

**UNIVERSIDAD NACIONAL DEL SANTA**  
**ESCUELA DE POSGRADO**  
**Programa de Doctorado en Matemática**



**UNS**  
**ESCUELA DE**  
**POSGRADO**

---

---

**Modelos matemáticos para el control de la incertidumbre  
en sensores de internet of things**

---

---

**Tesis para optar el grado de  
Doctor en Matemática**

**Autor:**

**Mg. Alcántara Ramírez, Manuel Abelardo**  
**Código Orcid: 0000-0001-9490-8447**

**Asesor:**

**Dr. Moore Flores, Teodoro**  
**Código ORCID: 0000-0002-1755-3459**  
**DNI N° 32763522**

**Línea de investigación**  
**Lógica difusa**

**Nuevo Chimbote - PERÚ**  
**2024**



**UNS**  
ESCUELA DE  
POSGRADO

## CERTIFICACIÓN DEL ASESOR

Yo, **Dr. Moore Flores, Teodoro**, por la presente certifico mi asesoramiento de la Tesis Doctoral titulado: **Modelos matemáticos para el control de la incertidumbre en sensores de internet Of Things**, elaborado por el **Mg. Alcántara Ramírez, Manuel Abelardo**, para obtener el grado de **Doctor en Matemática**, ha sido elaborado de acuerdo al Reglamento General de Grados y Títulos en la Escuela de Posgrado de la Universidad Nacional del Santa.

Nuevo Chimbote, noviembre del 2024

---

**Dr. Moore Flores, Teodoro**  
**ASESOR**  
**DNI N° 32763522**  
**Código ORCID: 0000-0002-1755-3459**



## AVAL DEL JURADO EVALUADOR

Tesis Doctoral titulado: **Modelos matemáticos para el control de la incertidumbre en sensores de internet Of Things**, elaborado por el Mg. Alcántara Ramírez, Manuel Abelardo.

Revisado y Aprobado por el Jurado Evaluador:

---

**Dr. Cedrón León, Ernesto Antonio**  
**Presidente**  
DNI: 32966495  
Código ORCID: 0000-0002-3198-831X

---

**Dr. Vera Obeso, Fidel Alejandro**  
**Secretario**  
DNI:32857547  
Código ORCID 0000-0002-9125-0464

---

**Dr. Moore Flores, Teodoro**  
**Vocal**  
DNI N° 32763522  
Código ORCID: 0000-0002-1755-3459



**UNS**  
ESCUELA DE  
POSGRADO

### ACTA DE EVALUACIÓN DE SUSTENTACIÓN DE TESIS

A los ocho días del mes de noviembre del año 2024, siendo las 13:00 horas, en el aula P-01 de la Escuela de Posgrado de la Universidad Nacional del Santa, se reunieron los miembros del Jurado Evaluador, designados mediante Resolución Directoral N° 545-2024-EPG-UNS de fecha 30.10.2024, conformado por los docentes: Dr. Ernesto Antonio Cedrón León (Presidente), Dr. Fidel Alejandro Vera Obeso (Secretario) y Dr. Teodoro Moore Flores (Vocal); con la finalidad de evaluar la tesis titulada **"MODELOS MATEMÁTICOS PARA EL CONTROL DE LA INCERTIDUMBRE EN SENSORES DE INTERNET OF THINGS"**; presentado por el tesista **Manuel Abelardo Alcántara Ramirez**, egresado del programa de **Doctorado en Matemática**.

Sustentación autorizada mediante Resolución Directoral N° 566-2024-EPG-UNS de fecha 05 de noviembre de 2024.

La presidenta del jurado autorizó el inicio del acto académico; producido y concluido el acto de sustentación de tesis, los miembros del jurado procedieron a la evaluación respectiva, haciendo una serie de preguntas y recomendaciones al tesista, quien dio respuestas a las interrogantes y observaciones.

El jurado después de deliberar sobre aspectos relacionados con el trabajo, contenido y sustentación del mismo y con las sugerencias pertinentes, declara la sustentación como APROBADO, asignándole la calificación de VEINTE.

Siendo las 13:50 horas del mismo día se da por finalizado el acto académico, firmando la presente acta en señal de conformidad.

  
Dr. Ernesto Antonio Cedrón León  
Presidente

  
Dr. Fidel Alejandro Vera Obeso  
Secretario

  
Dr. Teodoro Moore Flores  
Vocal

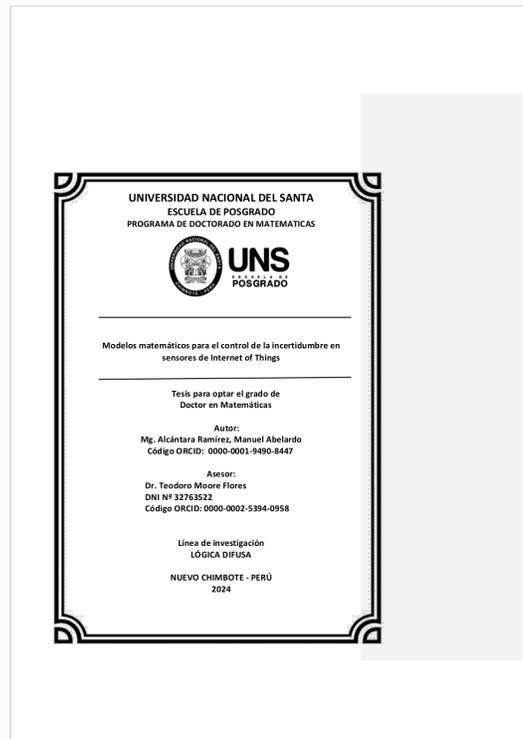


## Recibo digital

Este recibo confirma que su trabajo ha sido recibido por **Turnitin**. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega: Manuel Abelardo Alcantara Ramirez  
Título del ejercicio: tesis cueto  
Título de la entrega: tesis cueto  
Nombre del archivo: Informe\_final-tesis-Manuel\_Alcantara-15-01-25.docx  
Tamaño del archivo: 9.72M  
Total páginas: 176  
Total de palabras: 29,019  
Total de caracteres: 166,725  
Fecha de entrega: 21-ene.-2025 10:53a. m. (UTC-0500)  
Identificador de la entrega... 2568344597



# tesis cueto

## INFORME DE ORIGINALIDAD

12%

INDICE DE SIMILITUD

12%

FUENTES DE INTERNET

2%

PUBLICACIONES

5%

TRABAJOS DEL  
ESTUDIANTE

## FUENTES PRIMARIAS

1	<a href="http://repositorio.unac.edu.pe">repositorio.unac.edu.pe</a> Fuente de Internet	1%
2	<a href="http://hdl.handle.net">hdl.handle.net</a> Fuente de Internet	1%
3	<a href="http://repositorio.unap.edu.pe">repositorio.unap.edu.pe</a> Fuente de Internet	1%
4	<a href="http://repositorio.utec.edu.pe">repositorio.utec.edu.pe</a> Fuente de Internet	<1%
5	<a href="http://tesis.ipn.mx">tesis.ipn.mx</a> Fuente de Internet	<1%
6	<a href="http://mamacademy.es">mamacademy.es</a> Fuente de Internet	<1%
7	<a href="http://repositorio.ulasalle.edu.pe">repositorio.ulasalle.edu.pe</a> Fuente de Internet	<1%
8	<a href="http://www.poikonomou.com">www.poikonomou.com</a> Fuente de Internet	<1%
9	<a href="http://repositorio.upea.bo">repositorio.upea.bo</a> Fuente de Internet	<1%

## **DEDICATORIA**

A mis hijos Manuel Eduardo y Manuel Antonio como retribución al  
cariño mostrado por ambos

## **AGRADECIMIENTOS**

A mis colegas matemáticos que me apoyaron en la elaboración de este trabajo y a mi asesor Dr. TEODORO MOORE FLORES.

## ÍNDICE

DEDICATORIA .....	ii
AGRADECIMIENTOS .....	iii
ÍNDICE DE TABLAS .....	ix
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE ANEXOS.....	xiii
RESUMEN.....	xiv
ABSTRACT.....	xv
CAPÍTULO I. INTRODUCCIÓN .....	1
1.1 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN.....	1
1.2 DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN. ....	1
1.3 FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN. ....	3
1.4 OBJETIVOS.....	3
1.4.1 OBJETIVO GENERAL.....	3
1.4.2 OBJETIVOS ESPECÍFICOS.....	3
1.5 FORMULACION DE LAS HIPÓTESIS.....	4
1.5.1 HIPÓTESIS GENERAL.....	4
1.5.2 HIPÓTESIS ESPECÍFICAS.....	4
1.6 JUSTIFICACIÓN E IMPORTANCIA .....	5
1.6.1 JUSTIFICACIÓN.....	5
1.6.2 IMPORTANCIA.....	5
CAPITULO II. MARCO TEÓRICO .....	6
2.1 ANTECEDENTES DE LA INVESTIGACIÓN .....	6
2.1.1 ANTECEDENTES NACIONALES .....	6
2.1.2 ANTECEDENTES INTERNACIONALES .....	9
2.2 MARCO CONCEPTUAL.....	15
2.2.1 TEORÍA DE LA LÓGICA DIFUSA .....	15
2.2.1.1 INCERTIDUMBRE .....	16
2.2.1.2 VARIABLES LINGÜÍSTICAS.....	16
2.2.1.3 CONJUNTOS DIFUSOS .....	17
2.2.1.4 FUNCIONES DE MEMBRESÍA O DE PERTENENCIA.....	19
2.2.1.5 FUNCIÓN Z (FUNCIÓN HOMBRO DERECHO).....	21

2.2.1.6 FUNCIÓN GAMMA (FUNCIÓN HOMBRO IZQUIERDO) .....	22
2.2.1.7 FUNCIÓN TRIANGULAR .....	23
2.2.1.8 FUNCIÓN TRAPEZOIDAL .....	25
2.2.1.9 FUNCIÓN SIGMOIDAL .....	26
2.2.1.10 FUNCIÓN CAMPANA GENERALIZADA .....	26
2.2.1.11 FUNCIÓN GAUSSIANA .....	27
2.2.1.12 CÁLCULO DE LA FUNCIÓN DE MEMBRESÍA.....	30
2.2.1.13 PRINCIPIO DE EXTENSIÓN .....	32
2.2.1.14 OPERACIONES CON CONJUNTOS DIFUSOS.....	33
2.2.1.15 ÁLGEBRA DE BOOLE .....	37
2.2.1.16 ÁLGEBRA DIFUSA .....	39
2.2.1.17 NEGACIÓN FUERTE .....	40
2.2.1.18 NORMA TRIANGULAR .....	41
2.2.1.19 CONORMA TRIANGULAR .....	42
2.2.1.20 RELACIONES DIFUSAS.....	45
2.2.1.21 RELACIONES DE EQUIVALENCIA.....	47
2.2.1.22 RELACIONES DE ORDEN O COMPARACIÓN .....	47
2.2.1.23 PROYECCIÓN DIFUSA .....	48
2.2.1.24 EXTENSIÓN CILÍNDRICA.....	50
2.2.1.25 COMPOSICIÓN DE RELACIONES .....	51
2.2.1.26 EL CONDICIONAL DIFUSO.....	51
2.2.1.27 INFERENCIA DIFUSA .....	55
2.2.1.28 INFERENCIA TIPO MANDANI .....	57
2.2.1.29 INFERENCIA TIPO SUGENO .....	66
2.2.1.30 SISTEMAS DIFUSOS .....	68
2.2.2 TEORIA DE INTERNET OF THINGS (IOT).....	71
2.2.2.1 ARQUITECTURA DE UN SISTEMA IOT.....	71
2.2.2.2 CAPA DE DISPOSITIVOS: SENSORES .....	72
2.2.2.3 CAPA DE DISPOSITIVOS: ACTUADORES.....	72
2.2.2.4 CAPA DE DISPOSITIVOS: MICROCONTROLADORES.....	76
2.2.2.5 CAPA DE CONECTIVIDAD: CONEXIÓN A INTERNET .....	78
2.2.2.6 CAPA DE CONECTIVIDAD: PROTOCOLOS DE COMUNICACIÓN .....	79

2.2.2.7 CAPA DE COMPUTACION AL BORDE (EDGE COMPUTING) .....	80
2.2.2.8 CAPA DE AGREGACIÓN DE DATOS .....	81
2.2.2.9 CAPA DE PROCESAMIENTO .....	83
2.2.2.10 CAPA DE ALMACENAMIENTO .....	84
2.2.2.11 CAPA DE APLICACIÓN .....	88
CAPITULO III. METODOLOGÍA .....	90
3.1 ENFOQUE DE LA INVESTIGACIÓN .....	90
3.2 MÉTODO DE LA INVESTIGACIÓN .....	90
3.3 DISEÑO DE LA INVESTIGACIÓN .....	90
3.3.1 DEFINICIÓN DE VARIABLES .....	90
3.3.1.1 VARIABLE INDEPENDIENTE .....	90
3.3.1.2 VARIABLE DEPENDIENTE .....	90
3.3.2 SELECCIÓN DE MUESTRAS .....	90
3.3.3 IMPLEMENTACIÓN DEL EXPERIMENTO .....	91
3.3.4 ANÁLISIS DE DATOS .....	91
3.3.5 INTERPRETACIÓN DE RESULTADOS .....	91
3.4 POBLACIÓN .....	91
3.5 MUESTRA .....	91
3.6 OPERACIONALIZACIÓN DE LAS VARIABLES .....	92
3.6.1. VARIABLE INDEPENDIENTE: MODELO MATEMÁTICO .....	92
3.6.2. VARIABLE DEPENDIENTE: INCERTIDUMBRE EN DATOS DE SENSORES DE IOT .....	94
3.7 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS .....	96
3.8 TÉCNICAS DE ANÁLISIS DE LOS RESULTADOS. ....	97
3.8.1 ANÁLISIS ESTADÍSTICO DESCRIPTIVO .....	97
3.8.2 ANÁLISIS DE SENSIBILIDAD .....	97
CAPITULO IV RESULTADOS Y DISCUSIÓN .....	98
4.1 RESULTADOS .....	98
4.1.1 SELECCIÓN E IMPLEMENTACIÓN DE SENSORES .....	100
4.1.1.1 SELECCIÓN DE LOS SENSORES DE ESTUDIO .....	100
4.1.1.2 DISEÑO DE LOS CIRCUITOS DE LOS SENSORES .....	101
4.1.1.3 GEMELOS DIGITALES ASOCIADOS A LOS SENSORES .....	102

4.1.1.4 CODIGO ARDUINO GENERADO POR LOS GEMELOS DIGITALES.....	102
4.1.1.5 IMPLEMENTACIÓN FÍSICA DE LOS SENSORES.....	103
4.1.1.6 ALMACENAMIENTO EN LA NUBE DE LOS DATOS DE LOS SENSORES .....	104
4.1.1.7 MODELO DE BASE DE DATOS PARA ALMACENAMIENTO DE DATOS.....	105
4.1.1.8 PROCESAMIENTO DE DATOS EN UN ENTORNO LOCAL .....	106
4.1.2 FORMULACIÓN DE LOS MODELOS MATEMÁTICOS .....	108
4.1.2.1 VARIABLES LINGÜÍSTICAS.....	108
4.1.2.3 CONJUNTOS DIFUSOS ASOCIADOS A LOS SISTEMAS DIFUSOS.....	109
4.1.2.4 IMPLEMENTACIÓN DE LOS SISTEMAS DIFUSOS ASOCIADOS EN MATLAB. ....	112
4.1.2.5 DEFINICIÓN DE LAS REGLAS DE INFERENCIA EN MATLAB .....	112
4.1.2.6 SIMULACIÓN DEL SISTEMA DIFUSO CONTROL-AMBIENTAL.....	113
4.1.2.7 CODIGO ARDUINO DE LAS FUNCIONES DE PERTENENCIA.....	114
4.1.2.8 IMPLEMENTACIÓN EN LA CAPA DE BORDE DE LAS FUNCIONES DE PERTENENCIA.....	115
4.1.2.9 IMPLEMENTACIÓN DE LOS SISTEMAS DIFUSOS .....	116
4.1.2.10 MODELOS MATEMÁTICOS PROPUESTOS.....	120
4.1.2.11 IMPLEMENTACION COMPUTACIONAL DE LOS MODELOS MATEMÁTICOS .....	121
4.1.2.12 PARÁMETROS DE LA INCERTIDUMBRE DE SENSORES .....	122
4.1.2.13 EVALUACIÓN DE LOS MODELOS MATEMÁTICOS .....	122
4.2 DISCUSIÓN DE LOS RESULTADOS.....	126
4.2.1 CONTRASTACIÓN DE LAS HIPÓTESIS.....	126
4.2.2 COMPARACIÓN CON RESULTADOS DE OTRAS INVESTIGACIONES .....	127
4.2.3 FORTALEZAS EN EL DESARROLLO DEL PROYECTO .....	128
4.2.4 DEBILIDADES EN EL DESARROLLO DEL PROYECTO.....	129
4.2.5 OPORTUNIDADES DE DESARROLLO DE FUTURAS INVESTIGACIONES. ....	129
CAPITULO V. CONCLUSIONES Y RECOMENDACIONES .....	130
5.1 CONCLUSIONES.....	130
5.2 RECOMENDACIONES.....	131
REFERENCIAS BIBLIOGRÁFICAS .....	132
ANEXOS .....	139
ANEXO 1: DATASHEETS DE LOS SENSORES SELECCIONADOS .....	139

ANEXO 2: CIRCUITOS ELECTRÓNICOS DE LOS SENSORES SELECCIONADOS .....	141
ANEXO 3: CONEXIONES LÓGICAS DE LOS SENSORES SELECCIONADOS.....	143
ANEXO 4: GEMELOS DIGITALES DE LOS SENSORES SELECCIONADOS .....	145
ANEXO 5: CODIGO ARDUINO DE LOS SENSORES SELECCIONADOS .....	147
ANEXO 6: IMPLEMENTACIÓN FÍSICA DE LOS SENSORES SELECCIONADOS .....	148
ANEXO 7:CONJUNTOS DIFUSOS DE LAS OTRAS VARIABLES DE ESTUDIO.....	150
ANEXO 8: IMPLEMENTACIÓN DE LOS CONJUNTOS DIFUSOS EN MATLAB.....	157
ANEXO 9: DEFINICION DE LAS REGLAS DE INFERENCIA.....	159
ANEXO 10: SIMULACIÓN DE LOS OTROS SISTMAS DIFUSOS CONSIDERADOS .....	160

## ÍNDICE DE TABLAS

Cuadro 1: Valores de una función de membresía para un universo finito .....	19
Cuadro 2: Métodos para determinar las funciones de membresía.....	31
Cuadro 3: Operaciones del Álgebra de Boole.....	38
Cuadro 4: Propiedades del Álgebra difusa .....	39
Cuadro 5. Otras T-normas.....	43
Cuadro 6. Otras S-normas.....	44
Cuadro 7: Diversas formas del condicional difuso .....	54
Cuadro 8: Procesos en la inferencia tipo Mamdani .....	58
Cuadro 9: Principales métodos de defuzzificación .....	70
Cuadro 10: Arquitectura de un sistema IoT .....	71
Cuadro 11: Conversión de señales en los sensores .....	73
Cuadro 12: Ejemplos de sensores.....	74
Cuadro 13: Ejemplos de actuadores .....	75
Cuadro 14: Ejemplos de microprocesadores.....	77
Cuadro 15: Tecnologías de conexión a internet en IoT .....	78
Cuadro 16: Protocolos de comunicación en IoT .....	79
Cuadro 17: Herramientas para computación al borde .....	82
Cuadro 18: Herramientas para la capa de procesamiento IoT .....	86
Cuadro 19: Herramientas de la capa de almacenamiento IoT .....	87
Cuadro 20: Herramientas IoT en la nube .....	89
Cuadro 21: Selección de los sensores utilizados.....	100
Cuadro 22: Sistemas difusos de estudio.....	108
Cuadro 23: Variables y valores lingüísticos del estudio .....	108
Cuadro 24: Variables lingüísticas de los actuadores.....	109
Cuadro 25: Características de los modelos matemáticos propuestos .....	120
Cuadro 26: Características computacionales de los modelos matemáticos.....	121
Cuadro 27: Características de la incertidumbre de los sensores.....	122
Cuadro 28: Evaluación de los modelos MMT, MMH.....	123
Cuadro 29: Evaluación del modelo MMS .....	123
Cuadro 30: Evaluación del modelo MMG .....	124
Cuadro 31: Evaluación del modelo MMD .....	124
Cuadro 32: Evaluación del modelo MMM .....	125

## ÍNDICE DE FIGURAS

Figura 1: Gráfica de la función de membresía de un conjunto finito .....	20
Figura 2: Gráfica de la función de membresía para un conjunto infinito.....	21
Figura 3: Función Z (función hombro derecho) .....	22
Figura 4: Función gamma (Función hombro izquierdo) .....	23
Figura 5: Función triangular .....	24
Figura 6: Código en MATLAB para graficar una función triangular .....	24
Figura 7: Gráfica de la función triangular en MATLAB.....	25
Figura 8: Función trapezoidal.....	25
Figura 9: Función sigmoideal .....	26
Figura 10: Función campana generalizada .....	27
Figura 11: Función gaussiana .....	28
Figura 12: Funciones de membresía de los estados físicos del agua .....	29
Figura 13: Complemento de un conjunto difuso .....	35
Figura 14: Conjuntos difusos para ejemplificar operaciones gráficas.....	36
Figura 15: Unión de dos conjuntos difusos.....	36
Figura 16: Intersección de conjuntos difusos .....	37
Figura 17: Inferencia de Mamdani para dos reglas difusas y dos entradas nítidas .....	57
Figura 18 : Proceso de Fuzzificación en método de Mamdani .....	60
Figura 19. Obtención de los consecuentes de las reglas seleccionadas .....	61
Figura 20. Proceso de agregación en el método de Mamdani.....	63
Figura 21: Procesos de inferencia difusa tipo Mamdani .....	65
Figura 22: Procesos de la inferencia tipo Sugeno.....	66
Figura 23: Elementos de un sistema difuso.....	68
Figura 24: Arquitectura básica de la computación al borde .....	81
Figura 25: Diagrama de flujo del método de investigación usado.....	98
Figura 26: Esquema de un circuito electrónico para el sensor DHT11 .....	101
Figura 27: Conexión lógica del sensor de temperatura al microcontrolador .....	101
Figura 28: Gemelo digital (digital twin) del sensor DTH22 en Wokwi. ....	102
Figura 29: Código Arduino para el sensor DHT11 .....	103
Figura 30: Implementación física del sensor DHT11 .....	104
Figura 31: Almacenamiento de datos en la nube Arduino Cloud .....	104
Figura 32: Reporte de los datos almacenados en la nube.....	105
Figura 33: Base de datos de los sensores y actuadores seleccionados .....	105
Figura 34: Entorno de trabajo del servidor de archivos .....	106
Figura 35: Interfase de usuario para la gestión de sensores, actuadores.....	107
Figura 36: Interfase de usuario con daros captados por los sensores .....	107
Figura 37: Funciones de membresía de la variable humedad.....	110
Figura 38: Funciones de membresía de la variable temperatura.....	111
Figura 39: Funciones de membresía de la variable velocidad .....	111

Figura 40: Sistema difuso de control ambiental.....	112
Figura 41: Reglas de inferencia del sistema difuso control ambiental.....	113
Figura 42: Simulación del sistema Control-Ambiental .....	113
Figura 43: Una segunda simulación del sistema Control-Ambiental .....	114
Figura 44: Código de FuncionesMembresia.h .....	116
Figura 45: Código en C de las funciones de membresía .....	116
Figura 46: Código del archivo de cabecera InferenciaTipoMamdani.....	117
Figura 47: Código de la implementación de la inferencia tipo Mandani .....	118
Figura 48: Sistema difuso control ambiental en Arduino .....	119
Figura 49: Datasheet del sensor de gas .....	139
Figura 50: Datasheet del sensor de distancia .....	140
Figura 51: Circuito electrónico del sensor de distancia .....	141
Figura 52: Circuito electrónico del sensor de gas .....	141
Figura 53: Circuito electrónico del sensor de movimiento .....	142
Figura 54: Circuito electrónico del sensor de humedad del suelo .....	142
Figura 55: Conexión lógica del sensor de temperatura.....	143
Figura 56: Conexión lógica del sensor de humedad de la tierra.....	143
Figura 57: Conexión lógica del sensor de gases con Arduino .....	144
Figura 58: Conexión lógica del sensor de gas con ESP8266 .....	144
Figura 59: Conexión lógica del sensor de distancia con Arduino .....	145
Figura 60: Conexión lógica del sensor de distancia con ESP32 .....	145
Figura 61: Conexión lógica del sensor de movimiento .....	145
Figura 62: Gemelo digital del sensor de gas .....	146
Figura 63: Gemelo digital del sensor de movimiento.....	146
Figura 64: Gemelo digital del sensor de distancia .....	146
Figura 65: Código Arduino para el sensor de movimiento .....	147
Figura 66: Código Arduino para el sensor de humedad del suelo .....	147
Figura 67: Implementación física del sensor de humedad del suelo.....	148
Figura 68: Implementación física del sensor de distancia.....	148
Figura 69: Implementación física del sensor de humedad ambiental.....	149
Figura 70: Implementación física del sensor de gases .....	149
<b>Figura 71: Funciones de membresía de la variable humedad del suelo .....</b>	<b>150</b>
<b>Figura 72: Funciones de membresía de la velocidad de la bomba .....</b>	<b>151</b>
<b>Figura 73: Funciones de membresía de la variable concentración de gases .....</b>	<b>152</b>
<b>Figura 74: Funciones de membresía de la variable bomba extractora .....</b>	<b>153</b>
<b>Figura 75: Funciones de membresía de la variable distancia del objeto .....</b>	<b>154</b>
<b>Figura 76: Funciones de membresía de la variable volumen de alarma.....</b>	<b>154</b>
<b>Figura 77: Funciones de membresía de la variable distancia del objeto .....</b>	<b>155</b>
<b>Figura 78: Funciones de membresía de la variable posiciones del servomotor .....</b>	<b>156</b>
<b>Figura 79: Implementación del sistema Riego tecnificado en MATLAB.....</b>	<b>157</b>
<b>Figura 80: Implementación del sistema control de gases en MATLAB .....</b>	<b>157</b>
<b>Figura 81: Implementación del sistema control de movimiento en MATLAB .....</b>	<b>158</b>
<b>Figura 82: Implementación del sistema control de puertas en MATLAB .....</b>	<b>158</b>

<b>Figura 83:</b> Reglas de inferencia del sistema Riego tecnificado .....	159
<b>Figura 84:</b> Reglas de inferencia del sistema Riego tecnificado .....	159
<b>Figura 85:</b> Reglas de inferencia del sistema control de movimiento .....	159
<b>Figura 86:</b> Reglas de inferencia del sistema control de puertas .....	159
<b>Figura 87:</b> Simulación del sistema riego tecnificado .....	160
<b>Figura 88:</b> Simulación del sistema control de gases .....	160
<b>Figura 89:</b> Simulación del sistema control de movimiento .....	161
<b>Figura 90:</b> Simulación del sistema Control de puertas.....	161

## ÍNDICE DE ANEXOS

ANEXO 1: DATASHEETS DE LOS SENSORES SELECCIONADOS .....	139
ANEXO 2: CIRCUITOS ELECTRÓNICOS DE LOS SENSORES SELECCIONADOS .....	141
ANEXO 3: CONEXIONES LÓGICAS DE LOS SENSORES SELECCIONADOS.....	143
ANEXO 4: GEMELOS DIGITALES DE LOS SENSORES SELECCIONADOS .....	145
ANEXO 5: CODIGO ARDUINO DE LOS SENSORES SELECCIONADOS .....	147
ANEXO 6: IMPLEMENTACIÓN FÍSICA DE LOS SENSORES SELECCIONADOS .....	148
ANEXO 7: CONJUNTOS DIFUSOS DE LAS OTRAS VARIABLES DE ESTUDIO .....	150
ANEXO 8: IMPLEMENTACIÓN DE LOS CONJUNTOS DIFUSOS EN MATLAB .....	157
ANEXO 9: DEFINICIÓN DE LAS REGLAS DE INFERENCIA.....	159
ANEXO 10: SIMULACIÓN DE LOS OTROS SISTEMAS DIFUSOS CONSIDERADOS .....	160

## RESUMEN

El paradigma de Internet of Things (IoT) involucra la participación de diversos dispositivos conectados mediante internet para que interactuando entre sí resuelvan problemas concretos, los sistemas basados en IoT utilizan sensores para la captura de datos y actuadores para realizar tareas. Los sensores están expuestos a factores que pueden alterar su funcionamiento, la imprecisión o incertidumbre en los datos repercuten en la toma de decisiones. El objetivo es formular modelos matemáticos basados en lógica difusa para controlar la incertidumbre en los datos de los sensores y generar una salida adecuada para los actuadores. Se selecciona seis sensores representativos y se crean cinco sistemas según modelos matemáticos basados en lógica difusa, se simulan usando MATLAB, aplicando el paradigma de computación al borde se crean librerías y programas par Arduino y se graba el código generado en la memoria del microcontrolador ESP8266/ESP32 para realizar el control de manera autónoma, se hacen las implementaciones físicas de los sistemas y se prueba su funcionamiento. Como resultado se observa que las implementaciones físicas de los sistemas tratan la incertidumbre de los datos con la misma precisión que el sistema difuso simulado en MATLAB. Se concluye que es pertinente tratar la incertidumbre de los datos de los sensores mediante la adecuación a un sistema difuso en base a un modelo matemático que establezca las variables lingüísticas, los conjuntos difusos, las reglas condicionales y el método de inferencia, la computación al borde permite la implementación de las funciones de membresía en el microcontrolador. Se recomienda repetir la investigación con sensores de procesos más críticos como los de medicina e incorporar sistemas neuro difusos para conseguir aprendizaje autónomo.

### **Palabras claves:**

Internet of Things, Lógica difusa, Sensores, Actuadores, Funciones de membresía, Inferencia difusa.

## **ABSTRACT**

The Internet of Things (IoT) paradigm involves the participation of various devices connected via the internet to interact with each other to solve concrete problems. IoT-based systems use sensors for data capture and actuators to perform tasks. Sensors are exposed to factors that can alter their functioning, and inaccuracies or uncertainties in the data impact decision-making. The goal is to formulate mathematical models based on fuzzy logic to control the uncertainty in sensor data and generate an appropriate output for the actuators. Six representative sensors are selected, and five systems are created according to mathematical models based on fuzzy logic. These systems are simulated using MATLAB. Applying the edge computing paradigm, libraries and programs for Arduino are created, and the generated code is recorded in the ESP8266/ESP32 microcontroller memory to perform autonomous control. The physical implementations of the systems are made and their operation is tested. As a result, it is observed that the physical implementations of the systems handle data uncertainty with the same precision as the fuzzy system simulated in MATLAB. It is concluded that it is appropriate to address data uncertainty from sensors by adapting to a fuzzy system based on a mathematical model that establishes linguistic variables, fuzzy sets, conditional rules, and the inference method. Edge computing enables the implementation of membership functions in the microcontroller. It is recommended to repeat the research with sensors for more critical processes, such as those in medicine, and to incorporate neuro-fuzzy systems to achieve autonomous learning.

### **Keywords:**

Internet of Things, Fuzzy logic, Sensors, Actuators, Membership functions, Fuzzy inference.

# **CAPÍTULO I.**

## **INTRODUCCIÓN**

### **1.1 DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN.**

En este apartado se presenta el planteamiento y fundamentación del problema motivo de la investigación considerando la formulación del problema, los objetivos, la formulación de la hipótesis, la justificación e importancia.

### **1.2 DESCRIPCIÓN DEL PROBLEMA DE INVESTIGACIÓN.**

Actualmente Internet of Things (IoT) es una rama del conocimiento que está experimentando un crecimiento exponencial en las aplicaciones. La idea básica de esta disciplina consiste en usar dispositivos especializados denominados sensores para capturar los datos, procesarlos con microcontroladores y luego comunicarse con otros dispositivos denominados actuadores para realizar una actividad concreta (Prasetyo et al., 2023), (Xiao, 2018), (Maulana et al., 2023).

En este entorno de trabajo, la incertidumbre y la variabilidad de los datos captados por los sensores pueden distorsionar las salidas del sistema de Internet of Things y reportar datos no confiables que pueden dificultar la toma de decisiones (Hu, 2016), (Krushnan et al., 2023), (Bakar, 2022).

Los datos captados por los sensores a menudo están expuestos a diversos problemas que pueden generar diferentes tipos de incertidumbre como errores durante la medición, ruido, mala calibración de los equipos, existencia de interferencias electromagnéticas y condiciones ambientales cambiantes (Hu, 2016), (Krushnan et al., 2023), (Bakar, 2022).

Esta incertidumbre o ambigüedad puede afectar la precisión y confiabilidad de las decisiones basadas en estos datos. En muchas aplicaciones de IoT como sistemas de control, diagnóstico médico o gestión de recursos, las decisiones tomadas en función de los datos de sensores deben ser confiables y precisos. La falta de consideración del fenómeno de la incertidumbre puede llevar a decisiones subóptimas o incluso incorrectas. (Hillar, 2016), (Prasetyo et al., 2023), (Maulana et al., 2023).

Se conoce que existen algunos enfoques para el manejo de la incertidumbre en la captura de datos como las propuestas basadas en la estadística, pero estos enfoques tradicionales no son suficientemente flexibles para lidiar actualmente con la variabilidad y la vaguedad presentes en los datos de los sensores (Hillar, 2016), (Prasetyo et al., 2023), (Maulana et al., 2023).

La lógica difusa es una rama de las matemáticas que trata de manera formal y rigurosa las imprecisiones, incertidumbres o vaguedades a la que se le denomina borrosidad. Esta disciplina conforma una estructura de conocimientos que extiende el tratamiento de lo preciso a lo impreciso (Zadeh, 1965), (Klir y Yuan, 1995), (Mohammed y Bahjat, 2023).

Por lo anteriormente expuesto, es pertinente establecer como usar la lógica difusa para abordar la incertidumbre que se presenta en la captación de los datos a través de los sensores en sistemas IoT. Se trata de proponer modelos matemáticos basados en lógica difusa que permitan a los sistemas IoT interpretar, comprender y utilizar de manera efectiva los datos recopilados por sensores, mejorando la toma de decisiones, la precisión y la adaptabilidad del sistema IoT a entornos cambiantes y con datos ambiguos.

### **1.3 FORMULACIÓN DEL PROBLEMA DE INVESTIGACIÓN.**

Se considera buscar la forma de tratar la incertidumbre en la captura de datos a través de los sensores en sistemas de Internet of Things a fin de evitar posibles errores en la toma de decisiones futuras basadas en esta información, luego surge la siguiente interrogante, ¿De qué manera se puede utilizar los principios y técnicas de la lógica difusa para gestionar la incertidumbre que se presenta en la captación de datos a través de los sensores de los sistemas basados en Internet of Things?

### **1.4 OBJETIVOS**

Para resolver el problema se plantean los siguientes objetivos:

#### **1.4.1 OBJETIVO GENERAL**

Formular modelos matemáticos usando lógica difusa para gestionar la incertidumbre presente en la captación de los datos mediante los sensores en sistemas de Internet of Things.

#### **1.4.2 OBJETIVOS ESPECÍFICOS**

**OE1.** Utilizar técnicas de la lógica difusa adecuadas para proponer modelos matemáticos que permitan gestionar la incertidumbre en los datos de los sensores de un sistema de Internet of Things.

**OE2.** Aplicar los modelos matemáticos propuestos en sistemas difusos concretos y simular su funcionamiento con software especializado.

**OE3.** Implementar físicamente los sistemas difusos basados en los modelos matemáticos propuestos utilizando el paradigma de computación al borde.

**OE4.** Evaluar las características de los modelos matemáticos propuestos utilizando los resultados de los sistemas difusos simulados con los resultados de la implementación física de los mismos.

## **1.5 FORMULACION DE LAS HIPÓTESIS**

Para lograr los objetivos, se plantean las siguientes hipótesis:

### **1.5.1 HIPÓTESIS GENERAL**

Es posible establecer criterios para formular modelos matemáticos basados en lógica difusa que permitan gestionar la incertidumbre presente en la captación de los datos de los sensores en sistemas de Internet of Things.

### **1.5.2 HIPÓTESIS ESPECÍFICAS**

**HE1.** Las variables lingüísticas, conjuntos difusos, funciones de membresía, operaciones, relaciones e inferencias difusas constituyen la base para formular modelos matemáticos que permitan gestionar la incertidumbre en los datos de los sensores de un sistema de Internet of Things.

**HE2.** El uso del software MATLAB permite simular el funcionamiento de sistemas difusos diseñados en base a modelos matemáticos propuestos con técnicas de lógica difusa que gestionen la incertidumbre en los datos de sensores.

**HE3.** Las técnicas de la capa de computación al borde de la arquitectura de un sistema de Internet of Things hacen factible implementar físicamente las propuestas formuladas en los modelos matemáticos basados en lógica difusa para la gestión de la incertidumbre en datos de sensores.

**HE4.** Es perfectamente factible evaluar las características de los modelos matemáticos basados en lógica difusa para gestionar la incertidumbre en los datos de los sensores mediante la comparación de los resultados de los sistemas difusos simulados con los resultados de la implementación física de estos sistemas difusos.

## **1.6 JUSTIFICACIÓN E IMPORTANCIA**

### **1.6.1 JUSTIFICACIÓN**

La justificación de este proyecto de investigación se fundamenta en la necesidad de abordar las limitaciones y desafíos relacionados con la incertidumbre en los datos capturados por sensores en entornos de Internet of Things (IoT).

Además, es necesario contar con propuestas económicas para solucionar la incertidumbre en los datos de los sensores de Internet of Things que no implique la adopción de equipos especializados que aumentan el costo en las implementaciones de los proyectos.

La gestión de la incertidumbre en los datos de los sensores resulta crucial en las aplicaciones críticas, donde la toma de decisiones debe estar exenta de errores.

### **1.6.2 IMPORTANCIA**

La importancia de este proyecto radica en el aporte que se hace a la ciencia de Internet of Things a través de una propuesta para optimizar la performance de los proyectos basados en esta tecnología. Se trata de una solución que se fundamenta en modelos matemáticos creados a partir de la lógica difusa y que su implementación no requiere de dispositivos electrónicos adicionales para esta optimización. En suma, solamente se trata de la implementación de la programación de estas ideas matemáticas en lo que se denomina computación al borde.

## **CAPITULO II.**

### **MARCO TEÓRICO**

#### **2.1 ANTECEDENTES DE LA INVESTIGACIÓN**

En los repositorios nacionales y en las bases de datos bibliográficas internacionales se consultó sobre investigaciones anteriores relacionadas con el tema materia de estudio. Se presentan a continuación los antecedentes que a criterio del investigador resultan más afines con la propuesta de investigación.

##### **2.1.1 ANTECEDENTES NACIONALES**

En la tesis doctoral de título “Diseño del control de nivel y temperatura de un sistema mezclador para la industria textil usando lógica difusa” desarrollada por Utrilla Mego Limberg Walter, realizada en la Universidad Nacional San Agustín de Arequipa el año 2020, se abordó el problema de mejorar los procesos de la industria textil para lo cual se propone crear un sistema de nivel y temperatura para mezclar soluciones de manera proporcional para mejorar el rendimiento en los procesos de la industria textil. Se trata de crear un modelo matemático que describa el comportamiento general del sistema, para esto se diseñó tres controladores, uno tipo proporcional-integral (PI), otro tipo proporcional-integral-derivativo (PID) y un control basado en lógica difusa de modo muestren altos índices de performance, para realizar las comparaciones y validación. Como resultado se demostró que la performance obtenida por el diseño propuesto mostró un alto rendimiento ya que es capaz de mantener sobreimpulso nulo, menor tiempo de establecimiento y un error en estado estacionario nulo (Utrilla, 2020).

Como otro antecedente nacional se considera a “Diseño e implementación de controladores basados en lógica difusa para una planta de procesos multipropósito”,

una tesis realizada por Pampamallco Jara José Luis, el año 2022 en la Universidad de Ingeniería y Tecnología en Lima donde se plantea resolver la complejidad en los procesos en los cuales no se puede estimar el modelo dinámico y el cálculo matemático de un controlador proporcional integral y derivativo (PID), Para tal propósito se propone utilizar un controlador difuso desarrollando bloques de control para la implementación en un controlador lógico programable (PLC) Siemens. Se debe controlar los procesos de flujo, presión, temperatura y nivel, primero se determinan sus respectivos modelos dinámicos mediante la adquisición de datos experimentales usando software, se implementaron los sistemas de control PID y sistemas de control difuso en el software TIA Portal. Como resultados se obtuvo que el controlador PID presenta mejor desempeño basándonos en tiempo de establecimiento y error en estado estable, a diferencia del controlador difuso. Por tanto, en los procesos que requiera el control de la variable de flujo o presión de agua en una tubería resulta más conveniente la aplicación del controlador PID, debido a su rápida respuesta y mayor precisión. Por otro lado, el controlador difuso tiene un mejor desempeño que el controlador PID en procesos bajo ciertas condiciones. Finalmente, en la estrategia de control descentralizado los resultados muestran un menor tiempo de establecimiento, error en estado estable y porcentaje de sobreimpulso del controlador difuso a diferencia del controlador PID (Pampamallco-Jara, 2022).

La investigación realizada en la Universidad La Salle de Arequipa el año 2020 por Loza Peralta Christian Víctor denominada “Arquitectura de software basada en microservicios para el uso en dispositivos de Internet of Things” aborda la problemática en el desempeño de los proyectos de Internet of Things(IoT). El

objetivo fue implementar una arquitectura de software basada en microservicios para el uso en dispositivos IoT. Se propuso una solución en tres escenarios independientes que funcionen como servicios web. Para el desarrollo del software se usó Scrum bajo el diseño dirigido por el dominio, implementado Jakarta EE, MySQL y Docker, para el hardware se utilizó la placa Arduino UNO y otros dispositivos electrónicos, permitiendo que se pueda comunicar con los servicios web, manteniendo alta cohesión entre los diversos dispositivos utilizados, servicios y protocolos de comunicación. La arquitectura basada en microservicios fue evaluada a través de los atributos de calidad tales como la modificabilidad, disponibilidad y portabilidad. Mientras que los dispositivos IoT son evaluados por medio de un conjunto de pruebas para garantizar la fiabilidad de su funcionamiento y comunicación, obteniendo resultados significativos (Loza\_Peralta, 2020).

En la tesis doctoral “Modelo de recomendación de dietas saludables mediante algoritmos de optimización, lógica difusa y lógica de primer orden” de Flor Cagniy Cárdenas Mariño realizada en la Escuela de Posgrado del Doctorado en Ciencias de la Computación de la Universidad Nacional del Altiplano, Puno en el año de 2021. Se aborda el problema de la alimentación saludable. Se planteó como objetivo diseñar un sistema experto basado en lógica difusa y un algoritmo basado en lógica de primer orden. Mientras que el sistema difuso recomienda dietas saludables según la edad, el peso, la talla y el nivel de actividad física para obtener las kilocalorías diarias requeridas, el algoritmo de lógica de primer orden muestra un menú considerando el de desayuno, media mañana, almuerzo, media tarde y cena de acuerdo con los gustos de la persona. Una vez obtenida las kilocalorías y la lista de menús mediante un algoritmo de selección, el sistema experto difuso selecciona el

plan de menú que más se ajuste a las kilocalorías de la persona, además la persona puede decidir si quiere subir de peso o bajar de peso, entonces el sistema experto muestra el plan que más se ajuste a las necesidades de la persona. Para validar el modelo propuesto se ha tomado en cuenta 200 personas de la base de datos dataset kaggle, de los cuales se tomó los datos personales necesarios y además se simuló los gustos y preferencias de cada persona en la comida de cada persona, finalmente al aplicar el modelo, el sistema experto recomienda para cada persona el plan de menú de acuerdo con kilocalorías y gustos (Cárdenas, 2021).

En la tesis de título “Diseño del Fitotrón aplicando lógica difusa”, realizado por Reyna Salas Richard Ronaldo y Puente Camasca Raúl Ignacio el año 2021 en la Universidad Ricardo Palma se propone la simulación de un sistema difuso que tiene como entradas la temperatura, humedad e iluminación y como salida el clima adecuado para la germinación de plantas, se trazaron como objetivo mejorar las condiciones ambientales para la agricultura estudiando los factores que inciden en la germinación de plantas comestibles. Para tal propósito consideraron la humedad, la temperatura y la luminosidad como factores determinantes, utilizaron sensores, la placa Arduino como microcontrolador, el software Eclipse, y un calefactor para implementar el sistema difuso que lo denominaron Fitotrón, para la simulación hicieron uso de MATLAB (Reyna y Puente, 2021).

### **2.1.2 ANTECEDENTES INTERNACIONALES**

Cono un primer antecedente internacional se considera al trabajo denominado “Design and Development of Efficient Water Management for Agriculture using IoT and Fuzzy Logic” realizado el 2020 por Satis Dekka, Narasimha Raju, Satish Bonus, Mangalagiri Pallavi, Pudi Ganesh en la India. En esta investigación abordan

la problemática que existe en la agricultura respecto al suministro de agua y fertilizantes. Su objetivo fue establecer como utilizar las tecnologías de inteligencia artificial e Internet of Things en la solución de este problema, para este propósito aplican lógica difusa, una subrama de la inteligencia artificial en el control del funcionamiento del motor que permite lograr un suministro equilibrado de agua. Utilizaron Internet of Things para obtener información de sensores a fin de detectar datos de las fugas en las tuberías, datos de las lluvias y el suministro oportuno de agua. Como resultado lograron demostrar que las técnicas planteadas resuelven los problemas de fugas de agua que fluye a través de tuberías utilizando IoT y se controla inteligentemente el suministro de agua mediante lógica difusa. Se concluye que esta técnica es un proceso económico y ayuda a mejorar la productividad de los cultivos y por ende aumentar el producto bruto interno de su país (Dekka et al., 2020).

Se toma en cuenta como otro antecedente internacional a la investigación realizada por Dakhole Dipali, Thiruselvan Subramanian y Senthil Kumaran de título “A smart oyster mushroom cultivation using automatic fuzzy logic controller”, realizado en el departamento de ciencias de la computación e ingeniería de la Presidency University de la India y publicado el año 2023. Abordan el tema del cultivo de hongos tipo ostra que requiere mucha mano de obra y crece en un entorno sensible. Se plantearon como objetivo diseñar un sistema autónomo de control ambiental para optimizar y ofrecer las condiciones adecuadas para el cultivo. Para tal propósito utilizaron un nodo de Internet of Things que utilice sensores DHT22, una placa ESP32 para optimizar el uso adecuado de la temperatura y la humedad y para el riego utilizaron relés conectados a los actuadores controlados con lógica

difusa. Utilizaron como actuadores un aspersor, un ventilador de refrigeración, un humidificador y un calentador que son encendidos o apagados según los valores actuales de temperatura y humedad mediante la técnica de lógica difusa. Los datos son recopilados en cada nodo IoT y se visualizan en la plataforma ThingSpeak. Como resultado muestran que el sistema propuesto es más eficiente que el sistema manual de prendido y apagado de los actuadores (Dipali et al., 2023).

En la investigación denominada “An Industry 4.0 implementation of a condition monitoring system and IoT-enabled predictive maintenance scheme for diesel generators” realizado el año 2023 por Ambarish Gajendra Mohapatra, Anita Mohanty, Nihar Ranjan Pradhan, Sachi Nandan Mohanty, Deepak Gupta, Meshal Alharbi, Ahmed Alkhayyat y Ashish Khanna, reconocidos investigadores de diversas universidades de la India, Arabia Saudita e Irak, Se trata de monitorear de manera rutinaria los parámetros cruciales de los generadores diésel al usarse como una fuente de energía complementaria viable para garantizar un suministro de energía sin interrupciones. Plantearon sistemas de monitoreo de condición basados en Internet of Things para suplantar el método tradicional de mantenimiento de equipos. Primero presentan un esquema de adquisición de datos y monitoreo remoto para realizar el concepto de mantenimiento predictivo, luego analizan una estrategia para la observación en tiempo real de los parámetros del generador Diesel para hacer un análisis integral de varias métricas, además se incluye un esquema de monitoreo y análisis de factores cruciales en un generador Diesel como la velocidad del motor, la salida de voltaje, la corriente producida, el factor de potencia, el refrigerante requerido, el consumo de combustible y el estado de la batería. Se formulan diferentes modelos matemáticos correlacionando datos experimentales y

estimando los coeficientes. Finalmente, para crear advertencias adecuadas en tiempo real en circunstancias críticas, se presenta un sistema de soporte a la decisión basado en lógica difusa y elementos de integración basados en web (Mohapatra et al., 2023).

En la investigación “An iot-based system for fault detection and diagnosis in solar pv panels” llevada a cabo por Balakrishnan, Raja, Manikandan Rajagopal, Sudhakar, y Janani. en la India el año 2023 se estudia la detección y diagnóstico de fallas en paneles solares fotovoltaicos. Se trazaron como objetivo mejorar el rendimiento y la confiabilidad de los paneles solares fotovoltaicos, los cuales pueden verse afectados por diversas causas, como sombras, suciedad, degradación y fallas eléctricas. Proponen como solución algoritmos de detección de fallas basados en lógica difusa que se incorporan en un sistema conformado por nodos de sensores inalámbricos que se implementan en los paneles para recopilar datos sobre sus parámetros eléctricos y condiciones ambientales, como temperatura, irradiancia y humedad. Luego, los datos recopilados se transmiten a un servidor central para su procesamiento y análisis mediante algoritmos de aprendizaje automático. El sistema puede detectar y diagnosticar fallas en tiempo real y proporcionar alertas y recomendaciones al personal de mantenimiento para que tome las medidas adecuadas para evitar mayores daños o tiempo de inactividad. Como resultado se demostró que el sistema propuesto tiene varias ventajas sobre los métodos tradicionales de inspección y mantenimiento manual ya que incluye un menor tiempo de inactividad, menores costos de mantenimiento y una mejor eficiencia energética. El sistema propuesto ha sido validado mediante pruebas experimentales

y los resultados muestran que puede detectar y diagnosticar fallas en paneles solares fotovoltaicos con alta confiabilidad y eficiencia (Balakrishnan et al., 2023).

Como otro antecedente internacional se considera al trabajo académico denominado “Fuzzy logic implementation in internet of things technology for fogging greenhouse plants” cuyos autores son Suyud Widiono y Iwan Hartadi Tri Untoro, realizado en la Universidad Tecnológica de Yogyakarta en Indonesia el año 2023. En esta investigación se trata de resolver el problema de nebulización manual de plantas en un invernadero. El objetivo fue crear un sistema de nebulización automática para plantas en un invernadero utilizando técnicas de lógica difusa. Para captar datos de temperatura y humedad del ambiente se utilizó el sensor DHT11, los datos se almacenaron en la nube Firebase y se mostró en tiempo real con una aplicación de Android creada con MIT App Inventor 2. Se utilizó lógica difusa para producir valores PWM (modulación de ancho de pulso) para configurar la niebla. Se logró mapear con los resultados los problemas de datos de entrada con los datos de salida al realizar dicho control de niebla. De la prueba del sensor, los resultados obtenidos de la comparación de las lecturas del sensor DHT11 tienen un error promedio del 2,73%. La duración del riego se realiza mediante lógica difusa, la cantidad total de riego en un día es de 2.500 ml de agua con una duración de 75 segundos, para que la nebulización con el método de lógica difusa sea acorde a las condiciones necesarias en el invernadero (Widiono y Tri Untoro, 2023).

En el artículo científico “A digital twinning approach for the internet of unmanned electric vehicles (iouevs) in the metaverse” (Mohsen et al., 2023), se aborda el tema de la aplicación de Internet of Things y la tecnología del metaverso para mejorar la digitalización de los sistemas de transporte público. Plantearon la mejora del IoT y

los vehículos eléctricos no tripulados en el metaverso. Se proponen describir de un marco sobre cómo los vehículos eléctricos no tripulados pueden ser utilizados en el metaverso y la creación de un gemelo digital para un vehículo eléctrico no tripulado. Para lograr su objetivo definen un gemelo digital que simule un diferencial electrónico (EDS) en el que se ha mejorado la estabilidad, emplearon controladores de velocidad basados en algoritmos de lógica difusa robusta en el EDS para controlar independientemente las ruedas del vehículo eléctrico impulsadas por motores eléctricos de alto rendimiento sin escobillas (BLDC), la información de la posición del rotor de los motores, estimada a partir de sensores de efecto Hall de baja precisión montados en los ejes de los motores, se combina y convierte en un conjunto de señales de conmutación comunes para potenciar el EDS del sistema de tracción del vehículo eléctrico. El gemelo digital propuesto incluye una topología práctica y simple que no requiere infraestructuras auxiliares, lo cual permite reducir las pérdidas y tensiones mecánicas, y puede adaptarse a los vehículos no tripulados de manera más efectiva (Ebadpour et al., 2023)

En el artículo de investigación titulado “A novel method for energy efficiency in internet of thing (iot) through fuzzy logic and genetic-algorithm-based clustering applied for agriculture monitoring” publicado por Shayesteh Tabatabaei el 2023, Sugirieron un nuevo enfoque eficiente en energía basado en algoritmos genéticos y lógica difusa para aumentar la vida útil de una red de monitoreo para un campo agrícola. Para tal propósito colocaron sensores en el terreno y monitorearon las condiciones en un campo agrícola, incluyendo la temperatura, la humedad, la luz, el dióxido de carbono, la humedad del suelo y la acidez. La técnica propuesta utiliza una estructura de agrupación integrada para reducir las distancias entre los

nodos de los sensores a donde se deben transmitir los datos. Los resultados de las simulaciones realizadas con OPNET mostraron que el sistema propuesto tiene un mejor rendimiento que el protocolo IEEE 802.15.4 en términos de uso de energía y vida útil de las redes (Tabatabaei, 2023).

## **2.2 MARCO CONCEPTUAL**

Esta investigación se basa en el cuerpo de conocimientos relacionado con la lógica difusa e Internet of Things, insumos necesarios para la formulación de un modelo matemático que aborde la problemática planteada.

### **2.2.1 TEORÍA DE LA LÓGICA DIFUSA**

Fue creada el año 1965 por el ingeniero y matemático Lofti Zadeh natural de Azerbaiyán quien trabajó en el Departamento de Ingeniería Eléctrica y Ciencias de la Computación de la Universidad de California en Berkeley, Estados Unidos de Norteamérica. Esta rama de las matemáticas en la actualidad tiene mucha aceptación no solo en la matemática teórica sino también en las aplicaciones de ingeniería (Zapata, 2022).

La teoría de lógica difusa se desarrolla principalmente haciendo uso de los conceptos de conjunto difuso y función de membresía, el primer concepto es una generalización de un conjunto clásico (conjunto nítido o crisp set) y el segundo concepto se refiere a la función de membresía que es una generalización de la función característica clásica (Zapata, 2022), (Klir y Yuan, 1995).

Alrededor de estos conceptos se definen operaciones, relaciones, inferencias y otros conceptos que forman parte de un cuerpo de entes matemáticos consistentes que permiten abordar dominios del conocimiento que presentan imprecisiones, vaguedad e incertidumbre (Klir y Yuan, 1995), (Zadeh, 1965).

A diferencia de la lógica clásica, donde las proposiciones solo pueden ser verdaderas o falsas de forma absoluta, en la lógica difusa, las proposiciones pueden tener grados de verdad comprendidos desde completamente falso hasta completamente verdadero. Debido a esta caracterización, con la lógica difusa, se puede abordar el razonamiento con conceptos inexactos o confusos propios de la incertidumbre. Conceptualmente se puede entender que se trata de generalizar la teoría de conjuntos en una teoría de conjuntos difusos, la lógica proposicional en una lógica difusa y el Algebra de Boole en una algebra difusa (Diallo, 2023), (Zapata, 2022), (Zadeh, 1965).

#### **2.2.1.1 INCERTIDUMBRE**

La incertidumbre supone que en una determinada situación no se dispone de información cuantitativa o cualitativa apropiada para describir una variable, para prescribir o predecir de manera determinista y numérica el comportamiento y otras características de un sistema en particular (Chandrasekaran, 2021), (Nithya et al., 2022), (Zadeh, 1965).

En la vida cotidiana como en las actividades comerciales, ingenieriles, médicas, etc., es frecuente utilizar expresiones que presentan incertidumbre, estos hechos no pueden estudiarse con los métodos de la matemática que requieren precisión (Zadeh, 1965), (Chandrasekaran, 2021).

#### **2.2.1.2 VARIABLES LINGÜÍSTICAS**

Son variables cuyos valores no son números como frecuentemente sucede, estas variables asumen valores que son palabras de un lenguaje natural o artificial (Zadeh, 1965), (Chandrasekaran, 2021).

Se prefiere el uso de palabras u oraciones en lugar de números porque estas caracterizaciones son, en general, menos específicas que las numéricas y por consiguiente más adecuadas para la representación formal de la incertidumbre (Zadeh, 1996).

Por ejemplo, la variable numérica “temperatura” puede tomar los siguientes valores numéricos expresados en grados centígrados  $temperatura = \{-10, 25, 40\}$ , en cambio si se le considera como una la variable lingüística, “temperatura” puede tomar como valores las palabras “frio”, “templado” o “caliente”, es decir  $temperatura = \{\text{"frio"}, \text{"templado"}, \text{"caliente"}\}$ .

Las palabras o valores de las variables lingüísticas tienen naturalmente significados imprecisos o borrosos y pueden variar según la interpretación o apreciación de las personas, por tanto, están relacionadas con la incertidumbre y la imprecisión (Zadeh, 1996), (Zadeh, 1975).

### 2.2.1.3 CONJUNTOS DIFUSOS

Según Zadeh (Zadeh, 1965), un conjunto difuso  $A$  definido en el conjunto de referencia  $X$  se expresa como

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

donde  $X$  es el universo del discurso o conjunto referencial,  $\mu_A(x)$  es el valor de su función de membresía definida por:

$$\begin{aligned} \mu_A: X &\rightarrow [0,1] \\ x &\mapsto \mu_A(x) \end{aligned} \quad (2)$$

Observe en (2) que la función de membresía  $\mu_A$  del conjunto difuso  $A$ , establece que a cada elemento  $x \in X$ , se le asigna un número  $\mu_A(x) \in [0,1]$  que significa el grado de pertenencia al conjunto difuso  $A$  (Klir y Yuan, 1995), (Ross, 2010), (Zadeh, 1965).

El valor  $\mu_A(x)$  indica en qué medida el elemento pertenece o se identifica con el conjunto  $A$ , el valor 0 significa que el elemento definitivamente no se identifica o no pertenece al conjunto  $A$ , el valor 1 significa que definitivamente pertenece al conjunto o se identifica totalmente, los valores intermedios comprendidos entre el 0 y el 1 representan grados de pertenencia o porcentajes de identificación con el conjunto  $A$ , esta forma de pertenencia o identificación es la que permite modelar la incertidumbre o ambigüedad (Zadeh, 1965), (Mohammed y Bahjat, 2023).

Por ejemplo, sea  $X = \{0,1,2,3,4,5,6,7,8,9,10\}$  el conjunto universo del discurso, consideremos un conjunto difuso  $A$  definido en el conjunto finito  $X$ , utilizando la notación dada en (1), escribimos  $A$  como:

$$A = \{ (0,0), (1, 0.2), (2, 0.4), (3, 0.6), (4, 0.8), (5, 1), (6, 0.8), (7, 0.6), (8, 0.4), (9, 0.2), (10, 0) \}$$

Cuando no exista lugar a confusión este conjunto se suele expresar también de esta otra forma o notación

$$A = \left\{ \frac{0}{0} + \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.8}{4} + \frac{1}{5} + \frac{0.8}{6} + \frac{0.6}{7} + \frac{0.4}{8} + \frac{0.2}{9} + \frac{0}{10} \right\}$$

En esta notación debe entenderse por ejemplo que  $\frac{0.6}{3}$  está representando a (3,0.6), además no debe confundirse como suma o sumatoria, se trata solamente de una representación formal. En forma resumida la notación se puede escribir como

$$A = \sum_{x \in X} \frac{\mu_A(x)}{x} \tag{3}$$

La representación (3) se puede extender para cuando el universo del discurso  $X$  sea un conjunto infinito expresándose como

$$A = \int_{x \in X} \frac{\mu_A(x)}{x}$$

Donde tampoco debe entenderse como una integral sino como una representación formal de  $A$  (Klir y Yuan, 1995), (Ross, 2010), (Zadeh, 1965).

#### 2.2.1.4 FUNCIONES DE MEMBRESÍA O DE PERTENENCIA

Cabe hacer notar que según la expresión (1), los conjuntos difusos, se definen de manera explícita mediante el establecimiento su correspondiente función de membresía  $\mu_A: X \rightarrow [0,1]$ , se sabe que  $\mu_A(x)$  es el grado de pertenencia de  $X$  en  $A$ , es decir las funciones de membresía proporcionan para cada elemento  $x$  de  $X$  un valor de esta función que está en el intervalo  $[0,1]$ , entendiéndose que el grado 1 es el valor para la máxima pertenencia o identificación con el conjunto difuso  $A$ , mientras que el gado 0 representa la no pertenencia o no identificación absoluta con  $A$  (Ponce-Cruz, 2010), (Klir y Yuan, 1995).

Cabe hacer notar que, si el valor de esta función se restringiera solamente a 0 y 1, es decir,  $\mu_A: X \rightarrow \{0,1\}$ , se tendría un conjunto clásico o no difuso también denominado conjunto nítido o crisp set (Nithya et al., 2022).

Considerando nuevamente el conjunto difuso  $A$ ,

$$A = \left\{ \frac{0}{0} + \frac{0.2}{1} + \frac{0.4}{2} + \frac{0.6}{3} + \frac{0.8}{4} + \frac{1}{5} + \frac{0.8}{6} + \frac{0.6}{7} + \frac{0.4}{8} + \frac{0.2}{9} + \frac{0}{10} \right\}$$

se puede para efectos de graficar su función de membresía usar la tabulación que se muestra en el cuadro 1. La gráfica de la función de membresía se muestra en la figura 1.

#### **Cuadro 1:**

*Valores de una función de membresía para un universo finito*

$x$	0	1	2	3	4	5	6	7	8	9	10
$\mu_A(x)$	0	0.2	0.4	0.6	0.8	1	0.8	0.6	0.4	0.2	0

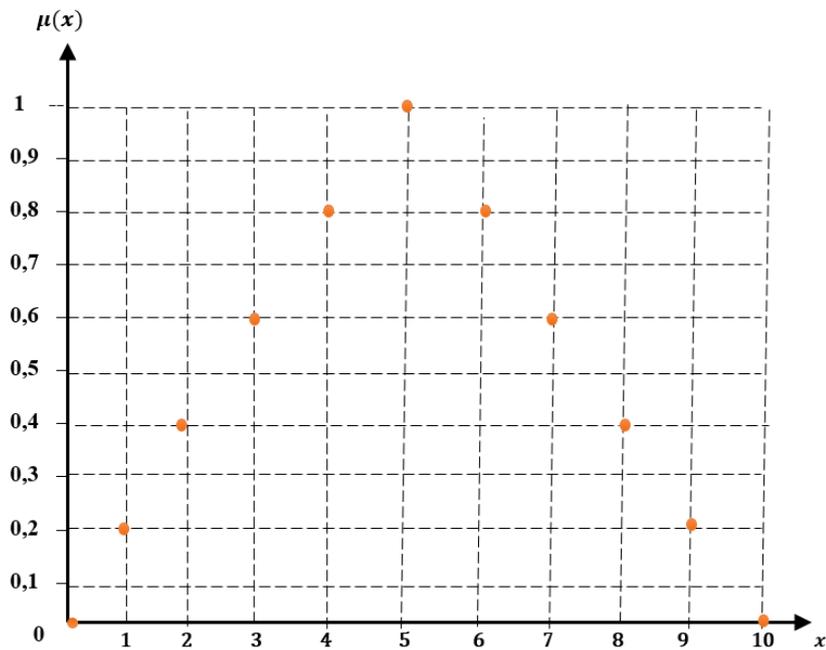


Figura 1:  
*Gráfica de la función de membresía de un conjunto finito*

En la figura 1 se observa que 5 tiene la máxima pertenencia 1, mientras que 0 y 10 tienen la mínima pertenencia 0, este conjunto difuso bien puede representar al valor lingüístico “cercaños al 5 “, ya que, además de los mencionado anteriormente, 4 y 6 tienen una cercanía de 0.8, mientras que 1 y 9 tienen una cercanía de 0.2, en general, de esta manera se podría modelar la variable lingüística “cercaños a un número entero” (Sivanandam et al., 2007).

Cuando el universo del discurso es un conjunto infinito, la gráfica de la función de membresía de un conjunto difuso es una curva continua, de manera formal estas funciones de membresía suelen expresar mediante funciones definidas por partes (Zadeh, 1983).

En la figura 2, se muestra la gráfica de una función de membresía para cuando el universo del discurso  $X$  es infinito.

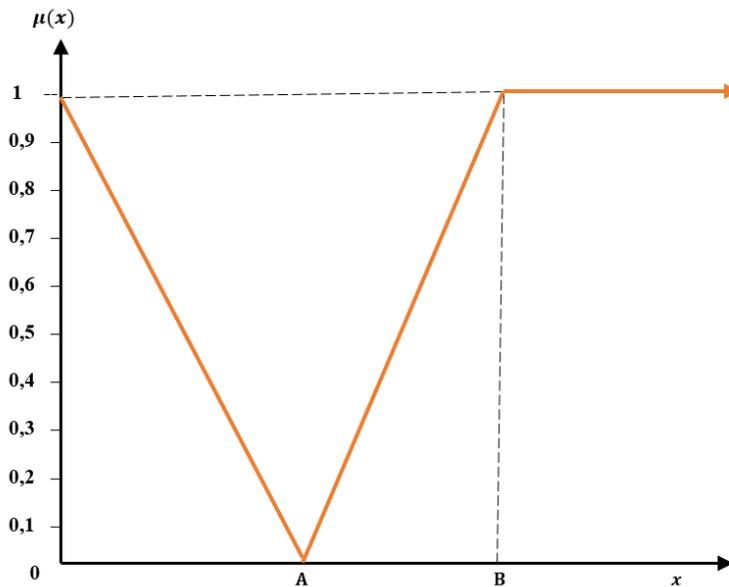


Figura 2:  
*Gráfica de la función de membresía para un conjunto infinito*

En las aplicaciones de la teoría de lógica difusa es frecuente utilizar algunas de las siguientes funciones de membresía descritas a continuación (Klir y Yuan, 1995).

### 2.2.1.5 FUNCIÓN Z (FUNCIÓN HOMBRO DERECHO)

Se define esta función de membresía mediante la siguiente expresión matemática

$$\mu(x) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases}$$

donde  $x, a, b$  son elementos del universo del discurso  $X$ .

Cuando sea necesario se puede escribir también de esta forma

$$\mu_{\text{Función Z}}(x; a, b) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases}$$

otra notación es la siguiente

$$\mu_{\text{Hombro derecho}}(x; a, b) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases}$$

En el caso de usar el software MATLAB, se expresa como

$$\text{linzmf}(x, [a, b]) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases}$$

La figura 3 muestra la gráfica de la función Z.

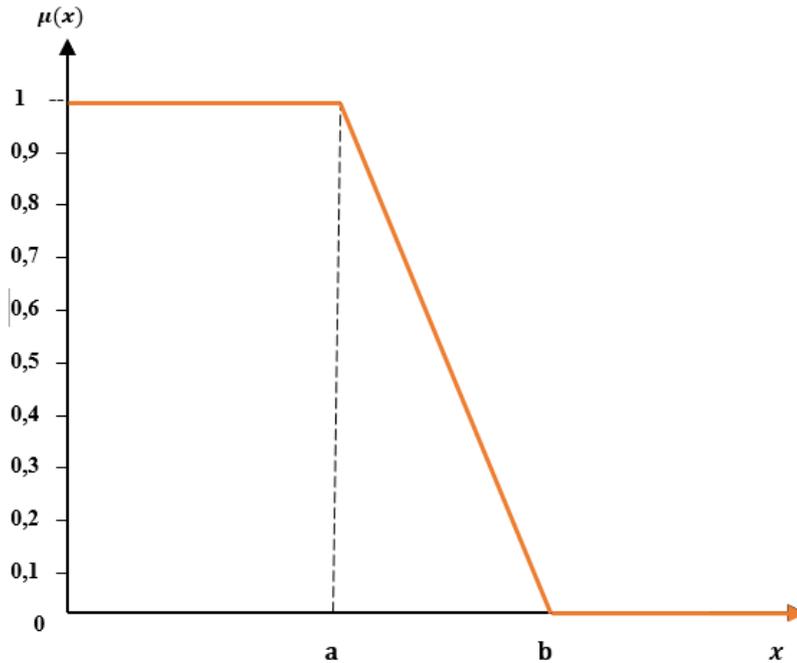


Figura 3:  
Función Z (función hombro derecho)

### 2.2.1.6 FUNCIÓN GAMMA (FUNCIÓN HOMBRO IZQUIERDO)

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & x \geq b \end{cases}$$

donde  $x, a, b$  son elementos del universo del discurso  $X$  (Klir y Yuan, 1995).

Cuando sea necesario se puede escribir también como.

$$\mu_{Gamma}(x; a, b) = \mu_{Hombre\ izquierdo}(x; a, b) = \begin{cases} 1, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases}$$

En el caso de usar el software MATLAB, se expresa como

$$\text{linsmf}(x, [a, b]) = \begin{cases} 1, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 0, & x \geq b \end{cases}$$

La figura 4 muestra la gráfica de la función gamma

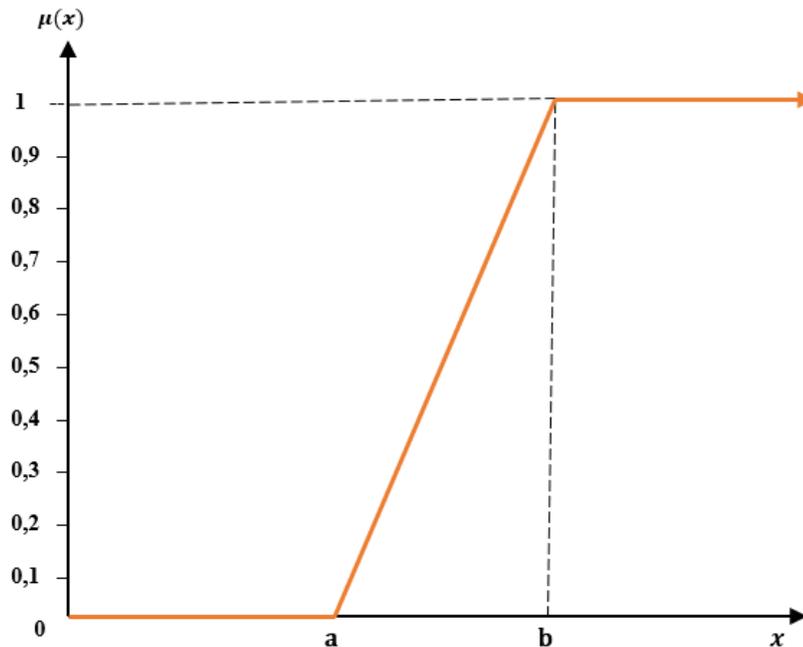


Figura 4:  
*Función gamma (Función hombro izquierdo)*

### 2.2.1.7 FUNCIÓN TRIANGULAR

$$\mu(x) = \begin{cases} 0, & x \leq a \vee x \geq c \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \end{cases}$$

donde  $x, a, b, c$  son elementos del universo del discurso  $X$  (Zadeh, 2005).

Cuando sea necesario se puede escribir también como.

$$\mu_{Triangular}(x; a, b, c) = \begin{cases} 0, & x \leq a \vee x \geq c \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \end{cases}$$

La figura 5 muestra la gráfica de la función triangular



Figura 5:  
*Función triangular*

Existen diversas herramientas software que permiten trabajar con lógica difusa, una de las más usadas es la librería FUZZY de MATLAB, en este entorno de trabajo se puede implementar la función triangular con el código mostrado en la figura 6 y su gráfica en la figura 7.

```
>> x = 0:0.2:20;  
>> y = trimf(x,[6 10 14]);  
>> plot(x,y)  
>> title('Función triangular(x;6,10,14)')  
>> xlabel('x')  
>> ylabel('Grado de membresía')  
>> ylim([-0.05 1.05])  
fx >>
```

Figura 6:  
*Código en MATLAB para graficar una función triangular*

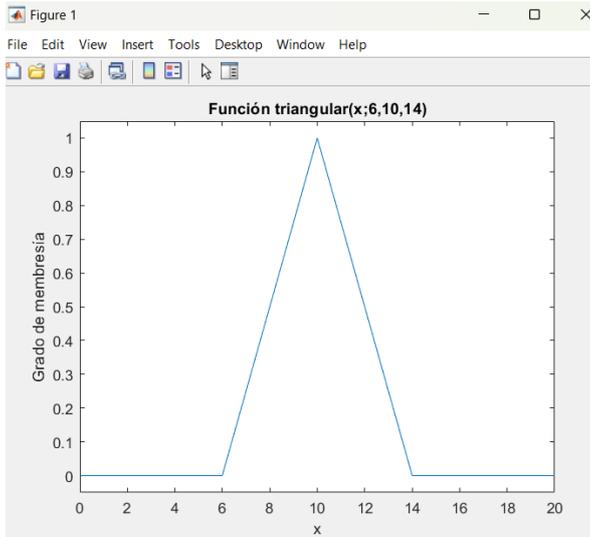


Figura 7:  
Gráfica de la función triangular en MATLAB

### 2.2.1.8 FUNCIÓN TRAPEZOIDAL

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

La figura 8 muestra la gráfica de la función de membresía trapezoidal

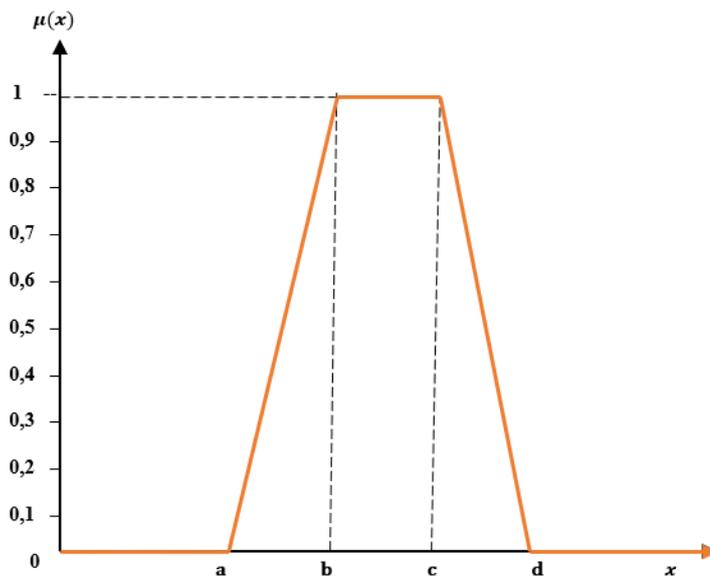


Figura 8:  
Función trapezoidal

### 2.2.1.9 FUNCIÓN SIGMOIDAL

$$\mu_{\text{sigmoidal}}(x) = \begin{cases} 0, & x \leq a \\ 2 \left( \frac{x-a}{c-a} \right)^2, & a \leq x \leq b \\ 1 - 2 \left( \frac{x-a}{c-a} \right)^2, & b \leq x \leq c \\ 1, & x \geq c \end{cases}$$
$$b = \frac{a+c}{2}$$

La figura 9, muestra la gráfica de la función de membresía sigmoidal

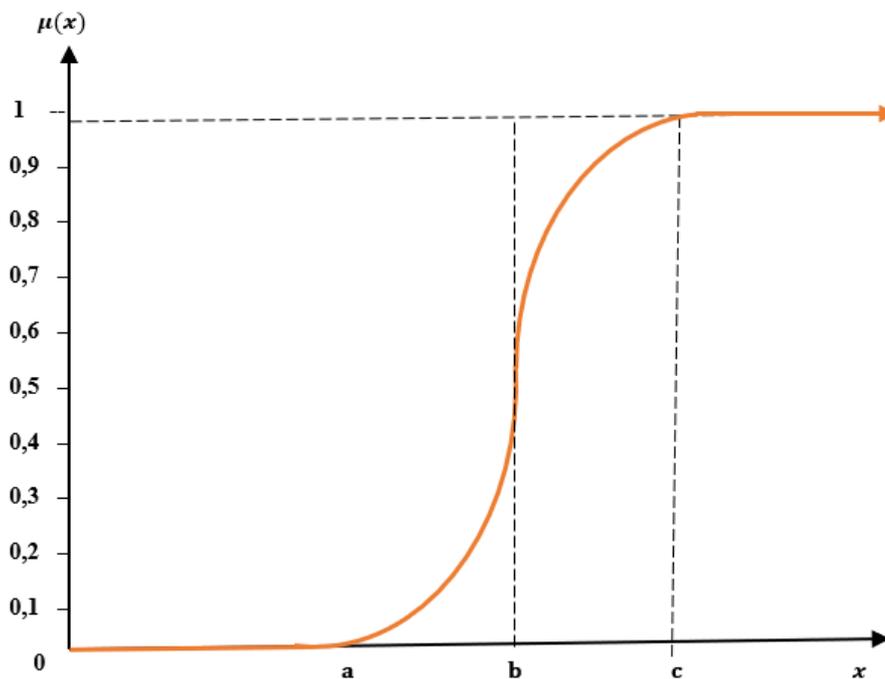


Figura 9:  
*Función sigmoidal*

### 2.2.1.10 FUNCIÓN CAMPANA GENERALIZADA

$$\mu_{\text{Campana generalizada}}(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}}$$

donde “c” establece el centro de la función de membresía, “a” define el ancho de la función de membresía con referencia al centro, donde un valor mayor crea una campana más amplia, “b” define la forma de la curva a cada lado de la meseta central, donde un valor mayor crea una transición más pronunciada (Jantzen, 2007).

Se utiliza el siguiente código en MATLAB para visualizar la gráfica de la función campana generalizada (gbellmf) como se muestra en la figura 10.

```
x = 0:0.1:10;  
y = gbellmf (x, [2 4 6]);  
plot (x, y)  
title ('Función campana generalizada, P=[2 4 6]')  
xlabel ('x')  
ylabel ('Grado de membresía')  
ylim ([-0.05 1.05])
```

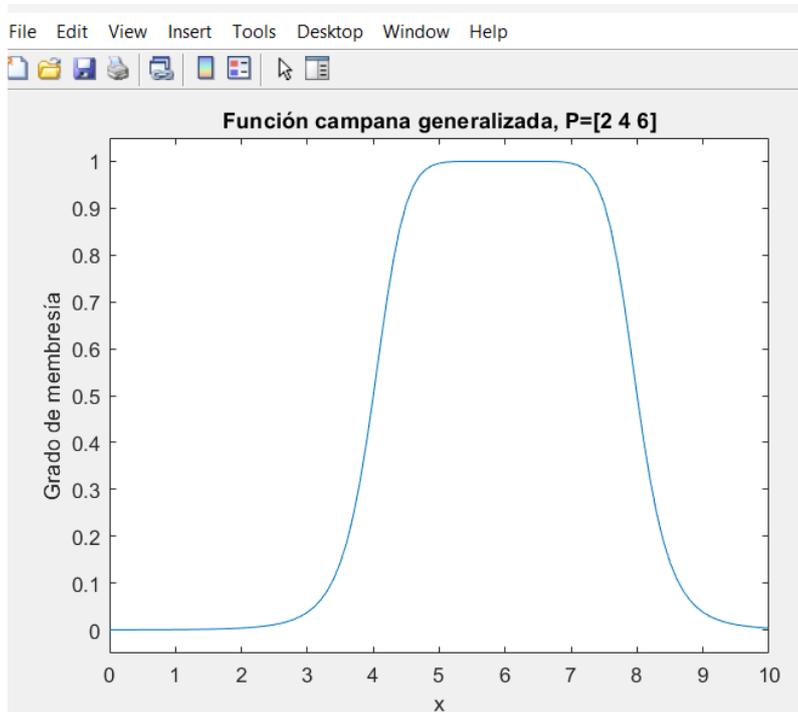


Figura 10:  
*Función campana generalizada*

### 2.2.1.11 FUNCIÓN GAUSSIANA

$$\mu_{Gaussiana}(x; \sigma, c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$$

Donde “c” es el centro de la curva gaussiana,  $\sigma$  es la desviación estándar que controla la anchura de la campana. Con el código MATLAB mostrado a

continuación se genera la gráfica de la función Gaussiana(gaussmf) mostrada en la figura 11.

```
x = 0:0.1:10;  
y = gaussmf (x, [2 5]);  
plot (x, y)  
title ('Función gaussiana(x;2,5)')  
ylabel ('Grado de membresía')  
xlabel ('x')  
ylim ([-0.05 1.05])
```

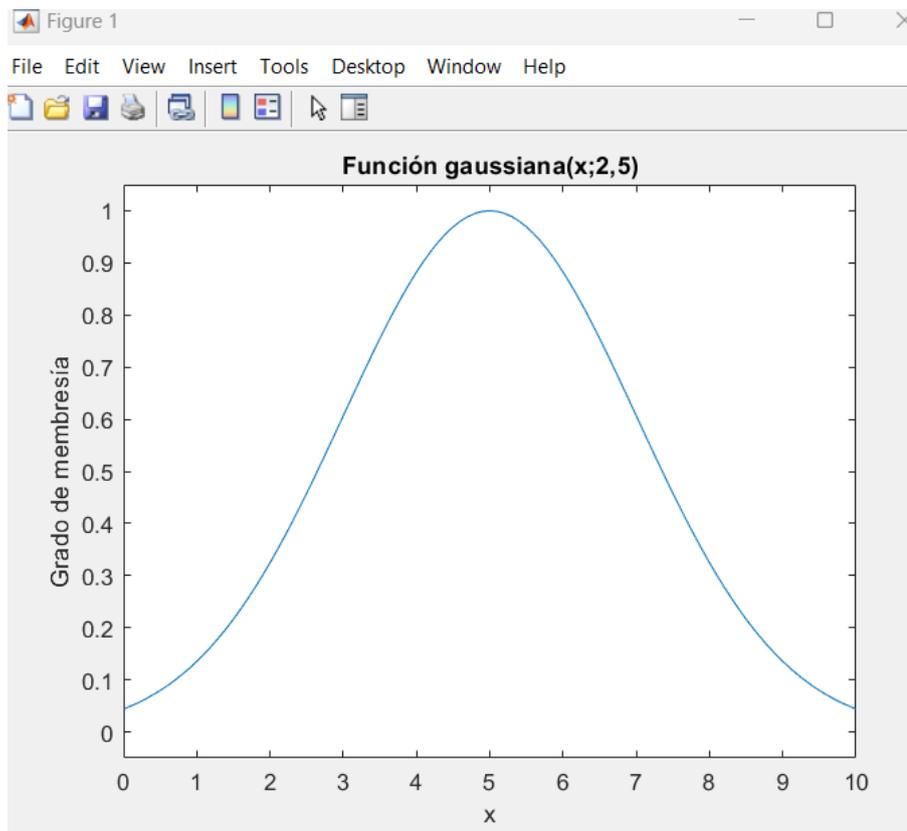


Figura 11:  
*Función gaussiana*

Se considera a continuación un ejemplo que involucra una variable lingüística, sus respectivos conjuntos difusos y las correspondientes funciones de membresía de cada uno de los conjuntos difusos (Nithya et al., 2022).

Sea la variable lingüística.  $\mathcal{L}$  = “Estados físicos del agua”, se considera como sus valores lingüísticos o etiquetas a  $S$  = “Sólido”,  $L$  = “Líquido” y  $V$  = “Vapor”, es decir podemos escribirlo como  $\mathcal{L} = \{S, L, V\}$ , considerando como universo del discurso  $X$ , el conjunto de grados Celsius ( $^{\circ}\text{C}$ ) de temperatura, se construye las siguientes funciones de membresía

$$\mu_{Sólido}(x) = \mu_{Función\ z}(x; 0, 20),$$

$$\mu_{Líquido}(x) = \mu_{Trapezoidal}(x; 0, 20, 80, 100),$$

$$\mu_{Vapor}(x) = \mu_{Función\ gamma}(x; 80, 100),$$

en la figura 12 se muestra las gráficas de las funciones de membresía de sus respectivos conjuntos difusos.

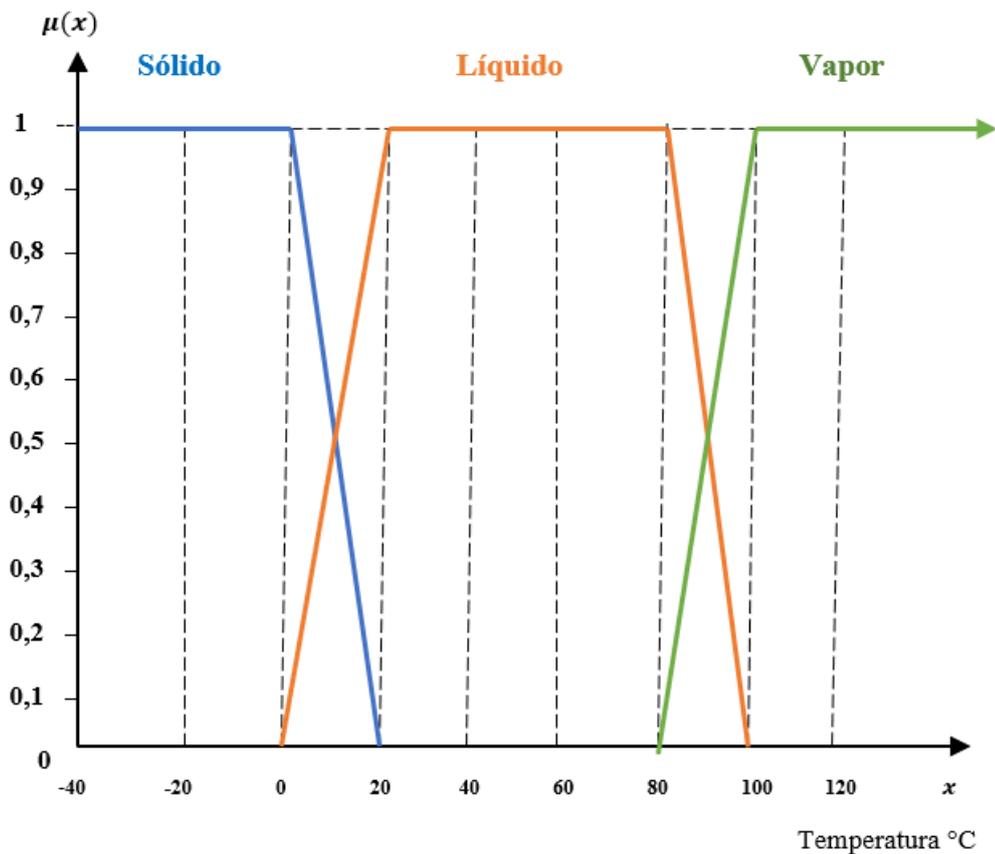


Figura 12:  
Funciones de membresía de los estados físicos del agua

### 2.2.1.12 CÁLCULO DE LA FUNCIÓN DE MEMBRESÍA

Generalmente, la manera de asignar el grado de membresía depende fundamentalmente de la percepción que se tiene respecto a los objetos que se están estudiando, es decir los conjuntos difusos son establecidos de manera subjetiva, algunas veces se utiliza métodos estadísticos o de inteligencia artificial para la construcción de las mismas (Klir y Yuan, 1995), (Kruse y Nürnberger, 1970), (Sivanandam et al., 2007).

El método para calcular las funciones de membresía depende de la aplicación en particular, del modo en que se manifieste la incertidumbre y como se medirá durante el experimento (Sivanandam et al., 2007).

Se describe a continuación algunos de los métodos más usuales:

**Método horizontal.** Consiste en considerar las respuestas de una encuesta a un grupo de  $N$  expertos sobre el tema de estudio (Ross, 2010). La pregunta tiene el formato ¿Puede  $x$  ser considerado compatible con el concepto  $A$ ?, con las únicas respuestas “SI”, “NO”. El cálculo se hace con la siguiente fórmula

$$\mu_A(x) = \frac{\text{Nro de respuestas afirmativas}}{N}.$$

**Método vertical.**

Se escogen varios valores para  $\alpha$  con el objeto de construir  $\alpha - \text{cortes}$ , para estos valores se pregunta ¿Identifique los elementos de  $X$  que pertenecen a  $A$ , con grado no menor que  $\alpha$ ?. Usando el principio de identidad o teorema de representación y a partir de los  $\alpha - \text{cortes}$  se identifica el conjunto difuso  $A$  (Ross, 2010).

El cuadro 2 muestra algunos métodos para obtener las funciones de membresía, sus aplicaciones ventajas y desventajas (Ross, 2010), (Morales y Méndez, 2011).

## Cuadro 2:

### Métodos para determinar las funciones de membresía

MÉTODO	DESCRIPCIÓN	APLICACIÓN	VENTAJAS	DESVENTAJAS
<b>BASADOS EN EL CONOCIMIENTO EXPERTO</b>				
<b>Heurísticos</b>	Se basa en el juicio experto para definir las funciones de membresía.	Cuando los expertos tienen conocimiento profundo.	Simple y directo.	Subjetivo y dependiente de la experiencia del experto.
<b>Entrevistas y Encuestas</b>	Recopilación de opiniones de varios expertos para formar una función de membresía consensuada.	Donde se pueden obtener múltiples opiniones expertas.	Proporciona una visión más completa y equilibrada.	Costoso y lleva tiempo.
<b>BASADOS EN DATOS</b>				
<b>Histograma</b>	Construcción de un histograma de los datos para definir la función de membresía.	Cuando se dispone de grandes conjuntos de datos.	Basado en datos reales, buena representación de la distribución.	Requiere una cantidad significativa de datos.
<b>Análisis de Agrupamiento (Clustering)</b>	Uso de algoritmos de agrupamiento como k-means para definir funciones de membresía basadas en grupos.	Cuando los datos forman grupos naturales.	Maneja grandes conjuntos de datos y encuentra estructuras subyacentes.	Elección del número de grupos puede ser arbitraria.
<b>Regresión Difusa</b>	Uso de técnicas de regresión para ajustar funciones de membresía a los datos.	Cuando hay una relación conocida entre variables.	Proporciona una relación matemática clara.	Requiere supuestos sobre la forma de la relación.
<b>MÉTODOS DE OPTIMIZACIÓN</b>				
<b>Optimización Evolutiva (Algoritmos Genéticos)</b>	Uso de algoritmos genéticos para encontrar funciones de membresía óptimas.	Problemas complejos donde otras técnicas no son viables.	Capaz de encontrar soluciones óptimas en grandes espacios de búsqueda.	Computacionalmente intensivo y puede ser lento.
<b>Optimización por Enjambre de Partículas (PSO)</b>	Uso de la inteligencia de enjambres para optimizar funciones de membresía.	Problemas de optimización multidimensional.	Eficiente y puede converger rápidamente.	Puede quedar atrapado en óptimos locales.
<b>BASADOS EN REDES NEURONALES</b>				
<b>- Redes Neuronales Artificiales (ANN)</b>	Entrenamiento de ANN para aproximar funciones de membresía a partir de conjuntos de datos.	Aplicaciones complejas con relaciones no lineales.	Modela relaciones complejas y no lineales.	Requiere una cantidad significativa de datos y puede ser difícil de interpretar.
<b>MÉTODOS HÍBRIDOS</b>				
<b>Combinación de Heurísticos y Datos</b>	Uso de conocimiento experto y datos empíricos para definir funciones de membresía.	Cuando se dispone de datos y experiencia de expertos.	Proporciona una solución robusta y precisa.	Complejo y lleva más tiempo implementar.

### 2.2.1.13 PRINCIPIO DE EXTENSIÓN

Este principio es de gran importancia en la lógica difusa ya que mediante él podemos formalmente extender funciones definidas en conjuntos nítidos a funciones definidas sobre conjuntos difusos, dicho de otro modo, es una manera de extender nociones y operaciones matemáticas del mundo nítido al mundo difuso (Klir y Yuan, 1995), (Ross, 2010).

Sea  $f: X \rightarrow Y$  una función definida de un conjunto nítido  $X$  a otro conjunto nítido  $Y$ , es decir, si  $x \in X \mapsto y = f(x) \in Y$

Se desea obtener una función difusa  $f^{\sim} : A \rightarrow B$  que aplica de un conjunto difuso  $A$  definido en  $X$  a otro conjunto difuso  $B$  definido en  $Y$ , es decir que aplica

$$(x, \mu_A(x)) \in A \mapsto f^{\sim}(x, \mu_A(x)) = (y, \mu_B(y)) \in B$$

Para ilustrar el principio de extensión, consideremos por ejemplo como  $A$  un conjunto difuso definido en  $X$  tal que  $A = \{(x, \mu_A(x)), x \in X, \mu_A(x) \in [0,1]\}$ , se pretende extender la función nítida  $f$ , para lo cual consideremos la función inversa de  $f$  definida como  $f^{-1}(y) = \{x \in X, y = f(x)\}$ , si tomamos como  $B$  un conjunto difuso definido en  $Y$  tal que  $B = \{(y, \mu_B(y)), y \in Y, \mu_B(y) \in [0,1]\}$  de tal manera que cumple que

$$\mu_B(y) = \sup_{x \in f^{-1}(y)} \{\mu_A(x)\}$$

De esta forma logra extender la función nítida  $f$  a la función difusa  $f^{\sim}$ , es decir

$$f^{\sim}(x, \mu_A(x)) = (y, \mu_B(y)), \text{ tal que } \mu_B(y) = \sup_{x \in f^{-1}(y)} \{\mu_A(x)\}$$

Por ejemplo

Sea  $f: \mathbb{R} \rightarrow \mathbb{R}$  definida por  $y = f(x) = 2x$ , consideremos el conjunto difuso  $A$  definido en  $\mathbb{R}$  cuya función de membresía es

$$\mu_A(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } 0 \leq x \leq 1 \\ 1 & \text{si } x > 1 \end{cases}$$

Usamos el principio de extensión para encontrar la función  $f^{-1}: A \rightarrow B$

Primero calculamos  $f^{-1}$ . Como  $f(x) = y = 2x$  entonces despejando tenemos

$x = \frac{y}{2}$ , es decir se ha obtenido que  $f^{-1}(y) = \frac{y}{2}$ , determinamos la función de

membresía del conjunto  $B$  a partir de considerar la inversa de la función de

membresía del conjunto  $A$ , en efecto:

Si  $\mu_A(x) = 0$  ,  $x < 0$  entonces reemplazando  $\frac{y}{2} < 0$  luego  $y < 0$  ,  $\mu_B(y) = 0$

Si  $\mu_A(x) = x$  ,  $0 \leq x \leq 1$  entonces reemplazando  $0 \leq \frac{y}{2} < 1$  luego  $0 \leq y < 2$ ,

$$\mu_B(y) = \frac{y}{2}$$

Si  $\mu_A(x) = 1$  ,  $x > 1$  entonces reemplazando  $\frac{y}{2} > 1$  luego  $y > 2$  ,  $\mu_B(y) = \frac{y}{2}$

Luego el conjunto difuso  $B$  tendría como función de membresía

$$\mu_B(y) = \begin{cases} 0 & \text{si } y < 0 \\ \frac{y}{2} & \text{si } 0 \leq y \leq 2 \\ 1 & \text{si } y > 2 \end{cases}$$

En este caso como  $y = 2x$  es una función inyectiva,  $f^{-1}(y) = \frac{y}{2}$  es también

inyectiva por tanto el cálculo del supremo se reduce a un solo valor (Klir y Yuan,

1995), (Ross, 2010).

#### 2.2.1.14 OPERACIONES CON CONJUNTOS DIFUSOS

Sean  $A, B$  dos conjuntos difusos definidos en  $X$ , se establecen las siguientes

operaciones con conjuntos difusos (Zadeh, 1965), (Klir y Yuan, 1995), (Ross,

2010).

**Inclusión de conjuntos difusos.**

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x), \quad \forall x \in X$$

**Igualdad de conjuntos difusos.**

$$A = B \Leftrightarrow \mu_A(x) = \mu_B(x), \quad \forall x \in X$$

**Unión de conjuntos difusos.**

$$A \cup B = \{(x, \mu_{A \cup B}(x)), \mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad \forall x \in X\}$$

**Intersección de conjuntos difusos.**

$$A \cap B = \{(x, \mu_{A \cap B}(x)), \mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \quad \forall x \in X\}$$

**Complemento de un conjunto difuso.**

$$\neg A = \{(x, \mu_{\neg A}(x)), \mu_{\neg A}(x) = 1 - \mu_A(x), \quad \forall x \in X\}$$

Para ilustrar las operaciones entre conjuntos difusos de universo  $X$  finito, consideremos el siguiente ejemplo.

Sean  $A, B$  dos conjuntos difusos definidos en  $X = \{2,3,4,5,6\}$

$$A = \left\{ \frac{1}{2} + \frac{0.5}{3} + \frac{0.6}{4} + \frac{0.2}{5} + \frac{0.6}{6} \right\}$$

$$B = \left\{ \frac{0.5}{2} + \frac{0.8}{3} + \frac{0.4}{4} + \frac{0.7}{5} + \frac{0.3}{6} \right\}$$

Se obtienen luego los siguientes resultados

$$\neg A = \left\{ \frac{0}{2} + \frac{0.5}{3} + \frac{0.4}{4} + \frac{0.8}{5} + \frac{0.4}{6} \right\}$$

$$\neg B = \left\{ \frac{0.5}{2} + \frac{0.2}{3} + \frac{0.6}{4} + \frac{0.3}{5} + \frac{0.7}{6} \right\}$$

$$A \cup B = \left\{ \frac{1}{2} + \frac{0.8}{3} + \frac{0.6}{4} + \frac{0.7}{5} + \frac{0.6}{6} \right\}$$

$$A \cap B = \left\{ \frac{0.5}{2} + \frac{0.5}{3} + \frac{0.4}{4} + \frac{0.2}{5} + \frac{0.3}{6} \right\}$$

Para el caso en que el conjunto universal  $X$  sea infinito con  $x \in X$ , las operaciones se ilustran a través de sus funciones de membresía. En la figura 13 se muestra el conjunto difuso  $B$  y su complemento  $\neg B$  (Zadeh, 1965).

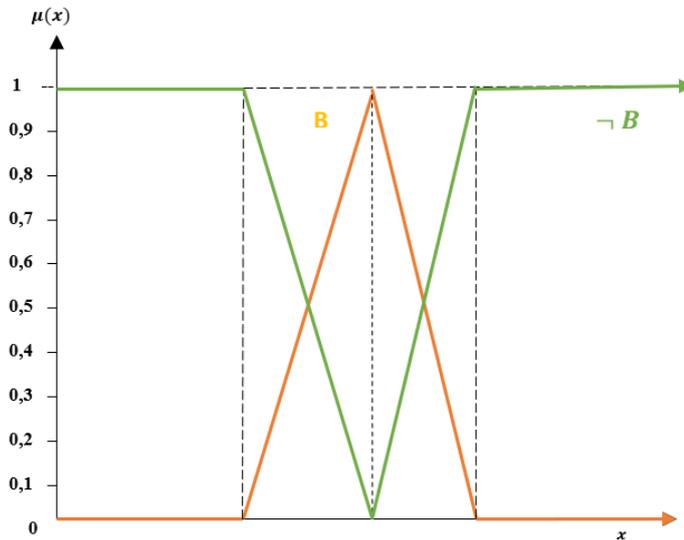


Figura 13:  
*Complemento de un conjunto difuso*

En la figura 14, se presentan las funciones de membresía de dos conjuntos difusos definidos sobre un conjunto universo del discurso infinito. Tomando como referencia las gráficas de las funciones de membresía de éstos dos conjuntos, en la figura 15 se muestra la gráfica correspondiente a la unión de estos dos conjuntos considerando el máximo entre los valores de la funciones de membresía de  $A$  y  $B$ , de igual forma en la figura 16 se muestra el resultado de la intersección de estos conjuntos considerando el mínimo entre los valores de las funciones de membresía de  $A$  y  $B$  respectivamente. El cálculo de operaciones de conjuntos difusos definidos sobre conjuntos infinitos, resulta preponderante en la realización de aplicaciones de ingeniería (Mohammed y Bahjat, 2023).

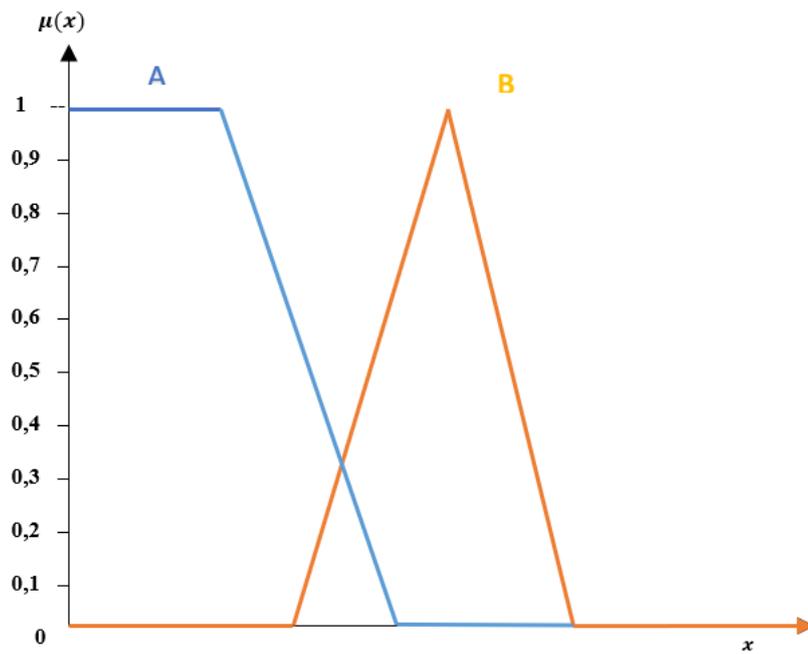


Figura 14:  
Conjuntos difusos para ejemplificar operaciones gráficas

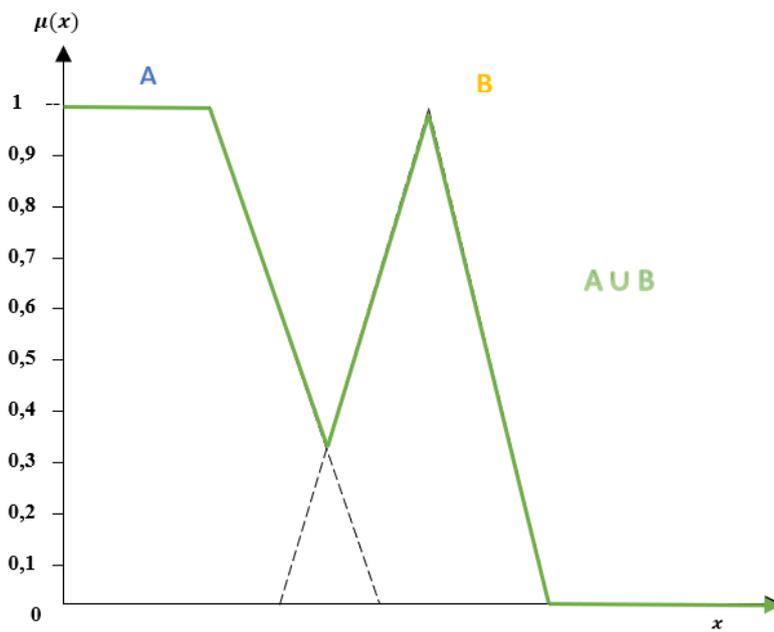


Figura 15:  
Unión de dos conjuntos difusos

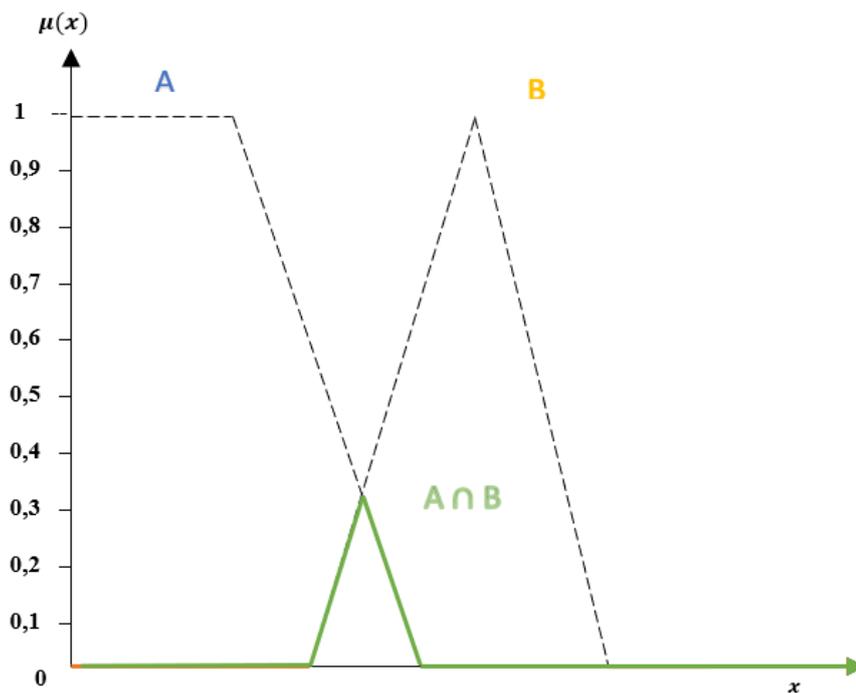


Figura 16:  
*Intersección de conjuntos difusos*

### 2.2.1.15 ÁLGEBRA DE BOOLE

Sea  $X$  un conjunto de referencia,  $\mathcal{P}(X)$  es el conjunto de partes de  $X$ , es decir el conjunto formado por todos aquellos subconjuntos nítidos del universo del discurso  $X$ , consideremos los símbolos  $\cup$ ,  $\cap$  y  $\neg$  para representar la unión, la intersección y el complemento de conjuntos nítidos respectivamente, la estructura algebraica  $(\mathcal{P}(X), \cup, \cap, \neg)$  se denomina Álgebra de Boole debido a que cumple las propiedades listas en el cuadro 3.

En el Álgebra de Boole, la propiedad 7 del tercio excluido  $A \cup \neg A = X$  garantiza que un elemento  $x \in X$  está o no está en el conjunto  $A$ . Mientras que su respectiva propiedad dual de la no negación  $A \cap \neg A = \emptyset$  garantiza que una proposición no puede ser al mismo tiempo verdadero y falso (Zadeh, 1965), (Klir y yuan, 1995).

**Cuadro 3:***Operaciones del Álgebra de Boole*

N°	PROPIEDAD	PARA LA UNION	PARA LA INTERSECCION
1	Idempotencia	$A \cup A = A$	$A \cap A = A$
2	Conmutativa	$A \cup B = B \cup A$	$A \cap B = B \cap A$
3	Asociativa	$A \cup (B \cap C) = (A \cup B) \cap C$	$A \cap (B \cup C) = (A \cap B) \cup C$
4	Absorción	$A \cup (A \cap B) = A$	$A \cap (A \cup B) = A$
5	Distributiva	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
6	Identidad	$A \cup \emptyset = A$	$A \cap X = A$
7	Tercio excluido (No contradicción)	$A \cup \neg A = X$	$A \cap \neg A = \emptyset$
8	De Morgan	$\neg(A \cup B) = \neg A \cap \neg B$	$\neg(A \cap B) = \neg A \cup \neg B$
9	Involución	$\neg\neg A = A$	$\neg\neg A = A$

Usando el principio de extensión, se puede pensar en extender un Álgebra de Boole de conjuntos nítidos usando la conceptualización y formalización de los conjuntos difusos (Zadeh, 1975), (Ross, 2010), (Kruse y Nürnberger, 1970).

Para definir esta estructura algebraica difusa consideremos en vez del conjunto de partes  $\mathcal{P}(X)$  el conjunto de las funciones de membresía de todos los subconjuntos de  $X$  que denotaremos como  $\{0, 1\}^X = \{\mu: X \rightarrow \{0, 1\}\}$ , considerando las funciones de membresía  $\mu: X \rightarrow \{0, 1\}$ . Para las operaciones de la unión, intersección y complemento de conjunto difusos consideremos el máximo, mínimo y complemento a 1 respectivamente, luego, se puede afirmar que la estructura algebraica  $(\{0, 1\}^X, \text{máx}, \text{mím}, 1 - x)$ ,  $x \in X$  es una Álgebra de Boole en el mundo de los conjuntos difusos (Zadeh, 1975), (Ross, 2010).

### 2.2.1.16 ÁLGEBRA DIFUSA

Las estructuras algebraicas  $(\mathcal{P}(X), \cup, \cap, \neg)$ ,  $(\{0, 1\}^X, \text{máx}, \text{mím}, 1 - x)$ ,  $x \in X$  son conceptualmente isomorfas con las funciones de membresía para las operaciones de unión, intersección y negación definidas como (Klir y Yuan, 1995):

$$\mu_{A \cup B}(x) = \text{máx} \{ \mu_A(x), \mu_B(x) \}$$

$$\mu_{A \cap B}(x) = \text{mín} \{ \mu_A(x), \mu_B(x) \}$$

$$u_{\neg A}(x) = 1 - u_A(x),$$

Esta extensión conceptual, es la más usual pero no es la única que soporta el mundo difuso como se hará notar más adelante. En el cuadro 4 se muestran algunas de las propiedades del Álgebra de Boole expresadas en términos de estas funciones de membresía. En esta notación difusa la propiedad del tercio excluido  $A \cup \neg A = X$  y de la no negación  $A \cap \neg A = \emptyset$  no se cumplen, esta es una propiedad importante de los conjuntos difusos. Es decir, ya no se garantiza que una proposición no puede ser al mismo tiempo verdadero y falso y que el todo sea igual a la unión de sus partes, pues ahora puede tener un grado de membresía de ambos valores, esta propiedad garantiza poder manejar las incertidumbres y estar de acorde con los principios del pensamiento sistémico (Zadeh, 1965), (Ross, 2010).

**Cuadro 4:**  
*Propiedades del Algebra difusa*

Notación nítida	Notación difusa
$A \cup A = A$	$\text{máx}\{u_A(x), u_A(x)\} = u_A(x)$
$A \cup B = B \cup A$	$\text{máx}\{u_A(x), u_B(x)\} = \text{máx}\{u_B(x), u_A(x)\}$
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$\text{máx}\{u_A(x), \text{máx}\{u_B(x), u_C(x)\}\}$ $= \text{máx}\{\text{máx}\{u_A(x), u_B(x)\}, u_C(x)\}$
$A \cup (A \cap B) = A$	$\text{máx}\{u_A(x), \text{mín}\{u_A(x), u_B(x)\}\} = u_A(x)$
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	$\text{máx}\{u_A(x), \text{mín}\{u_B(x), u_C(x)\}\}$ $= \text{mín}\{\text{máx}\{u_A(x), u_B(x)\}, \text{máx}\{u_A(x), u_C(x)\}\}$
$\neg(A \cup B) = \neg A \cap \neg B$	$1 - \text{máx}\{u_A(x), u_B(x)\} = \text{mín}\{1 - u_A(x), 1 - u_B(x)\}$
$\neg\neg A = A$	$1 - (1 - u_A(x)) = u_A(x)$

Se puede pensar el Álgebra difusa como una generalización del Álgebra de Boole o en su defecto ver desde la perspectiva que un Álgebra de Boole es un caso particular de un Álgebra difusa considerando operaciones como la unión, la intersección y la negación en un contexto más general (Zadeh, 1965), (Ross, 2010). Como se mencionó anteriormente en el Álgebra difusa existe la posibilidad de extender de diversas formas las operaciones que definen la estructura algebraica de una Álgebra de Boole. En esta perspectiva se utilizará la extensión de la negación o complemento nítido a la negación fuerte difusa, la intersección nítida a la norma triangular o T-norma, la unión nítida a la conorma triangular o S-conorma que se explicarán líneas abajo (Chandrasekaran, 2021), (Zadeh, 1965), (Klir y Yuan, 1995).

### 2.2.1.17 NEGACIÓN FUERTE

Consideremos la operación negación fuerte como la siguiente función

$$\begin{aligned} n: [0,1] &\rightarrow [0,1] \\ x &\mapsto n(x) \end{aligned}$$

que cumple que las siguientes propiedades (Zadeh, 1965), (Ross, 2010).

$$n(0) = 1$$

$$n(1) = 0$$

$$n(n(x)) = x, \quad \forall x \in [0,1]$$

$$\forall x, y \in [0,1], \quad x \leq y \rightarrow n(x) \geq n(y)$$

Existen diversos ejemplos de estas negaciones fuertes, pero la más usual es

$$n(x) = 1 - x$$

Se puede comprobar que las siguientes funciones también son negaciones fuertes

$$n(x) = \sqrt{1 - x^2}$$

$$n(x) = \frac{1 - t}{1 + \lambda t}, \lambda \geq -1$$

En efecto, comprobaremos que

$$n(x) = \sqrt{1 - x^2}$$

Es una negación fuerte que cumple

$$n(0) = \sqrt{1 - 0^2} = 1$$

$$n(1) = \sqrt{1 - 1^2} = 0$$

$$n(n(x)) = \sqrt{1 - n(x)^2} = \sqrt{1 - (\sqrt{1 - x^2})^2} = \sqrt{1 - 1 + x^2} = x, \quad \forall x \in [0,1]$$

$$\forall x, y \in [0,1], x \leq y \rightarrow x^2 \leq y^2 \rightarrow -x^2 \geq -y^2 \rightarrow 1 - x^2 \geq 1 - y^2 \rightarrow$$

$$\sqrt{1 - x^2} \geq \sqrt{1 - y^2} \rightarrow n(x) \geq n(y)$$

Luego el complemento difuso, se puede expresar mediante una negación fuerte  $n$  de tal forma que en términos de funciones de membresía se expresaría con la siguiente expresión (Zadeh, 1965), (Klir y Yuan, 1995):

$$\mu_{\neg A}(x) = n(\mu_A(x))$$

### 2.2.1.18 NORMA TRIANGULAR

Para extender la operación de intersección de conjuntos nítidos a conjuntos difusos, definamos la función  $T: [0,1]^2 \rightarrow [0,1]$ , que denominaremos Norma triangular o T-Norma que debe satisfacer las siguientes propiedades (Zadeh, 1965), (Klir y Yuan, 1995).

$$\text{Conmutatividad: } T(x, y) = T(y, x), \quad \forall x, y \in [0,1]$$

$$\text{Monotonía: } x \leq z, y \leq w \rightarrow T(x, y) = T(z, w), \quad \forall x, y, z, w \in [0,1]$$

$$\text{Asociatividad: } T(x, T(y, z)) = T(T(x, y), z) \quad \forall x, y, z, w \in [0,1]$$

$$\text{Elemento identidad: } T(x, 1) = x, \quad \forall x, y \in [0,1]$$

El número 1 actúa como elemento identidad.

Si tomamos como ejemplo de T-norma a la definida por

$$T(x, y) = \min\{x, y\}$$

Se comprueba que efectivamente cumple las cuatro propiedades anteriores.

Esta T-norma se puede utilizar definir la intersección de conjuntos difusos como:

$$\mu_{A \cap_T B}(x) = T(\mu_A(x), \mu_B(x)) = \text{mín}\{\mu_A(x), \mu_B(x)\}$$

La T-norma definida con el mínimo no es la única manera de expresar o generalizar la intersección de conjuntos difusos, existen otras T-normas que también cumplen las cuatro propiedades de la definición, podrá observar en el cuadro 5 que se listan otras de estas T-normas definidas por diferentes autores (Zadeh, 1965), (Klir y Yuan, 1995).

### 2.2.1.19 CONORMA TRIANGULAR

En el propósito de extender la unión de conjuntos nítidos a conjuntos difusos, definamos la función  $S: [0,1]^2 \rightarrow [0,1]$ , que denominaremos Conorma triangular o T-conorma o S-norma que satisface las siguientes propiedades (Zadeh, 1965), (Klir y Yuan, 1995).

Conmutatividad:  $S(x, y) = S(y, x), \quad \forall x, y \in [0,1]$

Monotonía:  $x \leq z, y \leq w \rightarrow S(x, y) = S(z, w), \quad \forall x, y, z, w \in [0,1]$

Asociatividad:  $S(x, S(y, z)) = S(S(x, y), z) \quad \forall x, y, z, w \in [0,1]$

Elemento identidad:  $S(x, 0) = x, \quad \forall x, y \in [0,1]$

El número 0 actúa como elemento identidad

Si tomamos como ejemplo de S-norma a la definida por

$$S(x, y) = \text{máx}\{x, y\}$$

Se comprueba que efectivamente cumple las cuatro propiedades anteriores.

Esta S-norma se puede utilizar definir la unión de conjuntos difusos como:

$$S\mu_{A \cup_S B}(x) = S(\mu_A(x), \mu_B(x)) = \text{máx}\{\mu_A(x), \mu_B(x)\}$$

La S-norma se utiliza para representar la disyunción en lógica difusa y la unión en conjuntos difusos (Zadeh, 1965), (Klir y Yuan, 1995).

Debe notarse que la S-norma definida con el máximo no es la única manera de expresar o generalizar la unión de conjuntos difusos, existen otras S-normas que también cumplen con las cuatro propiedades de la definición, en el cuadro 6 se listan otras de estas S-normas En el cuadro 6, se presentan otras S-normas, definidas por diferentes autores (Zadeh, 1965), (Klir y Yuan, 1995).

Es importante tener en cuenta que se puede probar que para cualquier T-Norma  $T(x, y)$  de las mostradas, se cumple que  $T(x, y) \leq \text{mín}(x, y)$ , es decir, el mínimo es la mayor de las T-normas y de manera similar se puede probar que para una S-norma  $S(x, y)$ ,  $S(x, y) \geq \text{máx}(x, y)$ ,  $\forall x, y \in [0,1]$ , es decir el máximo es la menor de las S-normas (Zadeh, 1965), (Klir y Yuan, 1995).

**Cuadro 5.**  
*Otras T-normas*

T-NORMA	DEFINICIÓN	OBSERVAC.
1 Dombi	$T(x, y) = \frac{1}{1 + \left[ \left( \frac{1}{x} - 1 \right)^\lambda + \left( \frac{1}{y} - 1 \right)^\lambda \right]^{\frac{1}{\lambda}}}$	$\lambda \in \langle 0, \infty \rangle$
2 Dubois-Prade	$T(x, y) = \frac{xy}{\text{máx}\{x, y, \alpha\}}$	$\alpha \in [0,1]$
3 Yager	$T(x, y) = 1 - \text{mín}\{1, ((1-x)^w + (1-y)^w)^{\frac{1}{w}}\}$	$w \in \langle 0, \infty \rangle$
4 Producto drástico	$T(x, y) = \begin{cases} x, & \text{si } y = 1 \\ y, & \text{si } x = 1 \\ 0, & \text{en otro caso} \end{cases}$	
5 Producto de Einstein	$T(x, y) = \frac{xy}{1 + (1-x)(1-y)}$	
6 Producto algebraico	$T(x, y) = xy$	
7 Mínimo	$T(x, y) = \text{mín}\{x, y\}$	
8 Frank	$T(x, y) = \log_p \left( 1 + \frac{(p^x - 1)((p^y - 1))}{p - 1} \right)$	$p > 0, p \neq 1$
9 Lukasiewicz	$T(x, y) = \text{máx}\{0, x + y - 1\}$	
10 Mínimo nilpotente	$T(x, y) = \begin{cases} \text{mín}\{x, y\}, & \text{si } x + y > 1 \\ 0, & \text{en otro caso} \end{cases}$	
11 Producto de Hamacher	$T(x, y) = \begin{cases} 0, & \text{si } x = y = 0 \\ \frac{xy}{x + y - xy}, & \text{en otro caso} \end{cases}$	

**Cuadro 6.**

*Otras S-normas*

S-NORMA	DEFINICIÓN	OBSERVAC.
1 Dombi	$S(x, y) = \frac{1}{1 + \left[ \left( \frac{1}{x} - 1 \right)^{-\lambda} + \left( \frac{1}{y} - 1 \right)^{-\lambda} \right]^{\frac{1}{\lambda}}}$	$\lambda \in \langle 0, \infty \rangle$
2 Dubois-Prade	$S(x, y) = \frac{x + y - xy - \min\{x, y, 1 - \alpha\}}{\max\{1 - x, 1 - y, \alpha\}}$	$\alpha \in [0, 1]$
3 Yager	$S(x, y) = \min\{1, ((x)^w + (y)^w)^{\frac{1}{w}}\}$	$w \in \langle 0, \infty \rangle$
4 Suma drástica	$S(x, y) = \begin{cases} x, & \text{si } y = 0 \\ y, & \text{si } x = 0 \\ 1, & \text{en otro caso} \end{cases}$	
5 Suma de Einstein	$S(x, y) = \frac{x + y}{1 + xy}$	
6 Suma algebraica	$S(x, y) = x + y - xy$	
7 Máximo	$S(x, y) = \max\{x, y\}$	
8 Frank	$S(x, y) = \log_p \left( 1 + \frac{(p^{1-x} - 1)((p^{1-y} - 1))}{p - 1} \right)$	$p > 0, p \neq 1$
9 Lukasiewicz	$S(x, y) = \min\{x + y, 1\}$	
10 Máximo nilpotente	$S(x, y) = \begin{cases} \max\{x, y\}, & \text{si } x + y < 1 \\ 1, & \text{en otro caso} \end{cases}$	

Podemos observar la flexibilidad para definir una teoría de conjuntos difusos, esto permite adaptarla a múltiples contextos en las aplicaciones. Formalmente basta crear la estructura algebraica  $([0,1]^X, T, S, n)$ , donde  $[0,1]^X = \{\mu | \mu : X \rightarrow [0,1]\}$  es el conjunto de funciones de membresía entre  $X$  y  $[0,1]$ ,  $T$  es una T-Norma,  $S$  es una S-norma y  $n$  es una negación fuerte (Zadeh, 1965), (Klir y Yuan, 1995).

No existe por tanto una única teoría de conjuntos difusos, sino múltiples teorías, dependiendo de la elección de los elementos de esta estructura algebraica, esto permite sin dudas poder formalizar la incertidumbre con teoría de conjuntos difusos adecuándolo al problema estudiado en las aplicaciones y aquí está la razón de

porque se han creado muchas patentes al usar lógica difusa en proyectos de investigación (Mohammed y Bahjat, 2023).

### 2.2.1.20 RELACIONES DIFUSAS

Las relaciones difusas constituyen una extensión del concepto de relaciones clásicas para poder manejar la incertidumbre y la imprecisión en la asociación entre elementos y de esta manera poder abordar aplicaciones donde las relaciones no son claramente definidas, proporcionando una forma flexible y robusta de modelar y analizar interacciones complejas (Mohammed y Bahjat, 2023), (Nithya et al., 2022). Una relación difusa es un subconjunto difuso del producto cartesiano de dos conjuntos difusos (Zadeh, 1969), (Zadeh, 1971).

Formalmente, sean dos conjuntos difusos  $U, V$ , definidos en los conjuntos de referencia  $X, Y$ , se define la relación  $R \subseteq U \times V$  estableciendo su función de membresía como  $u_R: U \times V \rightarrow [0,1]$  de forma tal que  $\forall (u, v) \in U \times V$  se el asigna  $u_R(u, v) \in [0,1]$  que significa o determina el grado de relación que existe entre  $u$  y  $v$  (Zadeh, 1969), (Zadeh, 1971), (Kahraman, 2008).

Es decir, si tenemos dos conjuntos difusos  $A$  y  $B$  definidos como:

$$A = \{(a_1, \mu_A(a_1)), (a_2, \mu_A(a_2)), \dots, (a_n, \mu_A(a_n))\}$$

$$B = \{(b_1, \mu_B(b_1)), (b_2, \mu_B(b_2)), \dots, (b_n, \mu_B(b_n))\}$$

Una relación  $R$  de  $A$  a  $B$  se define como:

$$R = \{(a_i, b_j), \mu_R(a_i, b_j)\}, a_i \in A, b_j \in B$$

Donde  $\mu_R(a_i, b_j)$  es el grado de pertenencia de cada par  $(a_i, b_j)$  en la relación  $R$ .

Mientras que, en una relación clásica, los elementos  $(a_i, b_j)$  están claramente relacionados o no, en una relación difusa permite grados de pertenencia a la relación

$R$ , expresando cuán fuertemente un elemento está relacionado con otro (Klir y Yuan, 1995), (Mohammed y Bahjat, 2023).

Como ejemplo consideremos

$$U = \{(u_i, \mu_U(u_i))\}, \quad i = 1,2,3,4$$

$$V = \{(v_j, \mu_V(v_j))\}, \quad j = 1,2,3,4,5$$

Cuando los conjuntos son finitos, se acostumbra definir la relación  $R$  mediante una matriz donde  $\mu_R(u_i, v_j)$  representa el grado de relación que existe entre el elemento  $u_i$  de la fila  $i$  con el elemento  $v_j$  de la columna  $j$

	$v_1$	$v_2$	$v_3$	$v_4$	$v_5$
$u_1$	1	0.2	0.2	0.8	0.3
$u_2$	0.2	1	0.6	0.2	0.7
$u_3$	0.2	0.6	1	0.2	0.5
$u_4$	0.8	0.2	0.2	1	0.4

En esta tabla o matriz asociada a la relación  $R$  se observa que

$$\mu_R(u_2, v_3) = 0.6, \mu_R(u_4, v_5) = 0.4$$

Por ejemplo, si  $U$  es el conjunto de ciertas personas y  $V$  el conjunto de ciertas actividades deportivas, un valor de  $\mu_R(u_i, v_j) = 0.7$  podría representar que la persona  $u_i$  está moderadamente interesada en la actividad  $v_j$ .

Las relaciones difusas permiten manejar y modelar situaciones en las que los límites entre categorías no están claramente definidos, lo que es común en el mundo real. Este enfoque es particularmente útil en campos como la inteligencia artificial, la

toma de decisiones, la economía y la ingeniería, donde la incertidumbre y la vaguedad son inherentes (Zadeh, 1969), (Zadeh, 1971).

#### **2.2.1.21 RELACIONES DE EQUIVALENCIA**

Sea  $X$  un conjunto de referencia,  $R$  una relación definida en  $X \times X$  con su función de membresía  $\mu_R : X \times X \rightarrow [0,1]$ ,  $R$  es una relación de equivalencia si cumple las siguientes propiedades (Zadeh, 1969):

Reflexiva:  $\mu_R(x, x) = 1$  ,  $\forall x \in X$

Simétrica:  $\mu_R(x, y) = \mu_R(y, x)$  ,  $\forall x, y \in X$

Transitiva:  $T(\mu_R(x, y), \mu_R(y, z)) \leq \mu_R(x, z)$ ,  $\forall x, y, z \in X$ ,  $T$  es una T-Norma

En términos simples, una relación difusa de equivalencia agrupa elementos en clases que están "difusamente" relacionadas entre sí, permitiendo grados de pertenencia en lugar de una asociación binaria (sí/no). Cada elemento pertenece a una clase de equivalencia difusa con un cierto grado, permitiendo representar la pertenencia parcial o la pertenencia múltiple a varias clases (Zadeh, 1969).

#### **2.2.1.22 RELACIONES DE ORDEN O COMPARACIÓN**

Una relación de orden difusa es una extensión del concepto clásico de relación de orden. En lugar de establecer un orden absoluto y preciso, como en las relaciones de orden  $\leq, <, \geq, >$  tradicionales, las relaciones de orden difusas permiten comparar elementos con grados de certeza, proporcionando un marco más flexible para representar preferencias, prioridades y clasificaciones en presencia de incertidumbre (Zadeh, 1969).

Sea  $X$  un conjunto de referencia,  $R$  una relación definida en  $X \times X$  con su función de membresía  $\mu_R : X \times X \rightarrow [0,1]$ ,  $R$  es una relación de orden si cumple las siguientes propiedades (Zadeh, 1969).

Reflexiva:  $\mu_R(x, x) = 1$ ,  $\forall x \in X$

Antisimétrica: Si  $\mu_R(x, y) = 1$ ,  $\mu_R(y, x) = 1$  entonces  $x = y$   $\forall x, y \in X$

Transitiva:  $T(\mu_R(x, y), \mu_R(y, z)) \leq \mu_R(x, z)$ ,  $\forall x, y, z \in X$ ,  $T$  es una T-Norma

### 2.2.1.23 PROYECCIÓN DIFUSA

Una relación de proyección difusa es un concepto en teoría de conjuntos difusos que se utiliza para reducir la dimensionalidad de una relación difusa en un espacio multidimensional. Este proceso es análogo a la proyección en espacios clásicos, pero se adapta a la naturaleza gradual y vaga de las relaciones difusas (Morales y Méndez, 2011), (Klir y Yuan, 1995).

En términos simples, una relación de proyección difusa toma una relación difusa definida sobre un producto cartesiano de conjuntos y "proyecta" esta relación en uno o más de sus subconjuntos. Esto se hace mediante la agregación de los grados de pertenencia a lo largo de las dimensiones que no se están considerando en la proyección (Morales y Méndez, 2011), (Klir y Yuan, 1995).

Sea  $R$  una relación difusa en el producto cartesiano de conjuntos difusos  $U_1 \times U_2 \times \dots \times U_n$  tal que asocia a cada vector  $u = (u_1, u_2, \dots, u_n) \in U_1 \times U_2 \times \dots \times U_n$  un grado de pertenencia  $\mu_R(u)$  que se puede interpretar como un grado de satisfacción de la relación  $R$ .

Definimos  $\pi_R$  la proyección de  $R$  en  $U_{i_1} \times U_{i_2} \dots \times U_{i_k}$  donde  $\{i_1, i_2, \dots, i_k\}$  es una subsucesión de  $\{1, 2, \dots, n\}$  como una relación difusa definida por su función de membresía

$$\mu_{\pi_R}(u_{i_1}, u_{i_2}, \dots, u_{i_k}) = \sup_{u: u_{i_1}, u_{i_2}, \dots, u_{i_k}} \mu_R(u)$$

En el caso particular que se realice la proyección de  $R$  en  $U_i$ , el conjunto difuso tendría la siguiente función de membresía

$$\mu_{\pi_R}(u_i) = \sup_{u_1, \dots, u_{i-1}, u_{i+1}, \dots, u_n} \mu_R(u)$$

Como ejemplo supongamos que tenemos la relación difusa  $R \subset X \times Y$ , definida mediante su función de membresía  $\mu_R(x, y)$ , denotemos la proyección difusa de  $R$  en el conjunto  $X$  como  $\mu_{\pi_R}(x)$  y lo definimos como:

$$\mu_{\pi_R}(x) = \max_{y \in Y} \mu_R(x, y)$$

Consideremos  $X = \{x_1, x_2, x_3\}$ ,  $Y = \{y_1, y_2, y_3\}$  y la relación difusa  $R \subset X \times Y$ , definida por la siguiente matriz

$$R = \begin{pmatrix} 0.8 & 0.6 & 0.4 \\ 0.5 & 0.9 & 0.7 \\ 0.3 & 0.2 & 0.9 \end{pmatrix}$$

Para calcular la proyección difusa de  $R$  en el conjunto  $X$ , calculamos el máximo grado de compatibilidad de cada elemento de  $Y$ .

$$\mu_{\pi_R}(x_1) = \max_{y \in Y} \mu_R(x_1, y) = \max\{0.8, 0.6, 0.4\} = 0.8$$

$$\mu_{\pi_R}(x_2) = \max_{y \in Y} \mu_R(x_2, y) = \max\{0.5, 0.9, 0.7\} = 0.9$$

$$\mu_{\pi_R}(x_3) = \max_{y \in Y} \mu_R(x_3, y) = \max\{0.3, 0.2, 0.9\} = 0.9$$

Luego el conjunto difuso proyección es:

$$\pi_R(X) = \{(x_1, 0.8), (x_2, 0.9), (x_3, 0.9)\}$$

### 2.2.1.24 EXTENSIÓN CILÍNDRICA

La extensión cilíndrica difusa se utiliza para extender conjuntos o relaciones difusas definida en un subespacio a un espacio de mayor dimensión (Morales y Méndez, 2011), (Klir y Yuan, 1995).

La extensión cilíndrica es muy útil cuando se trabaja con relaciones incluidas en espacios multidimensionales y se tiene la necesidad de integrar más información adicional, pero manteniendo la coherencia de estos datos con los de la relación original, esto significa seguir replicando la función de membresía de la relación original a través de las nuevas dimensiones añadidas (Morales y Méndez, 2011), (Klir y Yuan, 1995).

Formalmente, sea  $A$  un conjunto difuso definido en  $X$  con su función de membresía  $\mu_A(x): X \rightarrow [0,1]$ , hagamos la extensión cilíndrica difusa de  $A$  en la nueva dimensionalidad  $X \times Y$ . Para este propósito denotemos a esta extensión cilíndrica difusa de  $A$  en  $X \times Y$  con  $A \times Y$  se definamos la siguiente función de membresía:

$$\mu_{A \times Y}(x, y) = \mu_A(x) \quad \forall (x, y) \in A \times Y$$

Esta definición indica que la extensión cilíndrica solo depende del valor de  $x \in A$ , cualquiera sea el valor de  $y \in Y$  (Morales y Méndez, 2011), (Klir y Yuan, 1995).

Como ejemplo consideremos  $X = \{x_1, x_2, x_3\}$ ,  $Y = \{y_1, y_2\}$  y el conjunto difuso  $A$  definido en  $X$

$$A = \{(x_1, 0.8), (x_2, 0.6), (x_3, 1)\}$$

La extensión cilíndrica difusa de  $A$  en  $X \times Y$  es:

$$A \times Y = \{(x_1, y_1), 0.8), ((x_1, y_2), 0.8), ((x_2, y_1), 0.6), ((x_2, y_2), 0.6), ((x_3, y_1), 1), ((x_3, y_2), 1)\}$$

### 2.2.1.25 COMPOSICIÓN DE RELACIONES

Consideremos los conjuntos  $X, Y, Z$  y las relaciones difusas  $R \subset X \times Y$ ,

$S \subset Y \times Z$ , con sus respectivas funciones de membresía  $\mu_R: X \times Y \rightarrow [0,1]$ ,

$\mu_S: Y \times Z \rightarrow [0,1]$ , la relación  $T \subset X \times Z$  composición de  $R$  con  $S$  denotado por

$T = RoS$  es una relación difusa cuya función de membresía es (Klir y Yuan, 1995):

$$\mu_T(x, z) = \mu_{RoS}(x, z) = \max_{y \in Y} \min \{ \mu_R(x, y), \mu_S(y, z) \}$$

### 2.2.1.26 EL CONDICIONAL DIFUSO

El condicional difuso extiende el concepto de condicional o implicación de la lógica clásica al mundo difuso de tal forma que permita manejar situaciones donde las condiciones no son absolutas, en la práctica, la capacidad para trabajar con grados de verdad hace que las aplicaciones puedan abordar la incertidumbre (Morales y Méndez, 2011), (Zapata, 2022).

Una expresión de la forma “ $A$  entonces  $B$ ” se denota en el cálculo proposicional con el condicional  $A \rightarrow B$ , en el caso que los conjuntos fueran nítidos se puede escribir como  $\neg A \vee B$  lo que se denomina implicación material y se dice que son equivalentes por tener los mismos valores de verdad, es decir.

$$A \rightarrow B \equiv \neg A \vee B \quad (4)$$

Denotemos al condicional difuso o implicación difusa con  $I$ , entiéndase este condicional como una relación difusa entre dos conjuntos difusos  $A$  y  $B$ , definamos entonces la función de membresía  $\mu_{A \rightarrow B}(x, y)$  de esta relación. Para tal propósito,

de forma más explícita, se puede expresar  $I : [0,1] \times [0,1] \rightarrow [0,1]$  y podemos por conveniencia utilizar la siguiente notación:

$$\mu_{A \rightarrow B}(x, y) = I(\mu_A(x), \mu_B(y))$$

Esta notación en realidad nos permite visualizar que la relación condicional entre dos elementos  $x, y$  en el mundo difuso se hace a través de su condicional difuso quien a su vez se define en base a las funciones de membresía de los conjuntos  $A$  y  $B$  (Morales y Méndez, 2011), (Zapata, 2022).

Se pueden obtener diversas implicaciones difusas similares a la implicación material mostrada en (4), lo más usual es considerar una S-norma  $S$  y una negación fuerte  $n$  según la expresión:

$$I(x, y) = S(n(x), y)$$

A esta forma de definir un condicional difuso se le denomina una S-implicación, luego utilizando la S-norma del máximo y la negación fuerte  $n(x) = 1 - x$  se tiene que:

$$\mu_{A \rightarrow B}(x, y) = \text{máx}\{1 - \mu_A(x), \mu_B(y)\} = I(x, y)$$

Utilizando algunas otras S-normas conocidas, El condicional difuso se puede representar como:

$$\mu_{A \rightarrow B}(x, y) = 1 - \mu_A(x) + \mu_A(x)\mu_B(y)$$

$$\mu_{A \rightarrow B}(x, y) = \text{mín}\{1 - \mu_A(x) + \mu_B(y), 1\}$$

Es importante comprender para efectos del razonamiento difuso que el condicional difuso debe cumplir las siguientes propiedades (Morales y Méndez, 2011):

1.- Compatible con la tabla de verdad del condicional en el caso de manejar predicados precisos o nítidos.

$$I(0,0) = I(0,1) = I(1,1) = 1, \quad I(1,0) = 0$$

2.- Un incremento en la verdad del antecedente no conduce a un incremento en la verdad de la implicación.

$$\text{Si } x \leq x' \text{ entonces } I(x, y) \geq I(x', y), \forall x, y, x' \in [0,1]$$

3.- Un incremento en la verdad del consecuente conduce a un incremento en la verdad de la implicación.

$$\text{Si } y \leq y' \text{ entonces } I(x, y) \leq I(x, y'), \forall x, y, y' \in [0,1]$$

4.- Si el antecedente es completamente falso, la implicación es completamente cierta, independientemente del valor de verdad del consecuente.

$$I(0, y) = 1, \forall y \in [0,1]$$

5.- Si el consecuente es cierto, la implicación es completamente cierta, independientemente del valor de verdad del antecedente.

$$I(x, 1) = 1, \forall x \in [0,1]$$

6.- Si el antecedente es cierto, la implicación es tan cierta como lo es su consecuente.

$$I(1, y) = y, \forall y \in [0,1]$$

7.- Implementa el principio del intercambio nítido

$$(A \rightarrow (B \rightarrow C)) \equiv (B \rightarrow (A \rightarrow C)) \equiv ((A \wedge B) \rightarrow C)$$

$$I(x, I(y, z)) = I(y, I(x, z)), \forall x, y, z \in [0,1]$$

8.- Forma de representar numéricamente la fórmula nítida

$$B \rightarrow (A \rightarrow B)$$

$$I(x, y) \geq y, \forall x, y \in [0,1]$$

9.- Permite generalizar la tautología nítida

$$A \rightarrow A$$

$$I(x, x) = 1, \forall x \in [0,1]$$

10.- La implicación es completamente cierta en tanto que el consecuente es al menos tan cierto como el antecedente, Por otra parte, la implicación define un orden.

$$I(x, y) = 1 \text{ si y solo si } x \leq y, \quad \forall x, y \in [0,1]$$

11.- Representa la ley de la contraposición nítida

$$(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A)$$

$$I(x, y) = I(n(y), n(x)), \quad \forall x, y \in [0,1]$$

12.-  $I$  es una función continua

En la práctica existen varias propuestas que se han usado para extender la implicación nítida en una condicional difuso. En el cuadro 7 se presentan otras formas de definir el condicional difuso, usado en diversas aplicaciones (Morales y Méndez, 2011), (Zapata, 2022).

**Cuadro 7:**

*Diversas formas del condicional difuso*

AUTOR	CONDICIONAL DIFUSO $I(x, y) = \mu_{A \rightarrow B}(x, y)$
<b>Zadeh</b>	$\mu_{A \rightarrow B}(x, y) = \text{máx}\{1 - \mu_A(x), \mu_B(y)\}$
<b>Mamdani</b>	$\mu_{A \rightarrow B}(x, y) = \text{mín}\{\mu_A(x), \mu_B(y)\}$
<b>Godel</b>	$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1 & \text{si } \mu_A(x) \leq \mu_B(y) \\ \mu_B(y) & \text{si } \mu_A(x) > \mu_B(y) \end{cases}$
<b>Lukasiewicz</b>	$\mu_{A \rightarrow B}(x, y) = \text{mín}\{1, 1 - \mu_A(x) + \mu_B(y)\}$
<b>Rescher-Gaines</b>	$\mu_{A \rightarrow B}(x, y) = \begin{cases} 1 & \text{si } \mu_A(x) \leq \mu_B(y) \\ 0 & \text{si } \mu_A(x) > \mu_B(y) \end{cases}$
<b>Reichenbach</b>	$\mu_{A \rightarrow B}(x, y) = 1 - \mu_A(x) + \mu_A(x) \cdot \mu_B(y)$
<b>Yager</b>	$\mu_{A \rightarrow B}(x, y) = 1 - \text{mín}\{1, \mu_A(x) - \mu_B(y)\}$
<b>Kleene-Dienes</b>	$\mu_{A \rightarrow B}(x, y) = \text{máx}\{1 - \mu_A(x), \text{mín}\{\mu_A(x), \mu_B(y)\}\}$

### 2.2.1.27 INFERENCIA DIFUSA

El Modus Ponens es una de las principales leyes de la inferencia en la lógica deductiva clásica, se utiliza para derivar conclusiones lógicas verdaderas a partir de premisas verdaderas, tiene el siguiente formato

*Premisas*  
*Condición:*  $p \rightarrow q$   
*Observación:*  $p$   
*Conclusión:*  
 $q$

Si tanto la condición como la observación son verdaderas, se tiene entonces que la conclusión es verdadera.

Considerando que, en la lógica difusa, las proposiciones no son simplemente verdaderas o falsas, sino que tienen un grado de verdad entre 0 y 1, representado por funciones de membresía y tomando los conjuntos difusos  $A$  y  $B$ , definidos en los conjuntos de referencia  $X, Y$  respectivamente, La regla de inferencia del modus ponens difuso se puede formular de la siguiente manera:

*Premisas*  
*Condición:* *si  $x$  es  $A$  entonces  $y$  es  $B$*   
*Observación:*  *$x$  es  $A$*   
*Conclusión:*  
 *$y$  es  $B$*

Debe entenderse que la condición simbólicamente se puede representar con  $A \rightarrow B$ , el hecho de afirmar por ejemplo que  $x$  es  $A$  es afirmar que  $x$  tiene un grado de verdad representado por  $\mu_A(x)$  (Zadeh, 1975).

Con estas consideraciones, el modus ponens difuso se puede operacionalizar usando los conceptos de condicional difuso, relaciones difusas, composición de relaciones difusas, extensiones, proyecciones y operaciones de conjuntos difusos,

conceptos presentados anteriormente (Zadef, 1996), (Zadeh, 1988), (Klir y Yuan, 1995).

*Premisas*

*Condición: Una relación difusa  $R \subset X \times Y$  que represente al condicional difuso  $A \rightarrow B$  con su función de membresía  $\mu_R(x, y)$ .*

*Observación: Un conjunto difuso  $A$  definido en  $X$  con su función de membresía  $\mu_A(x)$*

*Conclusión:*

*Un conjunto difuso  $B$  definido en  $Y$  con su función de membresía  $\mu_B(y)$*

Luego la inferencia del modus ponens difuso se puede describir como la proyección difusa de la composición de  $A \circ R$  sobre  $Y$ , usando una T-norma, es decir se puede escribir  $B = A \circ (A \rightarrow B)$ , por ejemplo, la composición de  $A$  con  $R$  se obtiene con la operación de extensión en  $Y$

$$\mu_{A \circ R}(y) = \max_{x \in X} \min \{ \mu_A(x), \mu_R(x, y) \}$$

Esta expresión evalúa el grado de pertenencia de cada  $y \in Y$  con el valor máximo de la intersección (mínimo) de  $\mu_A(x)$  y  $\mu_R(x, y)$  a través de todos los valores  $x \in X$  (Morales y Méndez, 2011), (Zapata, 2022).

La proyección difusa de la relación compuesta  $A \circ R$  sobre  $Y$ , define la función de membresía de  $B$

$$\mu_B(y) = \mu_{A \circ R}(y) = \max_{x \in X} \min \{ \mu_A(x), \mu_R(x, y) \}$$

Esto significa que  $\forall y \in Y$ , el grado de pertenencia de  $y$  en  $B$  es determinado por la mayor intersección entre el grado de pertenencia de  $x$  en  $A$  y el grado de pertenencia de la relación  $\mu_R(x, y)$  (Zadef, 1996).

Una de las implementaciones del modus ponens difuso más sencilla y utilizada es la que propuso Mamdani que se basa en la T-norma del mínimo para modelar la relación entre el antecedente y el consecuente. La inferencia se realiza mediante la

composición de las funciones de membresía del antecedente y la implicación, obteniendo así el grado de pertenencia del consecuente. Mamdani propone un método de razonamiento difuso al que se le denomina Inferencia tipo Mamdani que se utiliza bastante en los sistemas difusos (Zadeh, 1988), (Zadeh, 2005).

### 2.2.1.28 INFERENCIA TIPO MANDANI

Propuesto por Ibrahim Mamdani en 1975. En la figura 17, se muestra de manera gráfica los procesos para hacer la inferencia con dos entradas nítidas  $x_0, y_0$  y con dos reglas difusas seleccionadas, pudiéndose generalizar para más de dos reglas difusas.

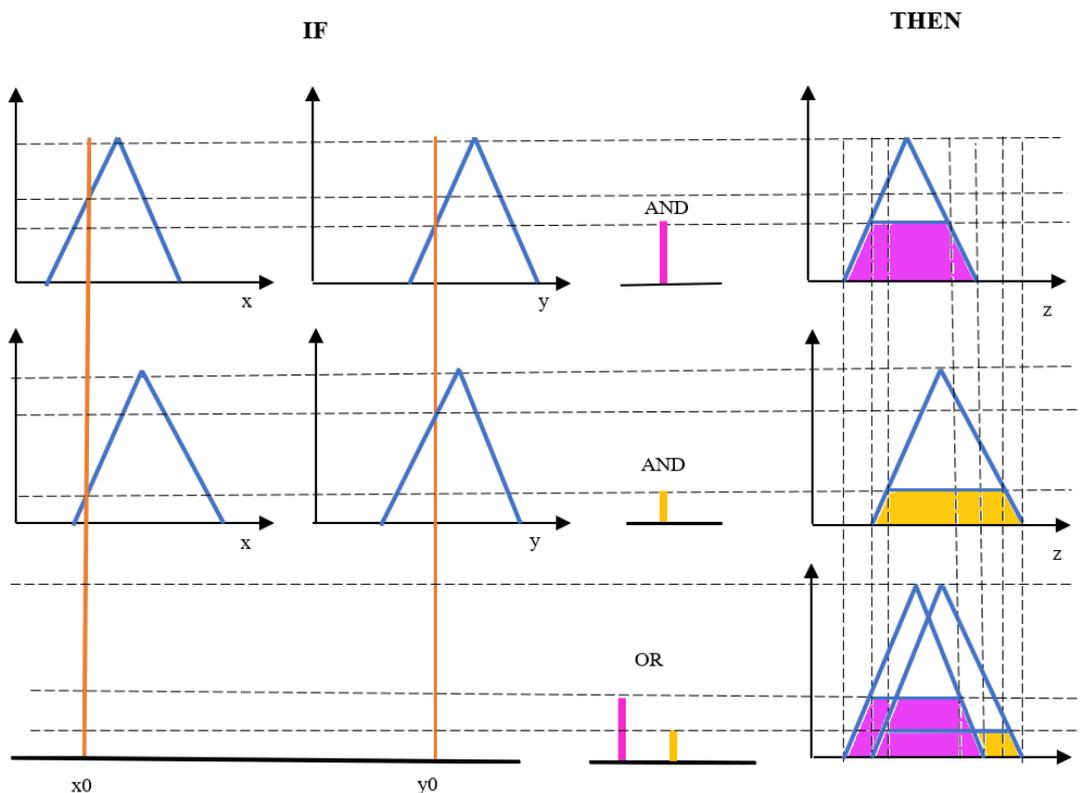


Figura 17:  
*Inferencia de Mamdani para dos reglas difusas y dos entradas nítidas*

De manera general esta inferencia se implementa en 6 pasos o procesos que son mostrados en el cuadro 8.

**Cuadro 8:**  
*Procesos en la inferencia tipo Mamdani*

PASO	PROCESO	BREVE DESCRIPCIÓN
1	Obtener el conjunto de reglas difusas	Establecer las reglas difusas que describen como se debe tomar la decisión teniendo en cuenta las entradas y salida
2	Fuzzificación	Obtener valores para las entradas según sus funciones de membresía
3	Seleccionar reglas	Combinar las entradas difusas usando el proceso de combinación difusa para seleccionar las reglas que cumplen.
4	Determinar consecuentes	Obtener el valor de la función de membresía de la salida según cada regla difusa seleccionada
5	Agregación	Combinar los consecuentes obtenidos para conseguir la distribución de salida
6	Defuzzificación	Convertir la distribución de salida en un valor nítido

**Paso 1. Obtener el conjunto reglas difusas.**

El conjunto de reglas difusas es una colección de expresiones lingüísticas que describen cómo se debe tomar una decisión con respecto a clasificar una entrada o controlar una salida. Las reglas difusas tienen la siguiente forma:

*IF (la entrada 1 es la función de membresía 1) y/o (la entrada 2 es la función de membresía 2) y/o ...ELSE  
(la salida n es la función de membresía de salida n).*

Por ejemplo, para definir la siguiente regla difusa:

Si la temperatura es alta y la humedad es alta, entonces la habitación está caliente.

Se tendría que haber establecido funciones de membresía que definan temperatura alta (entrada 1), humedad alta (entrada 2) y una habitación caliente (salida 1).

El conjunto de reglas de inferencias debe estar almacenado en una base de conocimientos.

### **Paso 2. Fuzzificación.**

El propósito de la fuzzificación es establecer para las entradas nítidas sus valores en el intervalo de 0 a 1 utilizando un conjunto de funciones de membresía de entrada. Consiste en tomar los valores numéricos nítidos de las entradas y determinar el grado de pertenencia de estas entradas a los conjuntos difusos asociados. El valor nítido está limitado en el universo del discurso de la variable. En la figura 18. Se muestra la fuzzificación de las entradas nítidas  $x_0$  obteniéndose gráficamente los valores de las funciones de membresía  $\mu_A(x_0), \mu_C(x_0)$  para los conjuntos difusos  $A$  y  $C$ , de igual manera para la entrada nítida  $y_0$  obteniéndose los valores  $\mu_B(y_0), \mu_D(y_0)$  para sus correspondientes conjuntos difusos  $B$  y  $D$ .

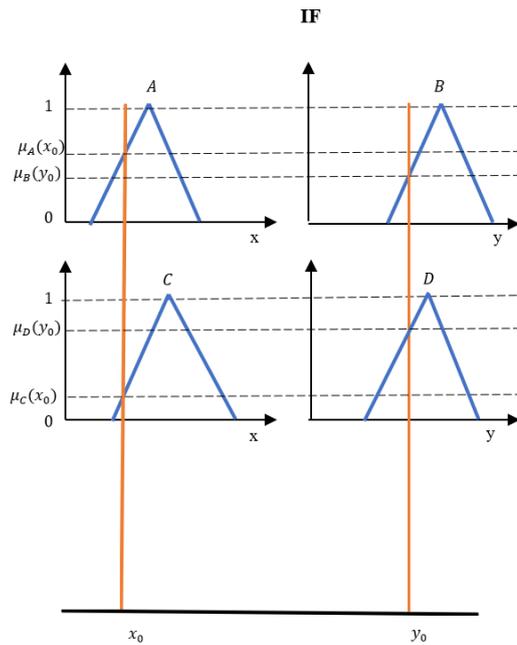


Figura 18 :  
*Proceso de Fuzzificación en método de Mamdani*

**Paso 3. Seleccionar reglas.**

De la base de conocimientos que tiene almacenado el conjunto de reglas difusas, se deben seleccionar las reglas que entrarán en la inferencia, para esto se toman las entradas fuzzificadas y se aplican a los antecedentes de las reglas difusas, siendo seleccionadas aquellas que cumplan con la estructura lógica en términos de AND u OR del antecedente.

**Paso 4. Determinar consecuentes**

Para determinar los consecuentes de cada una de las reglas difusas seleccionadas de la base de conocimientos, se evalúan los operadores lógicos que tienen sus antecedentes.

Para evaluar la disyunción (operador OR) habitualmente se emplea el máximo, definido como:  $\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}$  de igual forma, para el AND se usa habitualmente el mínimo.  $\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}$ . Ver figura 19.

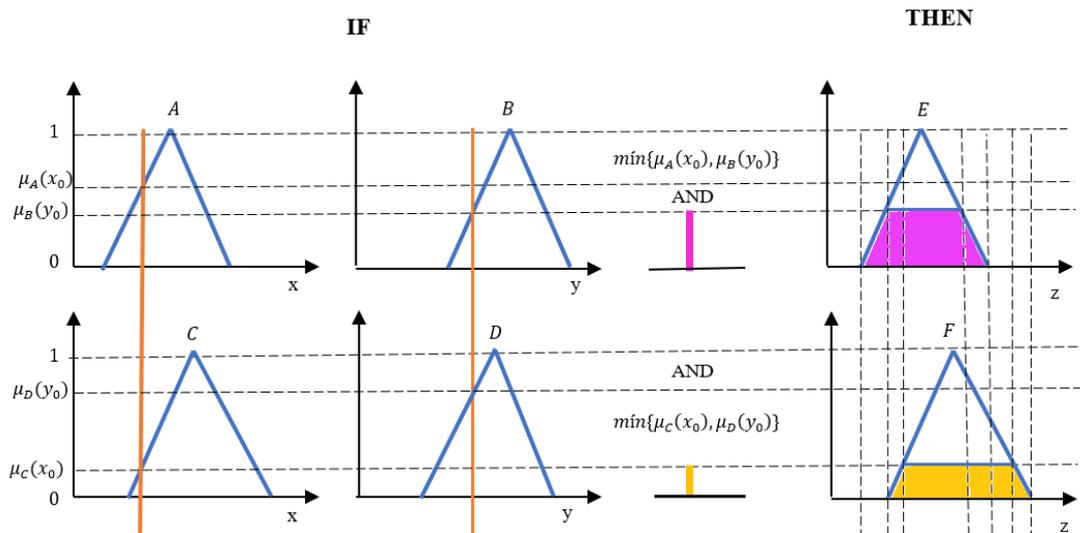


Figura 19.  
Obtención de los consecuentes de las reglas seleccionadas

El resultado de la evaluación del antecedente se aplica al consecuente, aplicando un recorte o escalado según el valor de verdad del antecedente. El método más comúnmente utilizado es el recorte (clipping) que corta el consecuente con el valor de verdad del antecedente. En la figura 19 se ha recortado el conjunto difuso  $E$  según el resultado de la primera regla, de igual manera se ha recortado el conjunto difuso de salida  $F$  según el resultado de la segunda regla.

### Paso 5. Agregación

La agregación es el proceso de unificación de las salidas de todas las reglas seleccionadas; es decir, se combinan las funciones de pertenencia de todos los consecuentes previamente recortados o escalados para obtener una distribución de salida difusa. Por lo general, aunque no siempre, esto se hace mediante el uso del

“OR” difuso. En la figura 20 se muestra el proceso de agregación de dos consecuentes obtenidos al evaluar dos reglas difusas con dos entradas nítidas.

### **Paso 6. Defuzzificación**

El resultado final habitualmente es necesario expresarlo mediante un valor nítido. En esta etapa se toma como entrada un conjunto difuso anteriormente obtenido para dar un valor de salida. Existen varios métodos de defuzzificación, pero probablemente el más ampliamente usado es el centroide; que calcula el punto donde una línea vertical divide el conjunto en dos áreas con igual masa utilizando la formula

$$Centroide = \frac{\sum_{x=a}^b \mu_A(x)x}{\sum_{x=a}^b \mu_A(x)} \quad o \quad \mathit{centroide} = \frac{\int y \cdot \mu_V(y) dy}{\int \mu_V(y) dy}$$

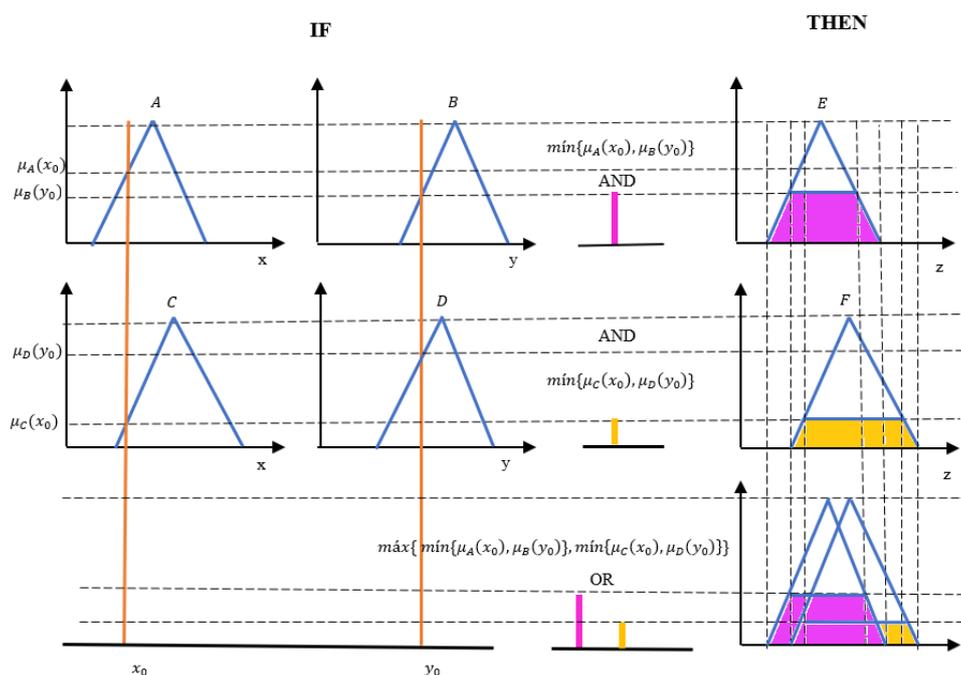


Figura 20.  
Proceso de agregación en el método de Mamdani

Se presenta a continuación un ejemplo de inferencia tipo Mamdani, para lo cual consideremos las variables lingüísticas de entrada “Temperatura” y “Humedad” del ambiente y la como variable de salida la “Velocidad” del ventilador, sus valores lingüísticos o conjuntos difusos respectivamente son:

Temperatura= {“baja”, “media”, “alta”},

$$\mu_{\text{baja}}(\mathbf{x}) = \mu_{\text{triangular}}(\mathbf{x}; -10, 10, 15)$$

$$\mu_{\text{media}}(\mathbf{x}) = \mu_{\text{triangular}}(\mathbf{x}; 10, 20, 30)$$

$$\mu_{\text{alta}}(\mathbf{x}) = \mu_{\text{triangular}}(\mathbf{x}; 25, 40, 50)$$

Humedad= {“baja”, “media”, “alta”},

$$\mu_{\text{baja}}(\mathbf{x}) = \mu_{\text{triangular}}(\mathbf{x}; 0, 20, 40)$$

$$\mu_{\text{media}}(\mathbf{x}) = \mu_{\text{triangular}}(\mathbf{x}; 30, 50, 70)$$

$$\mu_{\text{alta}}(\mathbf{x}) = \mu_{\text{triangular}}(\mathbf{x}; 60, 80, 100)$$

Velocidad = {"lenta", "media", "rápida"}

$$\mu_{lenta}(x) = \mu_{triangular}(x; 0, 15, 30)$$

$$\mu_{media}(x) = \mu_{triangular}(x; 20, 50, 80)$$

$$\mu_{rápida}(x) = \mu_{triangular}(x; 70, 85, 100)$$

Se explican los procesos según el cuadro 8.

#### Paso1. Obtener el conjunto de reglas difusas

*R1: Si (Temperatura es baja) y (Humedad es baja) entonces (velocidad es lenta)*

*R2: Si (Temperatura es media) y (Humedad es media) entonces (velocidad es media)*

*R3: Si (Temperatura es alta) y (Humedad es alta) entonces (velocidad es rápida)*

*R4: Si (Temperatura es alta) y (Humedad es baja) entonces (velocidad es media)*

*R5: Si (Temperatura es baja) y (Humedad es alta) entonces (velocidad es media)*

#### Paso2. Fuzzificación

Consideremos las entradas: Temperatura =  $x_0 = 22^\circ\text{C}$ , según sus funciones de membresía se obtiene:

$$\mu_{baja}(x_0) = 0, \quad \mu_{media}(x_0) = 0.8, \quad \mu_{alta}(x_0) = 0.1$$

Humedad= $y_0 = 65\%$ , de acuerdo a sus funciones de membresía se tiene:

$$\mu_{baja}(y_0) = 0, \quad \mu_{media}(y_0) = 0.25, \quad \mu_{alta}(y_0) = 0.75$$

#### Paso 3. Selección de reglas

*R1: Si ( $\mu_{baja}(x_0) = 0$ , ) y ( $\mu_{baja}(y_0) = 0$ ) entonces (velocidad es lenta)*

*R2: Si ( $\mu_{media}(x_0) = 0.8$ ) y ( $\mu_{media}(y_0) = 0.25$ ) entonces (velocidad es media)*

*R3: Si ( $\mu_{alta}(x_0) = 0.1$ ) y ( $\mu_{alta}(y_0) = 0.75$ ) entonces (velocidad es rápida)*

*R4: Si ( $\mu_{alta}(x_0) = 0.1$ ) y ( $\mu_{baja}(y_0) = 0$ , ) entonces (velocidad es media)*

*R5: Si ( $\mu_{baja}(x_0) = 0$ ) y ( $\mu_{alta}(y_0) = 0.75$ ) entonces (velocidad es media)*

#### Paso 4. Determinar consecuentes

Utilizando la T-norma del mínimo

$$R1: \text{mín}\{0, 0\} = 0 \text{ (no se considera)}$$

$$R2: \text{mín}\{0.8, 0.25\} = 0.25 \text{ (se considera velocidad media)}$$

$$R3: \text{mín}\{0.1, 0.75\} = 0.1 \text{ (se considera velocidad rápida)}$$

$$R4: \text{mín}\{0.1, 0\} = 0 \text{ (no se considera)}$$

$$R5: \text{mín}\{0, 0.75\} = 0 \text{ (no se considera)}$$

Paso 5. Agregación

$$R2: \mu_{media}(y) = \text{mínimo}(0.25, \mu_{media}(y))$$

$$R3: \mu_{rápida}(y) = \text{mínimo}(0.1, \mu_{rápida}(y))$$

Combinando las salidas difusas obtenidas mediante la S-norma del máximo

$$\mu_V(y) = \text{máx}\{\text{mínimo}(0.25, \mu_{media}(y)), \text{mínimo}(0.1, \mu_{rápida}(y))\}$$

Paso 6. Defuzzificación

Para convertir el resultado difuso en un valor nítido se usa el método del centroide

$$V_{crisp} = \text{centroide} = \frac{\int y \cdot \mu_V(y) dy}{\int \mu_V(y) dy}$$

En la figura 21, se muestra de manera gráfica los procesos que se llevan a cabo en

una inferencia difusa tipo Mamdani (Klir y Yuan, 1995)

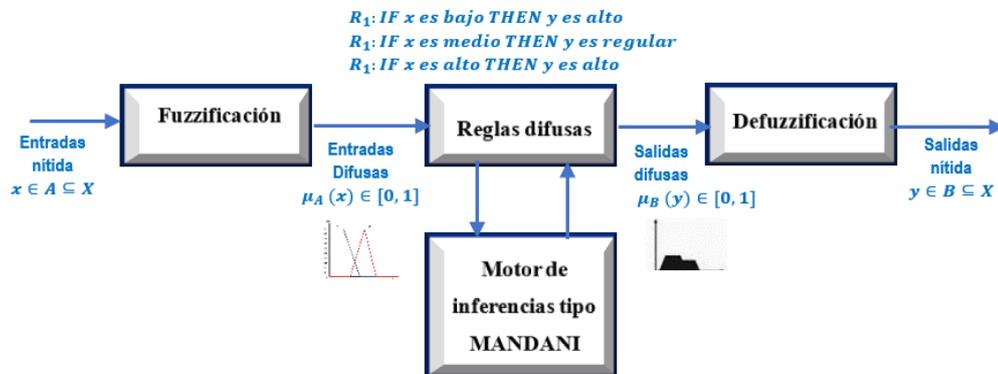


Figura 21:  
Procesos de inferencia difusa tipo Mamdani

### 2.2.1.29 INFERENCIA TIPO SUGENO

La inferencia tipo Sugeno, también conocida como inferencia TSK (Takagi-Sugeno-Kang), fue desarrollada por Takagi, Sugeno, y Kang en la década de 1980 y se utiliza ampliamente en aplicaciones donde se requiere un sistema de control preciso y rápido. En este tipo de inferencia difusa que utiliza funciones lineales o constantes para las salidas, en lugar de conjuntos difusos, resultando este método muy eficiente desde el punto de vista computacional comparado con el modelo de inferencia de Mamdani en lo referente a la defuzzificación. Se puede disminuir el tiempo de inferencia empleando una función matemática lineal o constante en el consecuente, de forma que el formato general en regla de inferencia TSK es:

IF  $x$  es A AND  $y$  es B THEN  $z$  es  $f(x, y)$ ,

en la figura 22 se muestra los procesos de la inferencia tipo Sugeno.

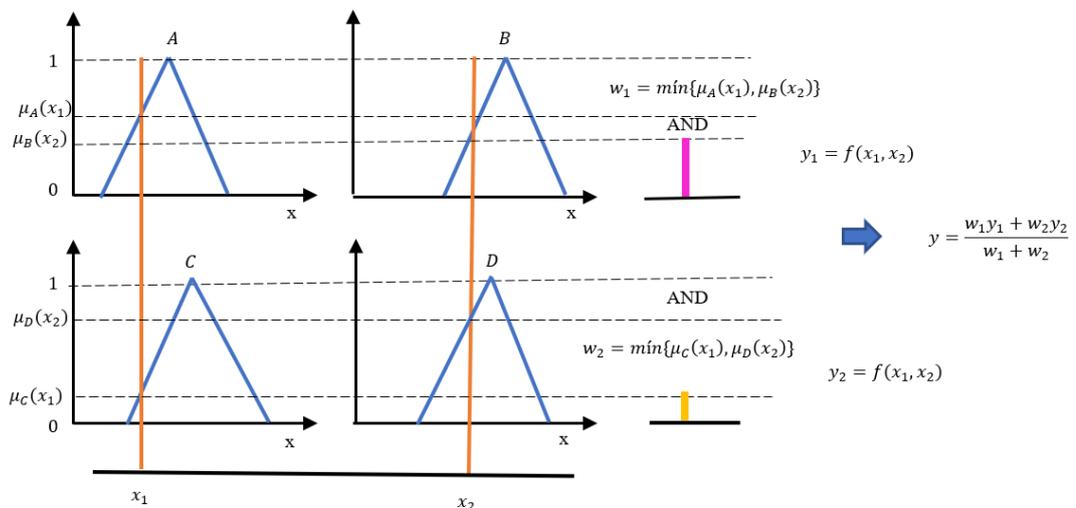


Figura 22:  
*Procesos de la inferencia tipo Sugeno*

Se presenta continuación un ejemplo para ilustrar una inferencia tipo Sugeno. Se considera como entradas son las variables lingüísticas

Temperatura= {"baja", "media", "alta"}, las funciones de membresía de los conjuntos difusos son:

$$\mu_{baja}(x) = \mu_{HombroDercho}(x; 10, 20)$$

$$\mu_{media}(x) = \mu_{Triangular}(x; 10, 20, 30)$$

$$\mu_{alta}(x) = \mu_{HombroIzquierdo}(x; 20, 30)$$

Humedad= {"baja", "media", "alta"}, con funciones de membresía

$$\mu_{baja}(x) = \mu_{HombroDerecho}(x; 40, 60)$$

$$\mu_{media}(x) = \mu_{Triangular}(x; 40, 60, 80)$$

$$\mu_{alta}(x) = \mu_{HombroIzquierdo}(x; 60, 80)$$

La salida es la "Calefacción" que no se presenta como conjunto difuso sino como el resultado de una función lineal o constante

Las reglas difusas son:

R1: Si (Temperatura es baja) y (Humedad es alta) entonces (Calefacción es  $y_1 = 30 - T + 0.2H$ )

R2: Si (Temperatura es media) y (Humedad es media) entonces (Calefacción  $y_2 = 50$ )

R3: Si (Temperatura es alta) y (Humedad es baja) entonces (Calefacción es  $y_3 = 30 - 0.5T - H$ )

Hacemos la fuzzificación de las entradas

Sea Temperatura= 18°C

$$\mu_{baja}(18) = 0.2, \mu_{media}(18) = 0.8, \mu_{alta}(18) = 0$$

Sea la Humedad= 55%

$$\mu_{baja}(55) = 0, \mu_{media}(55) = 0.75, \mu_{alta}(55) = 0.25$$

Evaluamos las reglas

R1: Si ( $\mu_{baja}(18) = 0.2$ , ) y ( $\mu_{alta}(55) = 0.25$ ) entonces  $w_1 = \min\{0.2, 0.25\} = 0.2$

$$y_1 = 30 - 18 + 0.2(55) = 23$$

R2: Si ( $\mu_{\text{media}}(18) = 0.8$ ) y ( $\mu_{\text{media}}(55) = 0.75$ ) entonces  $w_2 = \text{mín}\{0.8, 0.75\} = 0.75$

$$y_2 = 50$$

R3: Si ( $\mu_{\text{alta}}(18) = 0$ ) y ( $\mu_{\text{baja}}(55) = 0$ ) entonces  $w_3 = \text{mín}\{0, 0\} = 0$

$$y_3 = 30 - 0.5(18) - 55 = -34$$

Calculando la salida

$$y = \frac{0.2(23) + 0.75(50) + 0(-34)}{0.2 + 0.75 + 0} = 44.32$$

Lo que significa la calefacción adecuada es 44.32 cuando la temperatura es de 18°C y hay 55% de humedad.

### 2.2.1.30 SISTEMAS DIFUSOS

Un sistema difuso está constituido por un subsistema fuzzificador, un conjunto de reglas de inferencia, un motor de inferencia y un subsistema defuzzificador (Jantzen, 2007). La figura 23 muestra los elementos de un sistema difuso.

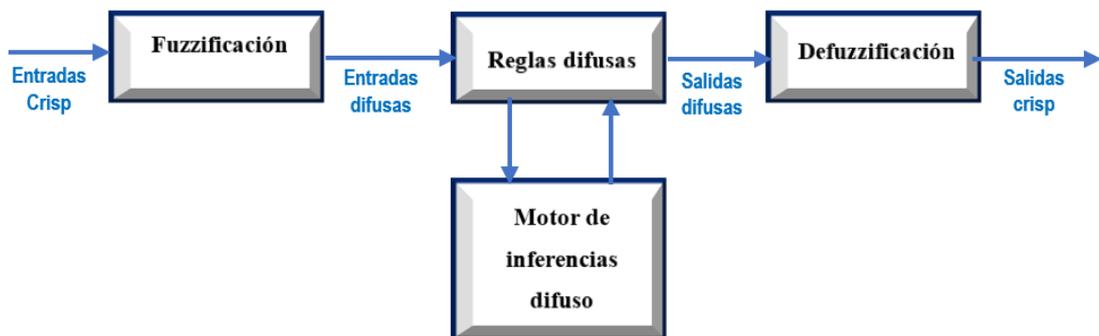


Figura 23:  
*Elementos de un sistema difuso*

#### Subsistema fuzzificador

Dadas las variables de entrada con valores crisp o nítidos, la fuzzificación consiste en asignar conjuntos difusos a cada una de estas variables y mapear sus valores según sus grados de membresía (Jantzen, 2007).

### **Reglas difusas o de inferencia**

Son expresadas en la forma SI-ENTONCES (IF-THEN), proporcionan la conexión necesaria entre los conjuntos difusos de entrada y salida del sistema difuso (Jantzen, 2007). Las reglas difusas son del tipo.

*IF (proposición difusa 1) AND (proposición difusa 2) AND ... AND (proposición difusa N)*  
*THEN (proposición difusa)*

Por ejemplo, si tenemos un sistema con dos variables de entrada y una variable de salida y se dispone de un conjunto constituido por  $N$  reglas de inferencia, entonces la  $j$  – ésima regla de inferencia para  $1 \leq j \leq N$  es escrita de la siguiente manera.

$$R_j : x \text{ is } A_j \text{ AND } y \text{ is } B_j \text{ THEN } z \text{ is } C_j$$

Donde  $x, y$  son las variables de entrada del sistema difuso y  $z$  es la variable de salida,  $A_1, A_2, \dots, A_N$  son los conjuntos difusos o valores de una variable lingüística  $L_1$ ,  $B_1, B_2, \dots, B_N$  son los conjuntos difusos o valores de una variable lingüística  $L_2$ ,  $C_1, C_2, \dots, C_N$  son los conjuntos difusos o valores de una variable lingüística  $L_3$ .

### **Motor de inferencias difuso**

La implementación de las reglas de inferencia se realiza en el motor de inferencia a través de alguna regla de composición que utiliza operaciones lógicas. El motor de inferencia difuso evalúa las reglas difusas en función de las entradas actuales y combina las reglas para producir una salida difusa, básicamente el proceso consiste en seleccionar las reglas, calcular el grado de verdad de la premisa para cada regla, combinar los resultados de todas las reglas aplicables e integrar los grados de pertenencia para formar una conclusión (Jantzen, 2007).

## Subsistema defuzzificador

Este subsistema convierte las salidas difusas que reporta el motor de inferencias a un valor nítido o numérico específico para que se pueda utilizar en la práctica. Se encarga de la conversión a un valor nítido las salidas difusas que reporta el motor de inferencias. El proceso consiste en considerar la función de pertenencia agregada de todas las salidas difusas producidas por el motor de inferencias y luego aplicar un método de defuzzificación para convertir el valor de esta función de pertenencia en un valor nítido o crisp, siendo los más comunes el método del centroide, de la media de máximos y de la máxima altura. En el cuadro 9 se listan los principales métodos de defuzzificación mencionando sus autores (Jantzen, 2007).

**Cuadro 9:**  
*Principales métodos de defuzzificación*

MÉTODO DE DEFUZZIFICACIÓN	AUTOR
Centroide (Centro de gravedad, COG)	Mamdani y Assilian (1975)
Promedio de máximos (Mean of Maximum, MOM)	Sugeno (1985)
Máximo valor de pertenencia	Bellman y Zadeh (1970)
Altura	Sugeno (1985)
Bisector del área	Ralston y Reilly (1993)
Defuzzificación alfa-corte ( $\alpha$ -Cut Defuzzification)	Zadeh (1965)
Promedio ponderado	Sugeno (1985)
Altura media	Dubois y Prade (1980)
Punto de máxima estabilidad	Nakamura (1992)
Suma ponderada de áreas	Mamdani y Assilian (1975)
Centroide modificado	Tsoukalas y Uhrig (1997)

## 2.2.2 TEORIA DE INTERNET OF THINGS (IOT)

Internet of things (IoT) es una disciplina del conocimiento que se refiere a la interconexión digital de dispositivos físicos usando internet, los cuales recopilan y comparten datos entre sí o interactúan con otros sistemas externos (Sunardi et al., 2023), (Gupta, 2019).

### 2.2.2.1 ARQUITECTURA DE UN SISTEMA IOT

Existen diversas formas de esquematizar las capas de la arquitectura de un sistema de Internet of Things, una de las más usuales es la que muestra en el cuadro 10.

#### Cuadro 10:

*Arquitectura de un sistema IoT*

CAPA	COMPONENTES	FUNCIÓN
<b>Dispositivos (Percepción)</b>	Sensores, actuadores, microcontroladores, dispositivos IoT	Recolección de datos del entorno físico y ejecución de acciones.
<b>Conectividad (Red)</b>	Conexiones, protocolos de comunicación, gateways, routers	Transmisión de datos desde dispositivos IoT hacia sistemas de procesamiento.
<b>Procesamiento en el borde (Edge computing)</b>	Gateways inteligentes, dispositivos de borde, servidores locales	Procesamiento preliminar de datos cerca de su origen.
<b>Agregación de datos</b>	Middleware, software de integración	Agregación y preprocesamiento de datos de múltiples fuentes.
<b>Procesamiento (Nube)</b>	Plataformas de procesamiento en la nube, servidores en la nube	Procesamiento avanzado y análisis de grandes volúmenes de datos.
<b>Almacenamiento</b>	Bases de datos (SQL, NoSQL), sistemas de almacenamiento en la nube	Almacenamiento de datos procesados y no procesados.
<b>Aplicación</b>	Aplicaciones de usuario, interfaces de usuario, dashboards, analítica, interfaz de programación de aplicaciones (API)	Interacción con los usuarios finales para visualización, monitoreo, control y análisis de datos.

### **2.2.2.2 CAPA DE DISPOSITIVOS: SENSORES**

Un sensor es un dispositivo que capta o detecta algún tipo de estímulo o cambio en su entorno físico y lo convierte en una señal generalmente eléctrica. Se define como transducción al proceso mediante el cual los sensores convierten una forma de energía (la señal física de entrada) en otra forma de energía (la señal eléctrica de salida) para lo cual utilizan diversos principios físicos y métodos de acuerdo al tipo de variable o magnitud que están midiendo (Raj y Raman, 2017), (Kranenburg, 2008), (Hu, 2016), (Krushnan et al., 2023). En el cuadro 11 se presentan algunos tipos de sensores clasificados según su principio de conversión o transducción

Los sensores son fundamentales en proyectos basados en IoT ya que constituyen la entrada al sistema mediante los cuales es posible proveer datos para una amplia variedad de aplicaciones en diversos sectores como la industria, la agricultura, la medicina, etc (Bakar, 2022), (Nyakuri et al., 2023). En el cuadro 12 se presentan algunos ejemplos de sensores, indicando el tipo de variable que censan e indicando su uso en algunas aplicaciones.

### **2.2.2.3 CAPA DE DISPOSITIVOS: ACTUADORES**

Son dispositivos que convierten una señal recibida que es generalmente eléctrica en una determinada acción física, mecánica o eléctrica concreta. En contraste con los sensores, que detectan y convierten estímulos físicos en generalmente señales eléctricas analógicas o digitales, los actuadores al recibir una señal de control de tipo eléctrico, la utilizan para iniciar algún tipo de movimiento o acción en algún dispositivo del mundo real como una bomba de agua, un extractor de gases, un sistema de ventilación, una bisturí, un brazo robótico, etc. Su uso resulta

fundamental en una amplia variedad de aplicaciones industriales, electrónicas y de automatización (Serpanos y Wolf, 2018), (Cirani et al., 2019), (Hillar, 2016), (Prasetyo et al., 2023). En el cuadro 13. Se presentan algunos ejemplos de actuadores, su clasificación según tipo y algunas de sus aplicaciones más usuales.

**Cuadro 11:**  
*Conversión de señales en los sensores*

TIPO DE SENSOR	PRINCIPIO DE CONVERSIÓN
<b>SENSORES RESISTIVOS</b>	La resistencia del material cambia en respuesta a una magnitud física como la temperatura o luz, este cambio se convierte en una variación de voltaje.
Termistores	La resistencia cambia con la temperatura.
LDR (Light Dependent Resistor)	La resistencia disminuye con el aumento de la intensidad de la luz.
<b>SENSORES PIEZOELÉCTRICOS</b>	La aplicación de una fuerza mecánica a un material piezoeléctrico genera una carga eléctrica proporcional a la fuerza aplicada.
Sensores de presión	La presión mecánica se convierte directamente en una señal eléctrica
Micrófonos	Las vibraciones del sonido se convierten en señales eléctricas.
<b>SENSORES CAPACITIVOS</b>	La capacitancia de un condensador varía con los cambios en las propiedades dieléctricas del medio entre las placas del condensador, generando una señal eléctrica.
Sensores de humedad	La humedad cambia la constante dieléctrica del aire entre las placas, modificando la capacitancia y, por ende, la señal eléctrica.
Sensores táctiles	La proximidad de un objeto cambia la capacitancia, alterando la señal eléctrica.
<b>SENSORES INDUCTIVOS</b>	La inductancia de una bobina cambia cuando un objeto metálico se aproxima, alterando el campo magnético y generando una señal eléctrica.
Sensores de proximidad	La proximidad de un objeto metálico cambia la inductancia, lo que se traduce en una señal eléctrica.
Sensores de posición	La posición de un objeto afecta la inductancia, generando una señal eléctrica
<b>SENSORES OPTOELECTRÓNICOS</b>	La luz incidente en un material fotosensible genera una corriente eléctrica mediante el efecto fotoeléctrico.
Fotodiodos	La luz genera pares de electrones-huecos, produciendo una corriente eléctrica proporcional a la intensidad de la luz.
Fototransistores	La luz incidente modula la corriente del transistor, generando una señal eléctrica.
<b>SENSORES TÉRMICOS</b>	Los termopares utilizan el efecto Seebeck (donde una diferencia de temperatura entre dos metales diferentes produce una tensión eléctrica)
Termopares	La diferencia de temperatura genera un voltaje proporcional a la diferencia de temperatura entre las dos uniones metálicas.

**Cuadro 12:**  
*Ejemplos de sensores*

TIPO DE SENSOR	APLICACIONES	EJEMPLOS
<b>Temperatura y humedad</b>	Monitoreo del clima, control ambiental en invernaderos, calefacción y aire acondicionado (sistemas HVAC)	DHT11, DHT22, BME280, SHT31, AM2320
<b>Movimiento y presencia</b>	Sistemas de seguridad, automatización del hogar, control de ambientes de almacén	HC-SR501 (PIR), HC-SR504 (PIR), VL53L0X (ToF)
<b>Luz</b>	Ajuste automático de iluminación, monitoreo de condiciones ambientales, agricultura de precisión	TSL2591, BH1750, VEML7700
<b>Calidad del aire</b>	Detección de contaminantes, monitoreo de CO2 en interiores, control de calidad del aire en fábricas	CCS811, BME680, SGP30
<b>Presión</b>	Monitoreo meteorológico, sistemas hidráulicos, medición de altitud	BMP390, MPL3115A2, MS5611
<b>Proximidad</b>	Detección de objetos, control de accesos, automatización industrial	HC-SR04, VCNL4010, TCS34725
<b>Nivel de agua</b>	Gestión de recursos hídricos, sistemas de riego, prevención de inundaciones	JSN-SR04T (ultrasónico), DFROBOT Gravity (flotador)
<b>Humedad del suelo</b>	Monitoreo de humedad del suelo, gestión de riego en agricultura, control de sistemas de agua	Capacitive Soil Moisture, Grove - Moisture
<b>Aceleración y giroscopios</b>	Monitoreo de vehículos, dispositivos portátiles, robótica	MPU9250, LSM9DS1, BNO055
<b>Corriente y voltaje</b>	Monitoreo de consumo energético, gestión de redes eléctricas, sistemas de energía renovable	INA219, ACS758, ZMPT101B
<b>PH y conductividad</b>	Análisis de calidad del agua, agricultura, procesos industriales	Atlas Scientific pH Sensor, DFRobot Gravity EC Sensor
<b>Gas</b>	Detección de gases peligrosos, monitoreo ambiental, seguridad industrial	MiCS-5524 (CO), MQ-7 (CO), MiCS-5524 (NH3, NOx)
<b>Sonido</b>	Monitoreo de ruido ambiental, detección de intrusos, control de calidad acústica	SPH0645LM4H-B, INMP441
<b>Vibración</b>	Monitoreo de maquinaria, detección de terremotos, análisis estructural	SW-420, ADXL355
<b>Radiación</b>	Detección de radiación UV, monitoreo de radiación nuclear, protección ambiental	GUVA-S12SD, Si1145
<b>Magnético</b>	Detección de campos magnéticos, aplicaciones en brújulas electrónicas, monitoreo industrial	LIS3MDL, MMC5603
<b>Infrarrojo (IR)</b>	Control remoto, detección de presencia, termografía	MLX90640, TSOP38238
<b>Peso</b>	Balanzas electrónicas, monitoreo de peso, control de procesos industriales	HX711 + celda de carga, NAU7802
<b>Frecuencia cardíaca</b>	Monitoreo de salud, dispositivos portátiles, telemedicina	MAX30100, MAX30102
<b>Oxígeno en sangre</b>	Monitoreo de salud, dispositivos portátiles, telemedicina	MAX30100, MAX30102, MAX30105
<b>Presión arterial</b>	Monitoreo de salud, dispositivos médicos, telemedicina	BP01, BPS Cuff Sensors
<b>Temperatura corporal</b>	Monitoreo de salud, dispositivos médicos, telemedicina	MLX90614, TMP117

**Cuadro 13:**  
*Ejemplos de actuadores*

<b>TIPO DE ACTUADOR</b>	<b>CLASIFICACIÓN</b>	<b>APLICACIONES</b>	<b>EJEMPLOS</b>
<b>Motores</b>	Motores eléctricos	Robótica, automatización de ventanas y puertas, vehículos eléctricos	NEMA 17, motor BLDC, motor DC
	Servomotores	Drones, robótica, automatización industrial	MG995, SG90, HS-311
	Motores de Paso	Impresoras 3D, CNC, sistemas de posicionamiento	NEMA 17, NEMA 23, A4988
<b>Válvulas</b>	Válvulas solenoides	Sistemas de riego, HVAC, procesos industriales	Válvula solenoide SMC Válvulas Rain Bird.
	Válvulas de control	Sistemas de calefacción y refrigeración, plantas de energía	Válvulas de Control Honeywell, Válvulas de control Siemens
<b>Relés y contactores</b>	Relés	Sistemas de iluminación, control de motores, electrodomésticos.	SRD-05VDC-SL-C, Omron G2R-1
	Contactores	Control de motores industriales, sistemas de potencia	Schneider Electric LC1, ABB A16-30-10
<b>Dispositivos térmicos</b>	Calentadores	Impresoras 3D, sistemas HVAC, dispositivos médicos	E3D V6 Heater Cartridge, Keenovo Silicone Heater
	Elementos Peltier	Refrigeración y calentamiento preciso en aplicaciones electrónicas y biomédicas	TEC1-12706, TEC1-12715
<b>Dispositivos electromecánicos</b>	Actuadores lineales	Muebles ajustables, maquinaria industrial, aplicaciones automotrices	Firgelli L12, Progressive Automations PA-14
	Actuadores rotativos	Robótica, automatización de puertas y ventanas	LINAK LA23, Hitec HS-805BB
<b>Bombas</b>	Bombas de agua	Sistemas de riego, acuarios, refrigeración	DC 12V Micro Water Pump, Peristaltic Pump
	Bombas de aire	Equipos médicos, sistemas de acuarios, automatización industrial	Tetra Whisper Air Pump, Medo AC0110-A
<b>Dispositivos de iluminación</b>	Leds y controladores de iluminación	Iluminación inteligente, señalización, pantallas	Philips Hue, WS2812B LED Strips
<b>Dispositivos de audio</b>	Altavoces y zumbadores	Sistemas de alarma, notificaciones sonoras, dispositivos de entretenimiento	Piezo Buzzer, JBL Bluetooth Speakers

#### **2.2.2.4 CAPA DE DISPOSITIVOS: MICROCONTROLADORES**

Son dispositivos electrónicos que integran en un solo chip un procesador, memoria y periféricos de entrada/salida, generalmente para las aplicaciones vienen acoplados en placas de desarrollo. Se puede afirmar que son similares a las computadoras convencionales, pero en dimensión y capacidades más pequeñas. Sus principales componentes son (Maulana et al., 2023), (Pfister, 2011):

##### **PROCESADOR**

Presentan un núcleo de procesamiento que ejecuta instrucciones y controla las operaciones del dispositivo, puede variar en potencia y arquitectura dependiendo del fabricante y del modelo del microcontrolador.

##### **MEMORIA**

Los microcontroladores poseen memoria integrada que se utiliza para almacenar programas de software, datos temporales y configuraciones del sistema. Esta memoria puede incluir tanto memoria de programa (flash) como memoria de datos (RAM).

##### **PERIFÉRICOS DE ENTRADA/SALIDA**

Los periféricos de entrada/salida hacen posible la comunicación con dispositivos externos y la interacción con el entorno. Se implementan a través de puertos de diversos tipos como puertos GPIO (entrada/salida de propósito general), puertos UART (comunicación serial), puertos SPI (interfaz serial periférica), puertos I2C (interfaz de comunicación de dos cables), puertos temporizadores, puertos conversores analógico-digitales (ADC) y puertos PWM (modulación por ancho de pulso), entre otros (Waluyo et al., 2023), (Dhumale, 2023).

En el cuadro 14 se presenta algunos ejemplos de microcontroladores, indicando sus principales características, su fabricante, la placa de desarrollo que lo alberga y sus principales usos.

**Cuadro 14:**  
*Ejemplos de microprocesadores*

Microcontrolador	Fabricante	Procesador	Memoria	Puertos	Aplicaciones comunes	Placa de desarrollo
<b>ATmega328</b>	Microchip (Atmel)	8-bit AVR	32 KB Flash, 2 KB RAM	23 I/O, ADC, UART, SPI, I2C	Proyectos básicos de IOT, prototipos, educación	Arduino Uno
<b>ATtiny85</b>	Microchip (Atmel)	8-bit AVR	8 KB Flash, 512 B RAM	6 I/O, ADC, UART, SPI	Proyectos pequeños, controladores simples	Digispark ATtiny85
<b>PIC16F877A</b>	Microchip	8-bit PIC	14 KB Flash, 368 B RAM	33 I/O, ADC, UART, SPI, I2C	Educación, automatización industrial, sistemas de control	PIC16F877A
<b>PIC18F45K22</b>	Microchip	8-bit PIC	32 KB Flash, 1.5 KB RAM	36 I/O, ADC, UART, SPI, I2C	Proyectos avanzados, sistemas de automatización, control de motores	PIC18F45K22
<b>ESP8266</b>	Espressif Systems	32-bit Tensilica L106	4 MB Flash, 160 KB RAM	17 I/O, ADC, UART, SPI, I2C, Wi-Fi	Proyectos IoT, dispositivos conectados, domótica	NodeMCU, WeMos D1 Mini
<b>ESP32</b>	Espressif Systems	32-bit dual-core Tensilica LX6	4 MB Flash, 520 KB RAM	36 I/O, ADC, DAC, UART, SPI, I2C, Wi-Fi, Bluetooth	Proyectos IoT avanzados, dispositivos conectados, aplicaciones en tiempo real	ESP32 DevKit, ESP32-WROOM-32
<b>MSP430</b>	Texas Instruments	16-bit MSP430	128 KB Flash, 10 KB RAM	48 I/O, ADC, UART, SPI, I2C	Dispositivos de bajo consumo, medición y control, sistemas portátiles	MSP430 LaunchPad
<b>Tiva C Series</b>	Texas Instruments	32-bit ARM Cortex-M4	256 KB Flash, 32 KB RAM	43 I/O, ADC, DAC, UART, SPI, I2C, USB	Aplicaciones industriales, robótica, dispositivos de alta performance	Tiva C Series LaunchPad
<b>RP2040</b>	Raspberry Pi	32-bit dual-core ARM Cortex-M0+	2 MB Flash, 264 KB RAM	30 I/O, ADC, UART, SPI, I2C, USB	Raspberry Pi Pico, educación, prototipos de alta performance	Raspberry Pi Pico
<b>MKL25Z</b>	NXP Semiconductors	32-bit ARM Cortex-M0+	128 KB Flash, 16 KB RAM	66 I/O, ADC, DAC, UART, SPI, I2C	Dispositivos de bajo consumo, aplicaciones médicas, dispositivos portátiles	FRDM-KL25Z
<b>ATSAME70</b>	Microchip (Atmel)	32-bit ARM Cortex-M7	2 MB Flash, 384 KB RAM	54 I/O, ADC, DAC, UART, SPI, I2C, USB	Aplicaciones industriales avanzadas, sistemas embebidos de alta performance	ATSAME70-Xplained

### 2.2.2.5 CAPA DE CONECTIVIDAD: CONEXIÓN A INTERNET

En los proyectos basados en IoT, los diferentes dispositivos que intervienen requieren estar conectados a internet para poder realizar la comunicación entre ellos o también con otros sistemas externos. Se hace necesario contar con diferentes tecnologías de conexión a internet, las más usuales en este tipo de proyectos son Wi-Fi, Ethernet, Bluetooth, Zigbee, LTE, Lora WAN, entre otras, la tecnología adecuada depende de las características específicas de la aplicación y de su entorno (Kerimkhulle et al., 2023), (Dhumale et al., 2023).

En el cuadro 15 se listan las principales tecnologías de conexión utilizadas en IoT, indicando su uso, ventajas y desventajas

**Cuadro 15:**  
*Tecnologías de conexión a internet en IoT*

TECNOLOGÍA	USO PRINCIPAL	VENTAJAS	DESVENTAJAS	EJEMPLOS
Wi-Fi	Conectividad de alta velocidad para aplicaciones domésticas e industriales	Alta velocidad, amplio soporte de dispositivos	Consumo de energía relativamente alto, alcance limitado	Dispositivos domésticos inteligentes, cámaras de seguridad
BLUETOOTH	Conexiones de corto alcance para dispositivos portátiles y wearables	Bajo consumo de energía, adecuado para dispositivos móviles	Alcance corto, menor velocidad en comparación con Wi-Fi	Pulseras de actividad, auriculares inalámbricos
ZIGBEE	Redes de baja potencia y alcance medio, especialmente en domótica	Bajo consumo de energía, redes de malla para mayor cobertura	Menor velocidad de datos	Sistemas de iluminación inteligente, sensores de hogar
LORA	Conectividad de largo alcance y baja potencia para aplicaciones de campo amplio	Alcance de hasta 15 km, bajo consumo de energía	Baja tasa de transferencia de datos	Agricultura inteligente, monitorización remota de infraestructuras
NB-IOT	Conexiones de baja potencia y largo alcance utilizando redes celulares	Gran cobertura, buena penetración en interiores, bajo consumo de energía	Requiere infraestructura de red móvil	Medición de agua/gas, seguimiento de activos
ETHERNET	Conectividad de alta velocidad y alta fiabilidad en entornos industriales y de oficina	Alta velocidad, fiabilidad y seguridad	Limitación en la movilidad de los dispositivos	Equipos de red, servidores
MODBUS	Protocolos de comunicación industrial para conectar dispositivos electrónicos	Sencillez y amplia adopción en entornos industriales	Velocidad limitada y sin soporte nativo para encriptación	PLCs, sistemas SCADA

## 2.2.2.6 CAPA DE CONECTIVIDAD: PROTOCOLOS DE COMUNICACIÓN

En IoT, se utilizan diferentes protocolos de comunicación para facilitar el intercambio de datos entre dispositivos. Algunos de los protocolos comunes incluyen MQTT (Message Queuing Telemetry Transport), HTTP (Hypertext Transfer Protocol), CoAP (Constrained Application Protocol), AMQP (Advanced Message Queuing Protocol), entre otros. Estos protocolos proporcionan estándares para la transmisión de datos, aseguran la fiabilidad y la integridad de la comunicación, y pueden adaptarse a diferentes requisitos de ancho de banda y latencia (Joseph y Baseer, 2024), (Xiao, 2018), (Javed, 2016).

En el cuadro 16 se presentan algunos protocolos de comunicación usados en IoT indicando sus usos, ventajas, desventajas y ejemplos de aplicación

**Cuadro 16:**  
*Protocolos de comunicación en IoT*

PROTOCOLO DE COMUNICACIÓN	USO PRINCIPAL	VENTAJAS	DESVENTAJAS	EJEMPLOS DE APLICACIONES
<b>MQTT (Message Queuing Telemetry Transport)</b>	Comunicación entre dispositivos IoT y servidores	Eficiente en el uso del ancho de banda, ligero, soporta QoS (Quality of Service), adecuado para redes de baja potencia	Requiere de un servidor de mensajería (broker), menos adecuado para grandes volúmenes de datos	Domótica, telemetría, monitorización remota
<b>CoAP (Constrained Application Protocol)</b>	Comunicación entre dispositivos IoT con recursos limitados	Ligero, optimizado para dispositivos con recursos limitados, interoperable con HTTP	Menos adecuado para transmisión de grandes cantidades de datos	Sensores y actuadores en redes restringidas, aplicaciones de IoT industriales
<b>HTTP/HTTPS</b>	Comunicación web estándar para dispositivos IoT	Ampliamente compatible, fácil de implementar, soporte para SSL/TLS	Mayor sobrecarga comparado con protocolos más ligeros, menos eficiente para dispositivos con recursos limitados	Aplicaciones web de monitoreo y control, APIs RESTful de IoT
<b>WebSockets</b>	Comunicación bidireccional en tiempo real entre dispositivos y servidores	Baja latencia, conexiones persistentes	Mayor complejidad de implementación, puede generar mayor consumo de recursos	Aplicaciones de tiempo real como chats en línea, paneles de control en tiempo real
<b>AMQP</b>	Mensajería entre aplicaciones y sistemas distribuidos	Fiabilidad y robustez, interoperabilidad, soporte para colas y mensajes persistentes, seguridad integrada	Complejidad de implementación, requiere más recursos que protocolos más simples	Integración de sistemas empresariales y financieros, comercio electrónico.

### **2.2.2.7 CAPA DE COMPUTACION AL BORDE (EDGE COMPUTING)**

Esta capa permite implementar el paradigma de computación al borde que es un tipo de computación distribuida que lleva el procesamiento de datos y la computación más cerca del lugar donde se generan los datos, en lugar de depender de un centro de datos remoto. La computación en el borde reduce la latencia al procesar datos y realizar computaciones cerca del lugar donde se generan los datos. Esto es especialmente importante en aplicaciones que requieren tiempos de respuesta rápidos, como el control industrial, los vehículos autónomos y la realidad aumentada (Sunardi et al., 2023), (Krushnan et al., 2023), (Bakar, 2022).

Otra ventaja de hacer computación al borde es poder reducir la cantidad de datos que necesita ser transmitida a través de la red hacia un centro de datos remoto. Esto puede ayudar a reducir la congestión de la red y a ahorrar ancho de banda, especialmente en redes con ancho de banda limitado o costoso. Se identifican también como ventajas, la mejora en la privacidad y seguridad, mejorar la carga de trabajo permitiendo la escalabilidad y adaptabilidad a entornos de trabajo diversos (Krushnan et al., 2023), (Bakar, 2022), (Prasetyo et al., 2023).

Puede ser implementada en dispositivos de borde de baja potencia, como sensores y dispositivos IoT, así como en dispositivos de borde más potentes, como servidores locales y gateways, computación en el borde no reemplaza completamente la computación en la nube, sino que complementa los servicios en la nube al proporcionar capacidades de procesamiento local y en tiempo real. Esto permite una integración fluida entre la computación en el borde y los servicios en la nube, lo que permite construir sistemas más complejos y sofisticados (Cirani et al., 2019) (Hillar, 2016), (Prasetyo et al., 2023).

La figura 24 muestra de forma resumida las interacciones de la capa de computación al borde con las otras capas de IoT.

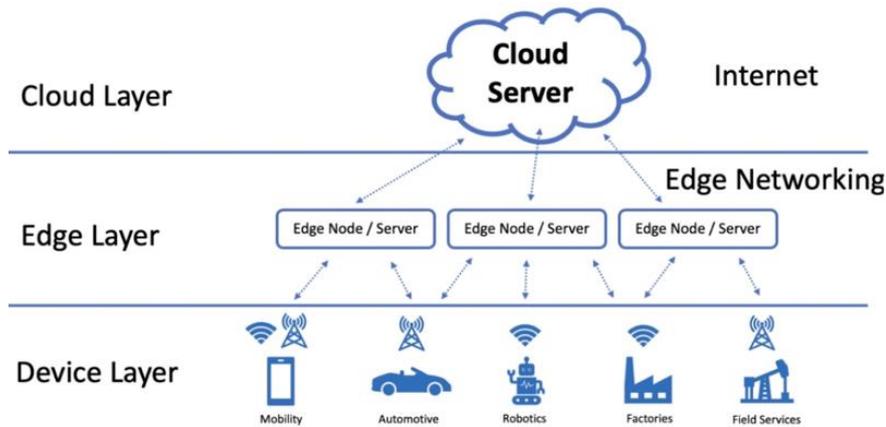


Figura 24:  
*Arquitectura básica de la computación al borde*

La computación en el borde en IoT utiliza una combinación de hardware potente y eficiente, software de gestión y orquestación, protocolos de comunicación optimizados, tecnologías de virtualización y contenedores, inteligencia artificial y bases de datos especializadas para procesar y gestionar datos de manera local. En el cuadro 17 se muestra algunas herramientas que hacen posible el trabajo de la computación al borde (Sunardi et al., 2023), (Dhumale et al., 2023), (Joseph y Baseer, 2024).

### **2.2.2.8 CAPA DE AGREGACIÓN DE DATOS**

Diseñada para consolidar y procesar los datos provenientes de múltiples dispositivos y sensores dispersos en un entorno IoT, actúa como una red de recolección, extrayendo datos crudos desde una diversidad de dispositivos y sensores IoT, que pueden abarcar desde sensores ambientales hasta sistemas de control industrial (Sunardi et al., 2023), (Gupta, 2019), (Raj y Raman, 2017).

## Cuadro 17:

### Herramientas para computación al borde

CATEGORÍA	TECNOLOGÍAS	DESCRIPCIÓN	EJEMPLOS
DISPOSITIVOS DE EDGE COMPUTING	<b>Gateways IoT</b>	Actúan como intermediarios entre dispositivos IoT y la nube, realizando procesamiento y filtrado de datos en el borde.	Raspberry Pi, NVIDIA Jetson
	<b>Edge servers</b>	Servidores ubicados en el borde de la red, capaces de manejar cargas de trabajo más intensivas en procesamiento.	HPE Edgeline, Dell Edge Gateway
SOFTWARE Y PLATAFORMAS DE EDGE COMPUTING	<b>Sistemas operativos y frameworks</b>	Proporcionan una plataforma flexible y personalizable para dispositivos edge.	Linux (Ubuntu, Yocto), Windows IoT
	<b>Plataformas de gestión y orquestación</b>	Facilitan la gestión eficiente de aplicaciones y dispositivos IoT en el borde.	Kubernetes, EdgeX Foundry, AWS IoT Greengrass
TECNOLOGÍAS DE VIRTUALIZACIÓN Y CONTENEDORES	<b>Docker</b>	Utiliza contenedores para aislar aplicaciones, facilitando el despliegue y gestión de software en dispositivos edge.	Eclipse mosquito, Nodod, Gramma
	<b>Kubernetes en el Edge</b>	Orquestación de contenedores, facilitando la gestión de aplicaciones distribuidas en entornos edge.	Kubectll, Helm, Kustomize
INTELIGENCIA ARTIFICIAL Y MACHINE LEARNING EN EL EDGE	<b>Frameworks de AI en el edge</b>	Facilitan el despliegue y ejecución de modelos de IA en dispositivos edge para análisis en tiempo real.	TensorFlow Lite (reconocimiento de imágenes), OpenVINO (análisis de video en tiempo real)
TECNOLOGÍAS DE ALMACENAMIENTO Y BASES DE DATOS	<b>Bases de Datos NoSQL</b>	Diseñadas para manejar grandes volúmenes de datos de eventos, ideales para el almacenamiento de datos de sensores.	InfluxDB (monitoreo de rendimiento)
	<b>Bases de Datos SQL Lite</b>	Base de datos ligera y embebida que no requiere un servidor, utilizada para almacenamiento local de datos en dispositivos edge.	SQLite (almacenamiento local de datos en dispositivos edge, aplicaciones móviles)

Una vez recopilados, los datos pasan por un proceso de refinamiento donde se filtran, se normalizan y se agregan para garantizar su integridad y coherencia, lo que allana el camino para una toma de decisiones precisa. En ciertos escenarios, los datos pueden enriquecerse con información adicional, como metadatos o análisis contextuales, aumentando su valor y utilidad para futuros análisis. Los datos procesados se almacenan en bases de datos optimizadas para acceso rápido y seguro, permitiendo su recuperación para análisis retrospectivos o en tiempo real. La capa de agregación no solo prepara los datos para su almacenamiento, sino que también los transmite de manera eficiente a sistemas de análisis avanzados, reduciendo el tráfico de red innecesario (Kerimkhulle et al., 2023), (Dhumale et al., 2023), (Joseph y Baseer, 2024).

#### **2.2.2.9 CAPA DE PROCESAMIENTO**

La capa de procesamiento es responsable de analizar los datos en tiempo real a medida que llegan desde los dispositivos IoT. Esto permite detectar patrones, tendencias y eventos importantes de manera inmediata. Además del análisis en tiempo real, esta capa también puede procesar grandes volúmenes de datos de forma periódica o por lotes. Esto es útil para análisis retrospectivos, generación de informes y entrenamiento de modelos de machine learning. Los datos originales se transforman y se les aplica una serie de operaciones para convertirlos en información útil. Esto puede incluir la limpieza de datos, la agregación, la normalización y la integración con otras fuentes de datos. Mediante el uso de algoritmos avanzados, la capa de procesamiento puede identificar anomalías en los datos que podrían indicar un mal funcionamiento del dispositivo, una situación de riesgo o una oportunidad de optimización. Basándose en los resultados del análisis,

esta capa puede generar alertas en tiempo real y activar acciones automatizadas, como ajustar la configuración de un dispositivo o enviar una notificación a un operador humano (Hu, 2016), (Bakar, 2022), (Nyakuri et al., 2023).

El cuadro 18 presenta un resumen de las principales herramientas que actualmente se usan para la implementación de esta capa de procesamiento.

#### **2.2.2.10 CAPA DE ALMACENAMIENTO**

Las bases de datos en IoT se utilizan para almacenar grandes cantidades de datos generados por los dispositivos conectados, como lecturas de sensores, eventos de dispositivos y registros de actividad. Estos datos pueden ser estructurados, semiestructurados o no estructurados, y se almacenan de manera eficiente para facilitar su acceso y recuperación (Sunardi et al., 2023), (Nyakuri et al., 2023), (Serpanos y Wolf, 2018), (Cirani et al., 2019).

Las bases de datos en IoT permiten el almacenamiento y la gestión de datos en tiempo real, lo que significa que los datos se pueden procesar y analizar a medida que se generan, permitiendo tomar decisiones rápidas y basadas en datos. Esto es crucial en aplicaciones donde la latencia es crítica, como sistemas de control industrial o sistemas de salud (Sunardi et al., 2023), (Gupta, 2019), (Joseph y Baseer, 2024), (Xiao, 2018).

Las bases de datos en IoT facilitan el análisis de datos para extraer información útil y tomar decisiones inteligentes. Esto puede incluir análisis de tendencias, detección de patrones, predicciones y análisis de anomalías, entre otros. Las bases de datos pueden integrarse con herramientas de análisis avanzado, como el aprendizaje automático y la inteligencia artificial, para obtener información más profunda y perspicaz (Hu, 2016), (Bakar, 2022), (Serpanos y Wolf, 2018).

Las bases de datos en IoT deben ser altamente escalables para manejar el crecimiento exponencial de datos en entornos IoT. Esto significa que deben poder manejar grandes volúmenes de datos y escalar horizontalmente para soportar un gran número de dispositivos conectados y transacciones simultáneas (Bakar, 2022), (Nyakuri et al., 2023), (Serpanos y Wolf, 2018).

Dado que los datos en IoT pueden ser sensibles y privados, las bases de datos deben implementar medidas de seguridad robustas para proteger la confidencialidad, integridad y disponibilidad de los datos. Esto incluye el cifrado de datos, la autenticación de usuarios, el control de acceso y la auditoría de registros (Sunardi et al., 2023), (Gupta, 2019), (Raj y Raman, 2017).

Las bases de datos en IoT deben integrarse fácilmente con sistemas existentes, como sistemas de gestión empresarial (ERP), sistemas de planificación de recursos (ERP), sistemas de control de procesos industriales (SCADA) y sistemas de gestión de la cadena de suministro (SCM), para compartir datos y facilitar la toma de decisiones integrada (Kranenburg, 2008), (Hu, 2016), (Krushnan et al., 2023).

La nube proporciona una plataforma escalable y rentable para almacenar grandes volúmenes de datos generados por dispositivos IoT. Los proveedores de servicios en la nube ofrecen soluciones de almacenamiento diseñadas específicamente para manejar datos estructurados y no estructurados, lo que permite a las organizaciones almacenar y gestionar fácilmente los datos de sus dispositivos IoT (Gupta, 2019), (Hu, 2016).

Docker es una plataforma de software que permite crear, desplegar y gestionar aplicaciones en contenedores. Un contenedor es una unidad estándar de software que empaqueta el código de una aplicación junto con todas sus dependencias,

bibliotecas y configuraciones necesarias para que se ejecute de manera uniforme en cualquier entorno. Esto asegura que las aplicaciones funcionen de manera consistente, independientemente de las diferencias entre los entornos de desarrollo, prueba y producción (Sunardi et al., 2023), (Xiao, 2018), (Javed, 2016).

**Cuadro 18:**  
*Herramientas para la capa de procesamiento IoT*

HERRAMIENTA TECNOLÓGICA	DESCRIPCIÓN	APLICACIONES	EJEMPLOS ACTUALES
<b>Apache Kafka</b>	Plataforma de streaming de eventos que permite la publicación, suscripción, almacenamiento y procesamiento de flujos de datos en tiempo real.	Captura y proceso de flujos de datos en tiempo real.	Monitoreo de actividad en línea, procesamiento de logs en tiempo real.
<b>Apache Spark</b>	Motor de procesamiento de datos rápidos y generales, diseñado para el procesamiento a gran escala.	Análisis de datos en tiempo real y por lotes, aprendizaje automático.	Análisis de grandes volúmenes de datos, procesamiento de datos de sensores IoT.
<b>Flink</b>	Sistema de procesamiento de flujos de datos en tiempo real y por lotes.	Procesamiento continuo de datos, detección de patrones y análisis de eventos.	Detección de fraudes, análisis en tiempo real de datos de dispositivos IoT.
<b>TensorFlow</b>	Biblioteca de código abierto para la inteligencia artificial y el aprendizaje automático.	Entrenamiento y despliegue de modelos de machine learning.	Predicción de fallos en dispositivos IoT, optimización de rutas en logística.
<b>Azure Stream Analytics</b>	Servicio de análisis en tiempo real de Microsoft Azure que permite procesar y analizar flujos de datos en movimiento.	Análisis de datos de telemetría en tiempo real, monitoreo de dispositivos IoT.	Monitoreo en tiempo real de maquinaria industrial, detección de anomalías en datos de sensores.
<b>AWS IoT Analytics</b>	Servicio de Amazon Web Services para el análisis de datos IoT que proporciona herramientas para la ingesta, procesamiento, almacenamiento y análisis de grandes volúmenes de datos.	Limpieza, enriquecimiento, procesamiento y análisis de datos IoT.	Análisis predictivo para mantenimiento preventivo, monitoreo de salud de dispositivos IoT.
<b>Google Cloud Dataflow</b>	Servicio de Google Cloud para el procesamiento de datos en tiempo real y por lotes, basado en Apache Beam.	Procesamiento de datos en tiempo real y por lotes, integración con otras fuentes de datos.	Procesamiento de flujos de datos de sensores IoT, integración con bases de datos de históricos.
<b>Hadoop</b>	Marco de trabajo de código abierto para el almacenamiento y procesamiento de grandes volúmenes de datos mediante un modelo de programación distribuida.	Almacenamiento y análisis de grandes volúmenes de datos, procesamiento por lotes.	Almacenamiento de datos históricos de IoT, análisis retrospectivo de datos de sensores.

Kubernetes es una plataforma de orquestación de contenedores de código abierto que automatiza la implementación, la gestión y el escalado de aplicaciones en contenedores. Fue originalmente desarrollado por Google y ahora es mantenido por la Cloud Native Computing Foundation (CNCF). Kubernetes permite gestionar clústeres de máquinas virtuales o físicas para ejecutar aplicaciones en contenedores de manera eficiente y escalable. El cuadro 19 presenta las principales herramientas utilizadas en esta capa.

**Cuadro 19:**  
*Herramientas de la capa de almacenamiento IoT*

HERRAMIENTA	DESCRIPCIÓN	APLICACIONES	EJEMPLOS
<b>Apache Cassandra</b>	Base de datos NoSQL distribuida diseñada para manejar grandes cantidades de datos en múltiples servidores.	Almacenamiento de grandes volúmenes de datos IoT, análisis de tendencias, detección de patrones.	Monitoreo de dispositivos IoT, gestión de logs y eventos en tiempo real.
<b>MongoDB</b>	Base de datos NoSQL orientada a documentos, flexible y escalable, ideal para datos semiestructurados y no estructurados.	Gestión de datos de sensores, almacen de registros de actividad, análisis en tiempo real.	Aplicaciones de salud, ciudades inteligentes, sistemas de gestión de activos.
<b>InfluxDB</b>	Base de datos de series temporales diseñada para manejar altas cargas de escritura y consultas rápidas de datos temporales.	Monitorización de datos de sensores en tiempo real, análisis de series temporales.	Monitorización industrial, aplicaciones de salud, análisis de rendimiento
<b>Amazon DynamoDB</b>	Base de datos NoSQL completamente gestionada que proporciona rendimiento de baja latencia a cualquier escala.	Almacenamiento, recuperación de datos IoT, integración con otros servicios de AWS.	Aplicaciones de comercio, gestión de dispositivos IoT, monitoreo de eventos en tiempo real.
<b>Azure Cosmos DB</b>	Base de datos distribuida globalmente y completamente gestionada por Microsoft Azure, que soporta múltiples modelos de datos.	Gestión de datos distribuidos globalmente, análisis en tiempo real, integración con servicios de Azure.	Aplicaciones en tiempo real, análisis predictivo, gestión de datos de sensores.
<b>Google Cloud Bigtable</b>	Base de datos NoSQL de alto rendimiento y baja latencia diseñada para manejar grandes cargas de trabajo analíticas y operacionales.	Análisis de grandes volúmenes de datos IoT, procesamiento de datos de series temporales.	Monitorización de infraestructura, análisis de datos de sensores en tiempo real.
<b>TimescaleDB</b>	Base de datos de series temporales basada en PostgreSQL, diseñada para escalabilidad y rendimiento en datos temporales.	Análisis y monitoreo de datos de sensores, almacenamiento de series temporales.	Monitorización de aplicaciones, análisis de datos IoT en tiempo real, gestión de infraestructura.
<b>Kudu (Apache Hadoop)</b>	Almacenamiento de datos para análisis en tiempo real, diseñado para funcionar con Hadoop y sistemas de procesamiento como Apache Impala.	Análisis en tiempo real de grandes volúmenes de datos IoT, integración con ecosistemas Hadoop.	Procesamiento de datos de sensores, análisis de registros, análisis de datos en tiempo real.

### **2.2.2.11 CAPA DE APLICACIÓN**

La nube permite el procesamiento y análisis de datos en escala, lo que permite extraer información valiosa y realizar análisis avanzados en los datos generados por dispositivos IoT. Los servicios en la nube proporcionan capacidades de procesamiento distribuido y herramientas de análisis avanzado, como aprendizaje automático y análisis predictivo, que permiten a las organizaciones descubrir patrones, tendencias y correlaciones en los datos de IoT (Hu, 2016), (Krushnan et al., 2023), (Xiao, 2018).

La nube ofrece una infraestructura altamente escalable que puede crecer y adaptarse según las necesidades cambiantes de una implementación de IoT. Los servicios en la nube permiten escalar verticalmente para aumentar la capacidad de procesamiento y almacenamiento, así como escalar horizontalmente para manejar un mayor número de dispositivos conectados y volúmenes de datos.

Los servicios en la nube ofrecen altos niveles de disponibilidad y redundancia, lo que garantiza que los datos de IoT estén disponibles cuando se necesiten. Los proveedores de servicios en la nube implementan redundancia en sus centros de datos y utilizan técnicas de replicación de datos para garantizar la integridad y disponibilidad de los datos en caso de fallas de hardware o desastres. La nube ofrece características de seguridad avanzadas para proteger los datos de IoT contra amenazas internas y externas. Los proveedores de servicios en la nube implementan medidas de seguridad física y lógica, como cifrado de datos, autenticación de usuarios, control de acceso y auditorías de seguridad, para proteger la seguridad de los datos de IoT (Dhumale et al., 2023), (Joseph y Baseer, 2024), (Xiao, 2018), (Javed, 2016).

En el cuadro 20 se presentan las herramientas IoT en la nube más usadas.

**Cuadro 20:**  
*Herramientas IoT en la nube*

Herramienta IoT en la nube	Descripción	Aplicaciones	Licencia	Herramientas de Analítica e IA	Empresa
<b>AWS IoT Core</b>	Plataforma de Amazon Web Services para conectar dispositivos IoT a la nube de manera segura.	Gestión de dispositivos, recopilación y análisis de datos, integración con otros servicios.	Pago	Amazon SageMaker, AWS IoT Analytics	Amazon
<b>Azure IoT Hub</b>	Servicio en la nube de Microsoft Azure para la conexión, monitoreo y gestión de millones de dispositivos IoT.	Conectividad bidireccional segura, monitoreo y control remoto de dispositivos.	Pago	Azure Machine Learning, Azure Stream Analytics	Microsoft
<b>Google Cloud IoT Core</b>	Plataforma de Google Cloud para la conexión y gestión de dispositivos IoT a escala global.	Ingesta y análisis de datos de dispositivos, gestión de dispositivos IoT.	Pago	Google AI Platform, BigQuery	Google
<b>IBM Watson IoT</b>	Plataforma de IBM que ofrece herramientas avanzadas para la gestión y análisis de datos IoT.	Análisis predictivo, gestión de dispositivos, integración con AI.	Pago	IBM Watson Machine Learning, IBM Watson Analytics	IBM
<b>ThingWorx</b>	Plataforma de PTC para el desarrollo de aplicaciones IoT, con capacidades de conectividad, análisis y visualización.	Desarrollo rápido de aplicaciones IoT, conectividad con múltiples dispositivos y sistemas.	Pago	ThingWorx Analytics, ThingWorx Machine Learning	PTC
<b>EdgeX Foundry</b>	Plataforma de código abierto para la interoperabilidad y gestión de dispositivos en el borde de la red.	Procesamiento de datos en el borde, integración con la nube.	Gratuita (Código Abierto)	No integrada, pero puede conectarse con herramientas de terceros	Fundación Linux
<b>Predix</b>	Plataforma de General Electric diseñada para aplicaciones industriales, ofreciendo análisis avanzado y gestión de activos.	Análisis de datos industriales, gestión de activos, mantenimiento predictivo.	Pago	Predix Machine Learning, Predix Analytics	General Electric
<b>Kaa IoT</b>	Plataforma de código abierto para la implementación de soluciones IoT escalables.	Gestión de dispositivos, análisis de datos, integración con servicios en la nube.	Gratuita (Código Abierto)	No integrada, pero puede conectarse con herramientas de terceros	KaaSoft
<b>Zerynth</b>	Plataforma que facilita el desarrollo de soluciones IoT con soporte para programación en Python y C.	Desarrollo de firmware IoT, conectividad con servicios en la nube, gestión de dispositivos.	Ambas (Gratuita y de Pago)	No integrada, pero puede conectarse con herramientas de terceros	Zerynth
<b>Losant</b>	Plataforma de desarrollo de aplicaciones IoT que permite la creación de soluciones escalables con conectividad en tiempo real.	Conectividad en tiempo real, análisis de datos, gestión de dispositivos.	Pago	Losant Edge Compute, Losant Data Visualization	Losant
<b>ThingSpeak</b>	Plataforma IoT de MathWorks para la recopilación y análisis de datos de sensores en tiempo real.	Análisis de datos, visualización en tiempo real, aplicaciones de monitoreo remoto.	Ambas (Gratuita y de Pago)	MATLAB Analytics, ThingSpeak Data Analysis	MathWorks
<b>Ubidots</b>	Plataforma IoT que permite la integración y visualización de datos de dispositivos conectados.	Gestión de dispositivos, visualización de datos, creación de aplicaciones personalizadas.	Ambas (Gratuita y de Pago)	Ubidots Data Analytics, Ubidots Machine Learning	Ubidots

## **CAPITULO III.**

### **METODOLOGÍA**

#### **3.1 ENFOQUE DE LA INVESTIGACIÓN**

Se utiliza el enfoque cuantitativo, se basa en la recolección y el análisis de datos numéricos y cuantificables producidos por los sensores para medir la variable incertidumbre y examinar las relaciones con los modelos matemáticos basados en lógica difusa planteados.

#### **3.2 MÉTODO DE LA INVESTIGACIÓN**

Se emplea el método experimental ya que se evalúa la incertidumbre en los datos producidos por los sensores, mediante la aplicación de diferentes modelos matemáticos.

#### **3.3 DISEÑO DE LA INVESTIGACIÓN**

Para realizar la investigación, se implementaron los siguientes procesos:

##### **3.3.1 DEFINICIÓN DE VARIABLES**

###### **3.3.1.1 VARIABLE INDEPENDIENTE**

Modelos matemáticos para controlar la incertidumbre en los datos de los sensores.

###### **3.3.1.2 VARIABLE DEPENDIENTE**

Incertidumbre en los datos de los sensores, que puede ser cuantificada mediante métricas específicas.

##### **3.3.2 SELECCIÓN DE MUESTRAS**

Se selecciona un conjunto representativo de sensores, los más representativos desde el punto de vista de las aplicaciones, se trabajó con la incertidumbre los datos asociados para el estudio.

### **3.3.3 IMPLEMENTACIÓN DEL EXPERIMENTO**

Aplicar los modelos matemáticos establecidos para evaluar la incertidumbre de los datos de los sensores, se registró y analizó los resultados en términos de su capacidad para controlar la incertidumbre en los datos.

### **3.3.4 ANÁLISIS DE DATOS**

Se utiliza técnicas estadísticas para comparar los resultados obtenidos, se evalúa la precisión, la fiabilidad y la eficacia en el control de la incertidumbre los datos de los sensores.

### **3.3.5 INTERPRETACIÓN DE RESULTADOS**

Se interpretaron los hallazgos del estudio para sacar conclusiones sobre la eficacia y la utilidad del modelo matemático en la determinación de la incertidumbre en los datos de los sensores, se identificaron posibles aplicaciones y limitaciones y se propone recomendaciones para investigaciones futuras.

## **3.4 POBLACIÓN**

En el contexto de la investigación, la población está compuesta por los diferentes tipos sensores de aplicaciones IoT.

## **3.5 MUESTRA**

Se considera un elemento de la muestra a un sensor que tenga más popularidad en las aplicaciones del control de medio ambiente, se tomó un sensor de captura datos de cada una de las siguientes variables: humedad, temperatura, presión atmosférica, humedad del suelo, gases, proximidades, movimientos.

### **3.6 OPERACIONALIZACIÓN DE LAS VARIABLES**

#### **3.6.1. VARIABLE INDEPENDIENTE: MODELO MATEMÁTICO**

##### **a. DEFINICIÓN CONCEPTUAL**

Un modelo matemático es la representación abstracta y formal de un sistema real o conceptual utilizando herramientas matemáticas. Se construyen para comprender, analizar o predecir el comportamiento de dicho sistema.

##### **b. DEFINICIÓN OPERACIONAL: DIMENSIONES**

Para operacionalizar el modelo matemático se realizan las siguientes etapas, componentes o dimensiones:

###### **b.1 DEFINICIÓN DEL PROBLEMA**

Se identifica con precisión el problema que se quiere abordar identificando los aspectos que se van a modelar matemáticamente.

###### **b.2 FORMULACIÓN DEL MODELO**

Significa traducir el problema del mundo real a términos matemáticos. Se logra identificando con precisión las variables, los parámetros, las constantes, las ecuaciones, las relaciones matemáticas, los algoritmos, etc. necesarios para interpretar la realidad materia del estudio.

###### **b.3 IMPLEMENTACIÓN COMPUTACIONAL**

Consiste en implementar algoritmos que constituyan el modelo computacional adecuado con software de simulación, lenguajes de programación o software especializado.

###### **b.3 CALIBRACIÓN Y VALIDACIÓN**

Ajustar los parámetros del modelo con datos empíricos, validarlo mediante la comparación con datos reales.

#### **b.4 ANÁLISIS Y APLICACIÓN.**

Hacer el análisis, predicciones o simulaciones que proporcionen información útil sobre el problema.

#### **b.5 INTERPRETACIÓN DE RESULTADOS**

Realizar la interpretación de los resultados y relacionarlos con el problema original, obteniendo conclusiones, identificando patrones o tendencias, y luego tomar decisiones basadas en el análisis realizado.

#### **c. DEFINICIÓN OPERACIONAL: INDICADORES**

Para evaluar el desempeño, precisión o utilidad del modelo se utilizan los siguientes indicadores

##### **c.1 AJUSTE A DATOS EMPÍRICOS**

Capacidad del modelo para ajustarse a datos empíricos o experimentales, usando técnicas estadísticas como el coeficiente de determinación (R-cuadrado) o el error cuadrático medio.

##### **c.2 PRECISIÓN DE LAS PREDICCIONES**

Que el modelo pueda predecir el comportamiento futuro del sistema, comparando las predicciones del modelo con datos observados en un período de tiempo posterior al que se utilizó para calibrar el modelo.

##### **c.3 ESTABILIDAD**

Es la capacidad del modelo para mantener su comportamiento a lo largo del tiempo o bajo diferentes condiciones.

##### **c.4 SENSIBILIDAD**

Observar cómo cambian las predicciones del modelo en respuesta a cambios en los parámetros o condiciones iniciales.

### **c.5 ROBUSTEZ**

Capacidad para funcionar correctamente incluso cuando se enfrenta a condiciones inesperadas o a datos incompletos o ruidosos.

### **c.6 EFICIENCIA COMPUTACIONAL**

La eficiencia computacional se refiere a proporcionar resultados precisos en un tiempo razonable y utilizando recursos computacionales razonables.

## **3.6.2. VARIABLE DEPENDIENTE: INCERTIDUMBRE EN DATOS DE SENSORES DE IOT**

### **d. DEFINICIÓN CONCEPTUAL**

Se refiere a la falta de precisión o confiabilidad de los datos proporcionados por los sensores debido a factores diversos.

### **e. DEFINICIÓN OPERACIONAL: DIMENSIONES**

Se definen las medidas que permitan cuantificar las diferentes dimensiones de la incertidumbre asociadas con los datos recopilados por los sensores. Las dimensiones asociadas son:

#### **e.1 PRECISIÓN DE LAS MEDICIONES**

Medir la discrepancia entre las mediciones de los sensores y un estándar de referencia conocido, por ejemplo, calcular el error medio, la desviación estándar de los errores, o el porcentaje de mediciones dentro de un cierto margen de error aceptable.

#### **e.2 ESTABILIDAD DE LOS SENSORES**

Calcular la variabilidad de las mediciones de los sensores en diferentes momentos o bajo diferentes condiciones, comparándolo con criterios predefinidos de estabilidad.

### **e.3 FIABILIDAD DE LOS DATOS**

Calcular la frecuencia de errores en las mediciones de los sensores, registrando la ocurrencia durante un período de tiempo determinado calculando la tasa de errores por unidad de tiempo.

### **e.4 CALIBRACIÓN DE LOS SENSORES**

Evaluar la calidad y la frecuencia de la calibración de los sensores, registrando la fecha y los resultados de las calibraciones realizadas y compararlas con los estándares de calibración establecidos.

### **e.5 INTERFERENCIA EXTERNA**

Evaluar susceptibilidad de los sensores a la interferencia externa, realizando pruebas controladas para determinar cómo diferentes fuentes de interferencia afectan las mediciones de los sensores, calculando la magnitud de la interferencia en relación con las mediciones originales.

### **e.6 VARIABILIDAD AMBIENTAL**

Analizar cómo varían las mediciones de los sensores en respuesta a cambios en las condiciones ambientales, registrando las mediciones de los sensores en diferentes condiciones ambientales y calcular la magnitud de la variación en relación con las condiciones ambientales.

## **f. DEFINICIÓN OPERACIONAL: INDICADORES**

Los indicadores de esta variable son:

### **f.1 ERROR MEDIO ABSOLUTO.**

Calcula el promedio de las diferencias absolutas entre las mediciones de los sensores y los valores de referencia conocidos.

## **f.2 DESVIACIÓN ESTÁNDAR DE LOS ERRORES.**

Mide la dispersión de los errores en las mediciones de los sensores, indicando la variabilidad de las mediciones y la consistencia en la precisión de las mediciones.

## **f.3 TASA DE ERRORES**

Calcula la frecuencia de errores en las mediciones de los sensores durante un período de tiempo determinado, proporcionando una medida de la fiabilidad de los datos recopilados por los sensores.

## **f.4 INTERVALO DE CONFIANZA**

Calcula un intervalo de confianza alrededor de las mediciones de los sensores, indicando el rango en el que se espera que se encuentren las mediciones reales con cierto nivel de confianza, dada la incertidumbre asociada con las mediciones.

## **f.5 VARIACIÓN DE LAS MEDICIONES EN DIFERENTES CONDICIONES AMBIENTALES**

Evaluar cómo varían las mediciones de los sensores en respuesta a cambios en las condiciones ambientales.

## **3.7 TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE DATOS**

Se emplearon diversas técnicas para la recolección de datos que permitieron obtener información relevante y precisa sobre los datos de los sensores y la incertidumbre asociada. Estas técnicas fueron:

### **a. FUNCIONAMIENTO DE SENSORES.**

Se utilizaron sensores de Internet of Things para recopilar datos en tiempo real sobre diversas variables ambientales como los que hacen medición de temperatura, humedad, presión, velocidad del viento, nivel de luz, entre otros.

## **b. REGISTRO Y ALMACENAMIENTO DE DATOS**

Se hicieron registros detallados de los datos recopilados por los sensores de IoT, incluyendo la fecha, hora, ubicación y otras metainformaciones relevantes, para garantizar la trazabilidad y la integridad de los datos para su posterior análisis.

## **c. OBSERVACIÓN DIRECTA**

Se hicieron observaciones directas en el campo para recopilar datos sobre el funcionamiento de los sensores de IoT, las condiciones del entorno y otros factores relevantes para la incertidumbre en los datos.

## **d. PRUEBAS PILOTO**

Se realizaron pruebas piloto para probar la eficacia de diferentes modelos matemáticos en la determinación de la incertidumbre en los datos de los sensores de IoT bajo condiciones controladas.

## **3.8 TÉCNICAS DE ANÁLISIS DE LOS RESULTADOS.**

### **3.8.1 ANÁLISIS ESTADÍSTICO DESCRIPTIVO**

Se hizo un análisis que permita resumir y describir las características principales de los datos recopilados con los sensores seleccionados, como la media, la mediana, la desviación estándar, estos estadísticos proporcionan una comprensión inicial de la distribución y la variabilidad de la incertidumbre en los datos reportados por los sensores de IoT.

### **3.8.2 ANÁLISIS DE SENSIBILIDAD**

Se hizo este tipo de análisis para evaluar cómo cambian los resultados del modelo cuando se modifican ciertos parámetros o condiciones, para entender la robustez y la fiabilidad de los modelos matemáticos en la determinación de la incertidumbre en los datos de los sensores.

## CAPITULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1 RESULTADOS

Los resultados de la investigación se han obtenido luego de aplicar la metodología de investigación propuesta en el proyecto que se resume en el diagrama de flujo mostrado en la figura 25.

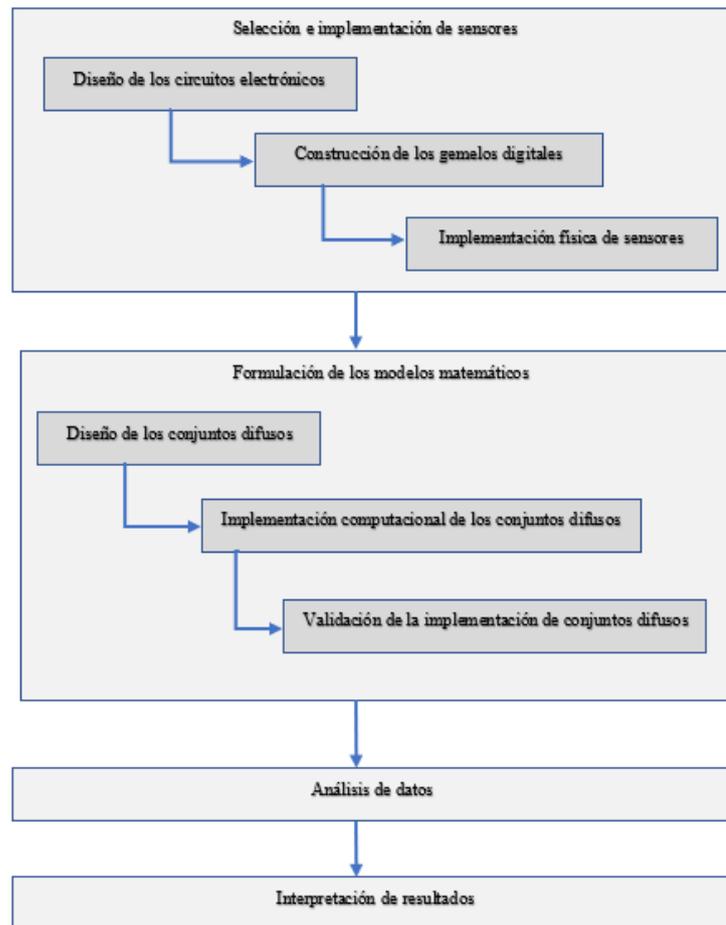


Figura 25:  
*Diagrama de flujo del método de investigación usado*

Se presentan 6 grupos de resultados según la metodológica de la figura 25.

1.- Grupo de resultados de la selección de los sensores de estudio teniendo en cuenta las variables que controlan de acuerdo a un criterio de usabilidad en las aplicaciones, estableciéndose los circuitos electrónicos correspondientes para su implementación, se evidencian en las secciones 4.1.1, 4.1.1.1, 4.1.1.2, 4.1.1.3, 4.1.1.4, el cuadro 21, las figuras 26, 27, 28 y 30.

2.- Grupo de resultados de la implementación física de los sensores para obtener la data de las variables controladas, que se almacenan en la nube para su posterior comparación, se evidencia en las secciones 4.1.1.5, 4.1.1.6, 4.1.1.7, 4.1.1.8, las figuras 30, 31, 32, 33, 34, 35 y 36.

3.- Grupo de resultados de la formulación de los modelos matemáticos, considerando las variables lingüísticas, los conjuntos difusos y los sistemas difusos asociados. Se evidencia en las secciones 4.1.2.1, 4.1.2.2, 4.1.2.3, las figuras 37, 38, 39, los cuadros 22, 23 y 24.

4.- Grupo de resultados de la implementación de los modelos matemáticos en MATLAB implementando las inferencias tipo Mamdani. Se evidencia en las secciones 4.2.2.4, 4.2.2.5, 4.2.2.6, 4.2.2.7, figuras 40, 41, 42 y 43.

5.- Grupo de resultados de la implementación de los modelos matemáticos considerados en los sistemas difusos correspondientes en la computación al borde. Se evidencia en las secciones 4.2.2.8, las figuras 44, 45, 46, 47, 48.

6.- Grupo de resultados de la evaluación de los resultados obtenidos en los grupos de resultados 2 y 4. Se evidencia en la sección 4.1.2.13, en los cuadros 28, 29, 30, 31, 32.

## 4.1.1 SELECCIÓN E IMPLEMENTACIÓN DE SENSORES

### 4.1.1.1 SELECCIÓN DE LOS SENSORES DE ESTUDIO

Para el desarrollo del proyecto se seleccionaron los sensores que más usos tienen en el desarrollo de proyectos basados en IoT. Se ha tenido en cuenta también diferentes variables de medición como la temperatura, la humedad ambiental, la humedad de la tierra, la presencia de gases, la distancia de objetos y la presencia de movimientos. En el cuadro 21, se muestran los sensores seleccionados y la variable que miden.

#### **Cuadro 21:**

*Selección de los sensores utilizados*

N°	Sensor	Variable medida
1	DHT22	Temperatura
2	DHT11	Humedad
3	Soil Moisture	Humedad de la tierra
4	MICCS-5524	Concentración de gases
5	HC-SR40	Distancia
6	HC-SR501	Movimiento

Es importante conocer las especificaciones técnicas de los sensores elegidos, para poder realizar una correcta instalación y uso, estas características son proporcionadas por los fabricantes en documentos denominados “datasheet”. En el Anexo 1, se adjuntan las hojas de datos (datasheets) con las características técnicas de cada uno de estos sensores seleccionados.

#### 4.1.1.2 DISEÑO DE LOS CIRCUITOS DE LOS SENSORES

Los sensores se conectaron con los microcontroladores de forma adecuada formando un correcto circuito electrónico, la figura 26 muestra el esquema de conexión de un circuito de sensor de temperatura. En el Anexo 2 se presentan los circuitos electrónicos de los otros sensores seleccionados.

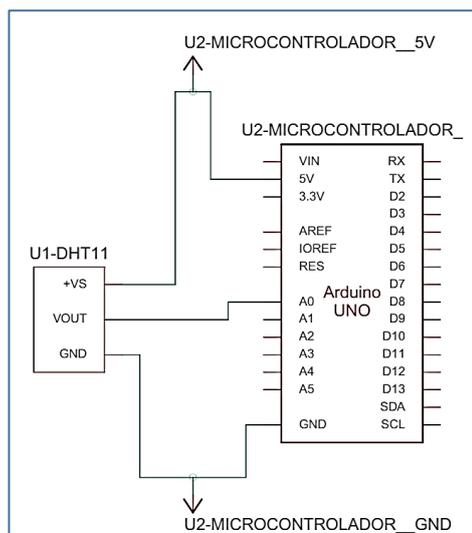


Figura 26:  
*Esquema de un circuito electrónico para el sensor DHT11*

Después de establecer el circuito electrónico, es necesario hacer la conexión lógica del sensor con el microcontrolador, La figura 27 muestra la conexión lógica del circuito electrónico para un sensor de temperatura. En el Anexo 3 se presentan las conexiones lógicas de los otros sensores elegidos.

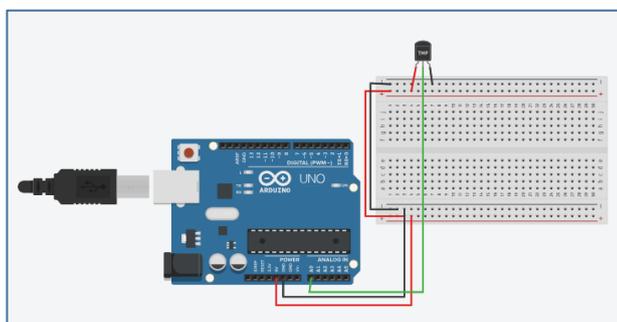


Figura 27:  
*Conexión lógica del sensor de temperatura al microcontrolador*

#### 4.1.1.3 GEMELOS DIGITALES ASOCIADOS A LOS SENSORES

Para simular el funcionamiento correcto de los sensores se utilizaron gemelos digitales (digital twins), usando Tinkercad, Fritzing, Wokwi, Proteus, etc., Estos simuladores permitieron hacer las calibraciones y ajustes y entrenamiento antes de hacer las implementaciones físicas reales. Estos simuladores presentan animaciones gráficas del funcionamiento y proporcionan en general el código para que luego se pueda implementar en un microcontrolador. La figura 28 muestra un gemelo digital para el sensor DHT22 hecho en Wokwi. En el anexo 4, se presentan los gemelos digitales de los otros sensores.

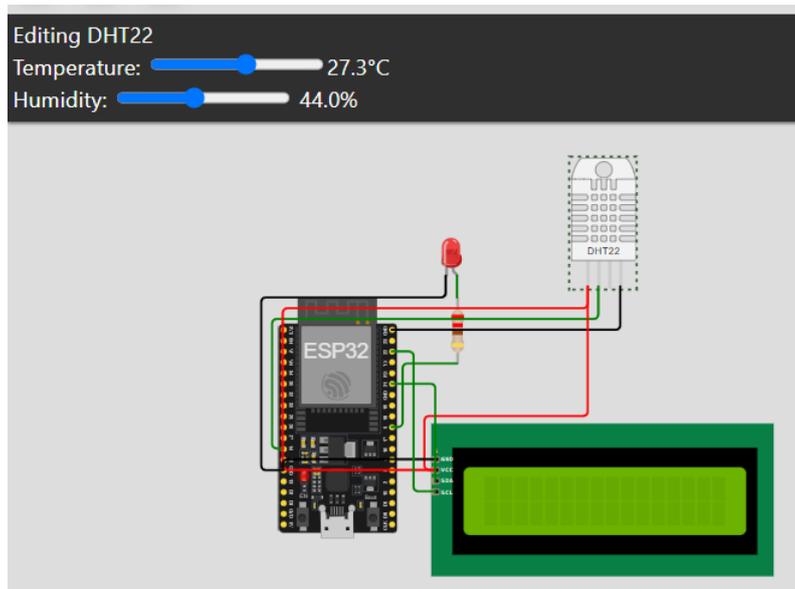
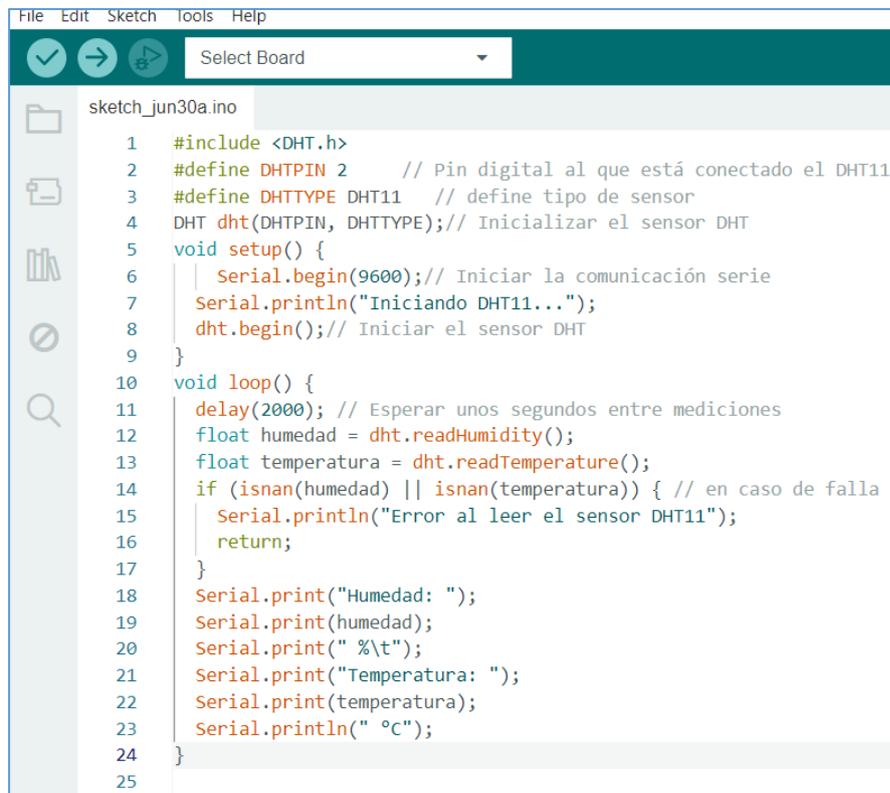


Figura 28:  
*Gemelo digital (digital twin) del sensor DTH22 en Wokwi.*

#### 4.1.1.4 CODIGO ARDUINO GENERADO POR LOS GEMELOS DIGITALES

En la figura 29, se muestra el código en lenguaje para el Arduino IDE del sensor de temperatura y humedad DHT11, generado por su gemelo digital.

The image shows the Arduino IDE interface with a sketch named 'sketch\_jun30a.ino'. The code is as follows:

```
File Edit Sketch Tools Help
Select Board
sketch_jun30a.ino
1 #include <DHT.h>
2 #define DHTPIN 2 // Pin digital al que está conectado el DHT11
3 #define DHTTYPE DHT11 // define tipo de sensor
4 DHT dht(DHTPIN, DHTTYPE); // Inicializar el sensor DHT
5 void setup() {
6   Serial.begin(9600); // Iniciar la comunicación serie
7   Serial.println("Iniciando DHT11...");
8   dht.begin(); // Iniciar el sensor DHT
9 }
10 void loop() {
11   delay(2000); // Esperar unos segundos entre mediciones
12   float humedad = dht.readHumidity();
13   float temperatura = dht.readTemperature();
14   if (isnan(humedad) || isnan(temperatura)) { // en caso de falla
15     Serial.println("Error al leer el sensor DHT11");
16     return;
17   }
18   Serial.print("Humedad: ");
19   Serial.print(humedad);
20   Serial.print(" %\t");
21   Serial.print("Temperatura: ");
22   Serial.print(temperatura);
23   Serial.println(" °C");
24 }
25
```

Figura 29:  
*Código Arduino para el sensor DHT11*

En el anexo 5 se presenta el código Arduino para los otros sensores seleccionados.

#### **4.1.1.5 IMPLEMENTACIÓN FÍSICA DE LOS SENSORES.**

Luego de haber probado con los gemelos digitales, se implementaron los sensores de manera física para poder para poder captar de manera real los datos. En la figura 29, se muestra la implementación física del sensor DHT11, en el anexo 5, se presenta la implementación física de los otros sensores seleccionados.

En el anexo 6 se presenta la implementación física de los otros sensores seleccionados

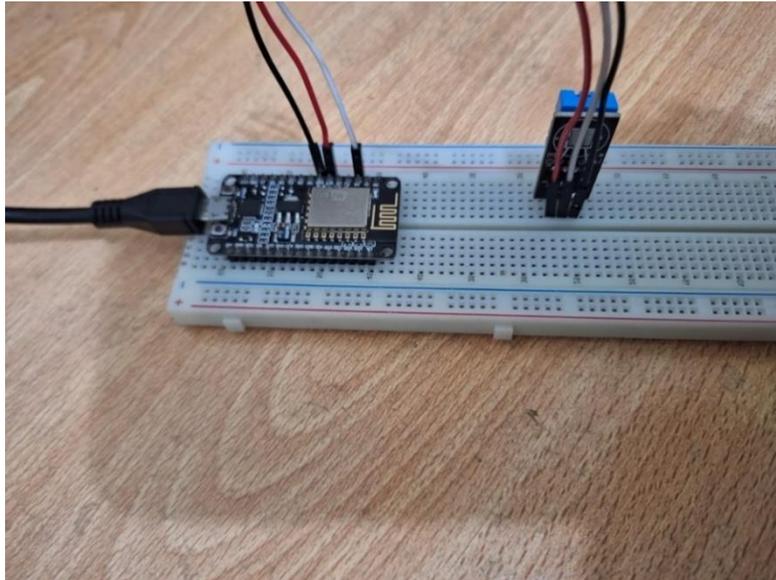


Figura 30:  
*Implementación física del sensor DHT11*

#### 4.1.1.6 ALMACENAMIENTO EN LA NUBE DE LOS DATOS DE LOS SENSORES

Se utiliza Arduino Cloud, una de las plataformas gratuitas de servicios en la nube, para reportar los valores captados por los sensores

Name ↓	Last Value	Last Update
<input type="checkbox"/> dht11_humedad float dht11_humedad;	83	25 Jun 2024 20:20:45
<input type="checkbox"/> dht11_temperatura float dht11_temperatura;	19,4	25 Jun 2024 20:20:45
<input type="checkbox"/> humedadSuelo float humedadSuelo;	47	25 Jun 2024 20:20:44
<input type="checkbox"/> relay bool relay;	false	25 Jun 2024 20:05:17

Figura 31:  
*Almacenamiento de datos en la nube Arduino Cloud*

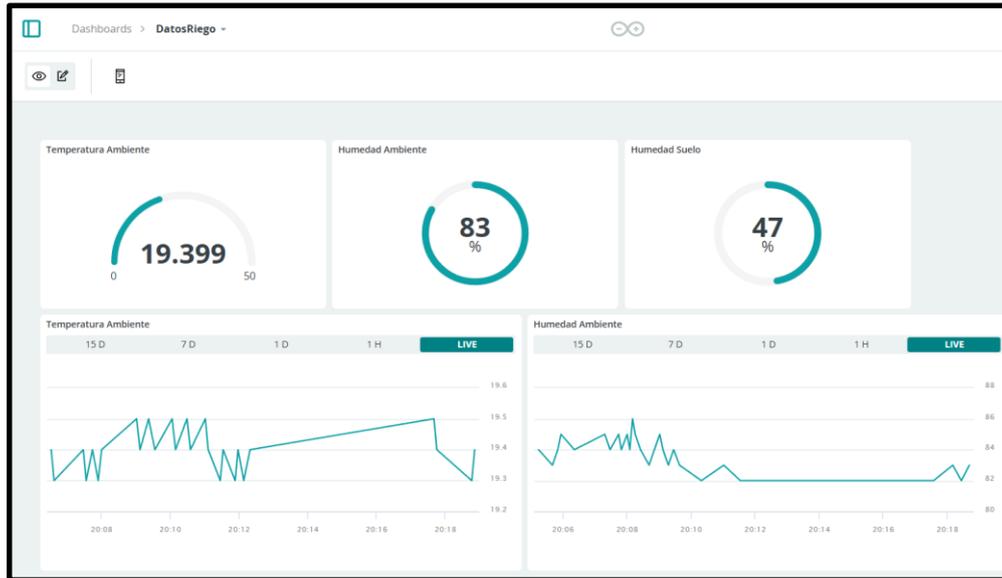


Figura 32:  
Reporte de los datos almacenados en la nube

#### 4.1.1.7 MODELO DE BASE DE DATOS PARA ALMACENAMIENTO DE DATOS

Para el almacenamiento de los datos de los sensores se utilizó una base de datos relacional implementada en el manejador de base de datos HeidiSQL como se muestra en la figura 33.

Nombre	Filas	Tamaño	Creado	Actualizado	Motor	Comentario	Tipo
actuador	12	2,4 KiB	2023-11-10 17:03:...	2023-11-10 17:07:...	MyISAM		Table
actuador1	7	2,3 KiB	2023-09-28 20:48:...	2023-09-29 16:58:...	MyISAM		Table
datos_sensores	2.235	70,8 KiB	2023-10-02 07:10:...	2023-11-06 17:50:...	MyISAM		Table
maquina	6	2,3 KiB	2023-09-29 22:07:...	2023-10-04 22:40:...	MyISAM		Table
sensor	5	2,4 KiB	2023-11-03 19:36:...	2023-11-10 10:19:...	MyISAM		Table
sensor11	0	1,0 KiB	2023-09-29 22:02:...	2023-09-29 22:02:...	MyISAM		Table
tipo_sensor	0	1,0 KiB	2023-10-01 11:50:...	2023-10-01 11:50:...	MyISAM		Table

Figura 33:  
Base de datos de los sensores y actuadores seleccionados

#### 4.1.1.8 PROCESAMIENTO DE DATOS EN UN ENTORNO LOCAL

Para la capa de procesamiento de a nivel local se creó un servidor de archivos que pueda conectarse con la base de datos que almacena los datos de los sensores, se utilizó el entorno de trabajo HTML-kit, para la programación de las páginas y librerías se hizo uso de los lenguajes PHP, JavaScript y HTML. En la figura 34 se muestra el entorno de trabajo, las carpetas de librerías y algunos programas en PHP.

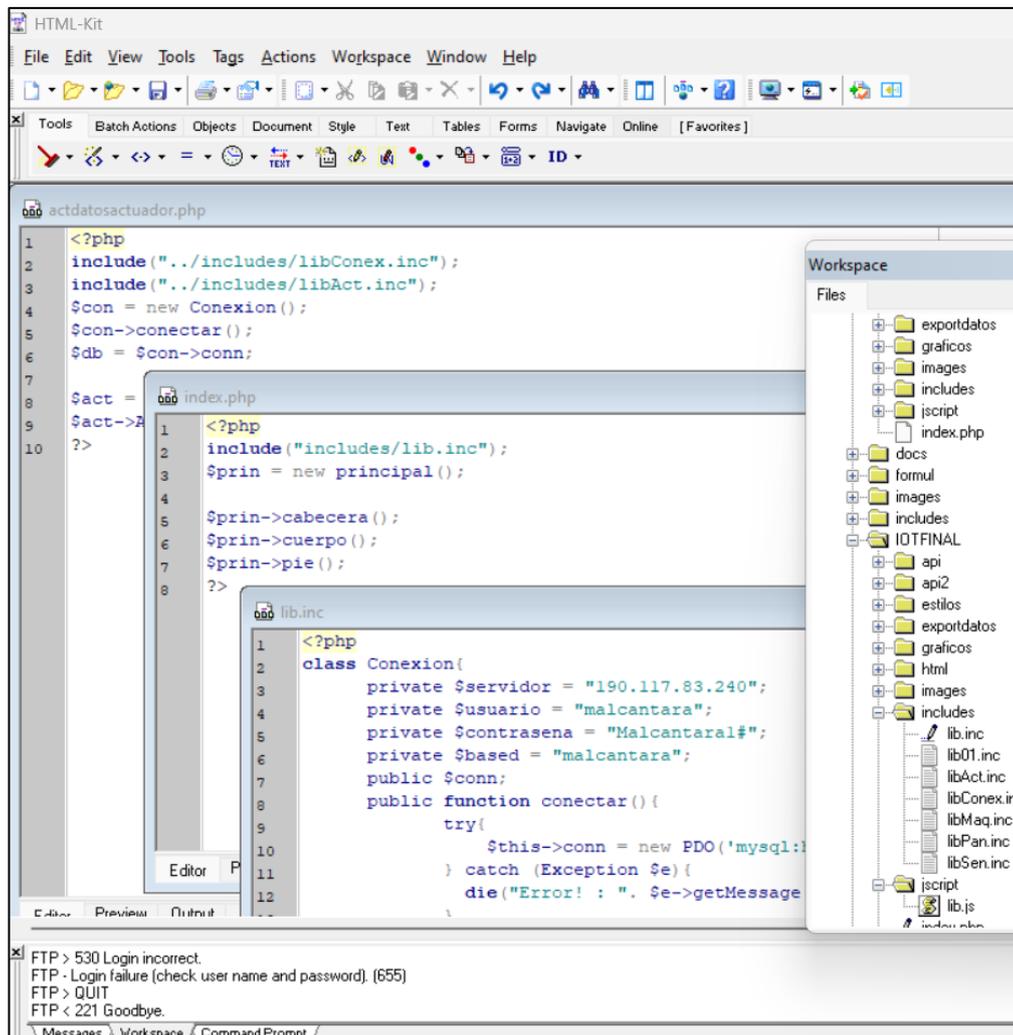


Figura 34:  
*Entorno de trabajo del servidor de archivos*

En las figuras 35 y 36 se muestran las interfases de usuario de la capa de procesamiento local donde se muestran la cantidad de datos captados por los sensores.

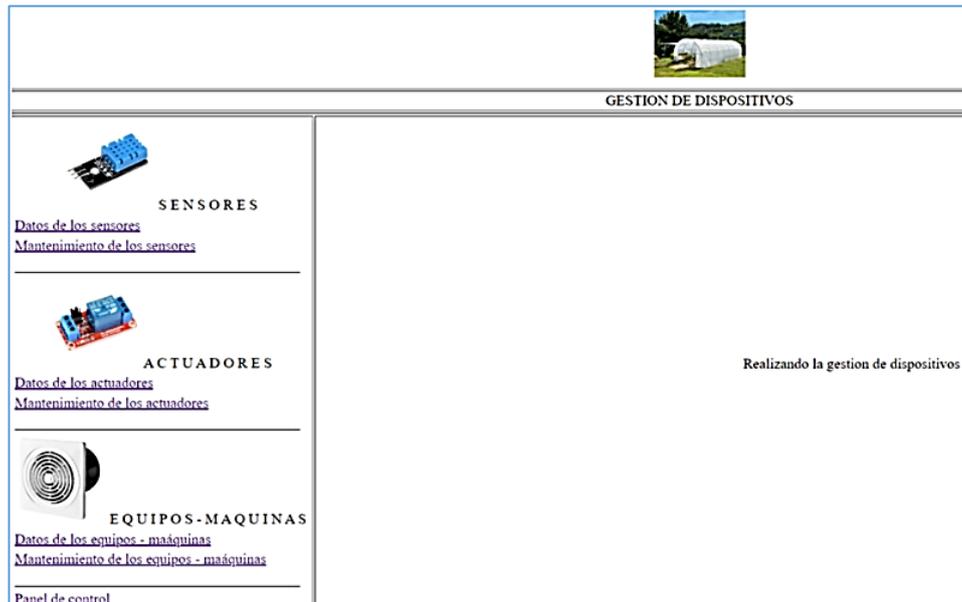


Figura 35:  
*Interfase de usuario para la gestión de sensores, actuadores*

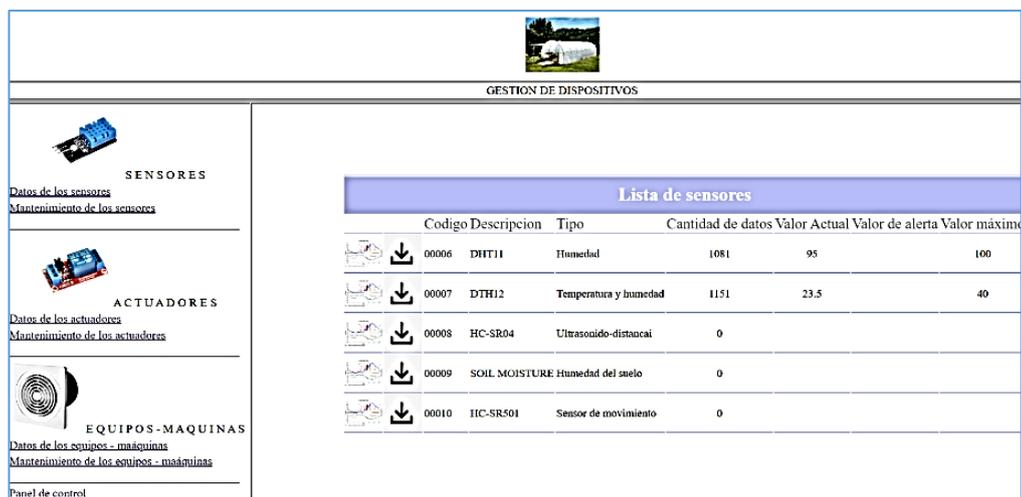


Figura 36:  
*Interfase de usuario con datos captados por los sensores*

#### 4.1.2 FORMULACIÓN DE LOS MODELOS MATEMÁTICOS

Con las condiciones técnicas establecidas en la arquitectura IoT implementada, se crean los modelos matemáticos asociados para el estudio. Para este propósito se establecen cinco sistemas que involucren las variables asociadas a los sensores, los actuadores y máquinas asociadas.

**Cuadro 22:**  
*Sistemas difusos de estudio*

SISTEMA DIFUSO	ENTRADAS	SALIDAS	MAQUINA
<b>Control ambiental</b>	Humedad Temperatura	Velocidad	Sistema de aire acondicionado
<b>Riego automatizado</b>	Humedad del suelo	Angulo de giro	Bomba de agua
<b>Control-polución</b>	Concentración de gas	Velocidad	Extractor de aire
<b>Control de puertas</b>	Distancia	Angulo de giro	Sistema de poleas
<b>Vigilancia</b>	Movimiento	Volumen	Alarma

##### 4.1.2.1 VARIABLES LINGÜÍSTICAS

Se definen las variables lingüísticas y sus valores que intervienen en los sistemas difusos propuestos.

**Cuadro 23:**  
*Variables y valores lingüísticos del estudio*

VARIABLES LINGÜÍSTICAS	SÍMBOLO	UNIVERSO DEL DISCURSO	VALORES LINGÜÍSTICOS		
Humedad	<i>H</i>	Porcentaje	Baja (B)	Normal(N)	Alta(A)
Temperatura	<i>T</i>	Grados	Baja (B)	Media(M)	Alta(A)
Humedad- suelo	<i>S</i>	Porcentaje	Baja (B)	Normal(N)	Alta(A)
Gases	<i>G</i>	Porcentaje	Baja (B)	Normal(N)	Alta(A)
Proximidad	<i>P</i>	Centímetros	Izquierda(I)	Centro(C)	Derecha(D)
Movimiento	<i>M</i>	Centímetros	Izquierda(I)	Centro (C)	Derecha(D)

Formalizando por ejemplo la variable lingüística humedad ( $H$ ), se observa que tiene tres valores lingüísticos bajo ( $B$ ), normal ( $N$ ) y alto ( $A$ ), un valor  $x$  del conjunto universal o universo del discurso  $X$ , es un determinado grado de temperatura en grados Celsius, el conjunto universal es un subconjunto del intervalo  $[i, s]$ , donde los extremos del intervalo, representan los valores mínimo y máximo que puede soportar el sensor según su hoja de datos (datasheet).

$$H = \{B, N, A\}, \quad x \in X, \quad X \subseteq [i, s], \quad i, s \in \mathbb{R}$$

Las otras variables lingüísticas se formalizan de manera similar

$$T = \{B, M, A\}, \quad x \in X, \quad X \subseteq [i, s], \quad i, s \in \mathbb{R}$$

$$S = \{B, N, A\}, \quad x \in X, \quad X \subseteq [i, s], \quad i, s \in \mathbb{R}$$

$$G = \{B, N, A\}, \quad x \in X, \quad X \subseteq [i, s], \quad i, s \in \mathbb{R}$$

$$P = \{I, C, D\}, \quad x \in X, \quad X \subseteq [i, s], \quad i, s \in \mathbb{R}$$

$$M = \{I, C, D\}, \quad x \in X, \quad X \subseteq [i, s], \quad i, s \in \mathbb{R}$$

**Cuadro 24:**

*Variables lingüísticas de los actuadores*

VARIABLES LINGÜÍSTICAS	SÍMBOLO	UNIVERSO DEL DISCURSO	VALORES LINGÜÍSTICOS		
Velocidad (Motor CC)	$V$	Rpm	Baja (B)	Normal(N)	Alta(A)
Rotación (Servomotor)	$R$	Ángulo de giro	Izquierda (I)	Normal(N)	Derecha(D)
Volumen	$W$	Nivel	Bajo(B)	Normal(N)	Alto(A)

**4.1.2.3 CONJUNTOS DIFUSOS ASOCIADOS A LOS SISTEMAS DIFUSOS**

Establecidas las variables lingüísticas se determinan los conjuntos difusos asociados a los sistemas difusos propuestos. De tal forma que para el sistema difuso control ambiental se tiene lo siguiente:

**a. HUMEDAD:**  $H = \{B, N, A\}$ ,  $x \in X$ ,  $X \subseteq [0, 100]$ , Se definen los conjuntos difusos:

$$B = \{(x, \mu_B(x), x \in X), \quad \mu_B(x) = \mu_{Función L}(x; 30,40),$$

$$N = \{(x, \mu_N(x), x \in X), \quad \mu_N(x) = \mu_{Trapezoidal}(x; 30,40,60,70)$$

$$A = \{(x, \mu_A(x), x \in X), \quad \mu_A(x) = \mu_{Gamma}(x; 60,80)$$

La figura 37 muestra las gráficas de las funciones de membresía de los conjuntos difusos de la variable lingüística humedad:

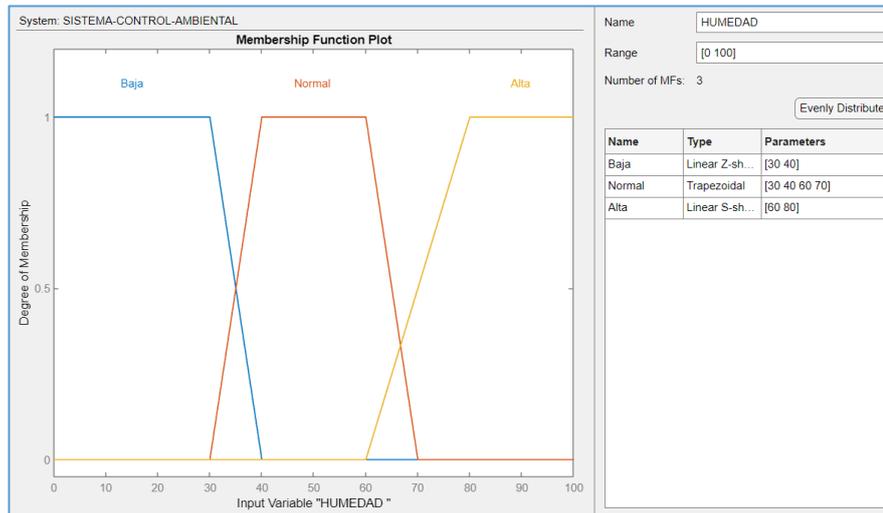


Figura 37:  
*Funciones de membresía de la variable humedad*

**b. TEMPERATURA:**  $T = \{B, M, A\}$ ,  $x \in X$ ,  $X \subseteq [0, 50]$ , Se definen los conjuntos difusos:

$$B = \{(x, \mu_B(x), x \in X), \quad \mu_B(x) = \mu_{Función L}(x; 10,20),$$

$$M = \{(x, \mu_N(x), x \in X), \quad \mu_M(x) = \mu_{Tranangular}(x; 10,25,40)$$

$$A = \{(x, \mu_A(x), x \in X), \quad \mu_A(x) = \mu_{Gamma}(x; 30,40)$$

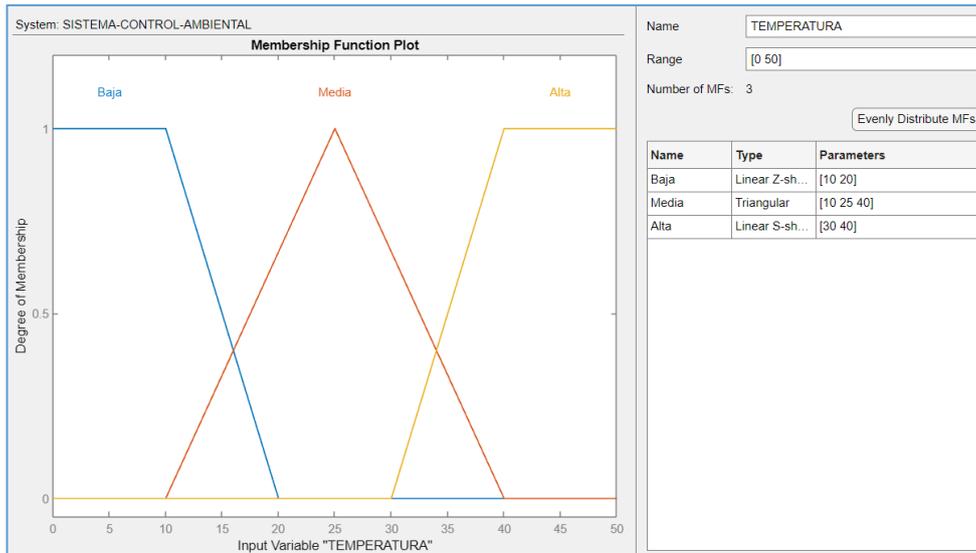


Figura 38:  
Funciones de membresía de la variable temperatura

c. VELOCIDAD:  $V = \{B, N, A\}$ ,  $x \in X$ ,  $X \subseteq [0, 255]$ ,

$$B = \{(x, \mu_B(x), x \in X), \quad \mu_B(x) = \mu_{Triangular}(x; 0, 50, 100),$$

$$N = \{(x, \mu_N(x), x \in X), \quad \mu_N(x) = \mu_{Triangular}(x; 50, 125, 200)$$

$$A = \{(x, \mu_A(x), x \in X), \quad \mu_A(x) = \mu_{Triangular}(x; 150, 200, 250)$$

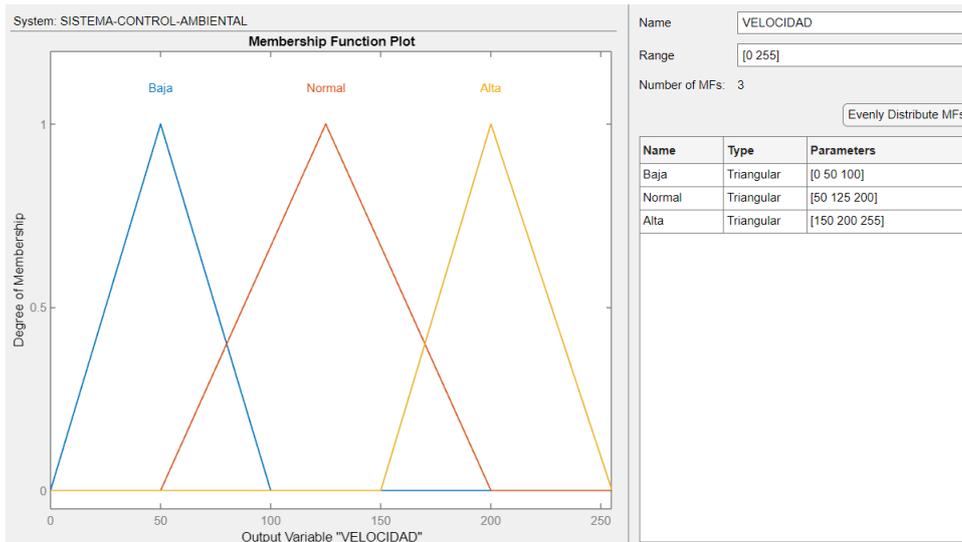


Figura 39:  
Funciones de membresía de la variable velocidad

En el anexo 7. Se presentan los conjuntos difusos de las otras variables de estudio.

#### 4.1.2.4 IMPLEMENTACIÓN DE LOS SISTEMAS DIFUSOS ASOCIADOS EN MATLAB.

En la figura 33 se muestra la implementación del sistema difuso que tiene como entradas la humedad y temperatura y como salida la velocidad del motor que activa un motor de corriente continua. En el anexo 8 se presentan los otros sistemas difusos considerados.

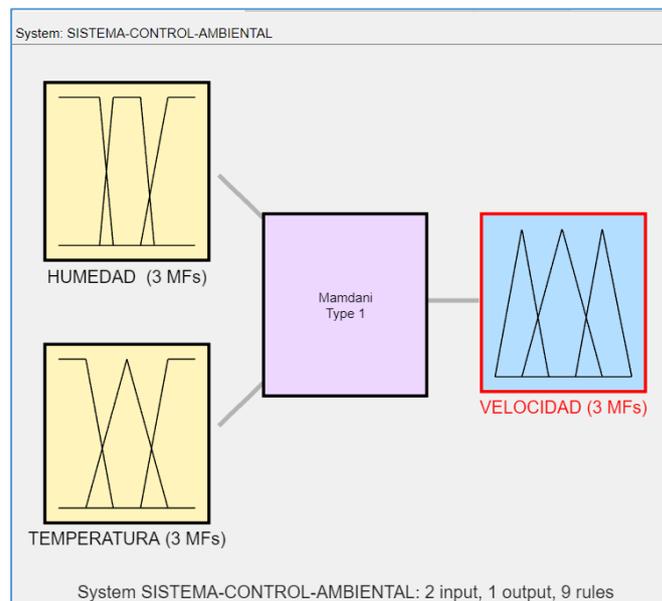


Figura 40:  
*Sistema difuso de control ambiental*

#### 4.1.2.5 DEFINICIÓN DE LAS REGLAS DE INFERENCIA EN MATLAB

En la figura 34 se presentan las 9 reglas que constituyen la base de reglas o conocimiento del sistema difuso control-ambiental. En el anexo 9, se presentan las reglas de inferencia de los otros sistemas difusos considerados.

System: SISTEMA-CONTROL-AMBIENTAL	
<input type="button" value="Add All Possible Rules"/> <input type="button" value="Clear All Rules"/>	
	Rule
1	If HUMEDAD is Baja and TEMPERATURA is Baja then VELOCIDAD is Alta
2	If HUMEDAD is Normal and TEMPERATURA is Baja then VELOCIDAD is Alta
3	If HUMEDAD is Alta and TEMPERATURA is Baja then VELOCIDAD is Normal
4	If HUMEDAD is Baja and TEMPERATURA is Media then VELOCIDAD is Normal
5	If HUMEDAD is Normal and TEMPERATURA is Media then VELOCIDAD is Normal
6	If HUMEDAD is Alta and TEMPERATURA is Media then VELOCIDAD is Normal
7	If HUMEDAD is Baja and TEMPERATURA is Alta then VELOCIDAD is Baja
8	If HUMEDAD is Normal and TEMPERATURA is Alta then VELOCIDAD is Baja
9	If HUMEDAD is Alta and TEMPERATURA is Alta then VELOCIDAD is Baja

Figura 41:  
*Reglas de inferencia del sistema difuso control ambiental*

#### 4.1.2.6 SIMULACIÓN DEL SISTEMA DIFUSO CONTROL-AMBIENTAL

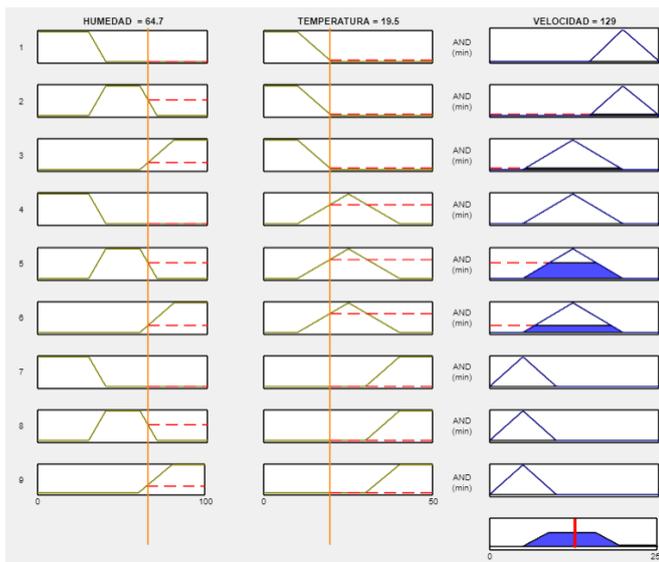


Figura 42:  
*Simulación del sistema Control-Ambiental*

El motor de inferencias del sistema difuso fuzzifica las entradas humedad=64.7, temperatura=19.5 y reporta como velocidad del motor de corriente continua 129 rpm. En la figura 36 se muestra otra simulación del sistema. En el anexo 10, se presentan las simulaciones de los otros sistemas difusos considerados,

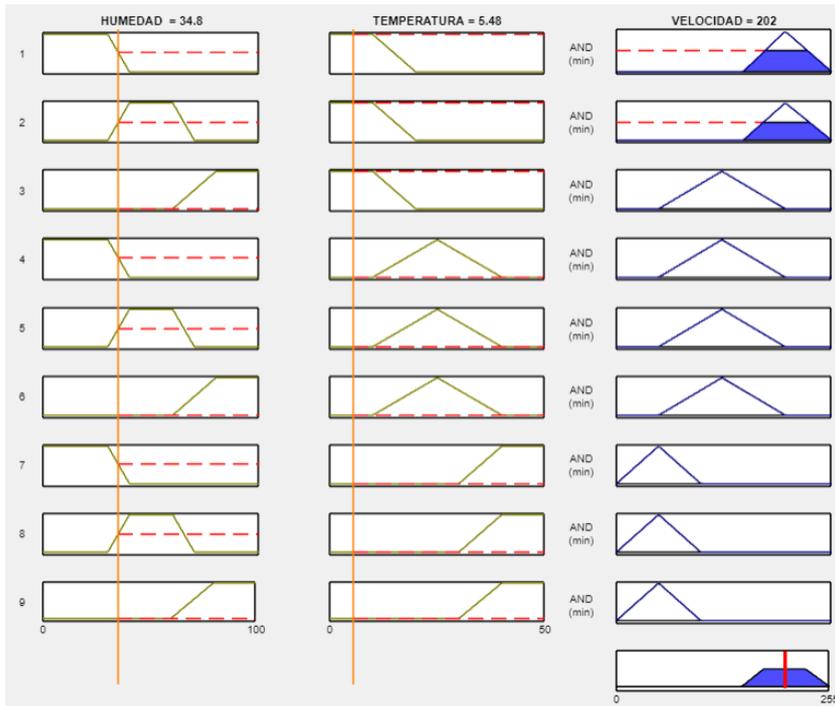


Figura 43:  
*Una segunda simulación del sistema Control-Ambiental*

#### 4.1.2.7 CODIGO ARDUINO DE LAS FUNCIONES DE PERTENENCIA

Se programaron el código Arduino de las funciones de membresía relacionadas con los conjuntos difusos de estudio.

##### a. FUNCIÓN TRIANGULAR

```
float triangular(float x, float a, float b, float c) {
  if (x <= a || x >= c) { return 0.0; }
  else if (x > a && x <= b) { return (x - a) / (b - a); }
  else if (x > b && x < c) { return (c - x) / (c - b); }
}
```

##### b. FUNCIÓN L

```
float funcionL(float x, float a, float b) {
  if (x <= a) { return 1.0; }
  else if (x > a && x < b) { return (b - x) / (b - a); }
  else if (x >= b) { return 0.0; }
}
```

### c. FUNCIÓN GAMMA

```
float funcionGamma(float x, float a, float b) {  
    if (x <= a) { return 0.0; }  
    else if (x > a && x < b) { return (x - a) / (b - a); }  
    else if (x >= b) { return 1.0; }  
}
```

### d. FUNCIÓN TRAPEZOIDAL

```
float funcionTrapezoidal(float x, float a, float b, float c, float d) {  
    if (x <= a || x >= d) return 0.0; }  
    else if (x > a && x <= b) { return (x - a) / (b - a); }  
    else if (x > b && x <= c) { return 1.0; }  
    else if (x > c && x < d) { return (d - x) / (d - c); }  
    else if (x >= d) { return 0.0; }  
}
```

## 4.1.2.8 IMPLEMENTACIÓN EN LA CAPA DE BORDE DE LAS

### FUNCIONES DE PERTENENCIA

Para hacer la implementación de las funciones de membresía en la capa de borde es necesario crear una librería de funciones que pueda incluirse en el código Arduino dentro del entorno de trabajo Arduino IDE para que con esta herramienta se pueda grabar en la memoria del microcontrolador. Para tal propósito se debe crear las librerías necesarias que se deben grabar dentro de la carpeta de trabajo según la instalación del Arduino IDE, por ejemplo, generalmente se crea la carpeta `../Documentos/Arduino/librares/FuncionesMembresia`.

Dentro de esta carpeta se grabarán los siguientes programas:

- a. El archivo “FuncionesMembresia.h” que establece la cabecera de las funciones de membresía que fue creado en el entorno de trabajo Dev-c++ para el lenguaje C. En la figura 44 se muestra este código.

```

1  #ifndef FUNCIONESMEMBRESIA_H
2  #define FUNCIONESMEMBRESIA_H
3  #include <Arduino.h>
4  class FuncionesMembresia {
5      public:
6          static float triangular (float x, float a, float b, float c);
7          static float trapezoidal (float x, float a, float b, float c, float d);
8          static float funcionL (float x, float a, float b);
9          static float funcionGamma (float x, float a, float b);
10 };
11 #endif
12

```

Figura 44:

*Código de FuncionesMembresia.h*

- b. El archivo “FuncionesMembresia.cpp” que implementa el código de las funciones de membresía. La figura 45 muestra este código

```

1  #include "FuncionesMembresia.h"
2  //..... Función de membresía triangular.....
3  float FuncionesMembresia::triangular(float x, float a, float b, float c) {
4      if (x <= a || x >= c) return 0.0;
5      else if (x == b) return 1.0;
6      else if (x < b) return (x - a) / (b - a);
7      else return (c - x) / (c - b);
8  }
9  // .....Función de membresía trapezoidal .....
10 float FuncionesMembresia::trapezoidal (float x, float a, float b, float c, float d) {
11     if (x <= a || x >= d) return 0.0;
12     else if (x >= b && x <= c) return 1.0;
13     else if (x > a && x < b) return (x - a) / (b - a);
14     else return (d - x) / (d - c);
15 }
16 // ..... Función de membresía funcionL .....
17 float FuncionesMembresia::funcionL(float x, float a, float b) {
18     if (x <= a) { return 1.0; }
19     else if (x > a && x < b) { return (b - x) / (b - a); }
20     else if (x >= b) { return 0.0; }
21 }
22 // .....Función de membresía funcionGamma .....
23 float FuncionesMembresia::funcionGamma(float x, float a, float b) {
24     if (x <= a) { return 0.0; }
25     else if (x > a && x < b) { return (x - a) / (b - a); }
26     else if (x >= b) { return 1.0; }
27 }
28

```

Figura 45:

*Código en C de las funciones de membresía*

#### 4.1.2.9 IMPLEMENTACIÓN DE LOS SISTEMAS DIFUSOS

Para la implementación de los sistemas difusos en la capa del borde, se hace necesario crear los carpetas y archivos requeridos para:

- a. Definir las funciones de membresía para las entradas y la salida.

Se comienza haciendo otro archivo de cabecera que se utilizará el sistema difuso al que al que se nombra como “InferenciaTipoMamdani.h” como el código mostrado en la figura 46.

```
1 // InferenciaTipoMamdani.h
2 #ifndef INFERENCIATIPOMAMDANI_H
3 #define INFERENCIATIPOMAMDANI_H
4
5 #include <Arduino.h>
6 #include "FuncionesMembresia.h"
7
8 class Mamdani {
9 public:
10     Mamdani();
11
12     // Métodos para establecer funciones de membresía de entrada y salida
13     void Entradas(float (*input1MF[])(float), int numMF1, float (*input2MF[])(float), int numMF2);
14     void Salidas (float (*outputMF[])(float), int numOutputMF);
15
16     // Método para definir las reglas difusas
17     void Reglas(int reglasDifusas[][3], int numReglas);
18
19     // Método para realizar la inferencia difusa
20     float inferencia(float input1, float input2);
21
22 private:
23     // Punteros a las funciones de membresía
24     float (**input1MF)(float);
25     int numMF1;
26     float (**input2MF)(float);
27     int numMF2;
28     float (**outputMF)(float);
29     int numOutputMF;
30
31     // Matriz de reglas difusas
32     int (*reglasDifusas)[3];
33     int numReglas;
34
35     // Método para defuzzificar usando el método del centroide
36     float defuzzificacion(float outputMFValues[]);
37 };
38 #endif
```

Figura 46:  
Código del archivo de cabecera *InferenciaTipoMamdani*

Luego definimos el archivo en C de la implementación de la inferencia de Mamdani. La implementación incluye las siguientes tareas:

- b. Definir un conjunto de reglas difusas.
- c. Aplicar la inferencia difusa para obtener una salida difusa.
- d. Defuzzificar la salida difusa para obtener un valor numérico nítido.

El código para hacer esta implementación utiliza las funciones de cabecera definidas anteriormente, en la figura 47 se muestra dicho código.

```

1 // InferenciaMamdani.cpp
2 #include "InferenciaTipoMamdani.h"
3 InferenciaMamdani::InferenciaMamdani() {}
4
5 // Configuración de funciones de membresía de entrada
6 void InferenciaMamdani::Entradas(float (*input1MF[])(float), int numMF1,
7     float (*input2MF[])(float), int numMF2) {
8     this->input1MF = input1MF;
9     this->numMF1 = numMF1;
10    this->input2MF = input2MF;
11    this->numMF2 = numMF2;
12 }
13
14 // Configuración de funciones de membresía de salida
15 void InferenciaMamdani::Salidas(float (*outputMF[])(float), int numOutputMF) {
16     this->outputMF = outputMF;
17     this->numOutputMF = numOutputMF;
18 }
19
20 // Configuración de las reglas difusas
21 void InferenciaMamdani::reglasIni(int reglas[][3], int numReglas) {
22     this->reglas = reglas;
23     this->numReglas = numReglas;
24 }
25
26 // Proceso de inferencia difusa
27 float InferenciaMamdani::inferencia(float input1, float input2) {
28     float outputMFValues[numOutputMF] = {0};
29
30     for (int i = 0; i < numReglas; i++) {
31         int input1Index = reglas[i][0];
32         int input2Index = reglas[i][1];
33         int outputIndex = reglas[i][2];
34
35         float input1Value = input1MF[input1Index](input1);
36         float input2Value = input2MF[input2Index](input2);
37
38         float ruleStrength = min(input1Value, input2Value);
39         outputMFValues[outputIndex] = max(outputMFValues[outputIndex], ruleStrength);
40     }
41
42     return defuzzificacion(outputMFValues);
43 }
44
45 // Desfuzzificación usando el método del centroide
46 float InferenciaMamdani::defuzzificacion(float outputMFValues[]) {
47     float numerador = 0;
48     float denominador = 0;
49
50     for (int i = 0; i < numOutputMF; i++) {
51         for (float x = 0; x <= 1.0; x += 0.01) { // Se asume que el dominio es [0, 1]
52             float membershipValue = min(outputMF[i](x), outputMFValues[i]);
53             numerador += x * membershipValue;
54             denominador += membershipValue;
55         }
56     }
57
58     return (denominador == 0) ? 0 : numerador / denominador;
59 }

```

Figura 47:  
Código de la implementación de la inferencia tipo Mamdani

Con la programación lista definida en los programas y librerías lo cargamos en el Arduino IDE para poder grabarlo en el microcontrolador. En la figura 48 se presenta el código de la implementación del sistema difuso control ambiental en Arduino IDE.

```

1  #include <inferenciaTipoMamdani.h>
2  // Definir funciones del sistema control - ambiental
3  // humedad
4  float humedadBaja(float x) {
5      return FuncionesMembresia::funcionl(x, 30, 40);}
6  float humedadNormal(float x) {
7      return FuncionesMembresia::funcionTrapezoidal(x, 30, 40, 60, 70);}
8  float humedadAlta(float x) {
9      return FuncionesMembresia::return FuncionesMembresia::triangular(x, 10, 25, 40);}
0  float temperaturaAlta(float x) {
1      return FuncionesMembresia::funcionGamma(x, 30, 40);}
2  // velocidad
3  float velocidadBaja(float x) {
4      return FuncionesMembresia::triangular(x, 0, 50, 100);}
5  float velocidadNormal(float x) {
6      return FuncionesMembresia::triangular(x, 50, 125, 200);}
7  float velocidadAlta(float x) {
8      return FuncionesMembresia::triangular(x, 150, 200, 225);}
9  Mamdani sistemaDifusc;
0  void setup() {
1      Serial.begin(9600);
2      // Definir las funciones de membresia de entrada
3      float (*input1MF[])(float) = {humedadBaja, humedadNormal, humedadAlta};
4      float (*input2MF[])(float) = {temperaturaBaja, temperaturaMedia, temperaturaAlta};
5      sistemaDifusc.Entradas(input1MF, 3, input2MF, 3);
6      // Definir las funciones de membresia de salida
7      float (*outputMF[])(float) = {velocidadBaja, velocidadNormal, velocidadAlta};
8      sistemaDifusc.Salidas(outputMF, 3);
9      // Definir las reglas difusas: {input1, input2, velocidad}
0      int rules[9][3] = {
1          {0, 0, 2}, // Si humedad es Baja y temperatura es Baja, entonces velocidad es Alta
2          {0, 2, 0}, // Si humedad es Baja y temperatura es Alta, entonces velocidad es Baja
3          {1, 0, 0}, // Si humedad es Alta y temperatura es Baja, entonces velocidad es Baja
4          {1, 1, 1}, // Si humedad es Alta y temperatura es Alta, entonces velocidad es Alta
5          {1, 0, 2},
6          {2, 0, 1},
7          {0, 1, 1},
8          {1, 1, 1},
9          {2, 1, 1}
0      };
1      sistemaDifusc.reglas(rules, 9);
2  }
3  void loop() {
4      float humedad = analogRead(A0) / 1023.0; // Lectura normalizada de 0 a 1
5      float temperatura = analogRead(A1) / 1023.0; // Lectura normalizada de 0 a 1
6      float velocidad = sistemaDifusc.inferencia(humedad, temperatura);
7      Serial.print("humedad: ");
8      Serial.print(humedad);
9      Serial.print(" | temperatura: ");
0      Serial.print(temperatura);
1      Serial.print(" | velocidad: ");
2      Serial.println(velocidad);
3      delay(1000);
4  }

```

Figura 48:  
Sistema difuso control ambiental en Arduino

#### 4.1.2.10 MODELOS MATEMÁTICOS PROPUESTOS

En el cuadro 25 se resume la formulación de los modelos matemáticos propuestos para el control de la incertidumbre de los diferentes sensores seleccionados.

**Cuadro 25:**  
*Características de los modelos matemáticos propuestos*

NUMERO	1	2	3	4	5	6
MODELO MATEMATICO	Modelo Temperatura	Modelo Humedad	Modelo Humedad Suelo	Modelo Gases	Modelo Distancia	Modelo Movimiento
ABREVIATURA	MMT	MMH	MMS	MMG	MMD	MMM
DEFINICION	Propuestas en base a lógica difusa para gestionar la incertidumbre en el sensor DHT22	Propuestas en base a lógica difusa para gestionar la incertidumbre en el sensor DHT11	Propuestas en base a lógica difusa para gestionar la incertidumbre en el sensor SOIL MOISTURE	Propuestas en base a lógica difusa para gestionar la incertidumbre en el sensor MICCS-554	Propuestas en base a lógica difusa para gestionar la incertidumbre en el sensor HC-SR40	Propuestas en base a lógica difusa para gestionar la incertidumbre en el sensor HC-SR501
VARIABLE DE ENTRADA	Temperatura	Humedad	Humedad del suelo	Concentrado de gases	Distancia	Presencia de movimiento
VARIABLE LINGÜÍSTICA DE ENTRADA	Temperatura	Humedad	HumedSuelo	Gases	Proximidad	Movimiento
CONJUNTO REFERENCIAL DE ENTRADA	Grados centígrados de temperatura ambiental	Porcentaje de humedad ambiental	Porcentaje de humedad del suelo	Concentrado de amoniaco	Centímetros	Movimiento
RANGO DE ENTRADA	0-50	0-100	0-100	0-100	0-700	V, F
VARIABLE DE SALIDA	Velocidad del motor de corriente continua	Velocidad del motor de corriente continua	Velocidad del motor de corriente continua	Velocidad del motor de corriente continua	Angulo de rotación del servomotor	Volumen del altavoz
VARIABLE LINGÜÍSTICA DE SALIDA	Velocidad	Velocidad	Velocidad	Velocidad	Angulo	Volumen
CONJUNTO REFERENCIAL DE SALIDA	Revoluciones por minuto	Revoluciones por minuto	Revoluciones por minuto	Revoluciones por minuto	Grados de rotación	Decibelios
RANGO DE SALIDA	0-255	0-256	0-257	0-258	0-90	0-100
MAQUINA ASOCIADA A LA SALIDA	Sistema de aire acondicionado	Sistema de aire acondicionado	Bomba de riego	Campana extractora de gases	Sistema giratorio de apertura de puertas	Altavoz

#### 4.1.2.11 IMPLEMENTACION COMPUTACIONAL DE LOS MODELOS

#### MATEMÁTICOS

En el cuadro 26, se describen las características de software y hardware utilizados para realizar la implementación computacional de los modelos matemáticos propuestos.

#### Cuadro 26:

*Características computacionales de los modelos matemáticos*

NUMERO	1	2	3	4	5	6
MODELO MATEMATICO	MMT	MMH	MMS	MMG	MMD	MMM
SENSOR	DHT22	DHT11	SOIL MOINSTURE	MICCS-5524	HC-SR40	HC-SR501
MICROCONTROLADOR	ESP8266/ESP32	ESP8266/ESP32	ESP8266/ESP32	ESP8266/ESP32	ESP8266/ESP32	ESP8266/ESP32
ACTUADOR (RELE)	SRD-12VDC-SL-C	SRD-12VDC-SL-C	SRD-12VDC-SL-C	JQC-3FF-S-Z	JQC-3FF-S-Z	JQC-3FF-S-Z
SOFTWARE PARA GEMELO DIGITAL	WOKWI	TINKERCAD	TINKERCAD	TINKERCAD	TINKERCAD	TINKERCAD
SOFTWARE PARA CONEXIÓN FISICA	ARDUINO IDE	ARDUINO IDE				
SOFTWARE PARA BASE DE DATOS	HEIDI-SQL	HEIDI-SQL	HEIDI-SQL	HEIDI-SQL	HEIDI-SQL	HEIDI-SQL
SOFTWARE PARA SERVIDOR DE ARCHIVO	HTML, PHP, JAVACRIPT	HTML, PHP, JAVACRIPT				
SOFTWARE PARA EDGE COMPUTING	C++, ARDUINO IDE	C++, ARDUINO IDE				
INFERENCIA	TIPO MAMDANI	TIPO MAMDANI				
SISTEMA DIFUSO	CONTROL AMBIENTAL	CONTROL AMBIENTAL	CONTROL RIEGO	CONTROL DE GASES	AUTOMATIZACION DE PUERTAS	VIGILANCIA
SOFTWARE PARA SIMULACION	MATLAB	MATLAB	MATLAB	MATLAB	MATLAB	MATLAB
LIBRERIAS PARA FUNCIONES DE MEMBRESIA	funcionesmembresia.h	funcionesmembresia.h	funcionesmembresia.h	funcionesmembresia.h	funcionesmembresia.h	funcionesmembresia.h
LIBRERIAS PARA INFERENCIA	inferenciatipomamdani.h	inferenciatipomamdani.h	inferenciatipomamdani.h	inferenciatipomamdani.h	inferenciatipomamdani.h	inferenciatipomamdani.h
PROGRAMA ARDUINO	controlambiental.ino	controlambiental.ino	controlriego.ino	controlgases.ino	controlpuertas.ino	vigilancia.ino

#### 4.1.2.12 PARÁMETROS DE LA INCERTIDUMBRE DE SENSORES

En el cuadro 27 se presentan las características de la incertidumbre de los sensores seleccionados.

**Cuadro 27:**  
*Características de la incertidumbre de los sensores*

SENSOR	DHT22	DHT11	SOIL	MICCS-5524	HC-SR40	HC-SR501
<b>MOINSTURE</b>						
VARIABLE CONTROLADA	Temperatura	Humedad	Humedad del suelo	Concentrado de gas	Distancia	Movimiento
PRECISION	±0.5	±0.1	±0.1	±0.5	±0.5	±0.1
ESTABILIDAD	SI	SI	SI	SI	SI	SI
CALIBRACION	SI	SI	SI	SI	SI	SI
INTERFERENCIA EXTERNA	SI	SI	SI	SI	NO	SI
VARIABILIDAD AMBIENTAL	SI	SI	NO	SI	NO	NO
ERROR MEDIO ABSOLUTO	0.2%	0.2%	0.1%	0.3%	0.1%	0.1%
TASA DE ERRORES	0.05	0.07	0.05	0.05	0.05	0.05
INTERVALO DE CONFIANZA	24.61°C-24.99	24.61°C-24.100	24.25% a 25.75%.	49.25 ppm a 50.75 ppm.	99.25 cm a 100.75 cm.	90.73% a 99.27%.

#### 4.1.2.13 EVALUACIÓN DE LOS MODELOS MATEMÁTICOS

Se presenta a continuación la evaluación de la implementación computacional de los modelos matemáticos propuestos.

**MODELOS MATEMATICOS:** MMT, MMH  
**SISTEMA DIFUSO:** CONTROL AMBIENTAL  
**SENSORES:** DHT22, DHT11  
**PRECISION:** ±0.5  
**ERROR MEDIO:** 0.2%  
**TASA DE ERROR:** 0.05

**Cuadro 28:***Evaluación de los modelos MMT, MMH*

CASO	TEMPERATURA (°C)	HUMEDAD (%)	VELOCIDAD SIMULADA (RPM)	VELOCIDAD EDGE COMPUTING (RPM)
1	6.22	13	202	202
2	12.6	25.1	117	117
3	20.5	21.4	125	125
4	8.6	45.4	202	202
5	23.9	50	125	125
6	30.8	15.5	119	119
7	34.5	58.5	94.3	94.3
8	42.5	18.8	50	50
9	36.8	50.5	89.2	89.2
10	26.1	81.2	125	125

**MODELO MATEMATICO:**

SISTEMA DIFUSO:

SENSORES:

PRECISION:

ERROR MEDIO:

TASA DE ERROR:

**MMS**

CONTROL DE RIEGO

SOIL MOINSTURE

 $\pm 0.5$ 

0.2%

0.05

**Cuadro 29:***Evaluación del modelo MMS*

CASO	HUMEDAD (%)	VELOCIDAD SIMULADA (RPM)	VELOCIDAD EDGE COMPUTING (RPM)
1	20.2	180	180
2	40.3	150	150
3	10.8	220	220
4	50.3	100	100
5	70.1	85	85
6	15	190	190
7	23.8	114.3	114.3
8	78	81.3	81.3
9	64.7	129.2	129.2
10	18.3	192.5	192.5

**MODELO MATEMATICO: MMG**

SISTEMA DIFUSO: CONTROL DE GASES  
 SENSORES: MICCS-5524  
 PRECISION:  $\pm 0.5$   
 ERROR MEDIO: 0.2%  
 TASA DE ERROR: 0.05

**Cuadro 30:***Evaluación del modelo MMG*

CASO	CONCENTRADO DE GAS (PPM)	VELOCIDAD SIMULADA (RPM)	VELOCIDAD EDGE COMPUTING (RPM)
1	80	189.4	189.4
2	50.3	140.5	140.5
3	61.2	160.6	160.6
4	30.2	80.1	80.1
5	10.5	20.4	20.4
6	70.4	170.6	170.6
7	63.7	163.3	163.3
8	41.7	90.5	90.5
9	22.7	40.5	40.5
10	89.6	200.1	200.1

**MODELO MATEMATICO: MMD**

SISTEMA DIFUSO: AUTOMATIZACIÓN DE PUERTAS  
 SENSORES: HC-SR40  
 PRECISION:  $\pm 0.5$   
 ERROR MEDIO: 0.2%  
 TASA DE ERROR: 0.05

**Cuadro 31:***Evaluación del modelo MMD*

CASO	DISTANCIA (CM)	ROTACIÓN SIMULADA (GRADOS)	ROTACION EN EDGE COMPUTING (GRADOS)
1	0	90	90
2	100	90	90
3	200	0	0
4	150	0	0
5	80	90	90
6	70	90	90
7	10	90	90
8	300	0	0
9	700	0	0
10	50	90	90

**MODELO MATEMATICO:** MMM  
**SISTEMA DIFUSO:** VIGILANCIA  
**SENSORES:** HC-SR501  
**PRECISION:**  $\pm 0.5$   
**ERROR MEDIO:** 0.2%  
**TASA DE ERROR:** 0.05

**Cuadro 32:**  
*Evaluación del modelo MMM*

CASO	PRESENCIA DE OBJETOS	VIGILANCIA SIMULADA (RPM)	VIGILANCIA EDGE COMPUTING (RPM)
1	0	0	0
2	1	100	101
3	0	0	0
4	0	0	0
5	1	100	100
6	1	100	102
7	1	100	101
8	0	0	0
9	1	100	100
10	1	100	100

## **4.2 DISCUSIÓN DE LOS RESULTADOS**

De manera resumida se hace un recuento de los resultados obtenidos. Selección de los sensores de humedad, temperatura, humedad del suelo, de gases, de distancia y de movimiento para el estudio. Implementación física de los sensores en sendos circuitos de funcionamiento. Diseño de los modelos matemáticos difusos para tratar la incertidumbre de los datos de los sensores seleccionados estableciendo las variables lingüísticas, conjuntos difusos, funciones de membresía, inferencia y sistemas difusos. Simulación de los sistemas difusos propuestos en base a los modelos matemáticos usando las librerías del paquete “fuzzyLogicDesigner” de MATLAB. Implementación en la nube de los sistemas difusos usando los servicios gratuitos IoT que provee la plataforma Arduino Cloud. Implementación en un entorno local en un sistema que gestiona los sistemas difusos, sus sensores, actuadores y máquinas asociadas, captando la información en una base de datos SQL conectándose con un servidor de archivos que fue diseñado e implementado en HTML-kit. Creación de las librerías y programas necesarios para implementar los sistemas difusos simulados y probados físicamente en los pasos anteriores en la memoria del microcontrolador usando el paradigma de computación al borde. Se comparan finalmente los datos obtenidos por las diversas implementaciones.

### **4.2.1 CONTRASTACIÓN DE LAS HIPÓTESIS**

Los resultados prueban lo afirmado en la hipótesis específica 1, se pueden observar en los cuadros 23 ,24, las figuras 37, 38 que los argumentos teóricos de la lógica

difusa permiten formular modelos matemáticos para el tratamiento de la incertidumbre de los sensores.

La hipótesis específica 2 se contrasta en este trabajo ya que los sistemas difusos propuestos por los modelos matemáticos simulan su funcionamiento usando las librerías de programas de MATLAB tal y conforme se evidencia en las figuras 40, 41, 42 y 43.

Para contrastar la hipótesis específica 3, se ha grabado en la memoria del microcontrolador ESP8266 conectado a los sensores, las librerías y programas en código C y código Arduino para el funcionamiento autónomo de los sistemas difusos, implementando de esta manera el paradigma de computación al borde, estos resultados se pueden observar en las figuras 44, 45, 46, 47, 48.

La contrastación de la hipótesis específica 4 se da al evaluar las características de los modelos matemáticos basados en lógica difusa para gestionar la incertidumbre en los datos de los sensores mediante la comparación de los resultados de los sistemas difusos simulados con los resultados de la implementación física de estos sistemas difusos como se pueden observar en los cuadros 25, 26 ,27 ,28, 29, 30, 31 y 32.

#### **4.2.2 COMPARACIÓN CON RESULTADOS DE OTRAS INVESTIGACIONES**

Este trabajo formaliza y profundiza lo abordado por “Design and Development of Efficient Water Management for Agriculture using IoT and Fuzzy Logic” realizado el 2020 (Dekka et al., 2020) en la India, que solo abordan la problemática que existe en la agricultura respecto al suministro de agua y fertilizantes.

Se amplia y formaliza lo estudiado en el trabajo realizado por (Dakhole et al., 2023) “A smart oyster mushroom cultivation using automatic fuzzy logic controller”, realizado en el departamento de ciencias de la computación e ingeniería de la Presidency University de la India y publicado el año 2023 que abordan el tema del cultivo de hongos tipo ostra que requiere mucha mano de obra y crece en un entorno sensible.

La investigación realizada puede servir para complementar lo realizado en la investigación “An iot-based system for fault detection and diagnosis in solar pv panels” llevada a cabo por (Balakrishnan et al., 2023) que estudian la detección y diagnóstico de fallas en paneles solares fotovoltaicos.

#### **4.2.3 FORTALEZAS EN EL DESARROLLO DEL PROYECTO**

Para el desarrollo del proyecto se han utilizado por un lado la lógica difusa, una teoría sólidamente establecida por Lofti Zadeh y empleada ampliamente en diversas aplicaciones ingenieriles y que a vez ha propiciado el logro de muchas patentes producto de las investigaciones relacionadas con esta teoría y por otro lado la teoría cada vez más evolucionada de Internet of Things, ambas teorías son utilizadas por el autor de esta investigación en su ejercicio profesional por poseer el doble título de Licenciado en matemáticas e Ingeniero de Sistemas.

Otra fortaleza observada en el desarrollo del proyecto es contar con dispositivos electrónicos para Internet of Things cada vez más económicos lo que hace factible la implementación física de los sistemas difusos.

La propuesta gratuita de soluciones en la nube para proyectos de Internet of Things se constituya como una fortaleza que ha permitido hacer el almacenamiento

temporal de los datos para luego hacer las respectivas comparaciones con los datos de las implementaciones físicas de los sistemas difusos.

#### **4.2.4 DEBILIDADES EN EL DESARROLLO DEL PROYECTO**

La principal debilidad en el desarrollo del proyecto lo constituye no haber contado con instituciones que permitan controlar sus sensores de manera real sobre todo en procesos críticos como los de la medicina, navegación aérea, etc. En estas condiciones se exige disponer de algunos permisos especiales como el consentimiento informado de los pacientes o el tratamiento de datos personales, hechos que no fueron considerados en el alcance de este proyecto pero que constituyen retos para futuras investigaciones.

#### **4.2.5 OPORTUNIDADES DE DESARROLLO DE FUTURAS INVESTIGACIONES.**

Una buena oportunidad para el desarrollo de futuras investigaciones consiste en gestionar la incertidumbre en la captura de datos con los sensores de sistemas de Internet of Things relacionados con aspectos críticos de la salud de las personas como la diabetes, hipertensión, osteoporosis, párkinson, etc.

Otra oportunidad es sin duda considerar la naturaleza cambiante de la incertidumbre en la captura de datos de los sensores de sistemas de Internet of Things lo que hace necesario manejar también esa variabilidad con técnicas de inteligencia artificial que propicien el aprendizaje automático como las redes neuronales para poder mejorar la adaptabilidad a los cambios.

## **CAPITULO V.**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 CONCLUSIONES**

La teoría de lógica difusa, provee de instrumentos matemáticos que permiten modelar propuestas matemáticas basadas en las variables lingüísticas, conjuntos difusos, funciones de membresía, operaciones, relaciones e inferencias adaptables a la naturaleza de la incertidumbre de sensores de temperatura, humedad, humedad del suelo, gases, distancia, movimiento y que bien pueden inferirse para otro tipo de sensor dada la naturaleza general de los conceptos difusos.

Los modelos matemáticos basados en lógica difusa son implementados adecuadamente en el paquete de librerías “fuzzyLogicDesigner”, permitiendo realizar la simulación según Mandani o Sugero de manera gráfica e intuitiva, estas implementaciones computacionales permiten evaluar la consistencia, pertinencia y robustez de los modelos propuestos.

El diseño de librerías en c++ y programas en Arduino permiten implementar físicamente las propuestas formuladas en los modelos matemáticos basados en lógica difusa para la gestión de la incertidumbre en datos de sensores y comprobar los resultados según las diversas implementaciones físicas o en la nube.

La evaluación de las características de los modelos matemáticos basados en lógica difusa para gestionar la incertidumbre en los datos de los sensores mediante la comparación de los resultados de los sistemas difusos simulados con los resultados de la implementación física de estos sistemas demuestra la perfecta coincidencia, lo que debe entenderse como que la incertidumbre está controlada.

## **5.2 RECOMENDACIONES**

Se recomienda repetir la experiencia, pero seleccionando otros sensores diferentes a los más usuales, de repente los que intervienen en procesos más críticos como la salud humana, vuelos espaciales, control industrial para corroborar esta propuesta metodológica siempre y cuando se cuente con la información o el acceso a ella contando por supuesto con todos los consentimientos informados necesarios.

Otra propuesta nueva interesante consiste en ampliar esta investigación con herramientas de analítica de datos i algoritmos inteligencia artificial que permitan el aprendizaje autónomo en los sistemas difusos.

## REFERENCIAS BIBLIOGRÁFICAS

- Bakar, S. M. (2022). *Iot-based system using fuzzy logic method for measuring salt quality*. <http://dr.ur.ac.rw/handle/123456789/2016>
- Balakrishnan, D., Raja, J., Rajagopal, M., Sudhakar, K. et al (2023). *An IoT-Based System for Fault Detection and Diagnosis in Solar PV Panels*. E3S Web of Conferences, 387, 05009. <https://doi.org/10.1051/e3sconf/202338705009>
- Cárdenas, F. (2021). *Modelo de recomendación de dietas saludables mediante algoritmos de optimización, lógica difusa y lógica de primer orden* [Universidad Nacional del Altiplano].  
<http://repositorio.unap.edu.pe/handle/20.500.14082/15546>
- Chandrasekaran, E. (2021). *Fuzzy intelligent systems: Methodologies, techniques, and applications*.
- Cirani, S., Ferrari, G., Picone, M. et al (2019). *Internet of things: Architectures, protocols and standards* (First edition). Wiley.
- Dekka, S., Raju, N., Satish, B. et al (2020). *Design and Development of Efficient Water Management for Agriculture using IoT and Fuzzy Logic*. International Journal of Innovative Technology and Exploring Engineering, 9, 1984–1987.  
<https://doi.org/10.35940/ijitee.C9117.019320>
- Dhumale, R. (2023). *Fuzzy water irrigation system using internet of things*. Agricultural Engineering International: CIGR Journal, 25(2), Article 2.  
<https://cigrjournal.org/index.php/Ejournal/article/view/5398>
- Dhumale, R., Dhumale, N., Umbrajkaar, A. et al (2023). *Fuzzy internet of things-based water irrigation system*.

- Diallo, G. (2023). *Neuro-symbolic digital twins for precision and predictive public health*.
- Dipali, D., Subramanian, T., y Kumaran, G. S. (2023). *A smart oyster mushroom cultivation using automatic fuzzy logic controller*. *Journal of Discrete Mathematical Sciences and Cryptography*, 26(3), 601–615.  
<https://doi.org/10.47974/JDMSC-1731>
- Ebadpour, M., Jamshidi, M. (Behdad), Talla, J. et al (2023). *A digital twinning approach for the internet of unmanned electric vehicles (iouevs) in the metaverse*. *Electronics*, 12(9), Article 9. <https://doi.org/10.3390/electronics12092016>
- Gupta, A. (2019). *The IoT Hacker's Handbook: A Practical Guide to Hacking the Internet of Things*. Apress. <https://doi.org/10.1007/978-1-4842-4300-8>
- Hillar, G. C. (2016). *Internet of Things with Python: Interact with the world and rapidly prototype IoT applications using Python*. Packt Publishing Limited.
- Hu, F. (2016). *Security and privacy in internet of things (iots): Models, algorithms, and implementations* (Primera edición). CRC Press.
- Jantzen, J. (2007). *Foundations of fuzzy control* (1ª ed.). Wiley.  
<https://doi.org/10.1002/9780470061176>
- Javed, A. (2016). *Building Arduino Projects for the Internet of Things*. Apress.  
<https://doi.org/10.1007/978-1-4842-1940-9>
- Joseph, B. M., y Baseer, K. K. (2024). *Fuzzy based reliable data collection and communication in artificial intelligence of things (aiot) networks*. *International Journal of Intelligent Systems and Applications in Engineering*, 12(3s), Article 3s.
- Kahraman, C. (Ed.). (2008). *Fuzzy multi-criteria decision making* (Vol. 16). Springer US. <https://doi.org/10.1007/978-0-387-76813-7>

- Kerimkhulle, S., Dildebayeva, Z., Tokhmetov, A. et al (2023). *Fuzzy logic and its application in the assessment of information security risk of industrial internet of things*. *Symmetry*, 15(10), Article 10. <https://doi.org/10.3390/sym15101958>
- Klir, G., y Yuan, B. (1995). *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice Hall PTR.
- Kranenburg, R. (2008). *The internet of things a critique of ambient technology and the all-seeing network of RFID*. Insitute of Network Cultures, Amsterdam.
- Kruse, R., y Nürnberger, A. (1970). *Learning methods for fuzzy systems*.
- Krushnan, S., Mohnish, S., Naveen, S. et al (2023). *Secured iot framework for soil moisture detection*. IEEE Silchar Subsection conference.
- Laurent, A., y Lesot, M.-J. (Eds.). (2010). *Scalable fuzzy algorithms for data management and analysis: Methods and design*. IGI Global. <https://doi.org/10.4018/978-1-60566-858-1>
- Li, H., Chen, C., y Huang, H. (2001). *Fuzzy neural intelligent systems: Mathematical foundation and the applications in engineering*. CRC Press.
- Loza\_Peralta, C. (2020). *Arquitectura de software basada en microservicios para el uso en dispositivos de Internet of Things*[Universidad LaSalle]. <http://hdl.handle.net/20.500.12953/100>
- Maulana, M., Rohana, T., y Mudzakir, T. (2023). *Implementasi fuzzy logic dalam monitoring infus berbasis internet of things (iot)*. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 7(2), Article 2. <https://doi.org/10.30645/j-sakti.v7i2.699>
- Mohammed, S., y Bahjat, M. (2023). *Fuzzy logic system for drug storage based on the internet of things: A survey*. *Indonesian Journal of Electrical Engineering and*

*Computer Science*, 29(3), 1382. <https://doi.org/10.11591/ijeecs.v29.i3.pp1382-1392>

Mohapatra, A., Mohanty, A., Pradhan, N., Mohanty, et al (2023). *An Industry 4.0 implementation of a condition monitoring system and IoT-enabled predictive maintenance scheme for diesel generators*. *Alexandria Engineering Journal*, 76, 525–541. <https://doi.org/10.1016/j.aej.2023.06.026>

Morales, R., y Méndez, J. (2011). *Inteligencia Artificial: Métodos, Técnicas y Aplicaciones*. McGraw-Hill España.

Nithya, R., Kumar, R., Singh, A., et al (2022). *Fuzzy logic implementation of internet of things and smart material management in the sector of automotive management*. 11, 404.

Nyakuri, J., Harerimana, F., Karanguza, F. et al (2023). *Iot and fuzzy logic applied to smart public hall for covid-19 monitoring system*. *American Journal of Computing and Engineering*, 6(1), Article 1. <https://doi.org/10.47672/ajce.1358>

Pampamallco-Jara, J. (2022). *Diseño e implementación de controladores basados en lógica difusa para una planta de procesos multipropósito* [Universidad de Ingeniería y Yecnología]. <https://repositorio.utec.edu.pe/handle/20.500.12815/291?show=full>

Pfister, C. (2011). *Getting started with the Internet of things: Connecting sensors and microcontrollers to the cloud* (1. ed). O'Reilly.

Ponce-Cruz, P. (2010). *Inteligencia Artificial. Con Aplicaciones a la Ingeniería* (Primera edición). Alfaomega grupo editor.

Prasetyo, A., Yusuf, A., Litanianda, et al (2023). *Implementation of fuzzy logic for chili irrigation integrated with internet of things*. *Journal of Computer Networks*,

*Architecture and High Performance Computing*, 5(2), Article 2.

<https://doi.org/10.47709/cnahpc.v5i2.2518>

Raj, P., y Raman, A. (2017). *The Internet of things: Enabling technologies, platforms, and use cases*. CRC Press/Taylor y Francis Group.

Reyna, R., y Puente, R. (2021). *Diseño del Fitotron aplicando lógica difusa*

[Universidad Ricardo Palma]. <https://hdl.handle.net/20.500.14138/4042>

Ross, T. (2010). *Fuzzy logic with engineering applications* (3rd ed). John Wiley.

Serpanos, D., y Wolf, M. (2018). *Internet-of-Things (IoT) Systems*. Springer

International Publishing. <https://doi.org/10.1007/978-3-319-69715-4>

Sivanandam, S., Sumathi, S., y Deepa, S.(2007). *Introduction to Fuzzy Logic using*

*MATLAB*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-35781-0>

Sunardi, Yudhana, A., y Furizal. (2023). *Tsukamoto fuzzy inference system on internet*

*of things-based for room temperature and humidity control*. *IEEE Access*, 11,

6209–6227. <https://doi.org/10.1109/ACCESS.2023.3236183>

Tabatabaei, S. (2023). *A novel method for energy efficiency in internet of thing (iot)*

*through fuzzy logic and genetic-algorithm-based clustering applied for*

*agriculture monitoring*. <https://doi.org/10.21203/rs.3.rs-2898962/v1>

Utrilla, L. W. (2020). *Diseño del control de nivel y temperatura de un sistema*

*mezclador para la industria textil usando lógica difusa* [Universidad Nacional de

San Agustín de Arequipa].

[https://repositorio.unsa.edu.pe/server/api/core/bitstreams/b7a82e75-8bf9-4dea-](https://repositorio.unsa.edu.pe/server/api/core/bitstreams/b7a82e75-8bf9-4dea-901a-35d606f2fa13/content)

[901a-35d606f2fa13/content](https://repositorio.unsa.edu.pe/server/api/core/bitstreams/b7a82e75-8bf9-4dea-901a-35d606f2fa13/content)

- Waluyo, A., Hadiatna, F., y Anugerah, D. (2023). *Fuzzy-based smart farming and consumed energy comparison using the internet of things*. *IEEE Access*, 11, 69241–69251. <https://doi.org/10.1109/ACCESS.2023.3291616>
- Widiono, S., y Tri Untoro, I.(2023). *Fuzzy Logic Implementation In Internet Of Things Technology For Foging Greenhouse Plants* / International Journal of Engineering Technology and Natural Sciences.  
<https://journal.uty.ac.id/index.php/IJETS/article/view/205>
- Xiao, P. (2018). *Designing Embedded Systems and the Internet of Things (IoT) with the ARM® Mbed™* (Primera edición). John Wiley.
- Zadef, L. (1996). *Fuzzy Logic = Computing with Words*. *IEEE Transactions on Fuzzy Systems*, 40(2).
- Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(8), 338–353.
- Zadeh, L. (1969). *Toward a theory of Fuzzy Systems*. Electronics research laboratory , Universidad de California, National Aeronautics and space administration, 1(1).
- Zadeh, L. (1971). *Similarity relations and fuzzy orderings*. *Information Sciences*, 3(2), 177–200. [https://doi.org/10.1016/S0020-0255\(71\)80005-1](https://doi.org/10.1016/S0020-0255(71)80005-1)
- Zadeh, L. (1975). *The concept of a linguistic variable and its application to approximate reasoning*. *Informations Sciences*, 9, 43–80.
- Zadeh, L. (1983). *A computational aproach to fuzzy quantifiers in natural languajes*. *Comp y Maths with appls*, 9(1), 149–184.
- Zadeh, L. (1988). *Fuzzy Logic*. *NASA Grant DCR 8513139*, 1(1), 83–91.
- Zadeh, L. (1999). *Fuzzy logic and the calculi of fuzzy rules, fuzzy graphs, and fuzzy probabilities*. *Computers y Mathematics with Applications*, 37(11–12), 35.  
[https://doi.org/10.1016/S0898-1221\(99\)00140-6](https://doi.org/10.1016/S0898-1221(99)00140-6)

Zadeh, L. (2005). *Toward a generalized theory of uncertainty* An outline. *Information Sciences*, 1(1).

Zapata, C. (2022). *Sistemas híbridos inteligentes basados en lógica difusa y aprendizaje profundo*.

<https://repositorio.uniandes.edu.co/entities/publication/2887befe-f953-4e14-9e2c-aa9ba2ddb303>

## ANEXOS

### ANEXO 1: DATASHEETS DE LOS SENSORES SELECCIONADOS

DESCRIPCIÓN GENERAL	
<p>El sensor MQ 135 es un sensor de calidad del aire que se utiliza para detectar gases nocivos como el amoníaco (NH<sub>3</sub>), el óxido nítrico (NO<sub>x</sub>), el alcohol, el benceno, el humo y los vapores de CO<sub>2</sub>. Es ampliamente utilizado en aplicaciones de detección de calidad del aire en interiores y monitoreo ambiental.</p>	
<b>Dimensiones</b>	Longitud: 32mm Anchura: 20mm Altura: 22mm
CARACTERÍSTICAS	
Tiempo de respuesta	< 10 segundos
Consumo de Corriente	< 150mA
Tensión de Alimentación	5V DC
Salida Analógica	0-5V
Temperatura de Operación	-10°C a 50°C
Humedad de Operación	< 95% RH (sin condensación)
RANGO DE DETECCIÓN	
Amoníaco (NH <sub>3</sub> )	10-300ppm
Óxido Nítrico (NO <sub>x</sub> )	10-1000ppm
Alcohol	10-300ppm
Benceno	10-1000ppm
CO <sub>2</sub>	10-5000ppm
SENSIBILIDAD	
NH <sub>3</sub>	1.5±0.5
NO <sub>x</sub>	0.6±0.2
Alcohol	1.5±0.5
Benceno	0.6±0.2
CO <sub>2</sub>	0.7±0.2
CONFIGURACIÓN Y USO	
<p><b>Alimentación:</b> Conecte el pin VCC del sensor MQ135 al pin de 5V de su microcontrolador o fuente de alimentación.</p> <p><b>Tierra (GND):</b> Conecte el pin GND del sensor MQ135 al pin GND de su microcontrolador o fuente de alimentación.</p> <p><b>Salida Analógica (AOUT):</b> Conecte el pin AOUT del sensor MQ135 al pin analógico de su microcontrolador (por ejemplo, A0 en Arduino).</p>	

Figura 49:  
*Datasheet del sensor de gas*

## Product Discription

HC-SR501 is based on infrared technology, automatic control module, using Germany imported LHI778 probe design, high sensitivity, high reliability, ultra-low-voltage operating mode, widely used in various auto-sensing electrical equipment, especially for battery-powered automatic controlled products.

### Specification:

- Voltage: 5V – 20V
- Power Consumption: 65mA
- TTL output: 3.3V, 0V
- Delay time: Adjustable (.3->5min)
- Lock time: 0.2 sec
- Trigger methods: L – disable repeat trigger, H enable repeat trigger
- Sensing range: less than 120 degree, within 7 meters
- Temperature: – 15 ~ +70
- Dimension: 32\*24 mm, distance between screw 28mm, M2, Lens dimension in diameter: 23mm

### Application:

Automatically sensing light for Floor, bathroom, basement, porch, warehouse, Garage, etc, ventilator, alarm, etc.

### Features:

- Automatic induction: to enter the sensing range of the output is high, the person leaves the sensing range of the automatic delay off high, output low.
- Photosensitive control (optional, not factory-set) can be set photosensitive control, day or light intensity without induction.
- Temperature compensation (optional, factory reset): In the summer when the ambient temperature rises to 30 ° C to 32 ° C, the detection distance is slightly shorter, temperature compensation can be used for performance compensation.
- Triggered in two ways: (jumper selectable)
  - non-repeatable trigger: the sensor output high, the delay time is over, the output is automatically changed from high level to low level;
  - repeatable trigger: the sensor output high, the delay period, if there is human activity in its sensing range, the output will always remain high until the people left after the delay will be high level goes low (sensor module detects a time delay period will be automatically extended every human activity, and the starting point for the delay time to the last event of the time).
- With induction blocking time (the default setting: 2.5s blocked time): sensor module after each sensor output (high into low), followed by a blockade set period of time, during this time period sensor does not accept any sensor signal. This feature can be achieved sensor output time "and" blocking time "interval between the work can be applied to interval detection products; This function can inhibit a variety of interference in the process of load switching. (This time can be set at zero seconds – a few tens of seconds).
- Wide operating voltage range: default voltage DC4.5V-20V.
- Micropower consumption: static current <50 microamps, particularly suitable for battery-powered automatic control products.
- Output high signal: easy to achieve docking with the various types of circuit.

Figura 50:  
*Datasheet del sensor de distancia*

**ANEXO 2:  
CIRCUITOS ELECTRÓNICOS DE LOS SENSORES SELECCIONADOS**

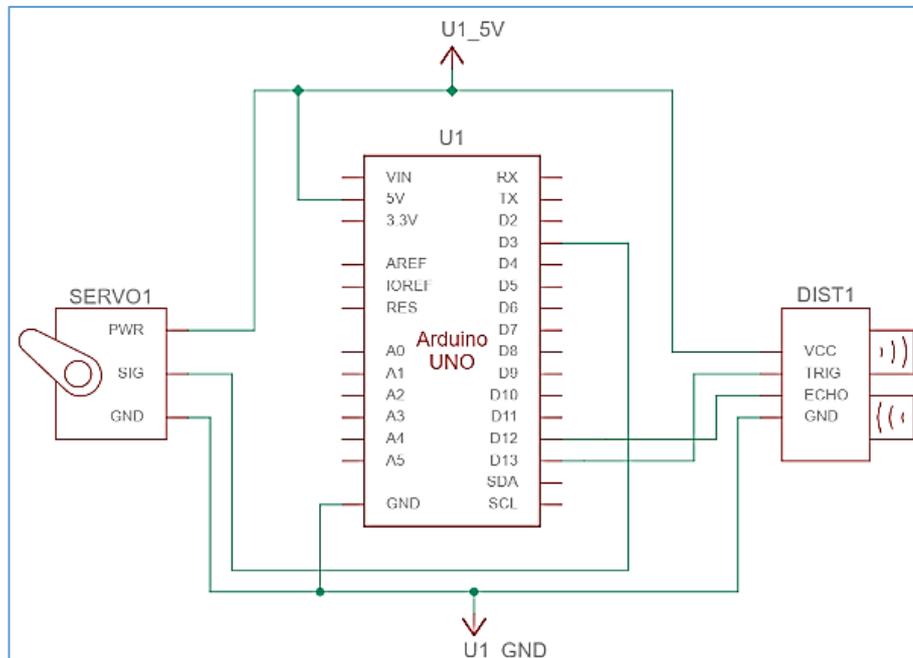


Figura 51:  
*Circuito electrónico del sensor de distancia*

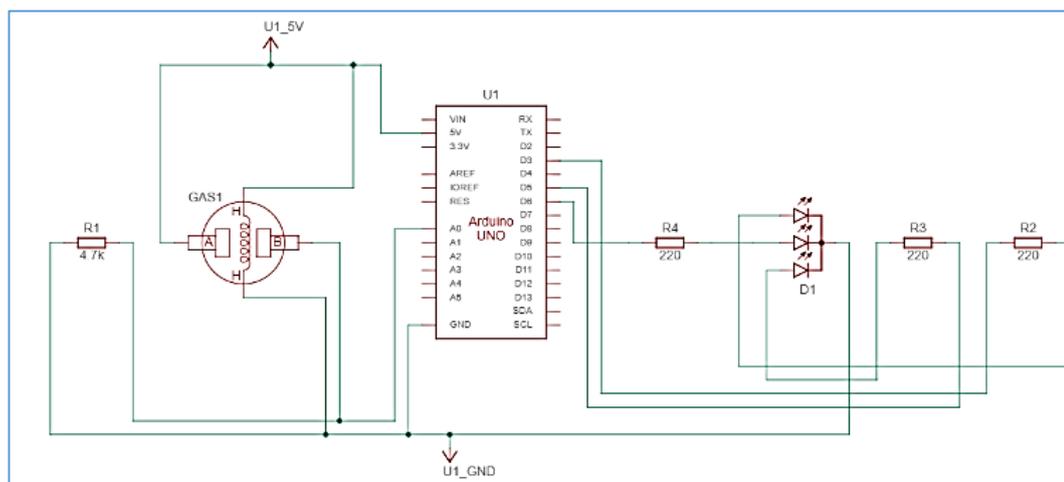


Figura 52:  
*Circuito electrónico del sensor de gas*

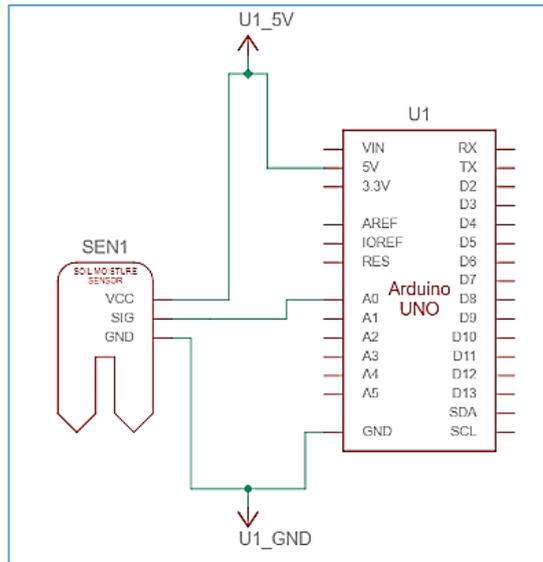


Figura 54:  
Circuito electrónico del sensor de humedad del suelo

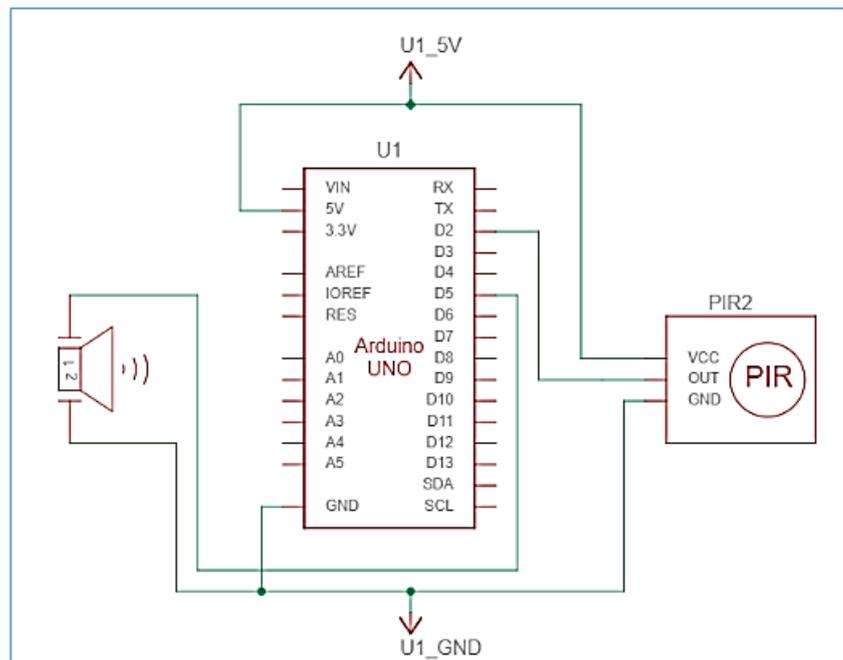


Figura 53:  
Circuito electrónico del sensor de movimiento

### ANEXO 3: CONEXIONES LÓGICAS DE LOS SENSORES SELECCIONADOS

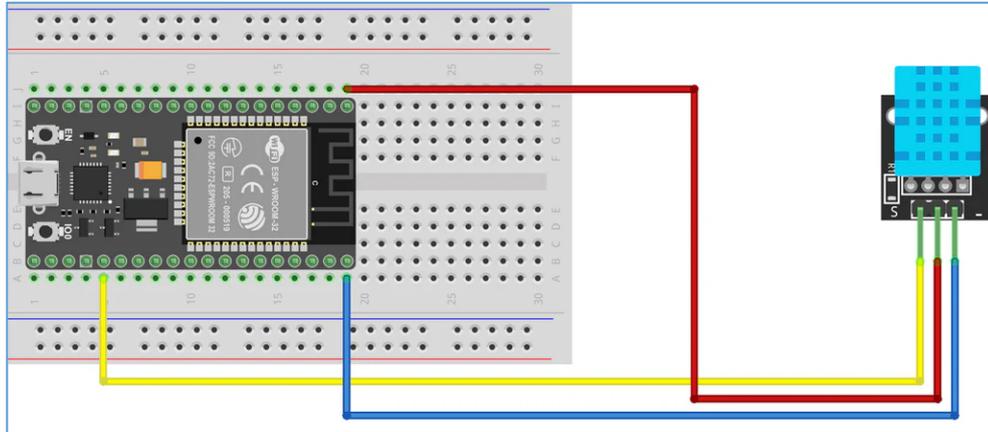


Figura 55:  
*Conexión lógica del sensor de temperatura*

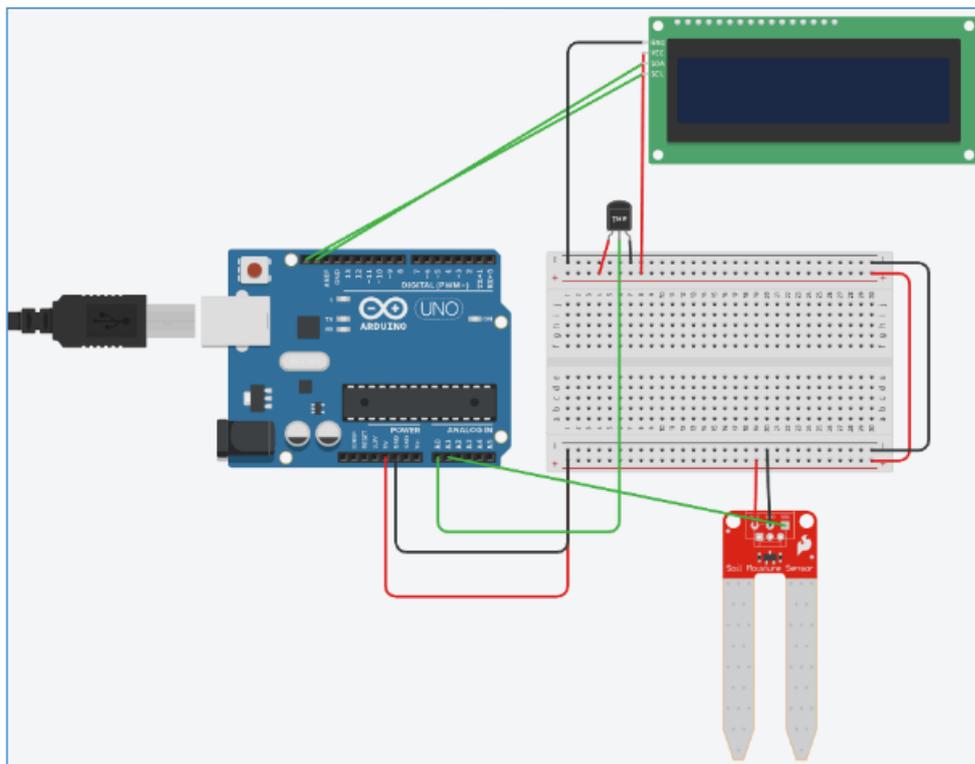


Figura 56:  
*Conexión lógica del sensor de humedad de la tierra*

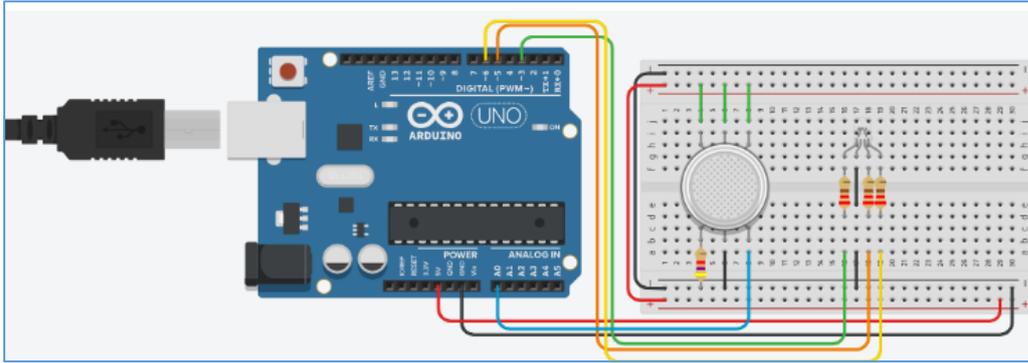


Figura 57:  
 Conexión lógica del sensor de gases con Arduino

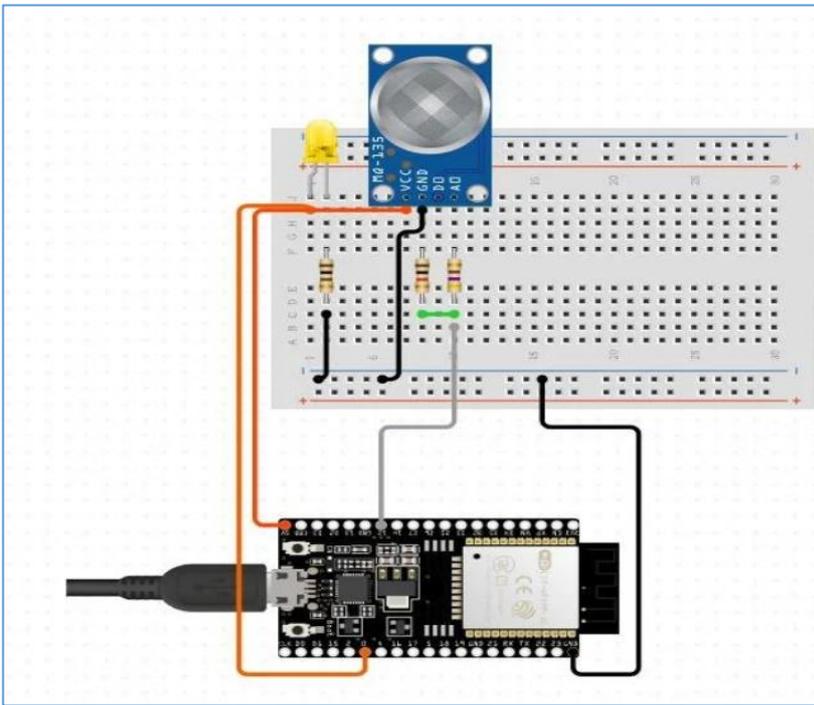


Figura 58:  
 Conexión lógica del sensor de gas con ESP8266

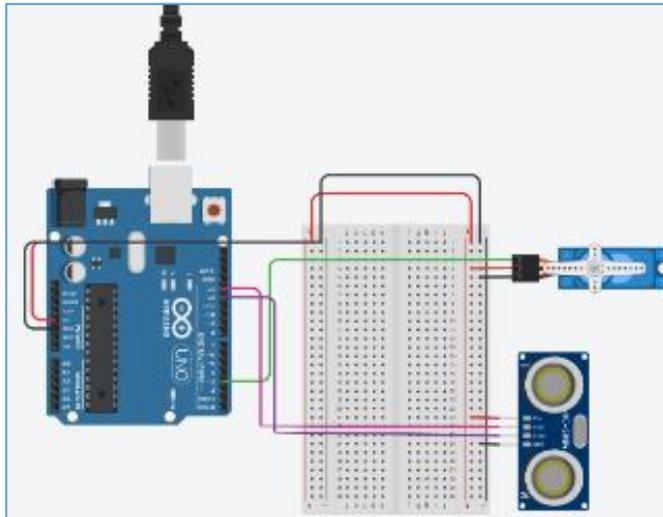


Figura 59:  
*Conexión lógica del sensor de distancia con Arduino*

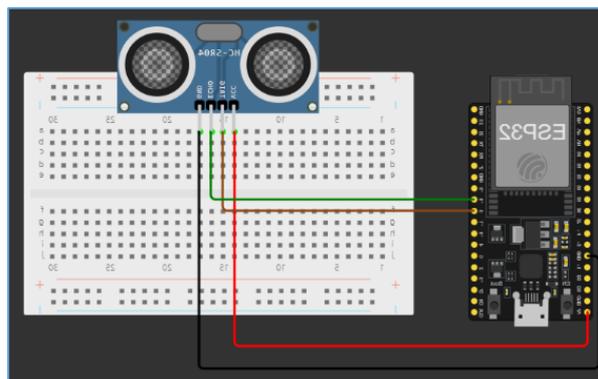


Figura 60:  
*Conexión lógica del sensor de distancia con ESP32*

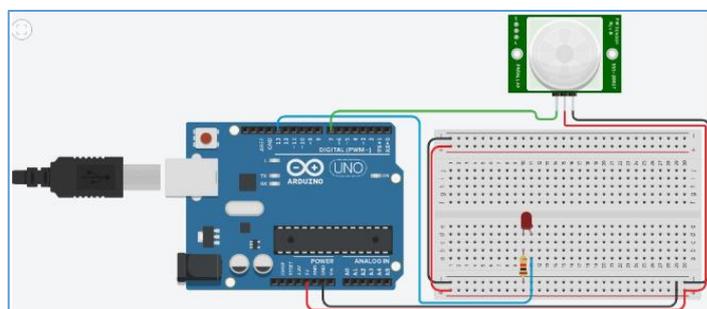


Figura 61:  
*Conexión lógica del sensor de movimiento*

## ANEXO 4: GEMELOS DIGITALES DE LOS SENSORES SELECCIONADOS

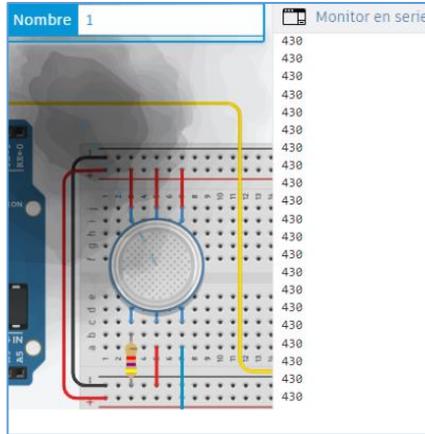


Figura 62:  
*Gemelo digital del sensor de gas*

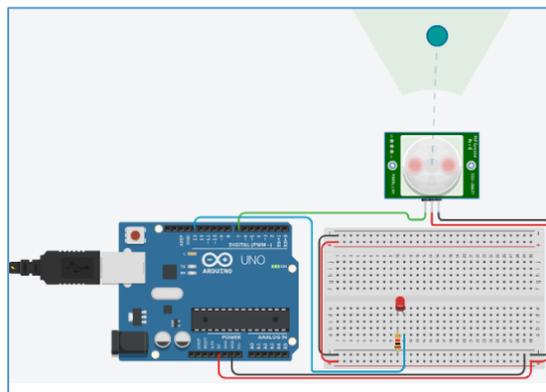
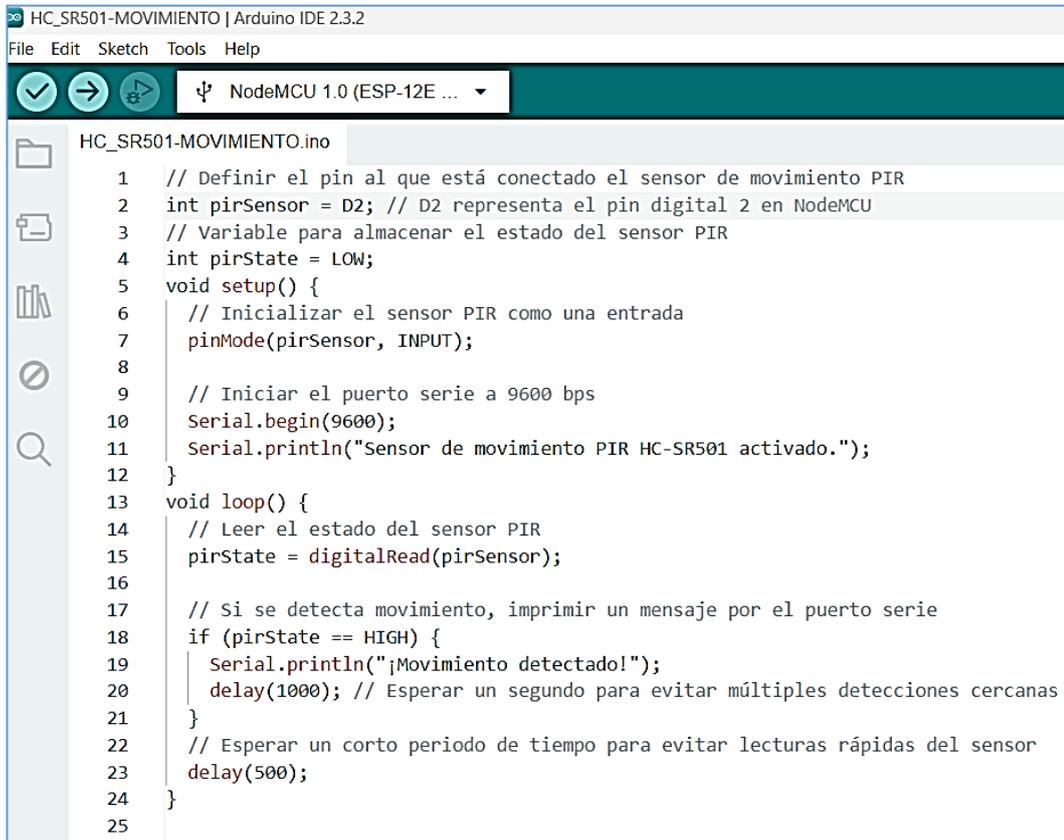


Figura 63:  
*Gemelo digital del sensor de movimiento*



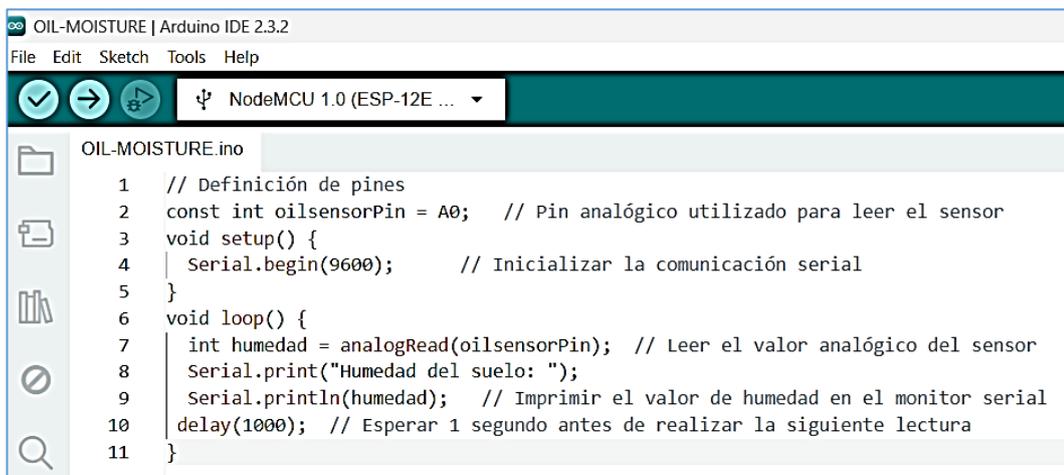
Figura 64:  
*Gemelo digital del sensor de distancia*

## ANEXO 5: CODIGO ARDUINO DE LOS SENSORES SELECCIONADOS



```
HC_SR501-MOVIMIENTO | Arduino IDE 2.3.2
File Edit Sketch Tools Help
NodeMCU 1.0 (ESP-12E ...)
HC_SR501-MOVIMIENTO.ino
1 // Definir el pin al que está conectado el sensor de movimiento PIR
2 int pirSensor = D2; // D2 representa el pin digital 2 en NodeMCU
3 // Variable para almacenar el estado del sensor PIR
4 int pirState = LOW;
5 void setup() {
6 // Inicializar el sensor PIR como una entrada
7 pinMode(pirSensor, INPUT);
8
9 // Iniciar el puerto serie a 9600 bps
10 Serial.begin(9600);
11 Serial.println("Sensor de movimiento PIR HC-SR501 activado.");
12 }
13 void loop() {
14 // Leer el estado del sensor PIR
15 pirState = digitalRead(pirSensor);
16
17 // Si se detecta movimiento, imprimir un mensaje por el puerto serie
18 if (pirState == HIGH) {
19 Serial.println("¡Movimiento detectado!");
20 delay(1000); // Esperar un segundo para evitar múltiples detecciones cercanas
21 }
22 // Esperar un corto periodo de tiempo para evitar lecturas rápidas del sensor
23 delay(500);
24 }
25
```

Figura 65:  
*Código Arduino para el sensor de movimiento*



```
OIL-MOISTURE | Arduino IDE 2.3.2
File Edit Sketch Tools Help
NodeMCU 1.0 (ESP-12E ...)
OIL-MOISTURE.ino
1 // Definición de pines
2 const int oilsensorPin = A0; // Pin analógico utilizado para leer el sensor
3 void setup() {
4 Serial.begin(9600); // Inicializar la comunicación serial
5 }
6 void loop() {
7 int humedad = analogRead(oilsensorPin); // Leer el valor analógico del sensor
8 Serial.print("Humedad del suelo: ");
9 Serial.println(humedad); // Imprimir el valor de humedad en el monitor serial
10 delay(1000); // Esperar 1 segundo antes de realizar la siguiente lectura
11 }

```

Figura 66:  
*Código Arduino para el sensor de humedad del suelo*

## ANEXO 6: IMPLEMENTACIÓN FÍSICA DE LOS SENSORES SELECCIONADOS

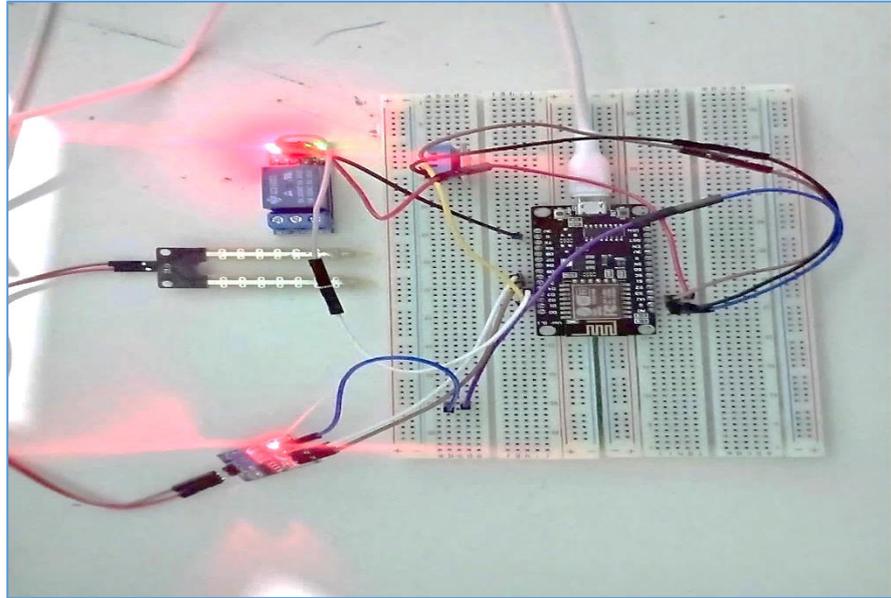


Figura 67:  
*Implementación física del sensor de humedad del suelo*



Figura 68:  
*Implementación física del sensor de distancia*

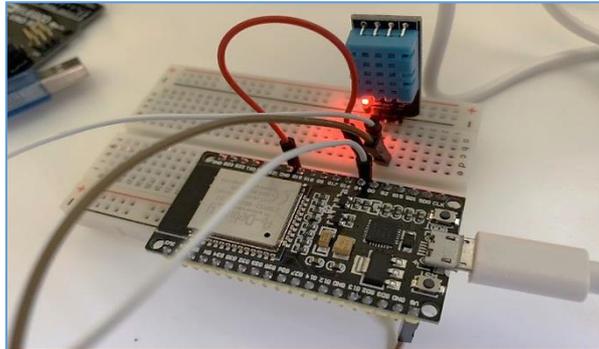


Figura 69:  
*Implementación física del sensor de humedad ambiental*

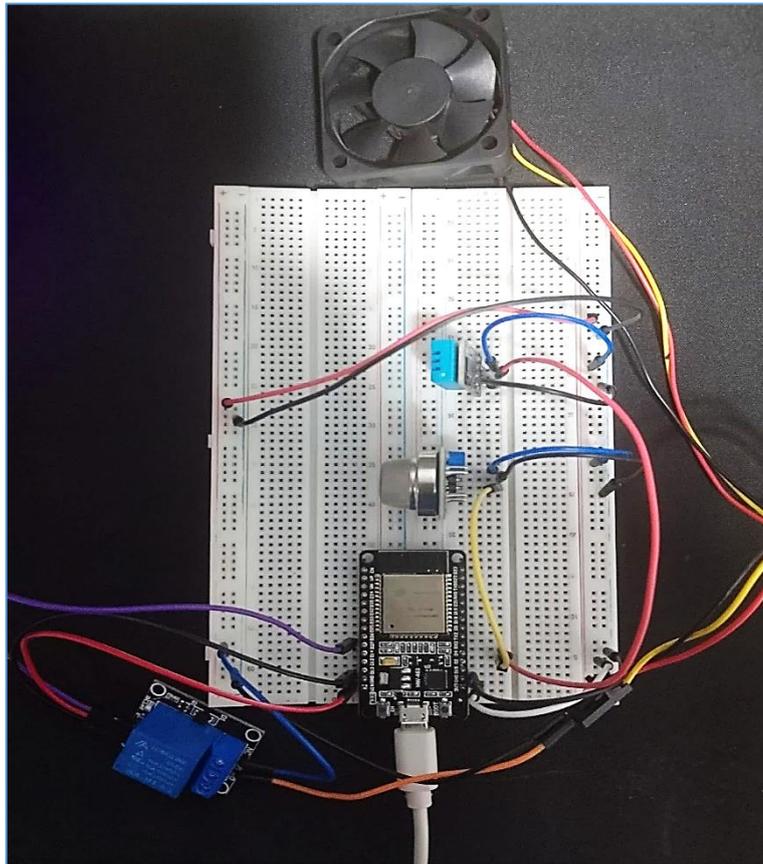


Figura 70:  
*Implementación física del sensor de gases*

## ANEXO 7: CONJUNTOS DIFUSOS DE LAS OTRAS VARIABLES DE ESTUDIO

### SISTEMA DE RIEGO TECNIFICADO

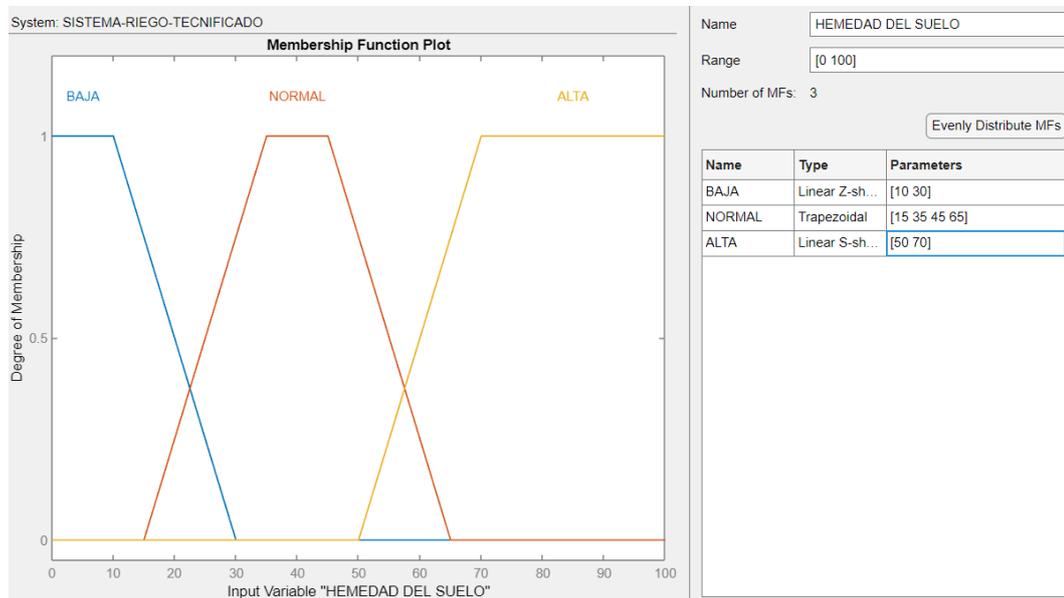
**HUMEDAD DEL SUELO:**  $H = \{B, N, A\}$ ,  $x \in X$ ,  $X \subseteq [0, 100]$ , Se definen los conjuntos difusos:

$$B = \{(x, \mu_B(x), x \in X), \quad \mu_B(x) = \mu_{Función L}(x; 10, 30),$$

$$N = \{(x, \mu_N(x), x \in X), \quad \mu_N(x) = \mu_{Trapezoidal}(x; 15, 35, 45, 56)$$

$$A = \{(x, \mu_A(x), x \in X), \quad \mu_A(x) = \mu_{Gamma}(x; 50, 70)$$

Las figuras 71, 72 muestran las gráficas de las funciones de membresía de los conjuntos difusos de la variable lingüística humedad del suelo, velocidad del motor de la bomba respectivamente:



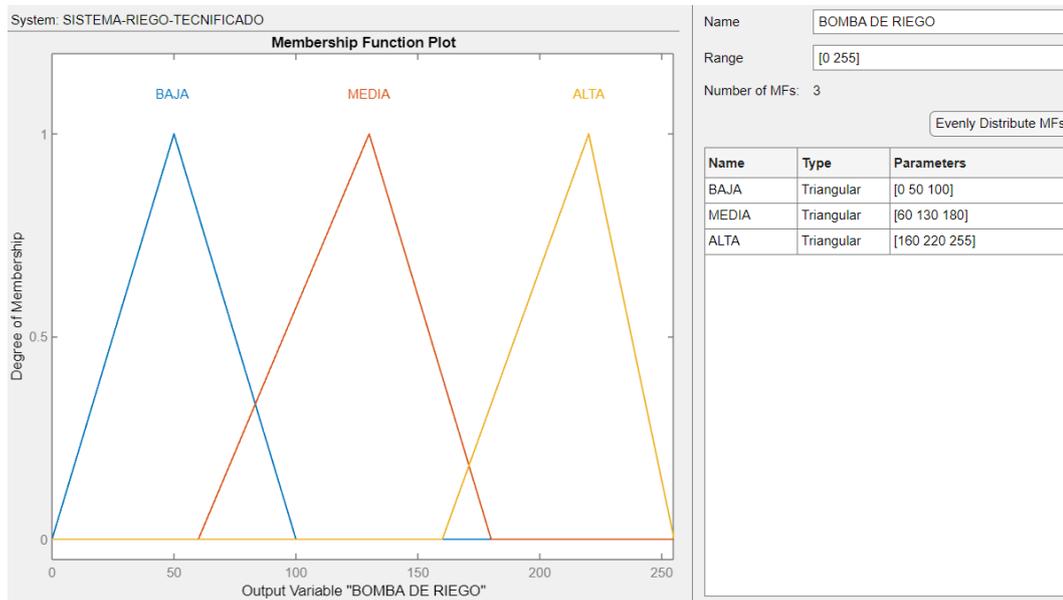
**Figura 71:**  
*Funciones de membresía de la variable humedad del suelo*

**VELOCIDAD MOTOR DE BOMBA:**  $V = \{B, N, A\}$ ,  $x \in X$ ,  $X \subseteq [0, 255]$ ,

$$B = \{(x, \mu_B(x), x \in X), \quad \mu_B(x) = \mu_{Triangular}(x; 0, 50, 100),$$

$$N = \{(x, \mu_N(x), x \in X), \quad \mu_N(x) = \mu_{Triangular}(x; 60, 130, 180)$$

$$A = \{(x, \mu_A(x), x \in X), \quad \mu_A(x) = \mu_{Triangular}(x; 150, 200, 250)$$



**Figura 72:**

*Funciones de membresía de la velocidad de la bomba*

## SISTEMA CONTROLADOR DE GASES

**CONCENTRACION DE GASES**  $C = \{B, N, A\}$ ,  $x \in X$ ,  $X \subseteq [0, 60]$ , Se

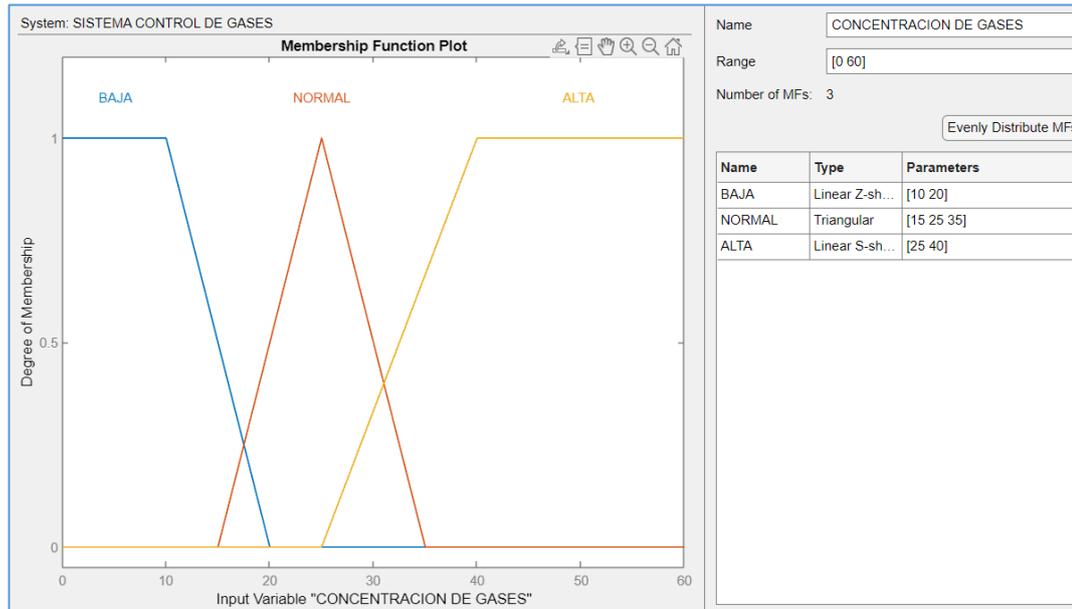
definen los conjuntos difusos:

$$B = \{(x, \mu_B(x), x \in X), \quad \mu_B(x) = \mu_{Función L}(x; 10, 20),$$

$$N = \{(x, \mu_N(x), x \in X), \quad \mu_N(x) = \mu_{Triangular}(x; 15, 25, 35)$$

$$A = \{(x, \mu_A(x)), x \in X\}, \quad \mu_A(x) = \mu_{Gamma}(x; 25, 40)$$

Las figuras 73, 74 muestran las gráficas de las funciones de membresía de los conjuntos difusos de la variable lingüística concentración de gases, velocidad del motor de la bomba extractora respectivamente:



**Figura 73:**

*Funciones de membresía de la variable concentración de gases*

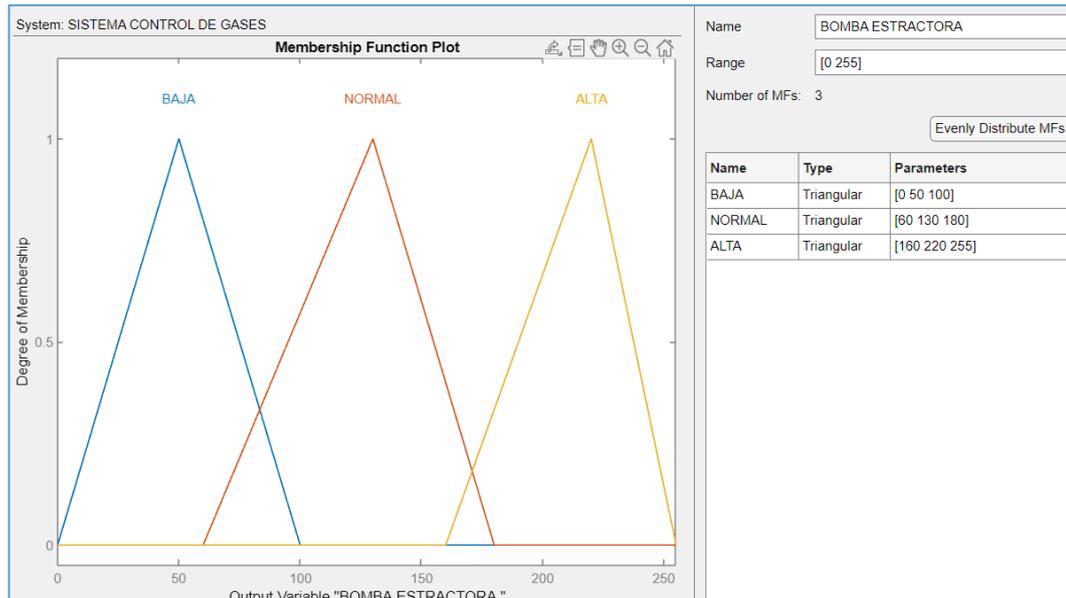
**VELOCIDAD MOTOR DE BOMBA EXTRACTORA:**  $V = \{B, N, A\}$ ,  $x \in X$ ,

$X \subseteq [0, 255]$ ,

$$B = \{(x, \mu_B(x)), x \in X\}, \quad \mu_B(x) = \mu_{Triangular}(x; 0, 50, 100),$$

$$N = \{(x, \mu_N(x)), x \in X\}, \quad \mu_N(x) = \mu_{Triangular}(x; 60, 130, 180)$$

$$A = \{(x, \mu_A(x)), x \in X\}, \quad \mu_A(x) = \mu_{Triangular}(x; 160, 220, 255)$$



**Figura 74:**  
*Funciones de membresía de la variable bomba extractora*

## SISTEMA DETECTOR DE MOVIMIENTOS

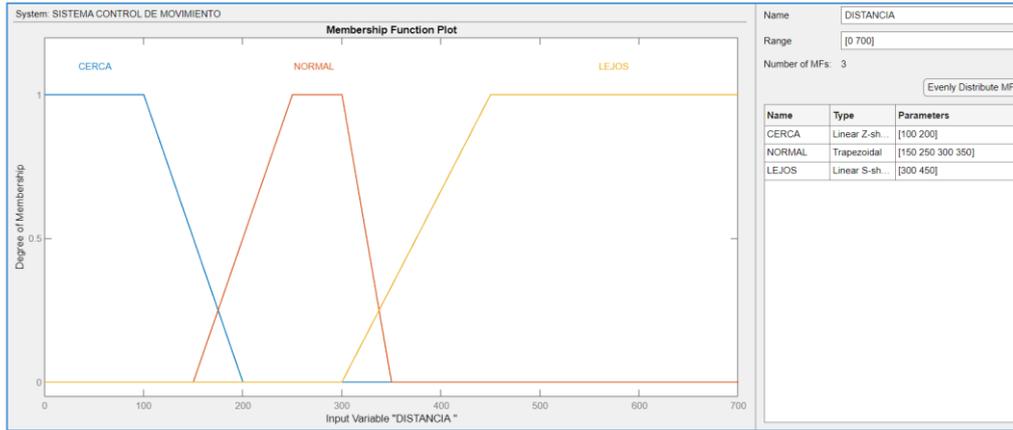
**DISTANCIA DEL OBJETO**  $D = \{C, N, L\}$ ,  $x \in X$ ,  $X \subseteq [0, 60]$ , Se definen los conjuntos difusos:

$$C = \{(x, \mu_C(x)), x \in X\}, \quad \mu_C(x) = \mu_{Función\ L}(x; 100, 200),$$

$$N = \{(x, \mu_N(x)), x \in X\}, \quad \mu_N(x) = \mu_{Triangular}(x; 150, 250, 350)$$

$$L = \{(x, \mu_L(x)), x \in X\}, \quad \mu_L(x) = \mu_{Gamma}(x; 300, 450)$$

Las figuras 75, 76 muestran las gráficas de las funciones de membresía de los conjuntos difusos de la variable lingüística distancia del objeto y volumen de la alarma respectivamente:



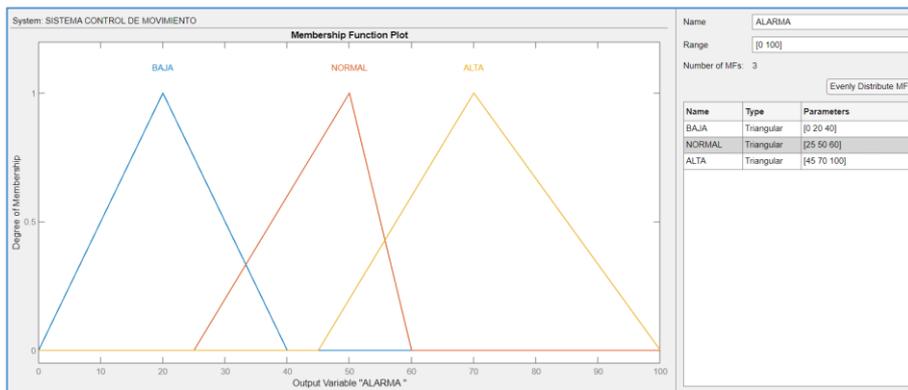
**Figura 75:**  
*Funciones de membresía de la variable distancia del objeto*

**VOLUMEN DE LA ALARMA:**  $V = \{B, N, A\}$ ,  $x \in X$ ,  $X \subseteq [0, 255]$ ,

$$B = \{(x, \mu_B(x), x \in X), \quad \mu_B(x) = \mu_{Triangular}(x; 0, 50, 100),$$

$$N = \{(x, \mu_N(x), x \in X), \quad \mu_N(x) = \mu_{Triangular}(x; 25, 50, 60)$$

$$A = \{(x, \mu_A(x), x \in X), \quad \mu_A(x) = \mu_{Triangular}(x; 45, 70, 100)$$



**Figura 76:**  
*Funciones de membresía de la variable volumen de alarma*

## SISTEMA CONTROL DE PUERTAS

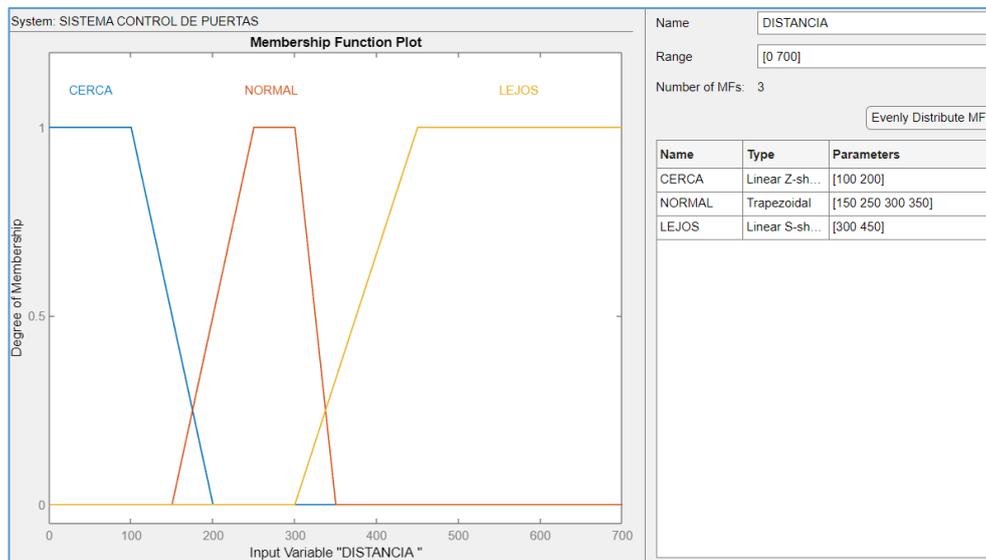
**DISTANCIA DEL OBJETO**  $D = \{C, N, L\}$ ,  $x \in X$ ,  $X \subseteq [0, 60]$ , Se definen los conjuntos difusos:

$$C = \{(x, \mu_C(x), x \in X), \quad \mu_C(x) = \mu_{Función L}(x; 100, 200),$$

$$N = \{(x, \mu_N(x), x \in X), \quad \mu_N(x) = \mu_{Triangular}(x; 150, 250, 350)$$

$$L = \{(x, \mu_L(x), x \in X), \quad \mu_L(x) = \mu_{Gamma}(x; 300, 450)$$

La figura 77,78 muestran las gráficas de las funciones de membresía de los conjuntos difusos de la variable lingüística distancia del objeto y posición del servomotor:



**Figura 77:**

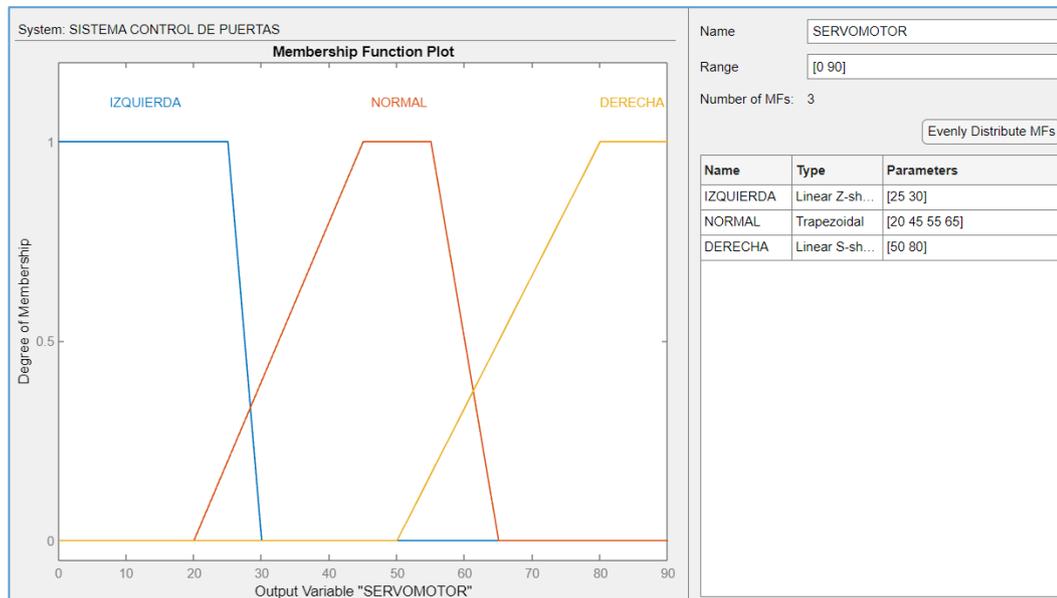
*Funciones de membresía de la variable distancia del objeto*

**SERVOMOTOR:**  $S = \{I, N, D\}$ ,  $x \in X$ ,  $X \subseteq [0, 90]$ ,

$$I = \{(x, \mu_I(x), x \in X), \quad \mu_I(x) = \mu_{Funcion L}(x; 25, 30),$$

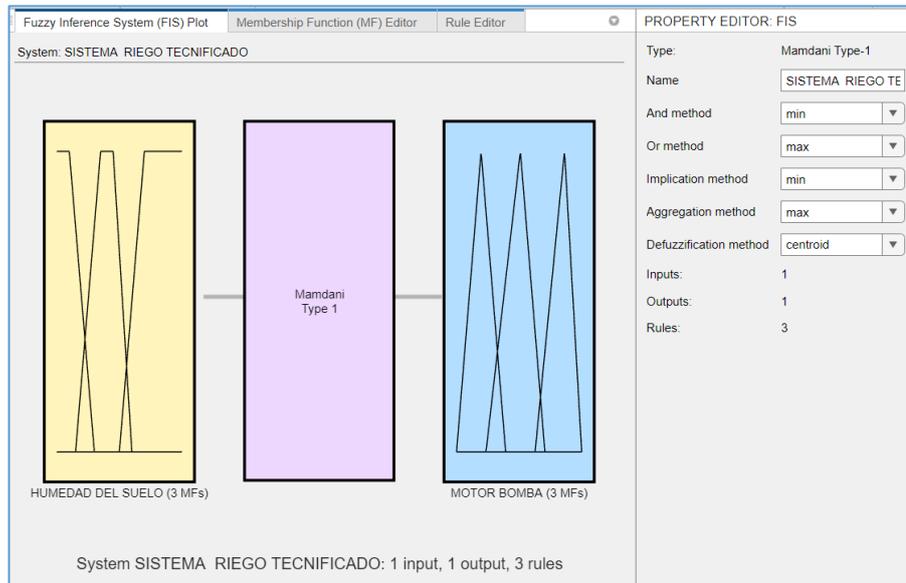
$$N = \{(x, \mu_N(x), x \in X), \quad \mu_N(x) = \mu_{Trapezoidal}(x; 20, 45, 55, 65)$$

$$D = \{(x, \mu_A(x)), x \in X\}, \quad \mu_A(x) = \mu_{Gamma}(x; 50, 80)$$

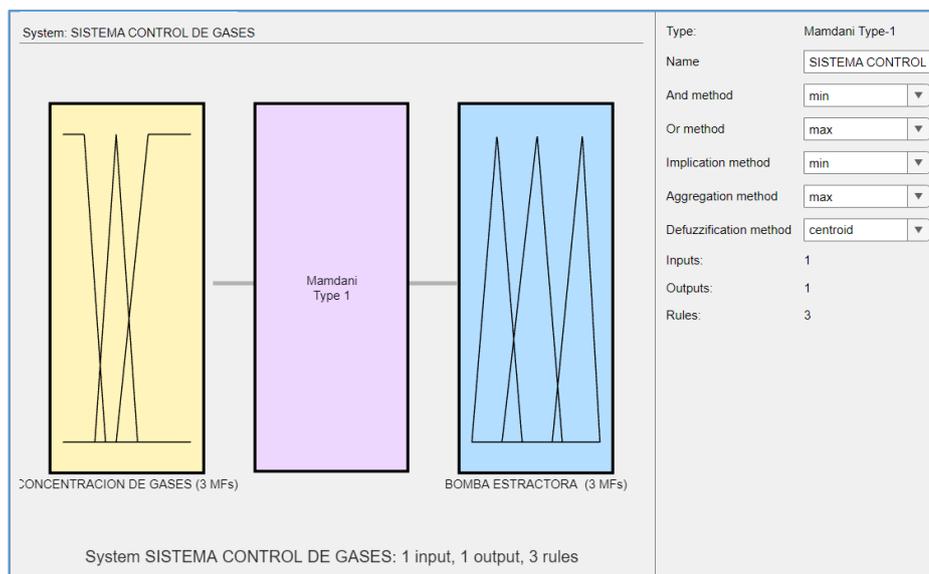


**Figura 78:**  
*Funciones de membresía de la variable posiciones del servomotor*

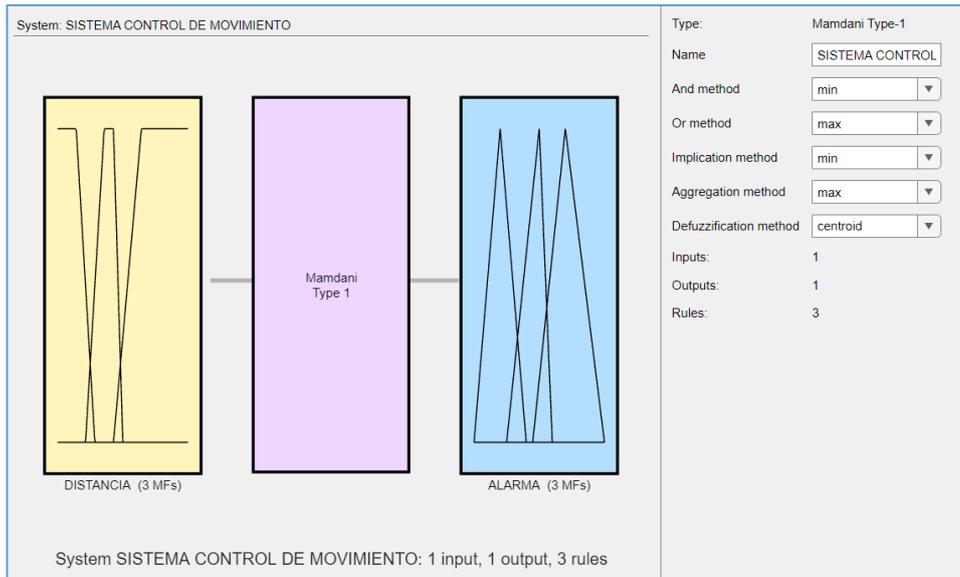
## ANEXO 8: IMPLEMENTACIÓN DE LOS CONJUNTOS DIFUSOS EN MATLAB



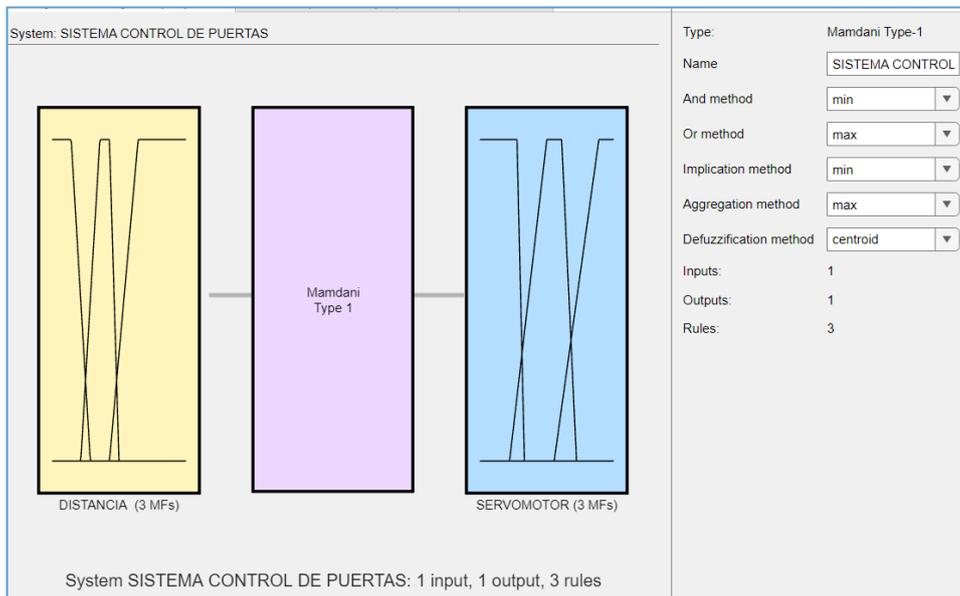
**Figura 79:**  
*Implementación del sistema Riego tecnificado en MATLAB*



**Figura 80:**  
*Implementación del sistema control de gases en MATLAB*



**Figura 81:**  
*Implementación del sistema control de movimiento en MATLAB*



**Figura 82:**  
*Implementación del sistema control de puertas en MATLAB*

## ANEXO 9: DEFINICION DE LAS REGLAS DE INFERENCIA

System: SISTEMA RIEGO TECNIFICADO

Add All Possible Rules Clear All Rules

Rule	Weight	Name
1 If HUMEDAD DEL SUELO is BAJA then MOTOR BOMBA is ALTA	1	rule1
2 If HUMEDAD DEL SUELO is NORMAL then MOTOR BOMBA is NORMAL	1	rule2
3 If HUMEDAD DEL SUELO is ALTA then MOTOR BOMBA is BAJA	1	rule3

Name: rule1  
Weight: 1  
Connection:  And  Or

If  
HUMEDAD D... is BAJA

Then  
MOTOR BOMBA is ALTA

**Figura 83:**  
*Reglas de inferencia del sistema Riego tecnificado*

System: SISTEMA CONTROL DE GASES

Add All Possible Rules Clear All Rules

Rule	Weight	Name
1 If CONCENTRACION DE GASES is BAJA then BOMBA EXTRACTORA is BAJA	1	rule1
2 If CONCENTRACION DE GASES is NORMAL then BOMBA EXTRACTORA is NORMAL	1	rule2
3 If CONCENTRACION DE GASES is ALTA then BOMBA EXTRACTORA is ALTA	1	rule3

Name: rule3  
Weight: 1  
Connection:  And  Or

If  
CONCENTRACI... is ALTA

Then  
BOMBA ESTRA... is ALTA

**Figura 84:**  
*Reglas de inferencia del sistema Riego tecnificado*

System: SISTEMA CONTROL DE MOVIMIENTO

Add All Possible Rules Clear All Rules

Rule	Weight	Name
1 If DISTANCIA is CERCA then ALARMA is ALTA	1	rule1
2 If DISTANCIA is NORMAL then ALARMA is NORMAL	1	rule2
3 If DISTANCIA is LEJOS then ALARMA is BAJA	1	rule3

Name: rule1  
Weight: 1  
Connection:  And  Or

If  
DISTANCIA is CERCA

Then  
ALARMA is ALTA

**Figura 85:**  
*Reglas de inferencia del sistema control de movimiento*

System: SISTEMA CONTROL DE PUERTAS

Add All Possible Rules Clear All Rules

Rule	Weight	Name
1 If DISTANCIA is CERCA then SERVO MOTOR is DERECHA	1	rule1
2 If DISTANCIA is NORMAL then SERVO MOTOR is NORMAL	1	rule2
3 If DISTANCIA is LEJOS then SERVO MOTOR is IZQUIERDA	1	rule3

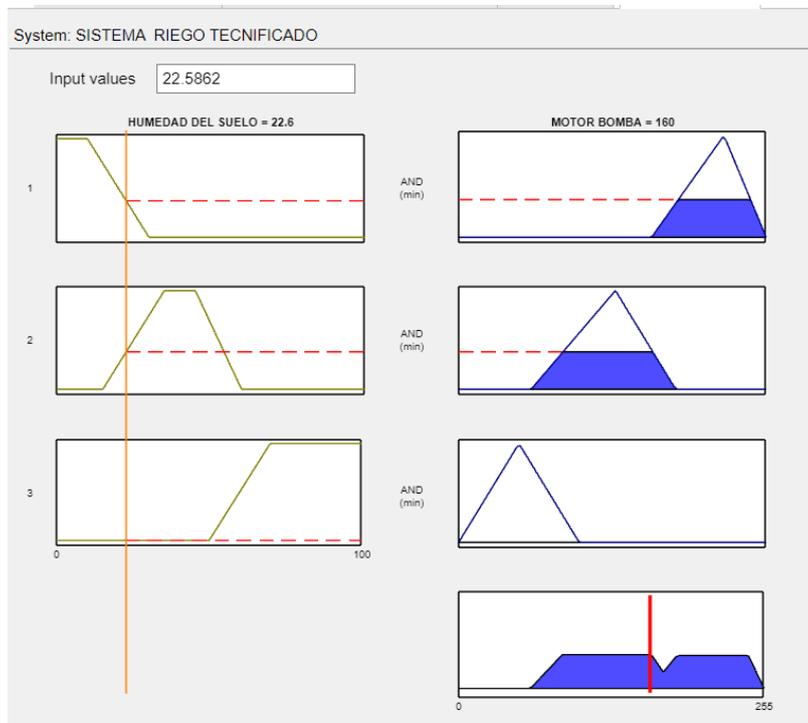
Name: rule1  
Weight: 1  
Connection:  And  Or

If  
DISTANCIA is CERCA

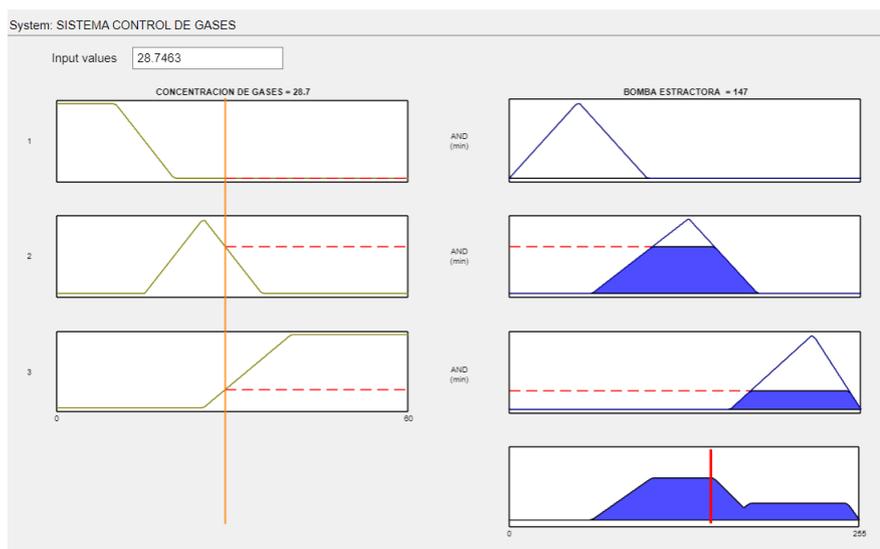
Then  
SERVO MOTOR is DERECHA

**Figura 86:**  
*Reglas de inferencia del sistema control de puertas*

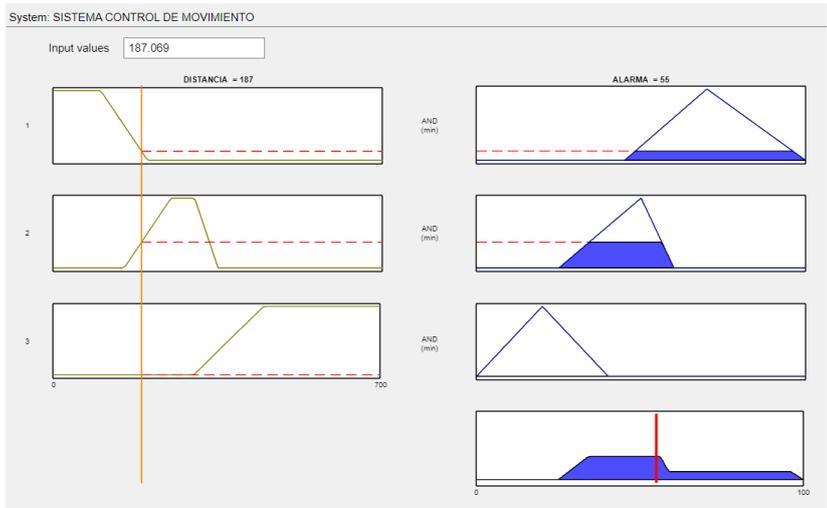
## ANEXO 10: SIMULACIÓN DE LOS OTROS SISTEMAS DIFUSOS CONSIDERADOS



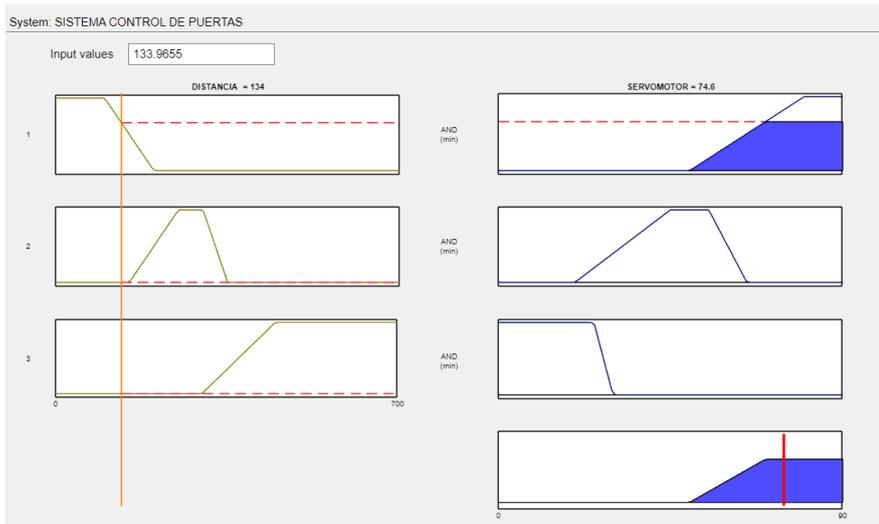
**Figura 87:**  
*Simulación del sistema riego tecnificado*



**Figura 88:**  
*Simulación del sistema control de gases*



**Figura 89:**  
*Simulación del sistema control de movimiento*



**Figura 90:**  
*Simulación del sistema Control de puertas*