buena transpiración y, por otro lado, niveles excesivamente altos de humedad pueden favorecer la aparición de enfermedades en la parte aérea de la planta y hasta puede llegar a dificultar el proceso de fecundación (Ruiz et al., 2013).

Luz solar

El chile jalapeño es una planta que requiere alta disponibilidad de luz solar, esto para realizar adecuadamente la fotosíntesis, no obstante, cabe mencionar que los frutos son vulnerables a la exposición directa de los rayos solares, por lo que se requiere que la planta tenga una buena cobertura de hojas o de algún tipo de lona para que los rayos UV no penetren directamente al cultivo. A pesar de que no existen datos numéricos de cuanta radiación es recomendable para el cultivo, algunos autores hablan de que la planta necesita de 5 hasta 8 horas de luz directa del sol al día. Una iluminación insuficiente podría tener como consecuencia plantas débiles, con menor floración y que los frutos sean de menor tamaño del promedio, sin embargo, niveles excesivamente altos de radiación junto con temperaturas altas pueden aumentar el estrés hídrico de la planta (INTAGRI, 2020).

Requerimientos hídricos y humedad del suelo.

El manejo del riego es un factor crítico para la producción del cultivo. El chile jalapeño requiere un suministro de agua constante, evitando tanto el déficit hídrico como el exceso de humedad en el suelo. Es por ello que, la humedad del suelo debe mantenerse preferentemente en un rango intermedio, que permita una buena aireación junto con una correcta absorción de nutrientes. Durante las etapas de floración y fructificación, la ausencia de riego puede provocar un estrés hídrico lo que tiene como consecuencia la caída de flores y frutos y, por otro lado, el exceso de agua puede generar problemas de asfixia en las raíces y con ello, favorecer la aparición de

enfermedades y hongos que afecten el rendimiento del cultivo. Por ello, el control preciso del riego resulta un aspecto fundamental para optimizar el uso del agua y asegurar el desarrollo del cultivo. Otra ventaja con la que cuenta este cultivo es que se puede sembrar en campo abierto y en invernadero, por lo que es posible sembrarlas en espacios pequeños, hasta 6 plantas por m^2 (Instituto Nacional de Investigaciones Forestales Agrícolas y Pecuarias, 2015).

2.2 Sistemas de riego

Un sistema de riego consiste en una combinación de componentes físicos con el objetivo de suministrar agua a los cultivos de un área específica, administrando el recurso hídrico tanto en cantidad adecuada como en flujo correcto. Estos sistemas incluyen diversos elementos, junto con una variedad de materiales como tuberías, mangueras y válvulas hidráulicas, dichos recursos y parámetros varían según las necesidades específicas con base al tipo de cultivo seleccionado (Cardenas, 2019).

Métodos y tipos de riego

Existen diferentes técnicas y métodos para el riego, los cuales están directamente relacionados con la forma en la que se suministra el agua a los cultivos y plantas, pero de acuerdo a la literatura podemos clasificarlas en tres categorías: riego por gravedad (superficial), riego a presión (aspersión) y riego subsuperficial, los cuales se mencionan y describen en la Tabla 3. (Cardenas, 2019).

Tabla 3 Principales métodos de riego

Método	Variables
Cajetas	- Tiempo de riego - Longitud de cajeta - Tiempo de aplicación - Profundidad de la cajeta
Melgas	- Tiempo de aplicación - Eficiencia de aplicación - Caudal unitario Condiciones de diseño: - Un solo caudal

	- Longitud de caudal - Profundidad del caudal
Surcos	- Tiempo de avance del agua - Tiempo de oportunidad - Infiltración acumulada - Eficiencia de aplicación - Ajuste del perímetro
Aspersión	- Cálculo de la frecuencia de riego - Capacidad del sistema - Número de cambios del sistema - Cálculo del caudal dependiendo del área requerida - Efecto del viento
Goteo	- Distribución de presiones - Descarga requerida - Profundidad de aplicación - Aplicación mínima y máxima - Cálculo del diámetro mojado

Además, la elección de una técnica de riego depende de muchos más factores como el tipo de cultivo, sus necesidades específicas que este requiera, qué tanta disponibilidad de agua exista y por supuesto, la infraestructura para poder dosificarla, así como también las condiciones climáticas en donde se ubique el cultivo (Cardenas, 2019).

Las técnicas de riego se diferencian principalmente por dos factores importantes: el costo de instalación e implementación y la eficiencia hablando en términos del cómo gestionen el agua.

2.3 Lógica difusa

El término de lógica difusa se introdujo a mediados de los años sesenta por el ingeniero eléctrico Lotfi A. Zadeh, quien en 1965 publicó un artículo en donde implementó este término el cual llevaba por nombre “fuzzy sets”, el objetivo principal de esta lógica es modelar el razonamiento clásico del ser humano cuando se tiene información imprecisa o incierta. A diferencia de la lógica binaria tradicional, donde una situación solo puede ser verdadera o falsa (0 a 1), la lógica difusa permite grados intermedios de pertenencia, es decir, ni completamente verdadero ni completamente falso, representando la naturaleza gradual de muchas situaciones del mundo real. Este tipo de lógica fue creando una gran popularización en muchas ramas de la ciencia/tecnología, quizás por su sencillez al momento de aplicarse, ya que es posible crear sistemas de control únicamente con expresiones lingüísticas relacionándolas con datos numéricos obteniendo de manera simple una conclusión a partir de datos ambiguos o imprecisos a diferencia de la lógica clásica que trabaja con datos precisos. Posteriormente, en 1974 Ebrahim Mamdani implementó los conceptos de lógica difusa para elaborar un control difuso para la automatización de un motor a base de vapor, a partir de ahí nació la base de las reglas que hoy conocemos del tipo if-then. Además, en 1985 Takagi y Sugeno lograron desarrollar un método nuevo, como otra nueva opción además del método Mamdani (Luna M., 2022).

En situaciones reales, particularmente en aplicaciones agrícolas y ambientales, las variables no suelen comportarse de manera estrictamente certera, por ejemplo, la afirmación “el suelo está seco” no necesariamente implica una condición absoluta, el suelo puede estar ligeramente seco, moderadamente seco o muy seco. La lógica difusa permite modelar esta transición progresiva mediante el uso de funciones de pertenencia y variables lingüísticas.

2.3.1 Conjunto difuso

Un conjunto difuso se trata de una colección de elementos en el cual cada uno posee un grado de pertenencia comprendido en un intervalo $[0,1]$. Este grado de pertenencia se representa mediante funciones llamadas funciones de membresía, con estas funciones se pueden relacionar valores lingüísticos con datos números y clasificarlos.

Formalmente, si x es un universo de discurso, un conjunto difuso A en X se define como:

$$A = \{(x, \mu_A(x)) | x \in X\} \quad (1)$$

donde $\mu_A(x)$ es la función de membresía que asigna a cada elemento x un valor real entre 0 y 1. En el contexto agrícola, un ejemplo de un conjunto difuso podría ser “Humedad del suelo baja”, donde un valor de humedad del 35% podría pertenecer parcialmente al conjunto “seco” con un grado de pertenencia de 0.6 y de manera simultánea al conjunto “húmeda” con un grado de pertenencia de 0.3.

Operaciones entre conjuntos difusos

Dentro de la teoría del control difuso se opera con repertorios de conjuntos difusos, entonces, se necesitan sistemas de operaciones útiles para combinar estos conjuntos. Estas operaciones tienen la ventaja de que comparten similitudes con los métodos aplicados para los conjuntos clásicos y generalmente se utilizan para conectar o agregar condiciones entre los conjuntos difusos. Por lo tanto, en lógica difusa las operaciones básicas entre conjuntos son las siguientes:

- Intersección (operador AND)

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

- Unión (operador OR)

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

Generalmente, la intersección se modela a través del operador mínimo, mientras que la unión se representa mediante el operador máximo. Estas operaciones permiten evaluar reglas compuestas que involucran múltiples valores de entrada ayudando a determinar el grado de activación de cada regla que involucra “N” grados de pertenencia.

Variable lingüística

La variable lingüística es aquella que representa con palabras o términos de lenguaje natural números o elementos complejos de los cuales no se tenga una definición precisa, estas variables permiten representar el conocimiento humano de manera estructurada.

Por ejemplo, en un sistema de riego, la variable “Temperatura” puede tomar valores lingüísticos como:

- Baja
- Media
- Alta

Cada uno de estos términos se asocia a una función de pertenencia que define su rango de influencia dentro del universo establecido. El uso de variables lingüísticas facilita la construcción de reglas del tipo:

“Si la humedad del suelo es seca y la temperatura es alta, entonces el riego es mucho”

Este tipo de reglas ayuda a imitar el razonamiento humano y permite construir sistemas de decisión interpretables.

Funciones de membresía.

Describen matemáticamente el grado de un valor dentro de un conjunto difuso. Las funciones de membresía más utilizadas para la teoría de control son la triangular, trapezoidal, gaussiana y sigmoideal. Sin embargo, en aplicaciones embebidas como sistemas aplicados en microcontroladores, es más común utilizar las triangulares y trapezoidales debido a su simplicidad computacional para la representación gráfica y las operaciones de conjuntos difusos ya que pueden ser construidas fácilmente con poca información.

Supongamos que tenemos una función triangular $A(x)$ la cual se encuentra definida por 3 parámetros a , b y c donde: “ a ” representa el límite inferior, “ b ” corresponde al punto de máxima pertenencia y “ c ” representa el límite superior. La función inicia en el punto $(a, 0)$ alcanza su valor máximo de pertenencia en (b, α) y finaliza en $(c, 0)$. La representación matemática de esta función se muestra en la Tabla 4 y su representación gráfica en la Figura 2.

Tabla 4 Representación matemática de una función de membresía triangular

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

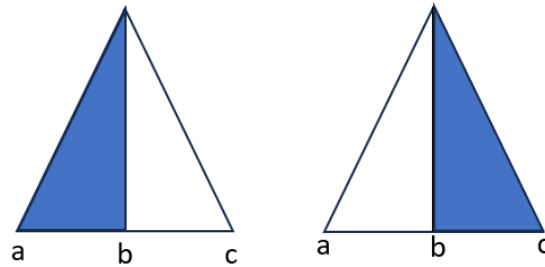


Figura 2 Representación gráfica de una función de membresía triangular: En la izquierda Si $a \leq x \leq b$, en la derecha Si $b \leq x \leq c$

2.3.2 Controlador difuso

Una de las grandes ventajas con las que cuenta este tipo de controladores es que no es necesaria la disposición de un modelo matemático del sistema, ya que estos controladores están basados en la creación y definición de reglas cualitativas. Con estos controladores es posible aplicar una amplia gama de problemas no lineales y dinámicas complicadas. Un controlador difuso decide la mejor opción de control, en este caso el riego, con respecto a los datos registrados por los sensores y de acuerdo a una base de conocimientos que están presentes en forma de reglas lingüísticas. Estas reglas lingüísticas están definidas a partir del conocimiento y se dividen en funciones de membresía, las cuales su principal objetivo es ser la referencia para cuantificar los valores lingüísticos, es decir, nos ayudan a definir rangos de pertenencia para nuestros valores y así, poderlos clasificar en un conjunto difuso. Este tipo de controladores resulta particularmente útil en sistemas donde existen incertidumbre, variabilidad constante y dificultades para obtener un modelo matemático exacto. En aplicaciones agrícolas, como el control de sistemas de riego, los controladores difusos permiten integrar múltiples variables ambientales y generar decisiones de riego más lógicas y flexibles.

Un controlador difuso generalmente está compuesto por cuatro etapas principales: fusificación, inferencia, base de reglas y defusificación como se puede observar en la Figura 3.

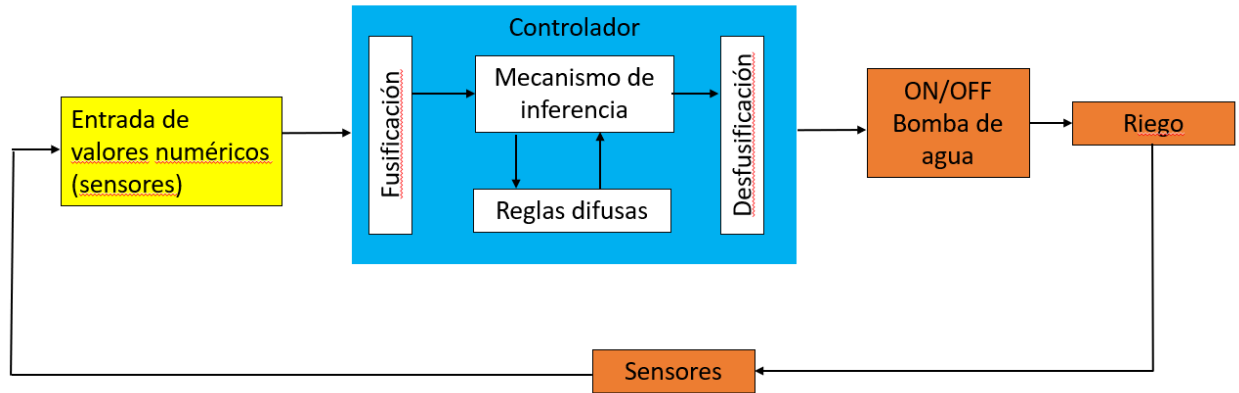


Figura 3 Diagrama de flujo del controlador difuso

Fusificación

Se trata de la primera parte de nuestro controlador difuso, se conoce como el proceso mediante el cual las diferentes lecturas de nuestros sensores serán convertidas a valores difusos con base a las funciones de membresía previamente establecidas para el controlador. Por lo tanto, se toman los datos de entrada, se evalúa con respecto a las funciones de membresía correspondientes y se plantea el grado en el que pertenecen de acuerdo a cada uno de los conjuntos difusos por medio de las funciones de pertenencia, obteniendo valores entre 0 y 1 que representan el grado de pertenencia a cada conjunto difuso.

Por ejemplo, un valor de temperatura de 29 °C pertenece al conjunto “temperatura media”, bastaría con trazar una línea perpendicular con la función o funciones de membresía activadas como se muestra en la Figura 4.

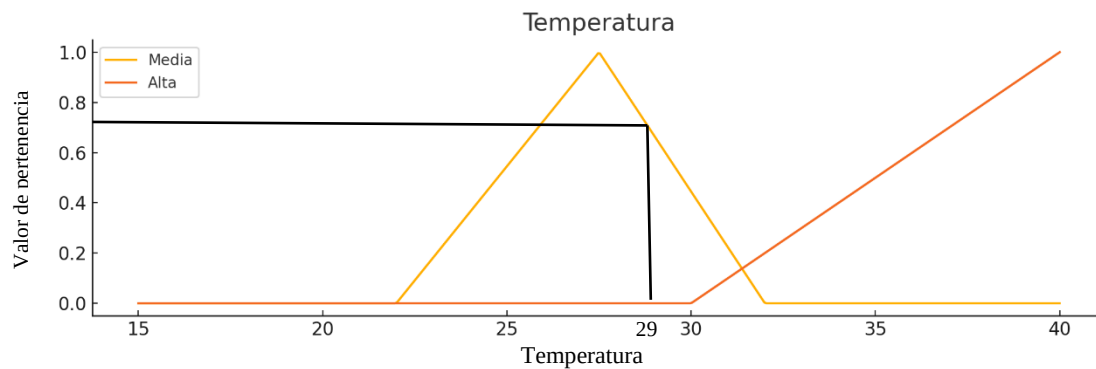


Figura 4 Representación en funciones de membresía para T=29°C

Base de reglas

En esta etapa se diseñan y construyen el conjunto de reglas lingüísticas que describen el comportamiento del sistema. Estas reglas generalmente adoptan la forma:

Si (condición) entonces (acción)

La base de reglas permite integrar múltiples variables de entrada y reflejar el conocimiento experto del dominio, permitiendo que el sistema tome decisiones similares a las que realizaría un humano, en este caso para decidir en que momento es necesario el riego y cuánta cantidad de agua suministrar.

Inferencia difusa

Se trata del proceso mediante el cual el sistema evalúa las reglas definidas en la base del conocimiento para determinar en qué grado se activa cada una de las reglas.

Durante esta etapa se combinan los grados de pertenencia obtenidos en la fusificación utilizando lógicos difusos, como el operador mínimo para la adición (AND) o el operador máximo para la disyunción (OR). El resultado de la inferencia es un conjunto difuso de salida que representa la decisión del sistema antes de ser convertida a un valor numérico (en este caso el tiempo de riego).

Desfusificación

Es el proceso inverso a la fusificación y es la etapa final del controlador difuso, este proceso permite asociar y conectar un conjunto difuso a un valor numérico y se realiza para calcular el valor de salida después de realizar el proceso de la inferencia difusa. Esta salida tiene que ser un valor numérico real y el cual debe de representar a todo el conjunto difuso, es decir, tiene que ser una conclusión de todo el controlador para tomar una decisión final y veraz.

Este proceso es necesario porque los sistemas de control reales necesitan valores concretos que puedan utilizarse para activar actuadores o ejecutar acciones específicas.

Existen diversos métodos de defusificación, entre los cuales se encuentran el método del máximo, el método de la media de máximos y el método del centroide.

Método del centro del área.

También conocido como método del centroide, es uno de los métodos de defusificación más utilizados debido a la facilidad para generar resultados estables y suaves. Este método consiste en calcular el centro de gravedad del área de la figura formada por la función de pertenencia resultante después de la suma de todas las reglas activadas.

Matemáticamente, el centro del área se obtiene mediante la relación entre el momento del área y el área total del conjunto difuso resultante a través de la siguiente fórmula:

$$Salida = \frac{\int x \cdot \mu(x) dx}{\int \mu(x) dx} \quad (2)$$

- x : Es el eje de la salida (por ejemplo, 0 a 15 segundos)
- $\mu(x)$: Es el grado de pertenencia recortado de la función activada.
- El numerador calcula el “momento” (peso por distancia)
- El denominador normaliza con el “peso total”

Este método permite obtener una salida representativa y equivalente a la contribución de todas las reglas activadas en el sistema, por lo que, en el presente trabajo se implementó un sistema de inferencia difusa tipo Mamdani, el cual utiliza funciones de membresía triangulares y el método del centroide para la defusificación de la salida.

2.4 Sistema embebido

Un sistema embebido es un sistema computacional diseñado para realizar funciones específicas dentro de un sistema menor. A diferencia de las computadoras de uso convencional y con propósito general, los sistemas embebidos están creados y diseñados para ejecutar tareas determinadas con recursos limitados de procesamiento, memoria, pero con menor consumo energético también.

Estos sistemas se utilizan ampliamente en aplicaciones de automatización, control, dispositivos electrónicos y sistemas agrícolas inteligentes. Su principal característica es la integración de

hardware y software para realizar tareas de monitoreo, procesamiento de información y control de actuadores.

En el contexto de los sistemas de riego automatizados, los sistemas embebidos permiten recopilar información proveniente de sensores los cuales monitorean las condiciones ambientales para después, procesar estos datos mediante algoritmos de control y tomar decisiones en tiempo real para activar o desactivar dispositivos de riego.

En el presente trabajo se utilizó un sistema embebido basado en un microcontrolador para integrar sensores ambientales, procesar datos mediante un algoritmo de control difuso y controlar el funcionamiento de una bomba de agua para controlar el riego.

2.5 Sensores

Los sensores son dispositivos capaces de detectar cambios en variables físicas del entorno y convertirlos en señales que pueden ser interpretadas por sistemas electrónicos. En aplicaciones agrícolas, los sensores permiten monitorear condiciones ambientales relevantes para el crecimiento de los cultivos, como temperatura, humedad, radiación solar y humedad del suelo.

El uso de sensores en sistemas de riego automatizados permite tomar decisiones basadas en información real del entorno, mejorando la eficiencia en el uso del agua y procurando las condiciones de crecimiento del cultivo.

Sensor de temperatura y humedad del aire.

El sensor de temperatura y humedad del aire permite monitorear de manera simultánea la temperatura ambiente y la humedad relativa del aire. Dichas variables influyen directamente en procesos fisiológicos de los cultivos sembrados, como la evapotranspiración.

El monitoreo de estas variables permite evaluar las condiciones ambientales las cuales pueden afectar de manera directa la demanda de recurso hídrico del cultivo y con ello, ajustar las decisiones del momento del riego y la cantidad que se debe suministrar.

Sensor de humedad del suelo

La inclusión de este sensor permite estimar el contenido de agua presente en el sustrato. Esta variable es la más importante para la gestión del riego, ya que proporciona información directa sobre la disponibilidad de agua para las raíces de la planta. El monitoreo continuo de la humedad del suelo permite evitar la ausencia del recurso hídrico, así como el exceso de este en el riego, contribuyendo a un suministro más inteligente y eficiente del agua.

Sensor de radiación solar.

La radiación solar es un factor fundamental en el proceso de fotosíntesis y en la tasa de evapotranspiración de las plantas, ya que, entre mayor radiación exista significa que hay demasiada exposición directa a los rayos UV, los cuales pueden provocar una tendencia a mayor evapotranspiración. El monitoreo de esta variable permite estimar indirectamente la demanda hídrica del cultivo, permitiendo ajustar las decisiones de riego con respecto a las condiciones ambientales reales.

2.6 *Plataforma Arduino*

Arduino es una plataforma de hardware y software de código abierto ampliamente utilizada para el desarrollo de sistemas embebidos y aplicaciones con ayuda de componentes electrónicos. Esta plataforma está basada en microcontroladores programables que permiten interactuar con sensores y actuadores mediante entradas y salidas digitales y analógicas.

Una de las principales ventajas de Arduino es su facilidad de uso, bajo costo y amplia disponibilidad de bibliotecas que facilitan la integración con diversos dispositivos electrónicos. Estas características han favorecido su uso en aplicaciones educativas, proyectos de investigación y sistemas de automatización.

En el presente trabajo se utilizó una placa Arduino como unidad de procesamiento principal del sistema de riego automatizado, encargada de recibir la información de los sensores, procesarla mediante el algoritmo de control difuso y activar el sistema de riego cuando las condiciones lo requieren.

2.7 MATLAB

MATLAB es un software de programación y cálculo numérico ampliamente utilizado en ingeniería y ciencias aplicadas para el análisis de datos, modelado matemático y simulación de sistemas. Dentro de sus herramientas se encuentra el Fuzzy Logic Toolbox, el cual permite diseñar, analizar y simular sistemas de control difuso.

Esta herramienta facilita la definición de variables lingüísticas, funciones de membresía, reglas difusas y métodos de inferencia, permitiendo evaluar el comportamiento del sistema antes de su implementación en hardware.

En este trabajo, MATLAB fue utilizado para el diseño y simulación del sistema de control difuso encargado de determinar el tiempo de riego en función de las variables ambientales monitoreadas. Posteriormente, el modelo desarrollado fue adaptado para su implementación en el sistema embebido.

3 Capítulo 3: Construcción e instrumentación del prototipo

En este capítulo se describe el proceso de construcción e instrumentación del prototipo experimental desarrollado para la implementación del sistema de riego automatizado basado en lógica difusa para el cultivo de chile jalapeño basado en las condiciones climáticas del estado de Morelos. Se detallan los componentes de hardware utilizados, un análisis comparativo costo/beneficio de los sensores utilizados, la integración de los sensores ambientales, la configuración del sistema embebido y el montaje del sistema de riego. Asimismo, se describe la forma en que se realizó la adquisición y almacenamiento de datos para su posterior análisis.

3.1 Diseño y construcción del prototipo.

El sistema está compuesto de dos macetas comerciales las cuales se encuentran elevadas y cuentan con las siguientes dimensiones: 82 cm de largo, 37.3 cm de ancho y 78 cm de altura aproximadamente (ver Figura 5) con el objetivo de comparar un esquema clásico de riego contra el esquema presentado basado en lógica difusa. Se presenta un diagrama base de la maceta que se trabajó en el software SolidWorks, posteriormente se montaron las macetas y finalmente a una de las macetas se le agregó un arco con varillas de plástico y una ligera cubierta de plástico con el objetivo de simular un invernadero y trabajar en un ambiente de agricultura protegida (Figura 6).

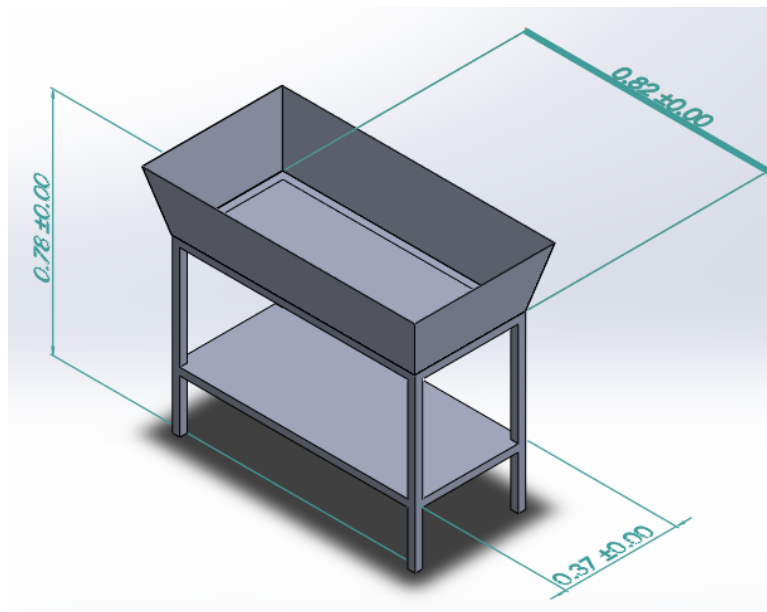


Figura 5 Diseño de la maceta comercial elaborada en SolidWorks



Figura 6 Montado y ensamblado de macetas comerciales con el arco y plástico protector.

Considerando las dimensiones de las macetas, podemos concluir que estas macetas nos permitirán plantar de 3 a 4 plántulas de chile jalapeño con suficiente espacio para que estas puedan crecer sin obstáculos ni complicaciones, además, las macetas se colocaron en un espacio por detrás del edificio de ciencias de la ingeniería ubicado en el CENIDET campus Apatzingán en donde por un lapso de alrededor de 4-5 horas recibían luz directa del sol, suficiente para que la planta de chile no sufra de olas de demasiado calor.

3.2 Selección de sensores

El sistema de riego desarrollado está compuesto por un conjunto de sensores encargados de medir variables ambientales que resultan relevantes para el crecimiento y producción del chile jalapeño, dichos sensores proporcionan información sobre las condiciones ambientales del entorno, la cual es procesada por el microcontrolador mediante un algoritmo de control difuso y a partir de esta información el sistema determina si es necesario activar el sistema de riego y durante cuánto tiempo.

Para la selección de todos los componentes del sistema de riego se realizó un análisis basándonos en el costo-beneficio de los sensores y actuadores que es posible encontrar en el mercado, contemplando un prototipo para un uso doméstico y que sea, hasta cierto punto, accesible. Considerando el entorno al que se aplican y el propósito de uso para este trabajo (aplicado a un sistema de riego), se buscó que los sensores permitieran realizar una medición ágil y con una

precisión adecuada, de manera que esta no sea invasiva o destructiva en el terreno, además, se planeó obtener respuestas de tipo eléctrico para su procesamiento y transmisión remota y, por supuesto, con un costo accesible.

A continuación, se presenta un análisis en la Tabla 5 de los sensores de humedad del suelo más empleados en trabajos reportados que involucran la lógica difusa en la agricultura, considerando su costo/beneficio y las ventajas con las que cuenta cada uno.

Tabla 5 Sensores de humedad del suelo comerciales más reportados en trabajos de agricultura.

Sensor	SoilWatch 10	FC-28	12C Soil Moisture	YL69 + YL38
Tipo	Capacitivo	Conductividad	Capacitivo	Resistivo
Tensión de alimentación	3.1 V - 5.0V	3.3 V - 5V	3.3 V – 5 V	3.3 V – 5 V
Consumo de corriente	-15 mA	-20 mA	14 mA a 5 V, 7.8 mA a 3.3 V	-20 mA
Temperatura de operación	-20 °C a 80 °C	-25 °C a 85° C	0° C – 85° C	0° C a 70° C
Encapsulado	Sí	No	Sí	No
Protocolo	Analógico	Analógico	DC	Analógico
Costo	\$237.78 Aprox	\$ 57.70 Aprox	\$ 60-115 Aprox	\$99 Aprox
Precisión	+ - 3	+ - 10 peor sin calibración	+ -5	+ - 10 o peor sin calibración

En la Tabla 5 se observa que los cuatro sensores cuentan con la tensión de alimentación más utilizada por los sistemas embebidos como Arduino (3.3 V y 5 V), por lo que cualquiera de las opciones de sensores es adecuada. Si hablamos del encapsulado, es primordial tomar en cuenta que la impermeabilidad de los sensores se relaciona con el tipo de sensor, por ende, los sensores tipo resistivos como el YL69+YL38 y los conductivos como el FC-28 es necesario que estén en contacto directo con el sustrato, por lo que los hace más vulnerables a la variedad de niveles de acidez del sustrato los cuales pueden deteriorar el material del sensor y perjudicar la medición de este. Con respecto al costo, se aprecia que los sensores I2C, FC-28 son los más económicos por lo que sacan ventaja de los demás en ese aspecto. Sin embargo, los sensores capacitivos son los más seleccionados según la literatura por su bajo costo y excelente rendimiento, además de que son fáciles de conectar y muy prácticos para sistemas de riego en específico, es por ello que se eligió para este trabajo.

Sensor capacitivo de humedad del suelo

Este sensor permite la medición de los niveles de humedad encontradas en el suelo o sustrato a través de un sensado tipo capacitivo. Contiene materiales lo suficientemente resistentes para evitar la corrosión (Figura 7) por lo que cuentan con un nivel de vida útil bastante aceptable. En la entrada cuenta con un regulador que puede soportar voltajes desde 3.3 V hasta 5.5 V, lo que lo hace ideal para implementarse con microcontroladores los cuales operan con el mismo nivel de voltaje.



Figura 7 Sensor capacitivo de humedad del suelo I2C

Sensor de temperatura ambiental

La temperatura se conoce como la intensidad de calor de un objeto, en este sentido, la temperatura ambiental es una variable crítica si se desea controlar los niveles de riego del cultivo, ya que niveles altos de temperatura pueden aumentar la demanda de recurso hídrico en el cultivo. Para la selección del sensor de temperatura se hizo el mismo análisis considerando algunos trabajos en la literatura encontrada, así como su costo/beneficio.

Tabla 6 Sensores de temperatura ambiente más reportados en trabajos de investigación				
Nombre	TMP36	DS18B20	MF523100	DHT11
Tipo	Estado solido	Estado solido	Termistor	Digital
Tensión de alimentación	2.7V-5.5V	3.3V-5V	3.3V-5V	3V-5V
Consumo de corriente	50 uA máximo	1,5 mA máximo	<1.58 mA	2,5 mA máximo
Temperatura de operación	-40 C a 125 °C	-10 C a 85 °C	-55 C a 125 °C	0-50 °C
Encapsulado	No	Si	No	Si
Protocolo	Analógico	One-wire	Analógico	Analógico
Costo	\$272.00 3 pines	\$184.00	\$300.00	\$105.00

Con base a la Tabla 6 podemos generar las siguientes conclusiones de nuestro análisis:

- La tensión de alimentación de los cuatro sensores que se compararon se encuentra dentro del rango para su funcionamiento e implementación en los sistemas embebidos.
- Los cuatro sensores cuentan con un consumo de corriente por debajo de los 30 mA.
- La temperatura en la que operan los cuatro sensores se encuentra dentro del rango de temperatura en el que se piensa trabajar (bajo condiciones climáticas del estado de Morelos).
- Respecto al costo, los cuatro se encuentran en un costo accesible, sin embargo, el DHT11 se destaca por su bajo costo y es uno de los más utilizados de acuerdo a la literatura.

DHT11

Una vez realizado el análisis de sensores de temperatura se decidió utilizar el DHT11 el cual se observa en la Figura 8, gracias a su costo accesible y que es un sensor fácil de implementar, además, con este sensor podemos medir la temperatura (de 0° C hasta 50° C) y al mismo tiempo la humedad relativa del aire (de 10% a 90%) lo cual también representa una ventaja contra los demás sensores. Internamente procesa señales analógicas, las convierte en digitales y envía un paquete de 40 bits de datos a través de un solo pin a sistemas embebidos como Arduino.

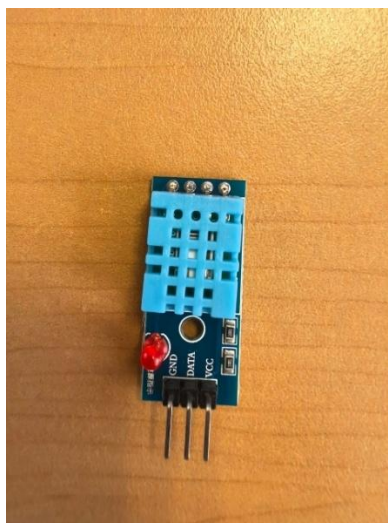


Figura 8 Sensor DHT11 para temperatura y humedad ambiente

Sensor de radiación solar

Para el desarrollo del sistema difuso se decidió integrar la variable de radiación solar, ya que en numerosas ocasiones un lapso demasiado largo a la exposición solar puede afectar el crecimiento del cultivo de chile jalapeño y sus frutos, creando una demanda de agua y, por ende, un mayor consumo de este recurso. A continuación, se presenta en la Tabla una comparativa entre los sensores de radiación/luminosidad solar más empleados en trabajos reportados de acuerdo a la búsqueda de la literatura.

Tabla 7 Sensores de radiación/iluminación solar más comerciales						
Sensor	Tipo de salida	Rango (lux)	Interfaz	Precio aproximado	Ventajas	Desventajas
LDR (fotoresistor)	Analógica	No definido	Analógica	\$10 – \$30 MXN	Muy económico	Baja precisión
TSL2561	Digital	0.1 – 40,000	I2C	\$120 – \$180 MXN	Buena precisión	Mayor costo
VEML7700	Digital	0 – 120,000	I2C	\$500-\$1,000 MXN	Muy alta precisión	Costo elevado
BH1750	Digital	1 – 65,535	I2C	\$35–\$60 MXN	Bajo costo, fácil integración, buena precisión	Saturación en luz extrema
ML8511 (UV)	Analógica	UV	Analógica	\$150–\$250 MXN	Mide radiación UV	No mide luz visible

Con base a la comparación presentada en la Tabla 7 podemos generar las siguientes conclusiones:

- Se seleccionó el sensor BH1750 (Figura 9) para la medición de radiación solar para este sistema debido a que presenta diversas ventajas al momento de implementarlo en sistemas embebidos basados en microcontroladores, además, presenta un bajo consumo energético y la entrega de valores de iluminación es en unidades lux y de manera digital, lo que facilita el procesamiento y análisis de datos.
- A diferencia de los sensores analógicos como los fotoresistores (LDR), el BH1750 ofrece una mayor precisión y estabilidad en las mediciones, evitando la necesidad de realizar procesos de calibración o de nuevas mediciones.
- El rango de medición del BH1750 es bastante aceptable, de hasta 65,500 lux lo cual resulta adecuado para aplicaciones agrícolas.
- El costo del sensor representa un criterio fundamental para el diseño del proyecto, por lo que el BH1750 significa una opción bastante considerable considerando que cuenta con un buen equilibrio costo/beneficio.

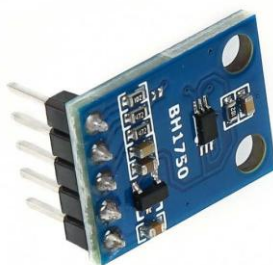


Figura 9 Sensor BH1750 para radiación solar

3.3 Otros materiales

Además de los sensores anteriormente expuestos, para llevar a cabo este proyecto se requirió de otros componentes y materiales secundarios los cuales brindan determinadas funciones y ayudan a que el prototipo funcione correctamente, considerando que el prototipo se encontraba en la intemperie. Dentro de este escenario podemos destacar las siguientes actividades y funciones que se realizaron con la lista completa de materiales que se puede observar en la Tabla 8.

- Energizar la bomba de agua y el microcontrolador empleado (Arduino Mega)
- Realizar pruebas del funcionamiento de todos los módulos empleados antes de llevarlos a la maceta y conectarlos.
- Instalar las conexiones de los circuitos en placas para desarrollar las interconexiones con los sensores.
- Verificar que el suministro del agua sea parejo y eficaz en el prototipo armado.

Tabla 8 Lista de materiales completa

Función	Sensor/Componente	Modelo elegido	Propósito/Uso
Medición de temperatura ambiente	Sensor de humedad y temperatura	DHT11	Monitoreo de temperatura y humedad relativa del aire
Humedad del suelo	Sensor de humedad del suelo	I2C sensor capacitivo	Detecta si el suelo está seco, húmedo o saturado
Radiación solar	Sensor de luz solar	BH1750	Mide la intensidad de luz solar en el ambiente (lux)
Reloj en tiempo real	RTC	DS3231	Registra la hora y fecha exactas de cada lectura
Almacenamiento de datos	Módulo microSD	Módulo lector de microSD (con SPI)	Guarda los datos de lectura en archivos .txt para análisis.
Control de riego	Bomba de agua	Bomba de agua sumergible de 110 V	Riego automáticamente cuando lo decide el sistema difuso
Activación de la bomba	Módulo relé	Módulo relé canal de 5 V	Controla el encendido/apagado de la bomba a través de Arduino
Procesamiento y control	Microcontrolador	Arduino Mega 2560	Ejecuta el sistema difuso, monitorea sensores y controla los actuadores
Fuente de alimentación	Adaptador 5 V Jack barreal	5V DC o 12V DC	Alimenta el Arduino y la bomba de forma externa

Cables/Conexión	Jumpers, protoboard		Conexiones generales entre sensores y Arduino
Dispersión de agua	Goteros y mangueras comerciales		Distribución del recurso hídrico hacia el pie de cada planta
Energizado entre la bomba y el relé	Clavija y conectores comerciales		Conexión segura entre la bomba de agua y el Arduino
Medición del flujo de agua	Flujómetro	YF-S401	Mide cuanta agua se dosifica a las plantas durante el tiempo que este activada la bomba.
Muestreo de la condición en la que se encuentra el sistema.	Pantalla LCD	Display LCD 1602	Muestra las condiciones ambientales en tiempo real y la activación de la bomba
Protección del microcontrolador	Caja de madera		Protege el Arduino de las condiciones ambientales extremas, como lluvia, vientos etc.

3.4 Montaje final del prototipo

Una vez adquirida toda la lista de materiales y la selección de sensores para el prototipo se montó el sistema de riego en las macetas comerciales, asimismo, se colocaron los sensores seleccionados de manera estratégica para aprovechar su máximo potencial en nuestro prototipo de invernadero (ver Figura 11); el sensor DHT11, el cual es el responsable de medir la temperatura y humedad relativa del ambiente se colocó en un poste del arco de plástico del huerto, el BH1750, el cual es el encargado de medir la radiación solar se colocó en una parte alta del invernadero y dentro de este, ya que el plástico utilizado como capa puede potenciar o disminuir la luz que entra al invernadero, finalmente, el sensor capacitivo para medir la humedad del suelo se colocó enterrado en el sustrato lo más cercano a una raíz de la planta sin invadir el espacio de crecimiento como se puede observar en la Figura 10.



Figura 10 Colocación del sensor capacitivo de humedad enterrado lo más cerca posible de la raíz de la planta



Figura 11 Fijación de sensores de temperatura ambiente (DHT11) y sensor de radiación solar (BH1750)

Además, se programó el Arduino para que en la pantalla LCD nos mostrara el valor de nuestras variables en tiempo real (ver Figura 12), se utilizaron abreviaciones para que todas las variables fueran visibles, dichas variables corresponden de la siguiente manera: T corresponde a Temperatura, Ha corresponde a Humedad ambiente, Hs corresponde a Humedad del suelo y L corresponde a Luz solar.



Figura 12 Visualización de condiciones ambientales en tiempo real a través de la pantalla LCD

Finalmente, se le adhirió el sistema de riego con los goteros comerciales y la bomba de agua a la maceta, la fuente de agua se trató de una cubeta de uso común la cual se observa en la Figura 13.



Figura 13 Bomba de agua sumergible para suministrar agua al cultivo

Los goteros se colocaron entre 1 y 2 cm del tallo de la planta (Figura 14), de manera que, cuando se activara la bomba las gotas cayeran lo más cercano posible al cultivo y, por ende, a las raíces de este, también considerando la ubicación del sensor capacitivo para que este pudiera detectar cuando hay presencia de agua.



Figura 14 Instalación de goteros comerciales a 1-2 centímetros del tallo de la planta

4 Capítulo 4: Diseño y planteamiento del controlador difuso

En este capítulo se presenta el diseño del controlador difuso implementado para el control del suministro de agua en un cultivo de chile jalapeño. Se describe el sistema de monitoreo de variables ambientales, la definición de entradas y salidas del controlador, la definición de las funciones de membresía, el proceso de fusificación, el mecanismo de inferencia y el método de defusificación.

Este enfoque permite modelar el comportamiento del sistema de riego ante las condiciones ambientales en tiempo real, de manera que, se comporte de manera similar al razonamiento humano, incorporando conocimiento empírico en forma de reglas lingüísticas.

4.1 Programación e implementación del esquema de monitoreo de las condiciones ambientales.

Para la implementación y codificación de los sensores se utilizó el software Arduino IDE el cual nos permite programar microcontroladores de manera accesible, para este trabajo se utilizó la tarjeta Mega 2560 para la lectura de las variables y la creación del sistema de monitoreo. En una primera etapa los sensores se programaron para que tengan lecturas constantes cada 5 minutos y con ayuda de un módulo de micro SD, los datos se registraron correctamente, de manera que, si las variables presentaban cambios significativos o muy bruscos, se pudieran detectar con anticipación, o bien, si la lectura no era tan necesaria cada 5 minutos, poder expandir el tiempo de lectura a 15 o 30 minutos.

A continuación, en la Figura 15 se puede observar el diagrama de flujo del funcionamiento del código programado en Arduino, dicho código se trata del monitoreo de variables con lectura cada determinado tiempo. Sin embargo, en el Anexo 8.6 se adjunta el código completo con explicación a detalle en cada bloque.



Figura 15 Diagrama de flujo para el sistema de monitoreo de condiciones ambientales

Los datos almacenados en la memoria SD se graficaron con el objetivo de observar cómo se comportaban las variables con respecto al tiempo y así, poder crear el sistema de monitoreo de variables ambientales para posteriormente crear nuestro controlador difuso basado en el comportamiento de estas variables, asimismo, se crearon las reglas difusas gracias a este monitoreo. En un inicio se graficaron los datos los cuales tienen un tiempo de muestreo cada 5 minutos y fueron graficados en una estación del año con lluvias y días nublados (Figura 16).

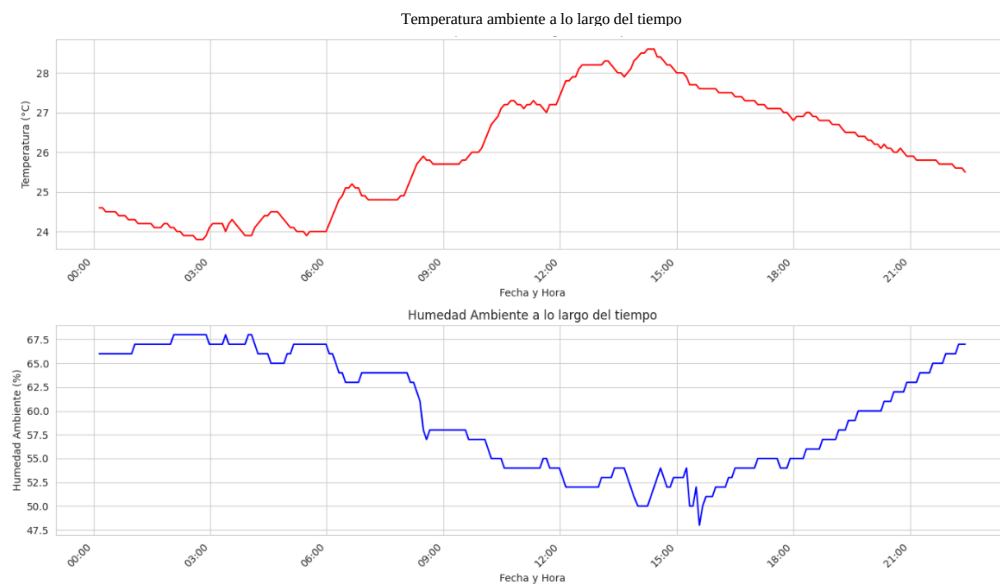


Figura 16 Comportamiento de condiciones ambientales a lo largo de un día con un tiempo de muestreo de 5 minutos

Una vez que se graficaron los datos, se determinó que no era estrictamente necesaria la lectura cada 5 minutos, ya que, por un lado, no se observaron cambios significativos en las mediciones y, por otro lado, eran demasiados datos registrados en la memoria. Así que, se decidió realizar un muestreo cada 15 minutos, dichos muestreos se pueden observar en la Figura 17.

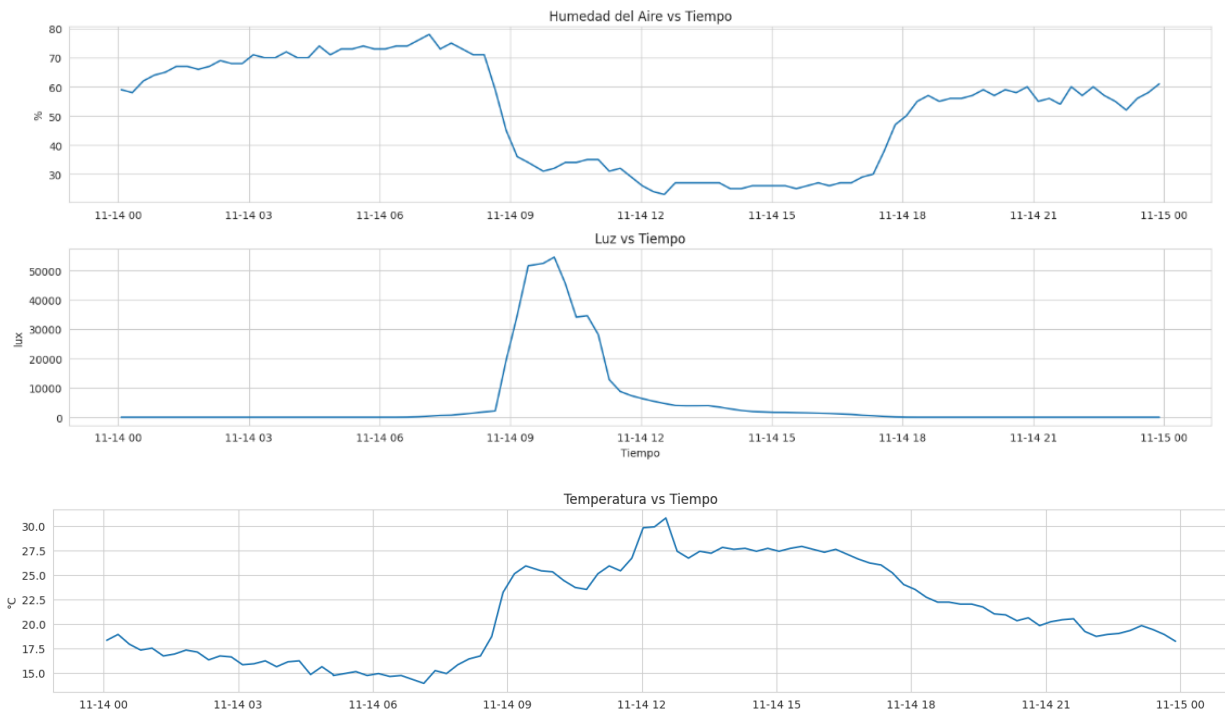


Figura 17 Comportamiento de condiciones ambientales con un tiempo de muestreo de 15 minutos

4.2 Definición de variables del sistema de control difuso

Basándonos en los hallazgos obtenidos de la literatura y de la recopilación de datos obtenidos por los sensores (ver Figura 16 y 17), se identificó que las condiciones climáticas registradas son adecuadas para el crecimiento del cultivo de chile jalapeño, por lo que el sistema de control difuso propuesto considera cuatro variables de entrada y una variable de salida, las cuales fueron seleccionadas con base a su influencia en la demanda hídrica del cultivo expuestas a continuación:

- Contamos con temperaturas de entre 20 °C a 32 °C, en ciertas ocasiones se llegaron a registrar temperaturas menores a los 20 °C, sin embargo, esto ocurría únicamente por las madrugadas cuando no era necesario un riego, lo cual es excelente para el crecimiento del cultivo a pesar de que en la literatura se ha encontrado que soporta temperaturas de hasta 40 °C, por

ende, se plantearon dos conjuntos difusos que van desde los 22 °C hasta los 40 °C y los cuales se asignaron como:

- Media (22 °C – 32 °C)
- Alta (30 °C – 40 °C)
- La humedad relativa del aire también es buena, ya que no es excesiva pero tampoco demasiado tenue y se definió en un rango de 0 a 100% con tres conjuntos difusos:
 - Baja (0% a 40%).
 - Media (30% a 70%)
 - Alta (60% a 100%)
- En cuanto a la luz del sol, se observa que existe un lapso de tiempo de alrededor de 4-5 horas las cuales son horas que recibe luz directa del sol y esto, en gran medida, beneficia al cultivo, se definió en un rango de 0 a 100,000 lux con tres conjuntos difusos:
 - Baja (0 a 10,000 lux)
 - Media (8,000 a 30,000 lux)
 - Alta (25,000 a 100,000 lux)
- La humedad del suelo es la variable más crítica para el control de riego dentro del sistema, por lo que es la única entrada que cuenta con 4 conjuntos difusos para tener mayor precisión al momento de decidir el riego y evitar riegos innecesarios, dicha variable va desde 0% hasta 100% como se muestra a continuación:
 - Muy seca (0% a 30%)
 - Seca (20% a 60%)
 - Húmeda (50% a 80%)
 - Muy húmeda (75% a 100%)

4.2.1 Definición de funciones de membresía del sistema de control difuso

Con base a lo anteriormente expuesto, se presentan las funciones de membresía empleadas para nuestro controlador difuso, las cuales están representadas con triángulos equiláteros debido a su simplicidad computacional y a que estas funciones se caracterizan por contar con una fácil comprensión y análisis al momento de implementar las variables en sistemas embebidos para obtener una respuesta numérica (ver Figura 18).

Cada variable fue dividida en conjuntos lingüísticos que permiten describir su comportamiento de manera cualitativa, por ejemplo, la humedad del suelo se representa mediante cuatro estados que describen el nivel de disponibilidad de agua.

Estas funciones permiten asignar a cada valor de entrada un grado de pertenencia, entre 0 y 1, lo que facilita una transición tenue entre estados y evita decisiones abruptas.

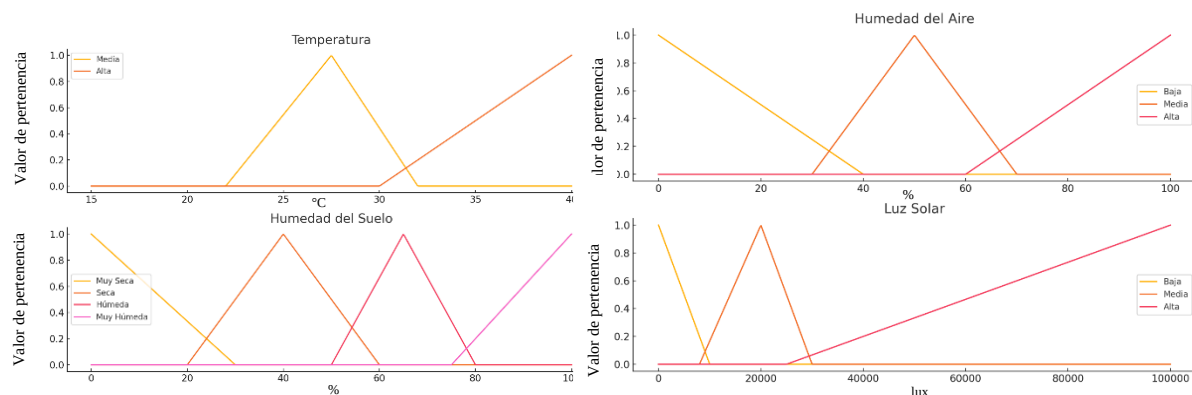


Figura 18 Funciones de membresía propuestas para el sistema difuso

4.3 Diseño y selección de reglas para el sistema de riego

De acuerdo al universo de entradas y salidas propuesto para el sistema se observa que es posible obtener hasta 72 reglas basadas en las condiciones ambientales y derivadas de las variables de entrada definidas para activar o desactivar la bomba que suministra el agua. Para el diseño de nuestras reglas se utilizó la lógica basada en condicionales del tipo SI-ENTONCES, no obstante, considerar las 72 reglas posibles representó un mayor costo computacional y redundancia en la toma de decisiones, por esta razón, se seleccionaron 45 más representativas y con mayor relevancia considerando los siguientes criterios:

- Se descartaron combinaciones en donde los escenarios eran poco comunes y que generan salidas similares.
- Se eliminaron escenarios de baja relevancia agronómica.
- Se priorizaron escenarios de condiciones críticas para el cultivo.
- Simplificación del sistema para su implementación en el sistema embebido.

En la Tabla 9 se presentan las 45 reglas implementadas en este trabajo de investigación, las cuales fueron llevadas al campo y al sistema de acuerdo a las condiciones ambientales. Sin embargo, en el Anexo 8.5 se presentan todas las combinaciones de reglas posibles para interés del lector.

Tabla 9 Reglas difusas para la automatización del sistema de riego

#	<i>Temperatura</i>	<i>Humedad Suelo</i>	<i>Humedad Aire</i>	<i>Luz</i>	<i>Riego</i>
1	Alta	MuySeca	Baja	Alta	Mucho
2	Alta	MuySeca	Media	Alta	Mucho
3	Alta	MuySeca	Alta	Alta	Medio
4	Alta	MuySeca	Baja	Media	Mucho
5	Alta	MuySeca	Media	Media	Mucho
6	Alta	MuySeca	Alta	Media	Medio
7	Alta	Seca	Baja	Alta	Mucho
8	Alta	Seca	Media	Alta	Medio
9	Alta	Seca	Alta	Alta	Medio
10	Alta	Seca	Baja	Media	Medio
11	Alta	Seca	Media	Media	Medio
12	Alta	Seca	Alta	Media	Poco
13	Alta	Húmeda	Baja	Alta	Medio
14	Alta	Húmeda	Media	Alta	Medio
15	Alta	Húmeda	Alta	Alta	Poco
16	Alta	Húmeda	Baja	Media	Medio
17	Alta	Húmeda	Media	Media	Poco
18	Alta	Húmeda	Alta	Media	Poco
19	Alta	MuyHúmeda	Baja	Alta	Poco
20	Alta	MuyHúmeda	Media	Alta	Poco
21	Alta	MuyHúmeda	Alta	Alta	Nada
22	Media	MuySeca	Baja	Alta	Mucho
23	Media	MuySeca	Media	Alta	Mucho
24	Media	MuySeca	Alta	Alta	Medio

25	Media	MuySeca	Baja	Media	Mucho
26	Media	MuySeca	Media	Media	Medio
27	Media	MuySeca	Alta	Media	Medio
28	Media	Seca	Baja	Alta	Medio
29	Media	Seca	Media	Alta	Medio
30	Media	Seca	Alta	Alta	Poco
31	Media	Seca	Baja	Media	Medio
32	Media	Seca	Media	Media	Poco
33	Media	Seca	Alta	Media	Poco
34	Media	Húmeda	Baja	Alta	Poco
35	Media	Húmeda	Media	Alta	Poco
36	Media	Húmeda	Alta	Alta	Nada
37	Media	Húmeda	Baja	Media	Poco
38	Media	Húmeda	Media	Media	Poco
39	Media	Húmeda	Alta	Media	Nada
40	Media	MuyHúmeda	Baja	Alta	Nada
41	Media	MuyHúmeda	Media	Alta	Nada
42	Media	MuyHúmeda	Alta	Alta	Nada
43	Media	MuyHúmeda	Baja	Media	Nada
44	Media	MuyHúmeda	Media	Media	Nada
45	Media	MuyHúmeda	Alta	Media	Nada

Se descartaron aquellas combinaciones en las que la radiación luminosa es baja, debido a que estas condiciones corresponden generalmente a periodos nocturnos y, por ende, de baja evapotranspiración, donde la demanda hídrica del cultivo es mínima o nula.

Asimismo, se eliminaron reglas en las que la humedad del suelo es alta o muy alta y las condiciones ambientales no favorecen la pérdida de agua, ya que en estos casos el riego no es necesario.

Esta estrategia permitió reducir la complejidad del sistema sin comprometer su desempeño, manteniendo todavía una buena capacidad de respuesta ante las condiciones más comunes y relevantes al momento de tomar decisiones de suministro de riego,

4.4 Mecanismo de inferencia difusa

El sistema utiliza un modelo de inferencia tipo Mamdani, el cual es ampliamente utilizado en aplicaciones de control debido a su interpretación intuitiva, las operaciones lógicas empleadas son:

- Operador AND: mínimo
- Operador OR: máximo.

Durante el proceso de inferencia se evalúan todas las reglas activas, se busca con qué intensidad se activan y se combinan sus resultados para generar una salida difusa agregada como se observa en la Figura 19.

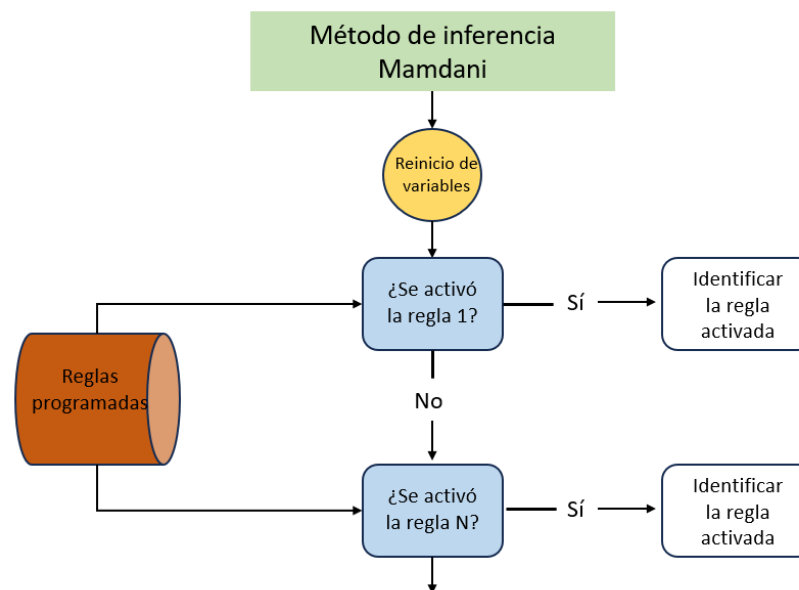


Figura 19 Diagrama de flujo del método de inferencia Mamdani

Consideremos como ejemplo la siguiente regla:

Si temperatura es Alta y Humedad del suelo es Seca y Luz solar es Alta ENTONCES Riego es Medio

Entonces, supongamos que se activan a este nivel:

- Temperatura Alta = 0.8
- Suelo Seco = 0.7

- Luz Alta = 0.6

Aplicando el operador AND (mínimo):

$$\alpha = \min(0.8, 0.7, 0.6) = 0.6$$

Esta regla se activa con un grado de 0.6

Se utiliza el operador “min” debido a que no se pueden tener “mayor certeza” acerca de la premisa de la regla. Por lo que el método de Mamdani permite que múltiples reglas contribuyan simultáneamente a la decisión final. En lugar de tomar decisiones precipitadas y abruptas, el sistema combina diferentes condiciones ambientales para generar una respuesta gradual.

4.5 Desfusificación (Toma de decisiones)

Para este trabajo de investigación se utilizó el método del centroide, también conocido como centro de gravedad, el cual es un método de desfusificación que permite convertir una salida difusa en un valor numérico representativo. Este método calcula el punto donde se equilibraría el área bajo la curva de la función de membresía resultante.

Consideremos el siguiente ejemplo:

Supongamos que el conjunto de salida **Medio** = trimf([7 10 13])

Activado con 0.375, el área resultante es una parte recortada de esa función (ver Figura 20).

El valor final se calcula con la Ecuación 2 dando como resultado lo siguiente:

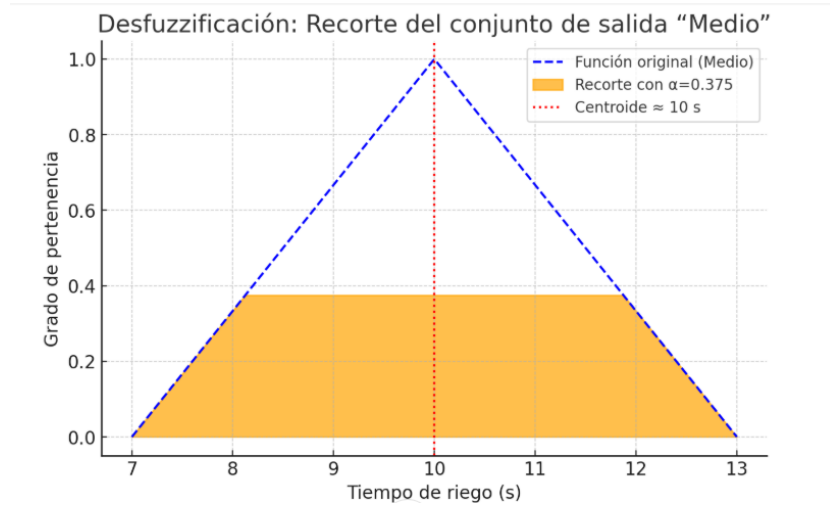


Figura 20 Recorte del conjunto de salida "Medio" a 0.375 después de la etapa de desfuzzificación

En la Figura 20 se aprecia que el nivel de activación es igual a 0.375 que es el área naranja dentro del conjunto Medio la cual permite obtener un valor representativo del conjunto difuso considerando toda la distribución de la salida. En este caso, aunque la activación de la regla es parcial, el sistema determina un tiempo de riego cercano a 10 segundos que es el pico máximo del conjunto que corresponde a un riego de intensidad media.

El método del centroide permite integrar la contribución de múltiples reglas activas en una sola salida continua, esto nos brinda una respuesta más realista y suave como se observa en la Figura 21 donde se realizaron las simulaciones en Matlab.

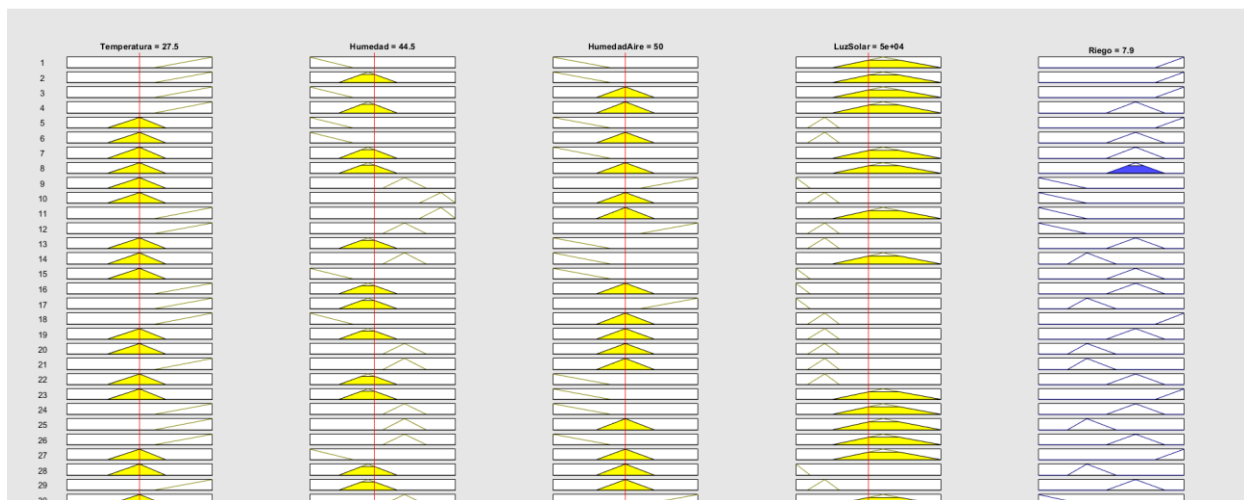


Figura 21 Simulaciones con las variables ambientales hechas con Matlab

5 Capítulo 5: Simulación y experimentación del prototipo.

5.1 Simulación en Matlab

Matlab cuenta con una herramienta llamada Fuzzy Logic Designer, dicha herramienta nos permite simular sistemas basados en lógica difusa definiendo nuestras entradas, funciones de membresía, reglas, métodos y salidas. Para nuestro trabajo de investigación se plantearon 4 entradas y una sola salida; cada una de las entradas cuenta con sus respectivas funciones de membresía, 2 para la temperatura, 3 para la humedad relativa del aire, 3 para la radiación solar y 4 para la humedad del suelo, por otro lado, nuestra única salida cuenta con 4 funciones de membresía, la cual se trata del tiempo de riego. Por ende, el número total de reglas posibles son 72, sin embargo, como anteriormente se mencionó en el Capítulo 4 se seleccionaron y diseñaron 45 reglas de manera estratégica, las cuales están previstas para cubrir los escenarios más comunes y controlar una buena humedad en la tierra, en las Figuras 22 y 23 se presentan las simulaciones en Matlab.

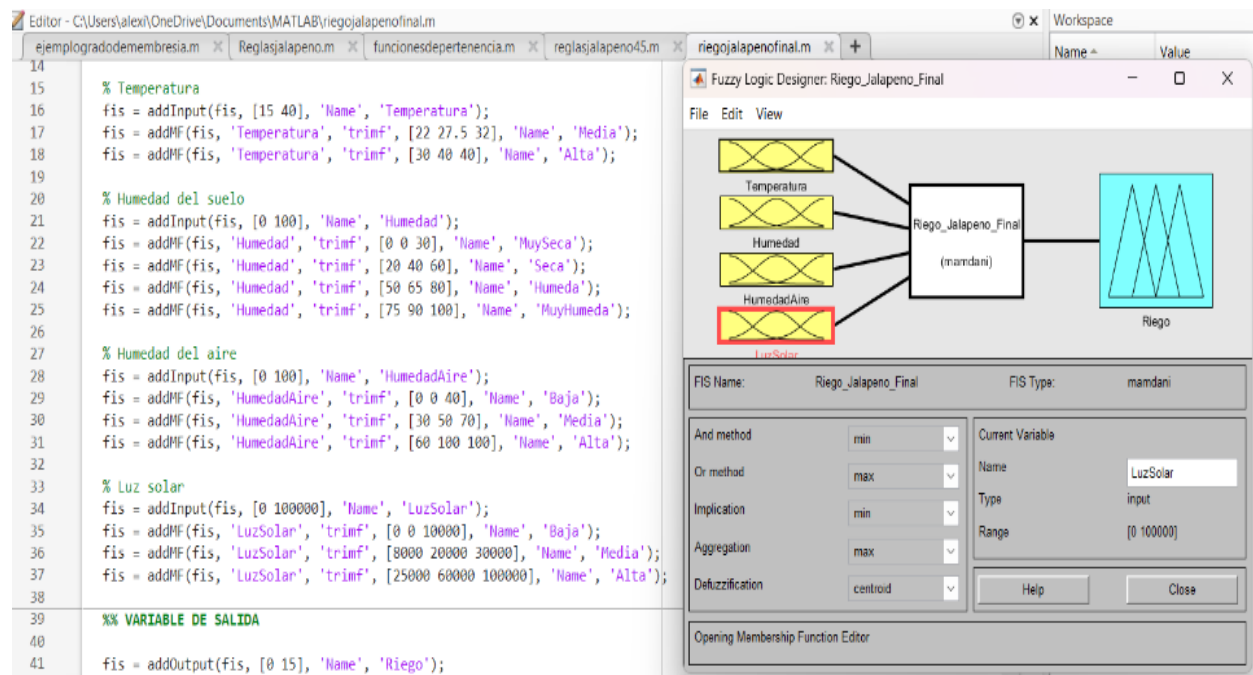


Figura 22 Creación del controlador difuso con funciones de membresía en Matlab

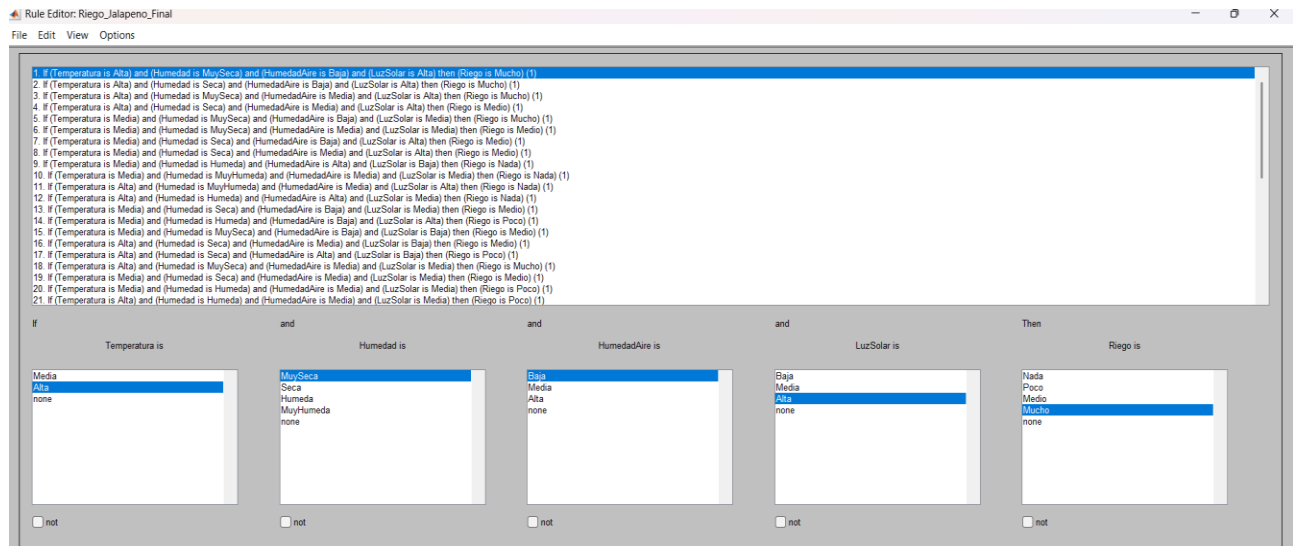


Figura 23 Creación de reglas difusas a través de Matlab

En la literatura se encontró una herramienta la cual nos facilita una transformación de código entre dos softwares, en este caso de Matlab a Arduino, dicha herramienta se trata de una página web llamada Arduino FIST la cual, nos ofrece un margen de error de 0.08 de acuerdo a Peralta-López et al. (2022).

5.2 Pruebas en el Protoboard

Antes de ser llevado al campo se realizaron pruebas del funcionamiento en el protoboard con el objetivo de rectificar su funcionamiento como el monitoreo y almacenamiento de datos ambientales, conexiones e instalación de sensores en el huerto de chile jalapeño, además, con el objetivo de mejorar la organización y compilación del sistema, el controlador difuso fue implementado en un archivo independiente denominado *fis_header.h*. Este archivo contiene la definición completa del sistema de inferencia difusa, incluyendo funciones de membresía, reglas y el método de defusificación por centroide, toda esta parte del programa funciona entonces como un complemento del archivo principal de Arduino, ya que esta parte se programó a través del Visual Studio Code como se observa en la Figura 24.

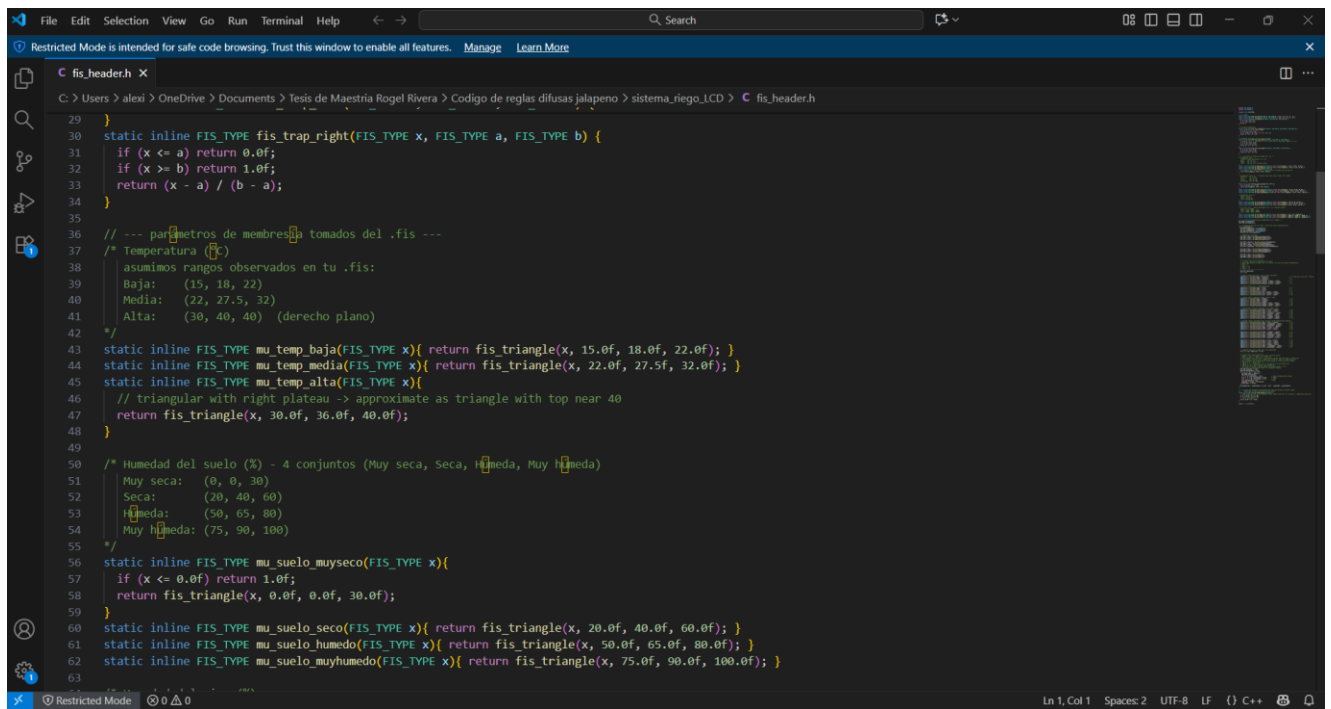


Figura 24 Programación del controlador difuso en Visual Studio

Posteriormente dicho archivo fue incluido en una carpeta junto con el archivo de Arduino donde esta programado el monitoreo con sensores, la lectura del archivo fis_header y el control de la bomba a través del controlador difuso. Es importante mencionar que el nombre de la carpeta tiene que ser el mismo del archivo .ino ya que de no ser así, el programa no se ejecutará como se aprecia en la Figura 25.

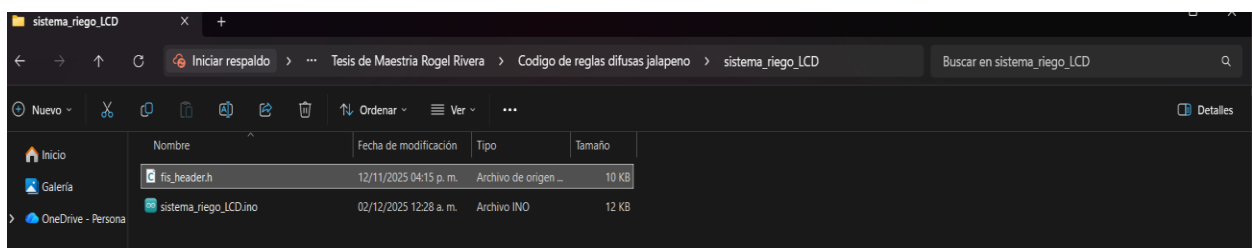


Figura 25 Agrupación del fis_header con el archivo de Arduino

Una vez que se adjuntaron los dos archivos en una misma carpeta, en el archivo principal de Arduino se utilizó la librería “fis_header.h” (ver Figura 26) para que en automático pueda detectar el archivo del controlador difuso programado en Visual Studio y con esto controlar la bomba para el suministro del recurso hídrico.

```

sistema_riego_LCD.ino  fis_header.h
1  /* -----
2  BLOQUE 1 – Inclusión de librerías
3  -----
4  Se incluyen todas las librerías necesarias para:
5  - Sensores (DHT11, BH1750)
6  - RTC DS3231
7  - MicroSD
8  - EEPROM (para guardar litros totales)
9  - LCD I2C
10 - Sistema difuso (fis_header)
11 -----*/
12 #include <Wire.h>
13 #include <LiquidCrystal_I2C.h>
14 #include <DHT.h>
15 #include <BH1750.h>
16 #include <RTCLib.h>
17 #include <SD.h>
18 #include <EEPROM.h>
19 #include "fis_header.h"
20
21
22 /* -----
23 BLOQUE 2 – Definición de pines y configuración
24 -----
25 Se configuran los pines donde están conectados:
26 - Sensor DHT11
27 - Sensor humedad de suelo
28 - Relé de bomba
29 - Sensor de flujo (interrupción)
30 - Micro SD
31 - Buzzer
32 -----*/

```

Figura 26 Implementación de librería en el programa principal para la lectura del fis_header

Esta estrategia permite una mejor estructuración del código, facilita la reutilización del sistema difuso para otros proyectos sin la necesidad de modificar todo el código y sin afectar el resto del sistema. Además, contribuye a una ejecución eficiente, limpia y depurada. Con esto se realizaron pruebas del funcionamiento a través de la pantalla LCD que se observa en la Figura.27.



Figura 27 Confirmación de almacenamiento de datos por medio de la pantalla LCD

5.3 Resultados experimentales y ajustes finales

5.3.1 Ajustes finales

Una vez montado todo el sistema de riego con sus componentes y contar con el código completo del controlador difuso (Figura 28), se observó que el sistema tenía algunas complicaciones al momento de compilar el código al Arduino, debido a que en un principio, se estuvo trabajando con un Arduino UNO, sin embargo, como se mencionó anteriormente, es necesario considerar la memoria y la capacidad del microcontrolador empleado, en este caso, se decidió emigrar a una tarjeta con mayores capacidades como lo es la tarjeta Arduino Mega 2560.



Figura 28 Montaje completo del prototipo

Además de esto, se decidió agregar un sensor de flujo al sistema (Figura 29) el cual nos brindó el apoyo para contabilizar los litros totales que se le aplicaban a nuestro riego cuando la bomba automatizada se activaba.



Figura 29 Sensor de flujo YF-S401

5.3.2 Primer experimento: Riego manual a todas las plántulas.

Objetivo: Analizar el comportamiento y crecimiento del cultivo con determinadas variaciones de riego para contemplar las necesidades hídricas de este.

Todos los experimentos de prueba realizados se iniciaron a partir de unas plántulas de chile jalapeño, las cuales se encontraban crecidas alrededor de 7 a 10 centímetros como se muestra en la Figura 30.



Figura 30 Plántulas de chile jalapeño

El riego que se les aplicó a las plántulas fue constante y de manera diaria, mayormente por las mañanas cuando los rangos de temperatura se encontraban entre 20 °C a 22° C y dentro de un horario de entre las 9 y 11 horas del día. En cuanto a la cantidad, el riego fue variado con la intención de experimentar con el cultivo y observar cómo se comportaba el sustrato cuando se le aplicaba un riego abundante o por el contrario un riego más lacónico. Esta primera prueba de experimentación se llevó a cabo durante 1 mes y medio aproximadamente y se llegó a un buen crecimiento y floración del cultivo como se observa en la Figura 31.



Figura 31 Floración del cultivo de chile jalapeño

En cada maceta se sembraron 3 plántulas de chile jalapeño y durante este primer experimento se les aplicó un riego diario de 300 ml por planta, es decir, 1.8 litros diarios, resultando en un riego semanal de 12.6 litros considerando los fines de semana. Con esto podemos concluir que el riego fue bueno solo si tomamos en cuenta el crecimiento, existía buena humedad y buenas condiciones, sin embargo, en algunas ocasiones la humedad en el sustrato se observó que era suficiente pudiendo crear algún sobre riego en el cultivo ya que no existía un control y monitoreo de esta variable.

5.3.3 Segundo experimento: Riego manual y adición del riego difuso.

La etapa comparativa inició en este segundo experimento, cuando se decidió regar el cultivo con 3 litros al día en una maceta, es decir, 1 litro para cada planta. Mientras que en la otra maceta se implementa el riego con el controlador difuso tomando decisiones inteligentes para suministrar el agua al cultivo. Una de las grandes ventajas que tienen las macetas industriales que se adquirieron es que cuentan con un filtro en la parte inferior, dicho filtro ayuda a que, si existe un exceso de agua, este no se quede acumulado y pueda drenarse en un contenedor para posteriormente depurarse. En la primera semana cuando se regó 3 litros al día se llegaron a obtener hasta 3.5 litros del recipiente que contiene la maceta (ver Figura 32), toda esta agua resulta desperdiciada ya que el cultivo no la aprovecha y puede llegar a existir un estrés hídrico, afectando el cultivo.

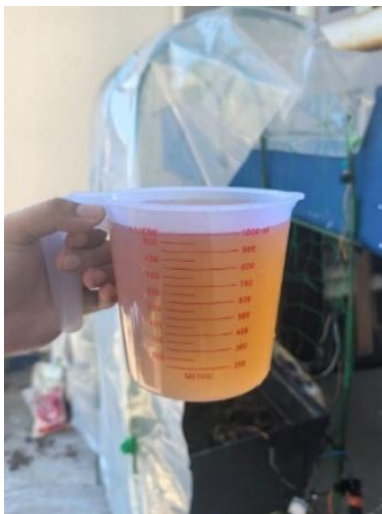


Figura 32 Agua no aprovechada por el cultivo debido a un exceso de riego

Durante este segundo experimento se observó que la tierra contaba con un exceso de humedad y que las plantas crecían más lento que las que tenían el riego controlado a pesar de que en este segundo experimento el riego se aplicó en un horario en donde existía mayor temperatura y más luz solar, alrededor de las 12 horas del día lo cual se puede observar representada con un punto rojo en la Figura 33, además, como se revisó en la búsqueda literaria, el exceso de humedad puede propiciar la aparición de alguna plaga o atraer insectos como hormigas, grillos y mantis que puedan intervenir en el crecimiento del cultivo deteriorando las hojas e incluso el tallo de la planta.

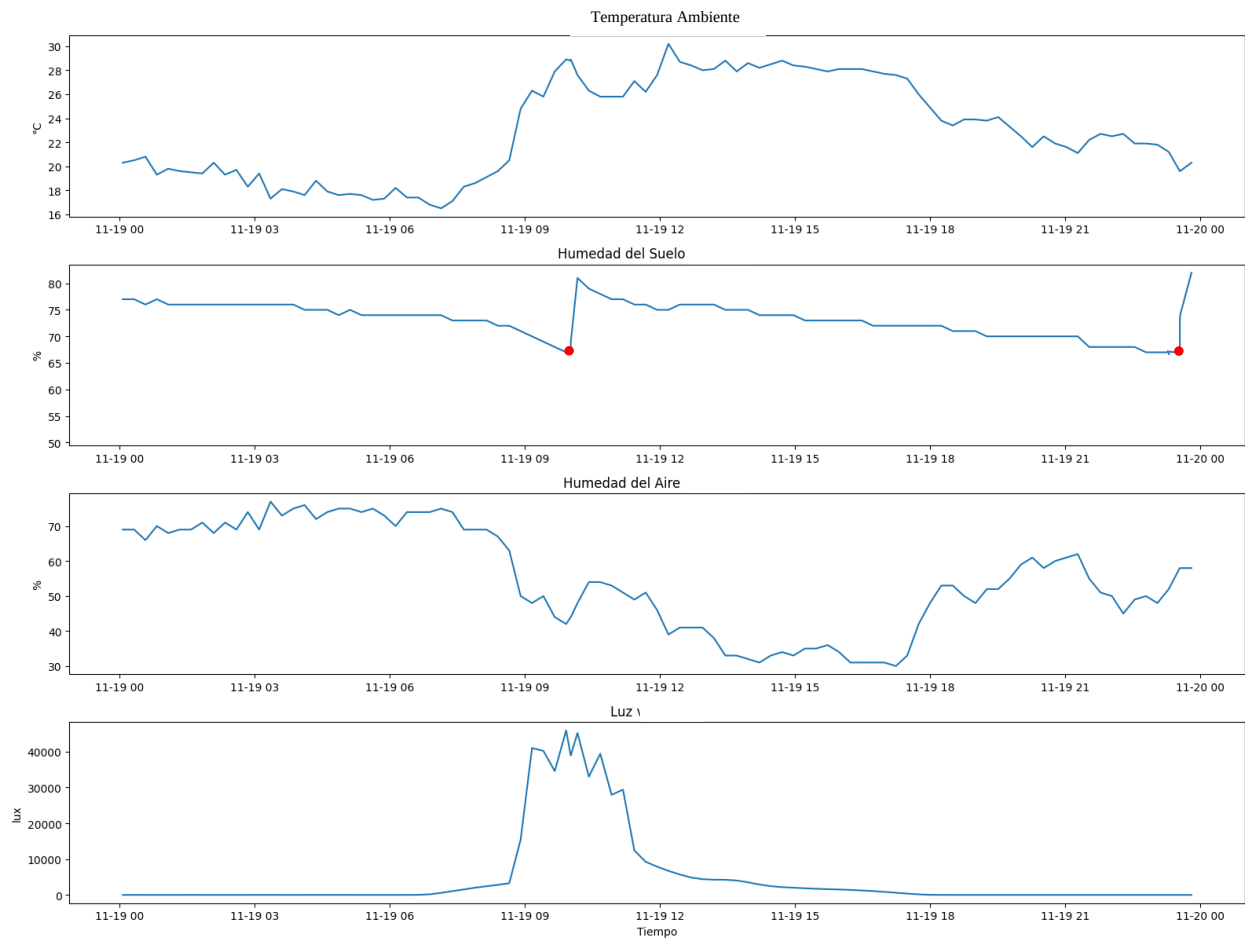


Figura 33 Gráficas de variables durante el segundo experimento

El riego difuso tomaba las condiciones ambientales y con base a los datos registrados se tomaba una decisión de cuánto tiempo regar, se tenían registros de que el riego suministraba en ocasiones cantidades de 100 ml a 400 ml y existían otros registros que marcaban días enteros sin riego los cuales se pueden observar en el histograma de la Figura 34.

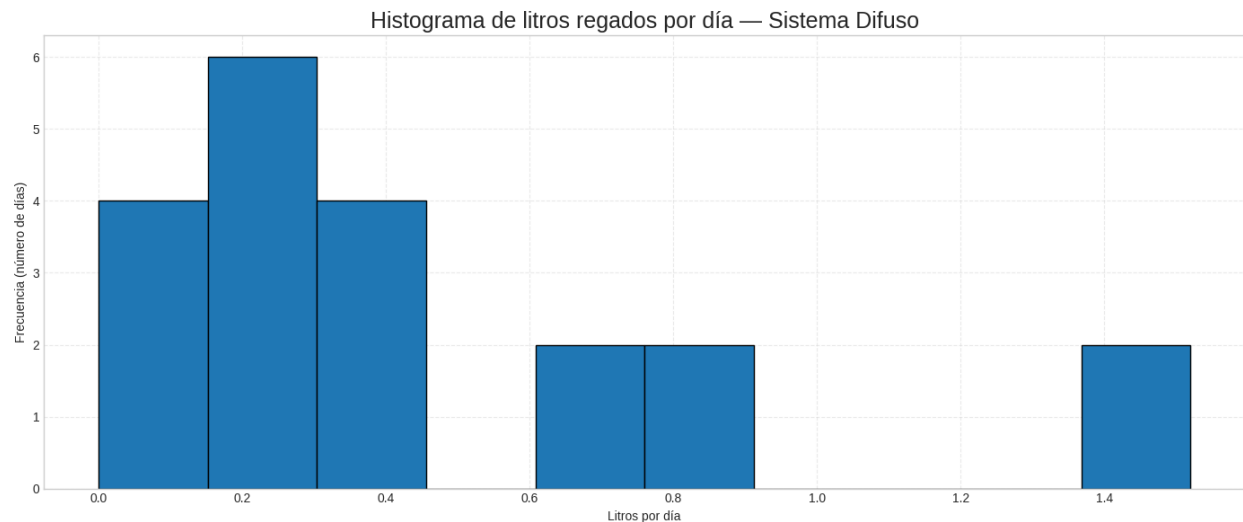


Figura 34 Histograma de cantidad de litros suministrados por el riego controlado mediante lógica difusa

5.3.4 Tercer experimento: Variación de riegos manuales

Posteriormente a lo observado en el segundo experimento, se decidió disminuir la cantidad de agua aplicada manualmente mientras que operaba el sistema difuso al mismo tiempo. Durante una segunda semana se aplicó un 50% menos del riego que se aplicó en el segundo experimento, es decir, 500 ml por cada planta.

Con esta nueva asignación se observó un mejor desempeño del cultivo al anterior experimento, no existía tanta humedad y el agua se veía mejor aprovechada por el cultivo sin existir algún exceso de humedad. Sin embargo, en el contenedor donde se drena el exceso de humedad se observó que aun existía agua acumulada y que no era aprovechada por el cultivo, nuevamente por un sobre riego.

En una nueva semana, se tomó la decisión de continuar con la cantidad de agua, con el cambio de que el riego iba a ser alternado, es decir, un día se le iba a aplicar y después otro día no y así sucesivamente intentando evitar un exceso de riego y considerando los experimentos anteriores donde se pudo observar que el sustrato mantenía buena retención de humedad a pesar de que existían días en donde no se les aplicaba ningún riego. No obstante, se tiene que recalcar que el riego del sistema difuso es un riego por goteo, este tipo de riego se caracteriza por suministrar de mejor manera el agua al cultivo, ya que el agua es aplicada directamente al pie de la planta como se observa en la Figura 35.



Figura 35 Visualización del riego por goteo en el sistema

Durante la etapa de experimentación se pudo observar el comportamiento tanto del sustrato como del cultivo con las variaciones que se tuvo al momento de aplicar el riego, sin embargo, uno de los comportamientos más considerables es que se observó que el cultivo creció ligeramente de manera más pausada con el riego manual (Figura 36) mientras que en el riego difuso se observó un mejor crecimiento, considerando que las plántulas se encontraban en un inicio crecidas alrededor de 7 a 10 cm, con el sistema difuso hubo un mayor crecimiento de hojas, alrededor de 10 hojas, y un crecimiento del tallo de 22 cm las primeras dos semanas, mientras que con el riego manual se observó incluso una decaída de hojas y poco crecimiento del tallo, alrededor de 15 cm.



Figura 36 Comparativa de crecimiento en ambos riegos, del lado izquierdo el riego manual y del lado derecho el riego controlado por lógica difusa

5.4 Evaluación del desempeño del sistema de riego difuso

El sistema registró información de manera periódica, generando una base de datos compuesta con las siguientes variables:

- Fecha y hora.
- Temperatura (°C)
- Humedad del suelo (%)
- Humedad relativa (%)
- Nivel de radiación solar (lux)
- Salida del sistema de riego (tiempo de riego)
- Litros de agua por ciclo
- Litros acumulados

Con base a la base de datos registrada, se realizaron y analizaron los resultados del comportamiento y desarrollo del controlador difuso. Las variables del sistema son presentadas a continuación. Se analizaron los comportamientos de las variables de entrada y se evaluó el sistema difuso contra esquemas clásicos de riego. Nuestras variables de interés se muestran en la Tabla.

Tabla 10 Valores de pertenencia de cada conjunto difuso en el sistema

	Temperatura (°C)	Humedad relativa (%)	Radiación solar (lux)	Humedad del suelo (%)
Rango/Valor de pertenencia	Media (22- 32°C)	Baja (0-40%)	Baja (0-10,000 lux)	Muy seca (0- 30%)
	Alta (30-40°C)	Media (30-70%)	Media (10,000- 30,000 lux)	Seca (20-60%)
		Alta (60-100%)	Alta (30,000- 100,000 lux)	Húmeda (50- 80%)
				Muy húmeda (75-100%)

El sistema de riego automático basado en lógica difusa permitió tomar decisiones de riego considerando de manera simultánea la temperatura, la humedad del suelo, la humedad relativa y la radiación solar. La bomba se activó únicamente cuando el conjunto de reglas determinaba un nivel de riego Medio o Alto, es decir, cuando el sistema requería un riego mayor a 7 segundos, lo cual permitió evitar riegos innecesarios cuando existían condiciones de alta humedad del suelo. De un universo de 72 reglas, para la construcción y desarrollo de base de datos se seleccionaron 45 como las más importantes para el controlador difuso y poner en marcha el riego automático. En primera instancia se plantaron 3 plántulas de chile jalapeño en cada maceta, se les estuvo aplicando un riego manual con una jarra medidora y se le aplicó una mezcla de humus de lombriz y fertilizante triple 17, los cuales nos ayudaron como apoyo al crecimiento de las plantas y llegar al fruto final. Posteriormente se le aplicó el riego difuso a una de las macetas mientras que a la otra se le continuó aplicando un riego tradicional. La etapa comparativa inicio la primera semana, cuando se decidió regar el cultivo con 3 litros al día, es decir, 1 litro aplicado a cada planta, posteriormente en la segunda semana se decidió disminuir el riego a la mitad del aplicado en la primera semana, es decir, 1.5 litros al día, finalmente, en la tercera semana se decidió aplicarle un riego intercalado, es decir, un día aplicar riego y otro día no, esto se puede observar en la Figura 37.

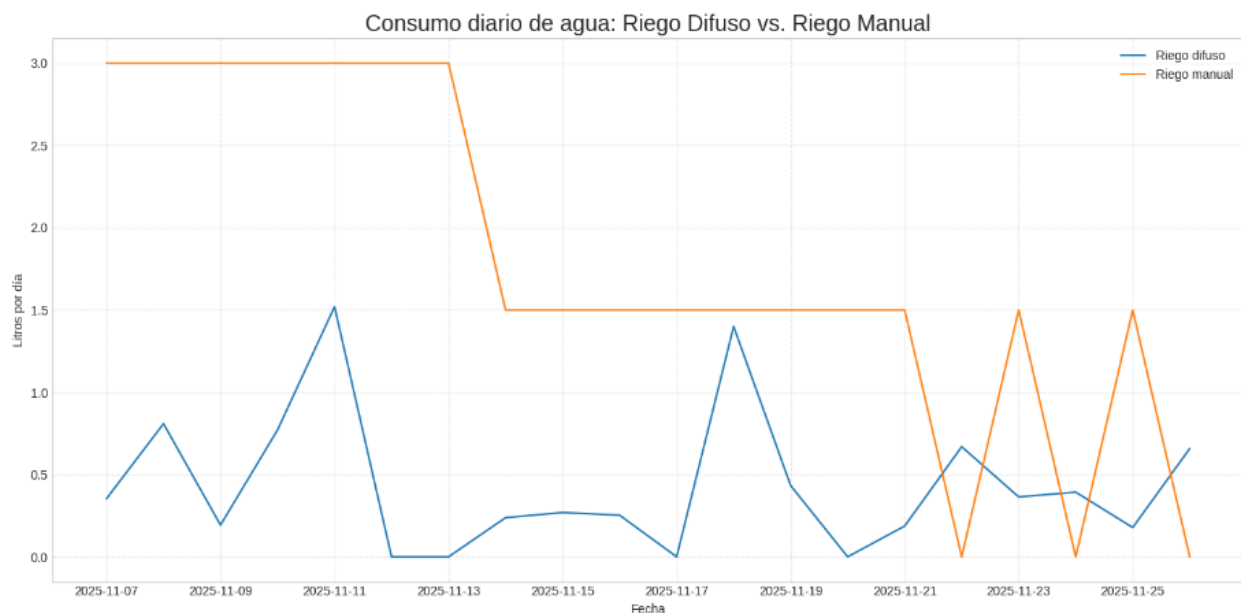


Figura 37 Consumo de agua de ambos riegos durante el periodo de experimentación

Se analizaron los litros de agua aplicados diariamente por el sistema difuso. Los resultados muestran que el sistema no mantiene un riego constante, sino que adapta dinámicamente la cantidad de agua suministrada en función de las condiciones ambientales, esto se puede observar en el histograma de la Figura 38, donde se observa que el sistema difuso aplicaba cantidades de agua necesarias, observándose un ahorro de agua considerable en contra de un riego típico de manera manual e intuitiva.

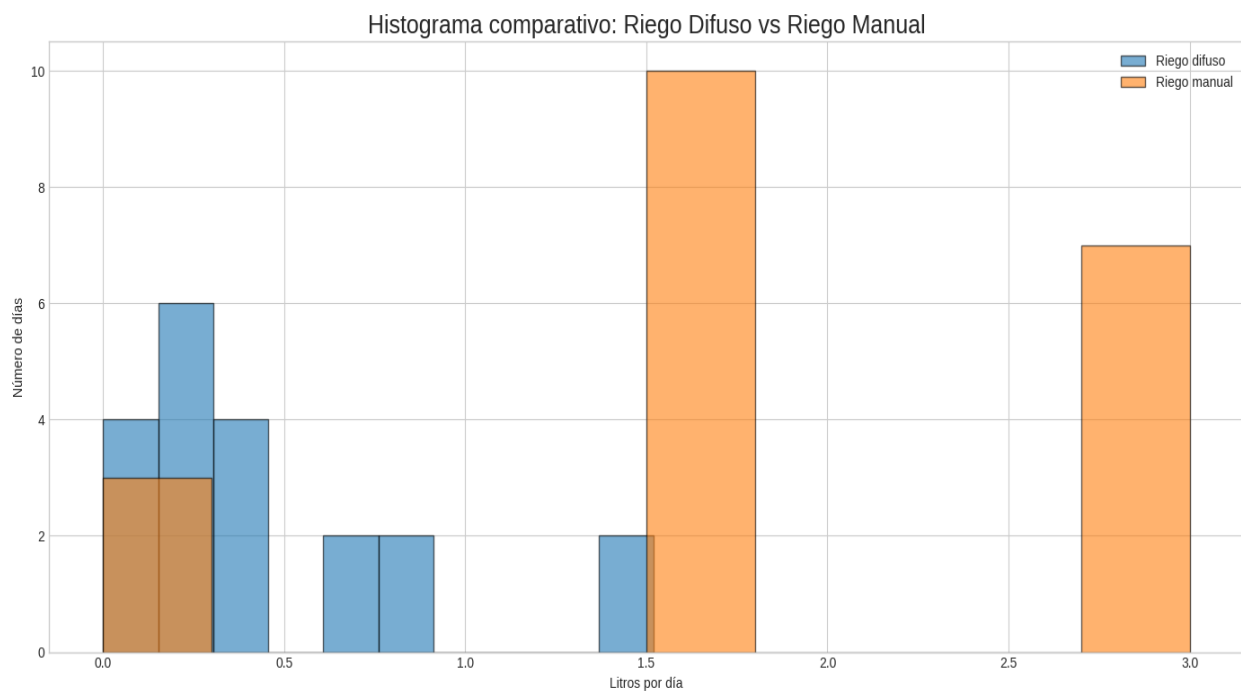


Figura 38 Histograma de los litros suministrados de ambos riegos durante el periodo de experimentación

Una de las grandes ventajas con las que cuenta nuestro sistema difuso es que solo riega cuando realmente lo necesita, esto se ve reflejado directamente en las variables de entrada del controlador y como intervienen en él. En la Figura 39 se puede observar el comportamiento de las variables a lo largo del tiempo en un día completo de experimentación, además, se puede observar las veces que la bomba se activaba con un punto rojo, observando la correlación que existe entre las variables para que las reglas difusas decidieran activar o no la bomba de agua, decidiendo activarse en días que contenían:

- Altas temperaturas y alta radiación, el sistema decidía no activarse ya que altas temperaturas con exposiciones altas de rayos UV pueden propiciar desperdicios de agua debido a la evapotranspiración.
- Alta humedad del suelo, el sistema redujo o anuló por completo el riego para evitar un estrés hídrico.
- El sistema aplicó riegos moderados.

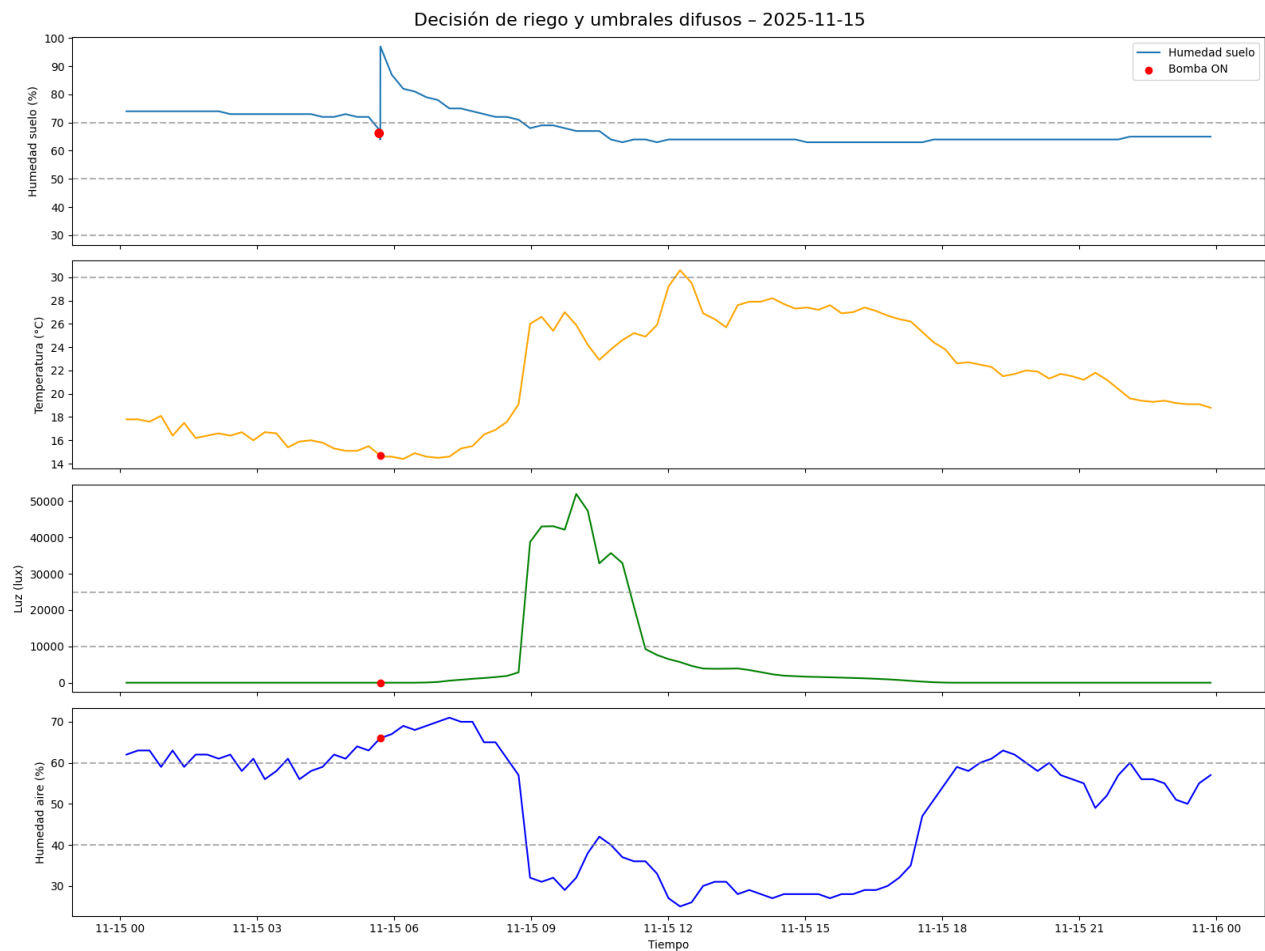


Figura 39 Comportamiento de variables en un día completo y la correlación con la activación de la bomba para suministrar agua

Se puede observar que el sistema no riega de forma arbitraria, sino que lo hace bajo ciertas condiciones ambientales específicas como por ejemplo que la humedad del suelo descienda de cierto nivel. Esto, por supuesto, representa directamente un menor consumo del recurso hídrico que, notablemente, se puede observar en la Figura 40.

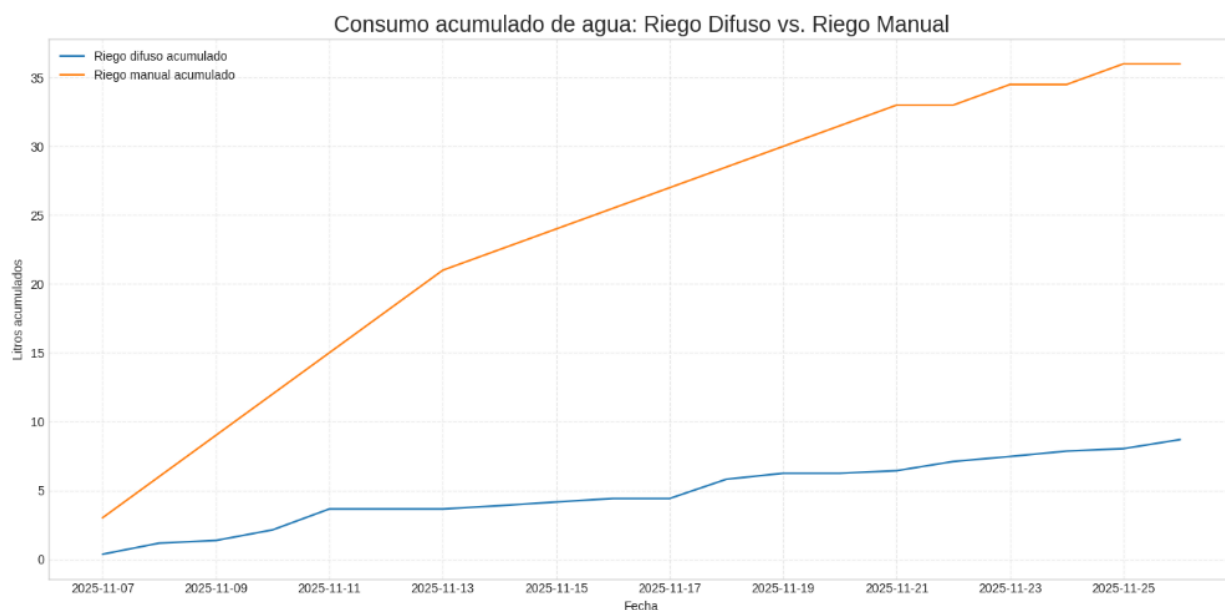


Figura 40 Gráfica del consumo de agua de ambos riegos (línea azul representa el riego difuso, línea naranja el riego manual)

5.4.1 Control de humedad del suelo

Dentro del universo de nuestras variables, consideramos la humedad del suelo como la más crítica para este sistema, debido a que tanto la ausencia como la presencia excesiva del recurso hídrico en el sustrato puede alterar o afectar directamente el crecimiento del cultivo, es por ello que, se consideró un mayor control para el monitoreo de esta variable. Empleando sistema difuso propuesto se pudo realizar este control logrando mantener la humedad del suelo dentro de los rangos aceptables para su crecimiento, en la Figura 41 se logra observar que la variable se mantuvo entre el 60% y 80% mayormente y que cuando lo requería, el sistema era capaz de detectar cuando los niveles de humedad descendían del 60% aplicando un riego moderado, demostrando una buena adaptabilidad respondiendo a múltiples variables y condiciones ambientales durante toda la etapa de experimentación.

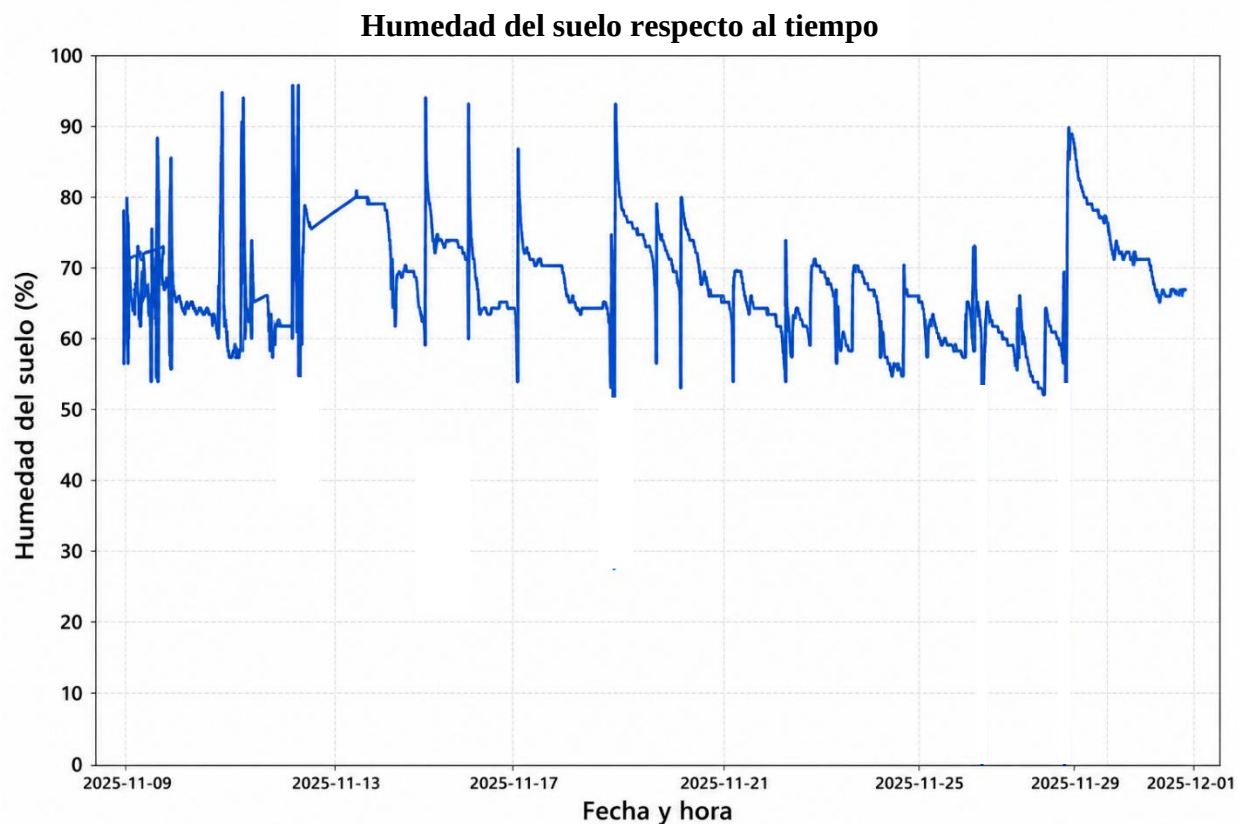


Figura 41 Comportamiento de la humedad del suelo durante el periodo de experimentación con el sistema de riego difuso

En la gráfica de la Figura 41, se visualiza cómo el sistema era capaz de detectar cuando los niveles de humedad descendían de cierto rango durante todo el periodo de experimentación, sin embargo, a continuación, se presenta en la Figura 42 el comportamiento en un solo día de esta etapa de experimentación.

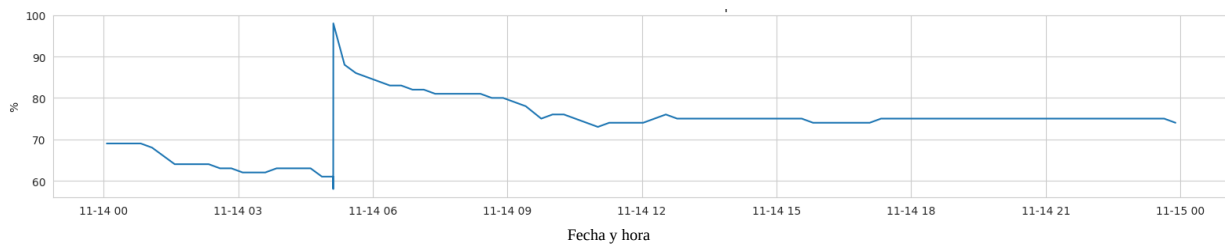


Figura 42 Comportamiento de la humedad del suelo en un día completo durante el periodo de experimentación

Con base a esto, se observa que, cuando los niveles de humedad descendían del 60% el sistema tomaba la decisión de regar y mientras se mantuvieran niveles altos de humedad, no lo hacía. Además, de acuerdo a los datos acumulados y registrados se puede observar un ahorro del recurso hídrico considerable, a pesar de que el riego manual estuvo en constante cambio debido a que en algunos escenarios podíamos provocar algún estrés hídrico el sistema difuso logró sobreponerse en cuanto al ahorro del agua, gracias a que no regaba de manera excesiva y solamente regaba cantidades necesarias, en momentos necesarios, con este análisis podemos saber y concluir que con el sistema difuso se pudieron ahorrar alrededor de 25 litros durante todo el periodo de experimentación como se puede observar en la Figura 43.

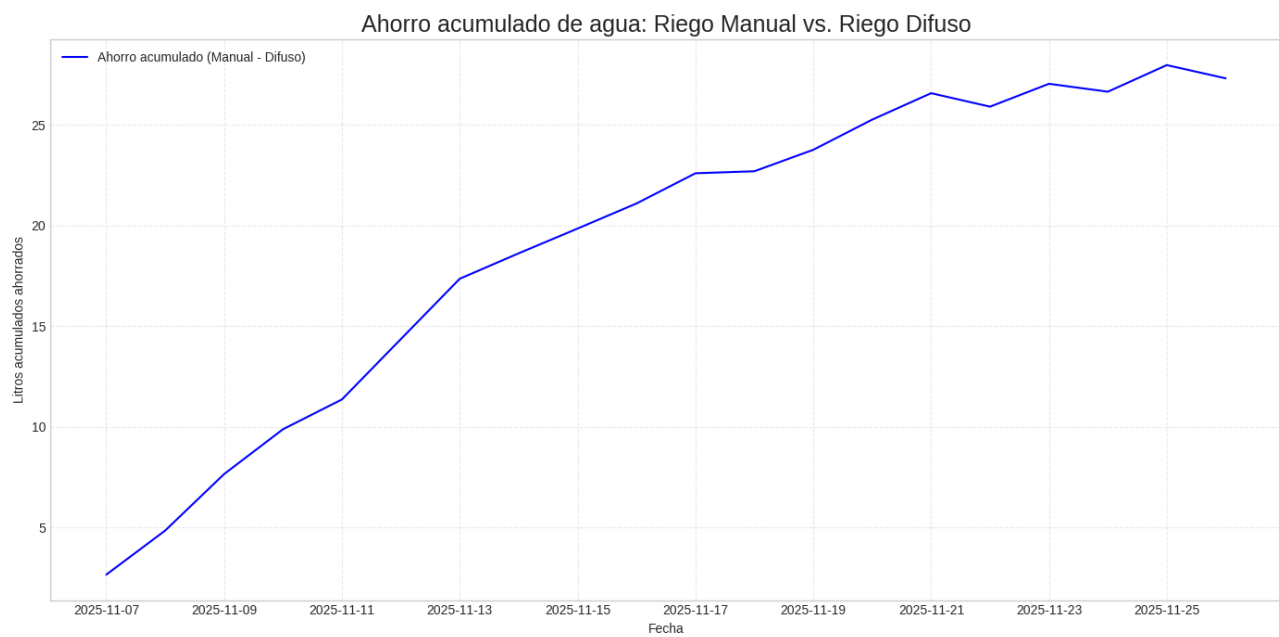


Figura 43 Ahorro acumulado de agua durante el periodo de experimentación entre ambos riegos

Al final de la etapa de experimentación se llegó al fruto del cultivo para ambos casos, tanto con el riego manual como con el riego difuso, con la diferencia de nivel de crecimiento, al regarlo manualmente se observó un estrés hídrico y como consecuencia se tuvo un crecimiento más lento, mientras que con el riego difuso fue constante y con un buen crecimiento hasta llegar a obtener los chiles jalapeños que se observan en la Figura 44 de buen tamaño y con un buen color.



Figura 44 Frutos finales posterior al periodo de experimentación

6 Capítulo 6: Conclusiones

6.1 Conclusiones Generales

El desarrollo del sistema de riego automatizado basado en lógica difusa permitió implementar un método de control capaz de responder dinámicamente a las condiciones ambientales presentes durante el cultivo de chile jalapeño. El controlador diseñado evaluó simultáneamente variables como temperatura, humedad relativa del aire, humedad del suelo y radiación solar, permitiendo ajustar el tiempo de activación de la bomba de agua de acuerdo con las necesidades hídricas del cultivo.

La implementación del sistema embebido Arduino permitió monitorear variables ambientales y controlar un sistema de riego basado en lógica difusa para la dosificación de agua aplicado al cultivo de chile jalapeño en el estado de Morelos, dicho sistema determina el tiempo de activación de una bomba de agua que suministra un riego por goteo a las plantas. Se cuenta con 4 entradas para nuestro control difuso las cuales son Temperatura (°C), Humedad relativa del aire (%), Humedad del suelo (%), Radiación solar (lux) y con estas entradas se diseñaron 72 reglas del tipo SI-ENTONCES, de las cuales se seleccionaron 45 reglas de manera estratégica cuyo objetivo es cumplir la mayor cantidad de escenarios posibles considerando las necesidades del cultivo.

El sistema propuesto logró mantener niveles de humedad en el suelo correctos para el crecimiento del chile jalapeño, durante gran parte del periodo experimental se observaron niveles de humedad entre 55% y 80% lo cual favoreció en gran medida a una buena administración del recurso hídrico. Además, el sistema activaba la bomba únicamente cuando era realmente necesario y en escenarios donde el agua fuera mejor aprovechada, como en las mañanas, cuando la radiación no es tan alta y se evitan periodos largos de evapotranspiración.

La instalación del sistema de riego nos brindó un mejor gestionamiento del recurso hídrico comparado a un riego común y tradicional de alguien que no cuenta con los conocimientos de un experto, esto se evidencia por el análisis comparativo de los dos esquemas de monitoreo y con los datos recabados se pudo concluir que para este caso de estudio se logró un ahorro de hasta 25 litros de agua aproximadamente.

6.2 Trabajos futuros

A pesar de los resultados satisfactorios obtenidos en la implementación del sistema de riego automático mediante lógica difusa, existen diversas áreas de oportunidad las cuales pueden ser abordadas en trabajos futuros con el objetivo de mejorar el desempeño, control y aplicabilidad del sistema en distintos entornos.

- **Incorporación de nuevas variables.**

El sistema actual considera temperatura, humedad del suelo, humedad del aire y radiación lumínica. Sin embargo, podrían integrarse variables adicionales como:

- Velocidad del viento
- Evapotranspiración
- Etapa fenológica del cultivo
- Tipo de suelo

La adición de estas variables podría permitir un control más preciso del riego.

- **Integración con tecnologías IoT**

La integración de esta tecnología permitiría un mejor monitoreo de variables en tiempo real, control del sistema desde algún dispositivo móvil y almacenamiento de datos en la nube con la implementación de routers.

- **Escalamiento a sistemas reales.**

El prototipo fue validado en condiciones controladas. Como trabajo futuro se propone su implementación en parcelas agrícolas reales y analizar su impacto en:

- Producción agrícola
- Consumo de agua a gran escala.

- **Aplicación a otros cultivos**

El prototipo fue diseñado específicamente para chile jalapeño, sin embargo, puede adaptarse a otro tipo de cultivo modificando las funciones de membresía y la programación de las reglas difusas con el objetivo de expandir el alcance del sistema en el sector agrícola.

7 Referencias bibliográficas

- Arie Altman, A. M. (2019). Understanding Agriculture within the Frameworks of Cumulative Cultural evolution, Gene-Culture, Co-evolution, and Cultural Niche Construction. *Human Ecology*, Vol. 13, No. 5, p. 483-497.
- Arunesh Kumar Singh, T. T. (2022). Intelligent Control of Irrigation Systems Using Fuzzy Logic Controller. *Energies*, Vol.15, No.19, p. 7199-7210
- Bright Keswant, A. G. (2018). Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms. *Neural Computing and Applications*, Vol. 31, p. 277-292
- Cardenas, L. J. (2019). *Diseño de un sistema de riego inteligente para cultivos de hortalizas basado en fuzzy logic en la granja la pradera en la universidad del norte*. Ecuador: Universidad Tecnica del Norte. Tesis de maestria
- David Vallejo-Gomez, M. O. (2023). Smart Irrigation Systems in Agriculture: A Systematic Review. *Agronomy*, Vol. 13, No. 2, p. 342-358
- Emmanuel Luna. (2022). *Diseño de un sistema digital de control basado en logica difusa para la automatizacion de un sistema de riego por goteo*. Bolivia: Universidad mayor de San Andres. Tesis de maestria.
- Eugenio Cedillo Portugal (2021) Manual de producción de chile jalapeño y poblano (*Capsicum annuum*) bajo invernadero, UNAM FES Aragón
- Faikul Umam, A. D. (2023). Implementation of fuzzy logic control method on chilli cultivation technology based smart drip irrigation. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, Vol.9, no 1, pp. 132-141
- Food and Agriculture Organization. (2023). *Goal 2: Zero Hunger*. New York, USA: United Nations Sustainable Development,
- Food and Agriculture Organization. (2022). *The state of food security and nutrition in the world 2022 - Repurposing food and agricultural policies to make healthy diets more affordable*. Rome, Italy: Food and Agriculture Organization of the United Nations.
- Frankelius, P. &. (2023). Chapter 11-Towards the fifth innovation era in agricultural innovation. *Agricultural Bioeconomy*, Vol.5 p. 205-229.
- Gackstetter, D. V. (2023). Autonomous field management–An enabler of sustainable future in agriculture. *Agricultural Systems*, Vol. 206, p. 103607.
- Goap, A. S. (2018). An IoT based smart irrigation management system using Machine learning and open source technologies. *Computers and electronics in Agriculture*, Vol. 155, p. 41-53.
- González, G., Gerardo, R., Ponse Hernández, J. L., Torres Pacheco, & González Chavira. (2018). Manual práctico para el cultivo del chile. Ediciones Mundi-Prensa.
- Guzmán-Tolosa (2019). Internet of Things for Irrigation System. *In Telematics and Computing: 8th International Congress, Mérida, México*, Vol. 8 p. 294-304.
- Hidalgo Aguaisa, B. G., & Tumbaco Sango, C. V. (2021). *Prototipo de riego automatizado para la producción de rosas usando la lógica difusa* [Tesis de licenciatura, Universidad Técnica de Cotopaxi].
- Instituto Nacional de Estadística y Geografía (2023) *Censo Agropecuario 2022: Resultados definitivos*. Morelos. INEGI
- Instituto Nacional de Investigaciones Forestales Agrícolas y Pecuarias (2015) *Paquete tecnológico para chile jalapeño*. Chihuahua, México. INIFAP.
- INTAGIRI (2020) Cultivo de chiles de México. Serie Hortalizas, Artículos técnicos de INTAGIRI, México, num 21, p. 6

- Jeni Molina-Puc. (2019). Fuzzy Logic Controller for Automation of an Autonomous Irrigation System Designed for Habanero Pepper. In *Telematics and Computing: 8th International Congress, WITCOM 2019, Merida, Mexico, Proceedings Vol. 8* (p. 284-293). Springer International Publishing
- Keswani, B., Mohapatra, A. G., Mohanty, A., Khanna, A., Rodrigues, J. J. P. C., & Gupta, D. (2018). Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms. *Neural Computing and Applications*, Vol. 31, p. 277–292..
- Konstantinos G. Liakos, P. B. (2018). Machine Learning in Agriculture: A review. *Sensors*. Vol. 18, No. 8, p. 2674-2684.
- M. A. López, D.G. Rivera, I. C. Satizábal y W. Y. Ortiz (2024) Comparative Analysis of Temperature, Humidity and Light Sensors for Use in Vermicompost Production Systems, *Entre Ciencia e Ingeniería*, Vol. 18, No. 35, p. 32-40
- Mizuno, R. G. (2020). Prediction Method of Plant Irrigation Timing Considering Data Imbalance. *Smartgreens*, Vol. 15 p. 177-184.
- MD Jiabul Hoque, M. S. (2023). A fuzzy logic-and Internet of Things-Based Smart Irrigation System. *Engineering Proceedings*, Vol. 58, No. 1, p. 93-103
- Mohammad Younesi, H. K. (2023). Design a solar system for fuzzy control of intelligent irrigation system. *CIGR Journal Open*, Vol. 25, No.1, p. 14-26
- Manuel Luna. (2022). *Controlador de riego basado en logica difusa para el cultivo de manzanas*. Durango: Instituto Tecnológico de Durango. Tesis de maestría
- Nayeli Montalvo-Romero Aaron Montiel-Gonzalez, R. P.-V. (2023). Agro-Technological Systems in Traditional Agriculture Assistance: A Systematic Review. *Topical Review*, Vol. 11, p. 123047-123069.
- Peralta-López, J. E., Lázaro-Mata, D., Padilla-Medina, J. A., Pérez-Pinal, F. J., & Barranco-Gutiérrez, A. I. (2022). An Arduino FIST evaluation for fuzzy system conversion from Matlab to Arduino. *Research in Computing Science*, Vol. 151 No. 12, pp. 67–73.
- Pezol, N. S. (2020). Design of an Internet of Things (Iot) Based Smart Irrigation and Fertilization System Using Fuzzy Logic for Chilli Plant. *2020 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Shah Alam, Malasya, Vol. 2 p. 69-73.
- Ruiz C. J.A., G. Medina G., I.J. González A., H.E. Flores L., G. Ramírez O., C. Ortiz T., K.E. Byerly M. y R.A. Martínez P. (2013). Requerimientos agroecológicos de cultivos. Segunda Edición. Libro Técnico Núm. 3. INIFAP. Instituto Nacional de Investigaciones Forestales Agrícolas y Pecuarias-CIRPAC-Campo Experimental Centro Altos de Jalisco. Tepatitlán de Morelos, Jalisco, México.
- Secretaría de Agricultura y Desarrollo Rural. (2024). *La importancia de la producción alimentaria en México*.
- Sjaak Wolfert, L. G.-J. (2017). Big Data in Smart Farming - A review. *Agricultural Systems*, Vol. 153 p.69-80.
- Stations, F. A. (2021). *The State of the World's Land and Water Resources for Food and Agriculture-Managing Systems at Risk*. Rome, Italy.
- Stræte, E. P. (2022). Critical support for different stages of innovation in agriculture: What, when, how? *Agricultural Systems*, Vol. 203, p. 103526.
- Subhashree Ghosh, S. S. (2016). Smart Irrigation: A smart drip irrigation system using cloud, Android and data Mining. *IEEE International Conference on Advances In Electronics, Communication and Computer Technology, Pune, India*, p. 236-239.

United Nations Educational, Scientific and Cultural Organization. (2019). *The United Nations World Water Development Report: Leaving No One Behind* . Paris, France: World Water Assessment Programme. UNESCO.

United Nations. (2022). *The Sustainable Development Goals Report*. New York, USA. Book,

Vanessa, H. A. (2021). *Prototipo de riego automatizado para la produccion de rosas usando la logica difusa*. Latacunga: Universidad Tecnica de Cotopaxi. Tesis de Licenciatura.

Y. Gamal, A. S. (2023). Smart Irrigation Systems: Overview. *in IEEE*, .Vol. 5, No. 13.

Yan-An Chen, W.-H. H.-S. (2021). An Ensemble Learning Model for Agricultural Prediction. *International Conference On Information Networking*, Vol. 2021, p. 311-316.

Zadeh, L.A. (1975). Fuzzy Logic and Approximate Reasoning. *Synthese*, Vol. 30 p. 407-428.

Zadeh, L.A (1965) Fuzzy sets. *Information and Control*, Vol. 8 No. 3, pp. 338-353

8 Anexos

8.1 Retribuciones sociales



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO

EL TECNOLÓGICO NACIONAL DE MÉXICO A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO

OTORGA LA PRESENTE

CONSTANCIA

A

Alexis Noé Rogel Rivera

Por su valiosa participación como Personal de Apoyo en la organización y desarrollo del
3er Congreso Internacional de Tecnología y Ciencia Aplicada (CITCA)
Llevado a cabo del 19 al 22 de noviembre de 2024 en las instalaciones del Tecnológico Nacional de
México campus Centro nacional de Investigación y Desarrollo Tecnológico (TecNM/CENIDET)

Cuernavaca, Morelos, 22 de noviembre de 2024



A079124
<http://constancias.cenidet.tecnm.mx>

LORENZO O. HERNÁNDEZ
DIRECTOR DEL CENTRO NACIONAL DE
INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO

ALICIA MARTÍNEZ REBOLLAR
COORDINADORA GENERAL
III CITCA

Sello Digital:

HaJwJON3s5lupFxrCUPQ/J6aCbhbNCrdLFRHzEcMmNI2mwP6TeLDEJa0VAXGRjM69uyxp1xAcpv3lC3EtS7SuT
XZ/gVoYysNCReVd25moHYk1OGiGDQCVqYkMf32AxKB0Y/cmNhhd+c23141QhCmqUfCVzVN7EVWYIwVR6n3coA
s5fKPqC7k4doZVXfnsvkUkmsdGCZJSJLCPc+yhlLkdXc5Z5CEhaIe5Gr4sHnLYZEjWyCr3BA7fxI7FO4PPLDBi
Iczk0IrgghzyppbSYvuJD/f1lT0H7GFJuAeE8LWE0x7p4Z62A5WANGCZ8ew+TzrCVSDsvxnN2qRBYB4PhbQ--

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico

CITCA



8.2 Producto académico



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO

EL TECNOLÓGICO NACIONAL DE MÉXICO A TRAVÉS DEL CENTRO NACIONAL DE INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO

OTORGA EL PRESENTE

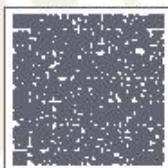
RECONOCIMIENTO

A

**Alexis Noé Rogel Rivera, Gloria Lilia Osorio Gordillo,
Vitervo López Caballero**

Por su destacada participación con la presentación del Artículo
en el área Ciencias de la Ingeniería, eje estratégico Sistemas Socioecológicos y Sustentabilidad titulado:
Implementación de Tecnologías en Sistemas de Riego Inteligente: Revisión de la Literatura.
En el marco del 3er Congreso Internacional de Tecnología y Ciencias Aplicadas (CITCA), celebrado en el
Tecnológico Nacional de México, campus Centro Nacional de Investigación y Desarrollo Tecnológico
(TecNM/CENIDET), del 19 al 22 de noviembre de 2024.

Cuernavaca, Morelos, 22 de noviembre de 2024



CITCA53724
<http://constancias.cenidet.tecnm.mx>

LORENZO O. HERNÁNDEZ
DIRECTOR DEL CENTRO NACIONAL DE
INVESTIGACIÓN Y DESARROLLO TECNOLÓGICO

Sello Digital:

IdXBzULvpo12TrjhAaAWmLSpjv1+SEwpy4QZ3j/mym/nUOhjSOmDWijQyFW4i57nYgiXfJ00bAQN04LE7YyDC
E3VXm2dG5c8CfaqveLxugQ6pq9+M64uza+TXbZ4/iAHwH6GGxBatgu4ca0R/Q6cJoJ5dErPt0jjPDTtjrupk0y
G4kN+NDWt0Sa58a0oNwrR2bmdMJOK7QF0GaIywjOmT2M4eYo8Cg2VFR6IGkaV9UtpgVwx15WYfWVVB9A4cxyT
Bj1Gx1q56maI+IeTp1wCSB+Ve7wFHS1r12HzMSd/1XM9/Xam0qO8nQmRCqBq9gR77rjLEKQ4GNRtMcby6vAA==

ALICIA MARTÍNEZ REBOLLAR
COORDINADORA GENERAL
III CITCA

cenidet
Centro Nacional de Investigación
y Desarrollo Tecnológico

CITCA
III CONGRESO INTERNACIONAL
DE TECNOLOGÍA Y CIENCIAS APLICADAS



8.3 Código para el monitoreo de variables

```
1 //Declaracion de librerias que se utilizaran
2 #include <Wire.h>
3 #include <RTClib.h>
4 #include <SPI.h>
5 #include <SD.h>
6 #include <DHT.h>
7 #include <BH1750.h>
8 #include <EEPROM.h>
9 #include <LiquidCrystal_I2C.h>
10
11 // --- Pines y módulos ---
12 // Agrupa los pines y el tipo de sensor que se van a usar.
13 #define DHTPIN 2
14 #define DHTTYPE DHT11
15 #define SD_CS 10
16 #define HUMEDAD_SUELO_PIN A0
17
18 RTC_DS3231 rtc;
19 DHT dht(DHTPIN, DHTTYPE);
20 BH1750 lightMeter;
21 LiquidCrystal_I2C lcd(0x27, 16, 2);
22
23 // --- Variables globales ---
24 // Variables de control, intervalos de medición y valores de calibración.
25 unsigned long tiempoPrevio = 0;
26 const long intervalo = 300000; // 5 minutos entre mediciones
27 bool ajustarHoraDesdePC = false; // Permite ajustar la hora del RTC desde la PC
28 int valorSeco, valorHumedo;
29
30 // --- CARGAR CALIBRACIÓN DESDE EEPROM ---
31 // Recupera los valores calibrados del sensor de humedad del suelo.
32 // Si no existen valores válidos, asigna valores por defecto.
33 void cargarCalibracion() {
34     EEPROM.get(0, valorSeco);
35     EEPROM.get(sizeof(int), valorHumedo);
36 }
37
38 if (valorSeco == 0xFFFF || valorHumedo == 0xFFFF || valorSeco >= valorHumedo) {
39     Serial.println("⚠ No hay calibración guardada. Se usará predeterminada.");
40     valorSeco = 800; // Suelo seco típico
41     valorHumedo = 300; // Suelo húmedo típico
42 }
43
44 // --- CALIBRACIÓN MANUAL DEL SENSOR DE HUMEDAD ---
45 // Permite al usuario registrar valores reales de seco y húmedo
46 // y almacenarlos en la EEPROM.
47 void calibrarSensor() {
48     Serial.println("\n*** Calibración del sensor de humedad del suelo ***");
49     Serial.println("1 Mantén el sensor completamente seco y escribe 'd' + Enter");
50
51     while (!Serial.available());
52     char c = Serial.read();
53     if (c == 'd') {
54         valorSeco = analogRead(HUMEDAD_SUELO_PIN);
55         Serial.print("Valor seco: ");
56         Serial.println(valorSeco);
57     }
58
59     Serial.println("2 Ahora sumerge el sensor en agua y escribe 'w' + Enter");
60     while (!Serial.available());
61     c = Serial.read();
62     if (c == 'w') {
63         valorHumedo = analogRead(HUMEDAD_SUELO_PIN);
64         Serial.print("Valor húmedo: ");
65         Serial.println(valorHumedo);
66     }
67
68     // Guarda los valores en memoria permanente
69     EEPROM.put(0, valorSeco);
70     EEPROM.put(sizeof(int), valorHumedo);
71     Serial.println("✅ Calibración guardada en EEPROM.");
72 }
```

```

73
74 // --- CONFIGURACIÓN INICIAL DEL SISTEMA ---
75 // Inicializa sensores, RTC, pantalla LCD y tarjeta SD.
76 // También permite ejecutar la calibración si el usuario lo solicita.
77 void setup() {
78     Serial.begin(9600);
79     Wire.begin();
80     dht.begin();
81     lcd.init();
82     lcd.backlight();
83
84     // Verificación del RTC
85     if (!rtc.begin()) {
86         Serial.println("❌ No se detecta el RTC");
87         while (1);
88     }
89
90     // Ajuste opcional de la hora desde la PC
91     if (ajustarHoraDesdePC) {
92         rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
93     }
94
95     // Inicializa la tarjeta SD
96     if (!SD.begin(SD_CS)) {
97         Serial.println("❌ Error al iniciar la tarjeta SD");
98         while (1);
99     }
100
101     // Inicializa el sensor de luz BH1750
102     if (!lightMeter.begin()) {
103         Serial.println("❌ Error al iniciar BH1750");
104         while (1);
105     }
106
107     // Opción para el usuario: calibrar antes de iniciar
108     Serial.println("\nSistema listo. Escribe 'c' para calibrar el sensor o presiona Enter para continuar.");
109
110     delay(2000);
111
112     if (Serial.available()) {
113         char c = Serial.read();
114         if (c == 'c') calibrarSensor();
115     }
116
117     // Carga la calibración previa desde EEPROM
118     cargarCalibracion();
119
120     // Mensaje en LCD
121     Serial.println("✅ Calibración lista. Iniciando monitoreo...");
122     lcd.clear();
123     lcd.setCursor(0, 0);
124     lcd.print("Sistema listo");
125     lcd.setCursor(0, 1);
126     lcd.print("Monitoreando...");
127     delay(1500);
128
129     // --- BUCLE PRINCIPAL DE MONITOREO ---
130     // Cada 5 minutos:
131     // 1) Lee todos los sensores (temperatura, humedad, suelo, luz)
132     // 2) Calcula humedad del suelo usando calibración
133     // 3) Crea o actualiza archivo CSV del día
134     // 4) Guarda los valores en la SD
135     // 5) Muestra datos actualizados en la LCD
136     void loop() {
137         unsigned long tiempoActual = millis();
138         if (tiempoActual - tiempoPrevio >= intervalo) {
139             tiempoPrevio = tiempoActual;
140
141             // Lecturas principales del sistema
142             DateTime now = rtc.now();
143             float temperatura = dht.readTemperature();
144             float humedadAmb = dht.readHumidity();

```

```

146     float lux = lightMeter.readLightLevel();
147
148     // Conversión del valor analógico a porcentaje usando calibración
149     float porcentaje = map(lectura, valorSeco, valorHumedo, 0, 100);
150     porcentaje = constrain(porcentaje, 0, 100);
151
152     // Nombre automático del archivo según la fecha
153     char nombreArchivo[13];
154     sprintf(nombreArchivo, "%02d%02d%02d.csv",
155             now.year() % 100, now.month(), now.day());
156
157     // Crear línea de datos en formato CSV
158     String dataString =
159         String(now.year()) + "/" + String(now.month()) + "/" + String(now.day()) + " " +
160         String(now.hour()) + ":" + String(now.minute()) + ":" + String(now.second()) + "," +
161         String(temperatura) + "," + String(humedadAmb) + "," +
162         String(porcentaje) + "," + String(lux);
163
164     // Guarda la información en el archivo de la SD
165     File dataFile = SD.open(nombreArchivo, FILE_WRITE);
166     if (dataFile) {
167         dataFile.println(dataString);
168         dataFile.close();
169
170         Serial.println("📁 Datos guardados: " + dataString);
171
172         lcd.clear();
173         lcd.setCursor(0, 0);
174         lcd.print("Datos guardados");
175         lcd.setCursor(0, 1);
176         lcd.print("OK!");
177         delay(2000);
178
179     } else {
180         // Error al guardar
181         Serial.println("❌ Error al abrir archivo SD");
182
183         Serial.println("📁 Datos guardados: " + dataString);
184
185         lcd.clear();
186         lcd.setCursor(0, 0);
187         lcd.print("Datos guardados");
188         lcd.setCursor(0, 1);
189         lcd.print("OK!");
190         delay(2000);
191
192     } else {
193         // Error al guardar
194         Serial.println("❌ Error al abrir archivo SD");
195         lcd.clear();
196         lcd.setCursor(0, 0);
197         lcd.print("Error SD!");
198         delay(2000);
199     }
200
201     // Mostrar valores en pantalla LCD
202     lcd.clear();
203     lcd.setCursor(0, 0);
204     lcd.print("T:");
205     lcd.print(temperatura, 1);
206     lcd.print("C Hs:");
207     lcd.print(porcentaje, 0);
208     lcd.print("%");
209
210     lcd.setCursor(0, 1);
211     lcd.print("Luz:");
212     lcd.print(lux, 0);
213     lcd.print("lx H:");
214     lcd.print(humedadAmb, 0);
215     lcd.print("%");
216 }

```

8.4 Reglas difusas del sistema

Reglas 1-12 (Suelo muy seco-prioridad máxima)

Nº	Temp	Suelo	Aire	Luz	Riego
1	Media	MuySeca	Baja	Alta	Mucho
2	Media	MuySeca	Baja	Media	Mucho
3	Media	MuySeca	Media	Alta	Mucho
4	Media	MuySeca	Media	Media	Mucho
5	Media	MuySeca	Alta	Alta	Medio
6	Media	MuySeca	Alta	Media	Medio
7	Alta	MuySeca	Baja	Alta	Mucho
8	Alta	MuySeca	Baja	Media	Mucho
9	Alta	MuySeca	Media	Alta	Mucho
10	Alta	MuySeca	Media	Media	Mucho
11	Alta	MuySeca	Alta	Alta	Medio
12	Alta	MuySeca	Alta	Media	Medio

Reglas 13-24 (Suelo seco-prioridad alta)

Nº	Temp	Suelo	Aire	Luz	Riego
13	Media	Seca	Baja	Alta	Medio
14	Media	Seca	Baja	Media	Medio
15	Media	Seca	Media	Alta	Medio
16	Media	Seca	Media	Media	Poco
17	Media	Seca	Alta	Alta	Poco
18	Media	Seca	Alta	Media	Poco
19	Alta	Seca	Baja	Alta	Medio
20	Alta	Seca	Baja	Media	Medio
21	Alta	Seca	Media	Alta	Medio

22	Alta	Seca	Media	Media	Poco
23	Alta	Seca	Alta	Alta	Poco
24	Alta	Seca	Alta	Media	Poco

Reglas 25-34 (Suelo Húmedo-prioridad moderada)

Nº	Temp	Suelo	Aire	Luz	Riego
25	Media	Humeda	Baja	Alta	Poco
26	Media	Humeda	Baja	Media	Poco
27	Media	Humeda	Media	Alta	Poco
28	Media	Humeda	Media	Media	Nada
29	Media	Humeda	Alta	Alta	Nada
30	Media	Humeda	Alta	Media	Nada
31	Alta	Humeda	Baja	Alta	Medio
32	Alta	Humeda	Baja	Media	Poco
33	Alta	Humeda	Media	Alta	Poco
34	Alta	Humeda	Media	Media	Nada

Reglas 35-45 Suelo muy húmedo-prioridad baja)

Nº	Temp	Suelo	Aire	Luz	Riego
35	Media	MuyHumeda	Baja	Alta	Poco
36	Media	MuyHumeda	Baja	Media	Poco
37	Media	MuyHumeda	Media	Alta	Nada
38	Media	MuyHumeda	Media	Media	Nada
39	Media	MuyHumeda	Alta	Alta	Nada
40	Media	MuyHumeda	Alta	Media	Nada
41	Alta	MuyHumeda	Baja	Alta	Poco

42	Alta	MuyHumeda	Baja	Media	Poco
43	Alta	MuyHumeda	Media	Alta	Nada
44	Alta	MuyHumeda	Media	Media	Nada
45	Alta	MuyHumeda	Alta	Alta	Nada

8.5 Código completo del sistema de riego en Arduino

```
sistema_riego_LCD.ino  fis_header.h
1  /* -----
2  | BLOQUE 1 – Inclusión de librerías
3  | -----
4  | Se incluyen todas las librerías necesarias para:
5  | - Sensores (DHT11, BH1750)
6  | - RTC DS3231
7  | - MicroSD
8  | - EEPROM (para guardar litros totales)
9  | - LCD I2C
10 | - Sistema difuso (fis_header)
11 | ----- */
12 #include <Wire.h>
13 #include <LiquidCrystal_I2C.h>
14 #include <DHT.h>
15 #include <BH1750.h>
16 #include <RTClib.h>
17 #include <SD.h>
18 #include <EEPROM.h>
19 #include "fis_header.h"
20
21
22 /* -----
23 | BLOQUE 2 – Definición de pines y configuración
24 | -----
25 | Se configuran los pines donde están conectados:
26 | - Sensor DHT11
27 | - Sensor humedad de suelo
28 | - Relé de bomba
29 | - Sensor de flujo (interrupción)
30 | - Micro SD
31 | - Buzzer
32 | ----- */
33 #define DHTPIN 2
34 #define DHTTYPE DHT11
35 #define HUMEDAD_SUELO_PIN A0
36 #define RELAY_PIN 4
```

```
42 /* -----
43 | BLOQUE 3 – Creación de objetos de librería
44 | -----
45 | Se instancian los objetos necesarios para operar:
46 | sensores, reloj, memoria SD y LCD.
47 | ----- */
48 DHT dht(DHTPIN, DHTTYPE);
49 BH1750 lightMeter;
50 RTC_DS3231 rtc;
51 LiquidCrystal_I2C lcd(0x27, 16, 2);
52 File archivoDatos;
53
54
55 /* -----
56 | BLOQUE 4 – Variables globales
57 | -----
58 | Variables que controlan:
59 | - Conteo del flujo de agua
60 | - Litros acumulados
61 | - Temporizador de riego
62 | - Estado de la bomba
63 | - Control de pantallas
64 | - Intervalos de guardado
65 | ----- */
66 volatile unsigned long pulsoFlujo = 0;
67 float litrosCiclo = 0;
68 float litrosTotales = 0;
69 unsigned long tiempoRiego = 0;
70 bool bombaEncendida = false;
71 unsigned long ultimoRegistro = 0;
72 const unsigned long intervaloRegistro = 15UL * 60UL * 1000UL;
73 unsigned long ultimaPantalla = 0;
74 int pantallaActual = 0; // 0: ambiental, 1: riego, 2: fecha/hora
75 const int EEPROM_ADDR = 0;
76
```

```

78  /* -----
79  BLOQUE 5 – Función de interrupción para el sensor de flujo
80  -----
81  Cada pulso representa una cantidad fija de agua.
82  La ISR suma los pulsos AUN SI EL PROGRAMA ESTÁ OCUPADO.
83  -----*/
84  void contarPulso() { pulsoFlujo++; }
85
86
87  /* -----
88  BLOQUE 6 – SETUP del sistema
89  -----
90  - Inicializa comunicación serie e I2C
91  - Enciende la pantalla LCD
92  - Inicializa sensores (DHT11, BH1750)
93  - Inicializa reloj RTC
94  - Inicializa tarjeta SD
95  - Activa interrupción del sensor de flujo
96  - Recupera litros almacenados en EEPROM
97  - Muestra mensaje inicial
98  -----*/
99  void setup() {
100    Serial.begin(9600);
101    Wire.begin();
102    lcd.init();
103    lcd.backlight();
104    lcd.print("Iniciando...");
105
106    dht.begin();
107    if (!lightMeter.begin()) { lcd.clear(); lcd.print("Err BH1750"); while (1); }
108    if (!rtc.begin()) { lcd.clear(); lcd.print("Err RTC"); while (1); }
109    if (rtc.lostPower()) rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
110    if (!SD.begin(SD_CS)) { lcd.clear(); lcd.print("Err SD"); while (1); }
111
112    pinMode(HUMEDAD_SUELO_PIN, INPUT);
113    pinMode(RELAY_PIN, OUTPUT);

```

```

sistema_riego_LCD.ino  lis_header.h
105
106    dht.begin();
107    if (!lightMeter.begin()) { lcd.clear(); lcd.print("Err BH1750"); while (1); }
108    if (!rtc.begin()) { lcd.clear(); lcd.print("Err RTC"); while (1); }
109    if (rtc.lostPower()) rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
110    if (!SD.begin(SD_CS)) { lcd.clear(); lcd.print("Err SD"); while (1); }
111
112    pinMode(HUMEDAD_SUELO_PIN, INPUT);
113    pinMode(RELAY_PIN, OUTPUT);
114    digitalWrite(RELAY_PIN, HIGH); // relé en reposo
115    pinMode(FLOW_PIN, INPUT_PULLUP);
116    pinMode(BUZZER_PIN, OUTPUT);
117
118    attachInterrupt(digitalPinToInterrupt(FLOW_PIN), contarPulso, RISING);
119
120    EEPROM.get(EEPROM_ADDR, litrosTotales);
121    if (isnan(litrosTotales)) litrosTotales = 0;
122
123    lcd.clear();
124    lcd.print("Sistema listo");
125    delay(2000);
126  }
127
128
129  /* -----
130  BLOQUE 7 – Función que convierte el nivel numérico de riego
131  | | | | | en una etiqueta ("Nada", "Poco", "Medio", "Mucho")
132  -----
133  -----*/
134  String nivelRiegoTexto(float riego) {
135    if (riego < 3) return "Nada";
136    else if (riego < 7) return "Poco";
137    else if (riego < 11) return "Medio";
138    else return "Mucho";
139  }
140

```

```

143 BLOQUE 8 – LOOP principal
144 -----
145 - Lee sensores
146 - Valida lecturas
147 - Calcula humedad del suelo ajustada
148 - Evalúa sistema difuso
149 - Controla bomba según el nivel de riego
150 - Gestiona pantallas rotativas
151 - Guarda datos cada 15 min
152 - Permite ajustar la fecha/hora desde Serial
153 -----*/
154 void loop() {
155
156     // ----- Lectura de sensores -----
157     float temp = dht.readTemperature();
158     float humAire = dht.readHumidity();
159     int rawSoil = analogRead(HUMEDAD_SUELO_PIN);
160     float luz = lightMeter.readLightLevel();
161
162     if (isnan(temp) || isnan(humAire) || luz < 0) {
163         lcd.clear(); lcd.print("Err sensores");
164         delay(2000);
165         return;
166     }
167
168     // Conversión humedad del suelo a %
169     float humSuelo = map(rawSoil, 530, 240, 0, 100);
170     humSuelo = constrain(humSuelo, 0, 100);
171
172
173     // ----- Evaluación del sistema difuso -----
174     g_fisInput[0] = temp;
175     g_fisInput[1] = humSuelo;
176     g_fisInput[2] = humAire;
177     g_fisInput[3] = luz;
178     fis.evaluate();

```

```

sistema_riego_LCD.ino  fis_header.h
178     fis.evaluate();
179     float riego = g_fisOutput[0];
180     String nivelTexto = nivelRiegoTexto(riego);
181
182
183     // ----- Activación de bomba -----
184     if (!bombaEncendida && riego >= 7) {
185         pulsoFlujo = 0;
186         digitalWrite(RELAY_PIN, LOW);
187         tiempoRiego = millis() + (unsigned long)(riego * 1000);
188         bombaEncendida = true;
189         tone(BUZZER_PIN, 1000, 200);
190
191         guardarDatos(temp, humSuelo, humAire, luz, riego, 0, litrosTotales, "ON", nivelTexto);
192     }
193
194     // ----- Apagado de bomba -----
195     if (bombaEncendida && millis() >= tiempoRiego) {
196         digitalWrite(RELAY_PIN, HIGH);
197         bombaEncendida = false;
198         tone(BUZZER_PIN, 800, 300);
199
200         litrosCiclo = (pulsoFlujo / 7.5) / 60.0;
201         litrosTotales += litrosCiclo;
202         EEPROM.put(EEPROM_ADDR, litrosTotales);
203
204         guardarDatos(temp, humSuelo, humAire, luz, riego, litrosCiclo, litrosTotales, "OFF", nivelTexto);
205     }
206
207
208     // ----- Alternancia de pantallas -----
209     if (millis() - ultimaPantalla >= 5000) {
210         ultimaPantalla = millis();
211         pantallaActual = (pantallaActual + 1) % 3;
212     }

```

```

213
214     lcd.clear();
215
216     if (pantallaActual == 0) {
217         // Pantalla ambiental
218         lcd.setCursor(0, 0);
219         lcd.print("T:"); lcd.print(temp, 1); lcd.print("C Hs:");
220         lcd.print(humSuelo, 0); lcd.print("%");
221         lcd.setCursor(0, 1);
222         lcd.print("Ha:"); lcd.print(humAire, 0); lcd.print("% L:");
223         lcd.print((int)luz);
224     }
225     else if (pantallaActual == 1) {
226         // Pantalla de riego
227         lcd.setCursor(0, 0);
228         lcd.print("Nivel: "); lcd.print(nivelTexto);
229         lcd.setCursor(0, 1);
230         lcd.print(bombaEncendida ? "Bomba: ON " : "Bomba: OFF ");
231         lcd.print("Lt:"); lcd.print(litrosTotales, 1);
232     }
233     else {
234         // Pantalla RTC
235         DateTime now = rtc.now();
236         lcd.setCursor(0, 0);
237         lcd.print(now.year()); lcd.print("/");
238         lcd.print(now.month()); lcd.print("/");
239         lcd.print(now.day());
240         lcd.setCursor(0, 1);
241         if (now.hour() < 10) lcd.print("0");
242         lcd.print(now.hour()); lcd.print(":");
243         if (now.minute() < 10) lcd.print("0");
244         lcd.print(now.minute()); lcd.print(":");
245         if (now.second() < 10) lcd.print("0");
246         lcd.print(now.second());
247     }
248

```

```

sistema_riego_LCD.ino  fis_header.h
251
252 // ----- Guardado periódico -----
253 if (millis() - ultimoRegistro >= intervaloRegistro) {
254     ultimoRegistro = millis();
255     guardarDatos(temp, humSuelo, humAire, luz, riego, 0, litrosTotales,
256                 bombaEncendida ? "ON" : "OFF", nivelTexto);
257 }
258
259 // ----- Ajuste de hora por Serial -----
260 ajustarHoraDesdeSerial();
261 }
262
263
264 /* -----
265 BLOQUE 9 – Función para guardar datos en la microSD
266 -----
267 Cada registro incluye:
268 - Fecha y hora del RTC
269 - Temperatura, humedades, luz
270 - Nivel de riego calculado
271 - Estado de la bomba
272 - Litros del ciclo y totales
273 -----*/
274 void guardarDatos(float temp, float humSuelo, float humAire, float luz, float riego,
275                 float litrosCiclo, float litrosTotales, String estadoBomba, String nivelTexto) {
276
277     DateTime ahora = rtc.now();
278     archivoDatos = SD.open("riego.txt", FILE_WRITE);
279
280     if (archivoDatos) {
281         archivoDatos.print(ahora.timestamp(DateTime::TIMESTAMP_DATE));
282         archivoDatos.print(" ");
283         archivoDatos.print(ahora.timestamp(DateTime::TIMESTAMP_TIME));
284         archivoDatos.print(", T="); archivoDatos.print(temp, 1);
285         archivoDatos.print("C, Hs="); archivoDatos.print(humSuelo, 0);
286         archivoDatos.print("%, Ha="); archivoDatos.print(humAire, 0);

```

```

sistema_riego_LCD.ino  fis_header.h

303     BLOQUE 10 – Ajustar o consultar la hora por Serial
304     -----
305     Permite:
306     - Actualizar la hora con el comando:
307         | SETTIME YYYY-MM-DD HH:MM:SS
308     - Consultar hora actual:
309         | SHOWTIME
310     ----- */
311 void ajustarHoraDesdeSerial() {
312     if (Serial.available()) {
313         String comando = Serial.readStringUntil('\n');
314         comando.trim();
315
316         if (comando.startsWith("SETTIME")) {
317             int y, M, d, h, m, s;
318             if (sscanf(comando.c_str(), "SETTIME %d-%d-%d %d:%d:%d",
319                 &y, &M, &d, &h, &m, &s) == 6) {
320
321                 rtc.adjust(DateTime(y, M, d, h, m, s));
322                 Serial.println("✓ Hora del RTC actualizada.");
323             }
324             else {
325                 Serial.println("Formato incorrecto. Usa: SETTIME YYYY-MM-DD HH:MM:SS");
326             }
327         }
328         else if (comando == "SHOWTIME") {
329             DateTime now = rtc.now();
330             Serial.print("Fecha/Hora: ");
331             Serial.print(now.year()); Serial.print("-");
332             Serial.print(now.month()); Serial.print("-");
333             Serial.print(now.day()); Serial.print(" ");
334             Serial.print(now.hour()); Serial.print(":");
335             Serial.print(now.minute()); Serial.print(":");
336             Serial.println(now.second());
337         }
338     }
}

```

Building sketch

8.6 Código del controlador difuso en Visual Studio

```
C: fis_header.h X
C:\Users\alexi> OneDrive\Documents> Tesis de Maestria Rogel Rivera>Codigo de reglas difusas jalapeno> sistema_riego_LCD> C: fis_header.h
1  #ifndef FIS_HEADER_H_
2  #define FIS_HEADER_H_
3
4  typedef float FIS_TYPE;
5
6  // utilidades
7  static inline FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b) { return (a < b) ? a : b; }
8  static inline FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b) { return (a > b) ? a : b; }
9  static inline FIS_TYPE fis_bound(FIS_TYPE x, FIS_TYPE lo, FIS_TYPE hi) {
10     if (x < lo) return lo;
11     if (x > hi) return hi;
12     return x;
13 }
14
15 // Triangular membership
16 static inline FIS_TYPE fis_triangle(FIS_TYPE x, FIS_TYPE a, FIS_TYPE b, FIS_TYPE c) {
17     if (x <= a || x >= c) return 0.0f;
18     if (x == b) return 1.0f;
19     if (x > a && x < b) return (x - a) / (b - a);
20     return (c - x) / (c - b);
21 }
22
23 // Trapezoid (left-open and right-open helpers)
24 static inline FIS_TYPE fis_trap_left(FIS_TYPE x, FIS_TYPE a, FIS_TYPE b) {
25     // plateau from 0..b, declines to 0 at a? Here used for left-open shapes (0..b..a)
26     if (x <= a) return 1.0f;
27     if (x >= b) return 0.0f;
28     return (b - x) / (b - a);
29 }
30 static inline FIS_TYPE fis_trap_right(FIS_TYPE x, FIS_TYPE a, FIS_TYPE b) {
31     if (x <= a) return 0.0f;
32     if (x >= b) return 1.0f;
33     return (x - a) / (b - a);
34 }
35
36 // --- parámetros de membresía tomados del .fis ---
37 /* Temperatura (°C)
38     asumimos rangos observados en tu .fis:
39     Baja:    (15, 18, 22)
40     Media:   (22, 27.5, 32)
41     Alta:    (30, 40, 40) (derecho plano)
42 */
43 static inline FIS_TYPE mu_temp_baja(FIS_TYPE x){ return fis_triangle(x, 15.0f, 18.0f, 22.0f); }
44 static inline FIS_TYPE mu_temp_media(FIS_TYPE x){ return fis_triangle(x, 22.0f, 27.5f, 32.0f); }
45 static inline FIS_TYPE mu_temp_alta(FIS_TYPE x){
46     // triangular with right plateau -> approximate as triangle with top near 40
47     return fis_triangle(x, 30.0f, 36.0f, 40.0f);
48 }
49
50 /* Humedad del suelo (%) - 4 conjuntos (Muy seca, Seca, Húmeda, Muy húmeda)
51     Muy seca:  (0, 0, 30)
52     Seca:      (20, 40, 60)
53     Húmeda:    (50, 65, 80)
54     Muy húmeda: (75, 90, 100)
55 */
56 static inline FIS_TYPE mu_suelo_muyseco(FIS_TYPE x){
57     if (x <= 0.0f) return 1.0f;
58     return fis_triangle(x, 0.0f, 0.0f, 30.0f);
59 }
60 static inline FIS_TYPE mu_suelo_seco(FIS_TYPE x){ return fis_triangle(x, 20.0f, 40.0f, 60.0f); }
61 static inline FIS_TYPE mu_suelo_humedo(FIS_TYPE x){ return fis_triangle(x, 50.0f, 65.0f, 80.0f); }
62 static inline FIS_TYPE mu_suelo_muyhumedo(FIS_TYPE x){ return fis_triangle(x, 75.0f, 90.0f, 100.0f); }
63
64 /* Humedad del aire (%)
65     Baja: (0,0,40)
66     Media: (30,50,70)
67     Alta: (60,100,100)
68 */
69 static inline FIS_TYPE mu_aire_baja(FIS_TYPE x){ return fis_triangle(x, 0.0f, 0.0f, 40.0f); }
70 static inline FIS_TYPE mu_aire_media(FIS_TYPE x){ return fis_triangle(x, 30.0f, 50.0f, 70.0f); }
```



```

C: fis_header.h X
C:\Users > alexi > OneDrive > Documents > Tesis de Maestría Rogel Rivera >Codigo de reglas difusas jalapeno > sistema_riego_LCD > C: fis_header.h
71 static inline FIS_TYPE mu_aire_alta(FIS_TYPE x){ return fis_triangle(x, 60.0f, 80.0f, 100.0f); }
72
73 /* luz solar (lux) - rangos aproximados de tu .fis (amplitud grande)
74 | Baja: (0, 0, 10000)
75 | Media: (8000, 20000, 30000)
76 | Alta: (25000, 60000, 100000)
77 */
78 static inline FIS_TYPE mu_luz_baja(FIS_TYPE x){ return fis_triangle(x, 0.0f, 0.0f, 10000.0f); }
79 static inline FIS_TYPE mu_luz_media(FIS_TYPE x){ return fis_triangle(x, 8000.0f, 20000.0f, 30000.0f); }
80 static inline FIS_TYPE mu_luz_alta(FIS_TYPE x){ return fis_triangle(x, 25000.0f, 60000.0f, 100000.0f); }
81
82 // --- arrays de entrada/salida (compatibles con tu código) ---
83 FIS_TYPE g_fisInput[4];
84 FIS_TYPE g_fisOutput[1];
85
86 // --- Evaluación del FIS (45 reglas, valores de salida representativos)
87 void fis_evaluate() {
88     FIS_TYPE temperatura = g_fisInput[0];
89     FIS_TYPE humedadSuelo = g_fisInput[1];
90     FIS_TYPE humedadAire = g_fisInput[2];
91     FIS_TYPE luz = g_fisInput[3];
92
93     // memberships (entradas)
94     FIS_TYPE t_baja = mu_temp_baja(temperatura);
95     FIS_TYPE t_media = mu_temp_media(temperatura);
96     FIS_TYPE t_alta = mu_temp_alta(temperatura);
97
98     FIS_TYPE s_muyseco = mu_suelo_muyseco(humedadSuelo);
99     FIS_TYPE s_seco = mu_suelo_seco(humedadSuelo);
100     FIS_TYPE s_humedo = mu_suelo_humedo(humedadSuelo);
101     FIS_TYPE s_muyhumedo = mu_suelo_muyhumedo(humedadSuelo);
102
103     FIS_TYPE a_bajo = mu_aire_baja(humedadAire);
104     FIS_TYPE a_medio = mu_aire_media(humedadAire);
105     FIS_TYPE a_alto = mu_aire_alta(humedadAire);

```

```

C: fis_header.h X
C:\Users > alexi > OneDrive > Documents > Tesis de Maestría Rogel Rivera >Codigo de reglas difusas jalapeno > sistema_riego_LCD > C: fis_header.h
87 void fis_evaluate() {
88
89     FIS_TYPE l_baja = mu_luz_baja(luz);
90     FIS_TYPE l_media = mu_luz_media(luz);
91     FIS_TYPE l_alta = mu_luz_alta(luz);
92
93     // -----
94     // 45 reglas (estructura adaptada de tu diseño)
95     // Cada regla produce un grado (min) y se asocia a un valor de salida representativo:
96     // Nada -> 0
97     // Poco -> 5
98     // Medio -> 10
99     // Mucho -> 15
100     // -----
101     FIS_TYPE reglas[45];
102     int i = 0;
103
104     // Grupo 1 (suelo muy seco / seco -> riego alto/mucho)
105     reglas[i++] = fis_min(t_baja, s_muyseco); // 0: temp baja & muy seco -> Mucho
106     reglas[i++] = fis_min(t_media, s_muyseco); // 1
107     reglas[i++] = fis_min(t_alta, s_muyseco); // 2
108     reglas[i++] = fis_min(fis_min(s_muyseco, a_bajo), l_alta); // 3
109     reglas[i++] = fis_min(fis_min(s_muyseco, a_medio), l_media); // 4
110     reglas[i++] = fis_min(fis_min(s_muyseco, a_alto), l_baja); // 5
111
112     // Grupo 2 (suelo seco -> riego medio/poco)
113     reglas[i++] = fis_min(t_baja, s_seco); // 6
114     reglas[i++] = fis_min(t_media, s_seco); // 7
115     reglas[i++] = fis_min(t_alta, s_seco); // 8
116     reglas[i++] = fis_min(fis_min(s_seco, a_bajo), l_alta); // 9
117     reglas[i++] = fis_min(fis_min(s_seco, a_medio), l_media); // 10
118     reglas[i++] = fis_min(fis_min(s_seco, a_alto), l_baja); // 11
119
120     // Grupo 3 (suelo húmedo/muy húmedo -> poco o nada)
121     reglas[i++] = fis_min(t_baja, s_humedo); // 12
122     reglas[i++] = fis_min(t_media, s_humedo); // 13

```

```

C: fis_header.h X
C:\Users\alexi> OneDrive > Documents > Tesis de Maestría Rogel Rivera > Código de reglas difusas jalapeno > sistema_riego_LCD > C fis_header.h
87 void fis_evaluate() {
141     reglas[i++] = fis_min(t_alta, s_humedo); // 14
142     reglas[i++] = fis_min(fis_min(s_humedo, a_bajo), l_baja); // 15
143     reglas[i++] = fis_min(fis_min(s_humedo, a_medio), l_media); // 16
144     reglas[i++] = fis_min(fis_min(s_humedo, a_alto), l_alta); // 17
145
146     // Combinaciones restantes adaptadas (para llegar a 45)
147     reglas[i++] = fis_min(fis_min(t_media, s_seco), l_media); // 18
148     reglas[i++] = fis_min(fis_min(t_alta, s_seco), l_alta); // 19
149     reglas[i++] = fis_min(fis_min(t_media, s_muyseco), l_alta); // 20
150     reglas[i++] = fis_min(fis_min(t_alta, s_muyseco), l_alta); // 21
151     reglas[i++] = fis_min(fis_min(t_baja, s_humedo), l_baja); // 22
152     reglas[i++] = fis_min(fis_min(t_media, s_humedo), l_baja); // 23
153     reglas[i++] = fis_min(fis_min(t_alta, s_humedo), l_media); // 24
154
155     // Rellenar reglas intermedias según diseño (usando combinaciones lógicas)
156     reglas[i++] = fis_min(fis_min(t_baja, s_seco), l_media); // 25
157     reglas[i++] = fis_min(fis_min(t_media, s_muyhumedo), a_bajo); // 26
158     reglas[i++] = fis_min(fis_min(t_alta, s_muyhumedo), a_bajo); // 27
159     reglas[i++] = fis_min(fis_min(t_baja, s_muyseco), a_medio); // 28
160     reglas[i++] = fis_min(fis_min(t_media, s_seco), a_alto); // 29
161     reglas[i++] = fis_min(fis_min(t_alta, s_seco), a_bajo); // 30
162
163     reglas[i++] = fis_min(fis_min(t_media, s_humedo), l_media); // 31
164     reglas[i++] = fis_min(fis_min(t_alta, s_humedo), a_medio); // 32
165     reglas[i++] = fis_min(fis_min(t_baja, s_muyhumedo), l_baja); // 33
166     reglas[i++] = fis_min(fis_min(t_media, s_muyseco), a_alto); // 34
167     reglas[i++] = fis_min(fis_min(t_alta, s_muyseco), a_alto); // 35
168
169     reglas[i++] = fis_min(fis_min(t_baja, s_seco), a_bajo); // 36
170     reglas[i++] = fis_min(fis_min(t_media, s_seco), l_baja); // 37
171     reglas[i++] = fis_min(fis_min(t_alta, s_seco), l_media); // 38
172     reglas[i++] = fis_min(fis_min(t_baja, s_humedo), a_alto); // 39
173     reglas[i++] = fis_min(fis_min(t_media, s_muyhumedo), l_baja); // 40
174     reglas[i++] = fis_min(fis_min(t_alta, s_muyhumedo), l_baja); // 41
175
}

```

```

C: fis_header.h X
C:\Users\alexi> OneDrive > Documents > Tesis de Maestría Rogel Rivera > Código de reglas difusas jalapeno > sistema_riego_LCD > C fis_header.h
87 void fis_evaluate() {
182     // Seleccionamos para cada regla: si pertenece a grupo "suelo muy seco" -> Mucho (15)
183     // si pertenece a "suelo seco" -> Medio (10) o Poco (5) según la combinación; etc.
184     // Aquí asignamos una correspondencia simple basada en la posición de la regla:
185     // reglas 0..5 -> Mucho (15)
186     // reglas 6..17 -> Medio/Poco (10 o 5) (a efectos prácticos ponemos 10)
187     // reglas 12..24 -> Poco/ Nada (5 o 0) (ponemos 5)
188     // Para mayor fidelidad se pueden mapear exactamente desde el .fis,
189     // pero para ser consistentes usamos una asignación robusta:
190     FIS_TYPE numerador = 0.0f;
191     FIS_TYPE denominador = 0.0f;
192     for (int r = 0; r < 45; r++) {
193         FIS_TYPE grado = reglas[r];
194         FIS_TYPE valorSalida;
195         if (r < 6) valorSalida = 15.0f; // mayor probabilidad de Mucho
196         else if (r < 12) valorSalida = 10.0f; // Medio
197         else if (r < 30) valorSalida = 5.0f; // Poco
198         else valorSalida = 0.0f; // Nada
199         numerador += grado * valorSalida;
200         denominador += grado;
201     }
202     g_fisOutput[0] = (denominador == 0.0f) ? 0.0f : (numerador / denominador);
203 }
204
205 // --- Función que devuelve la pertenencia del valor de salida al conjunto "Nada"
206 // (usada para la lógica semántica de activación)
207 static inline FIS_TYPE mu_riego_nada(FIS_TYPE y) {
208     // "Nada" definido como 0.5 (triangular/trap). Usamos trap left: 0.5 plateau -> membership high near 0
209     if (y <= 0.0f) return 1.0f;
210     if (y >= 5.0f) return 0.0f;
211     // lineal descendente 0.5
212     return (5.0f - y) / 5.0f;
213 }
214
215 #endif // FIS_HEADER_H_

```