

UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA MECATRÓNICA



**DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR
DIFUSO UTILIZANDO ARDUINO PARA LA
AUTOMATIZACIÓN DE UN MINI INVERNADERO DE
ROSAS**

**TESIS
PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO MECATRÓNICO**

PRESENTADA POR

Bach. GOMEZ PACCI, GIOVANNA NATALIA

Asesor: Ing. RIVAS LEÓN JAVIER HIPÓLITO

LIMA-PERÚ

2019

DEDICATORIA

Don't get me wrong

And you can speak your mind

But not on my time

(Billy Joel, 1978)

AGRADECIMIENTOS

Un conjunto de palabras no basta para expresar mi agradecimiento de manera idónea, pero hago el mejor intento para plasmarlo en el siguiente texto.

A mi familia cercana, mis padres, mi tía y mi hermana, por su apoyo emocional. Sin ellos esto no hubiera sido posible.

Al ingeniero Javier Rivas León, por brindarme su apoyo y guía durante el trabajo de tesis, así como algunas recomendaciones en la redacción.

Finalmente a todas aquellas personas que me brindaron sus ánimos en cualquier momento.

ÍNDICE GENERAL

RESUMEN	xix
ABSTRACT	xx
INTRODUCCIÓN	1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA	2
1.1.Presentación de la problemática	2
1.2.Formulación del problema	2
1.2.1.Problema General	2
1.2.2.Problemas Específicos	3
1.3.Objetivos.....	3
1.3.1.Objetivo general.....	3
1.3.2.Objetivos específicos	3
1.4.Alcance y limitaciones de la investigación.....	3
1.5.Justificación de la investigación	4
1.5.1.Justificación económica.....	4
1.5.2.Justificación técnica.....	4
1.6.Estructura de la tesis	5
CAPÍTULO II: MARCO TEÓRICO	7
2.1.Antecedentes de la Investigación.....	7
2.1.1.Antecedentes nacionales	7
2.1.2.Antecedentes Internacionales	7
2.2.Selección de variables.....	8
2.3.Bases teóricas.....	8
2.3.1.Características del invernadero y del cultivo	8
2.3.1.1.Concepto de invernadero.....	8
2.3.1.2.Orientación del invernadero	9
2.3.1.3.Estructura del invernadero	10

2.3.1.4.Material de la estructura.....	11
2.3.1.5.La cubierta del invernadero.....	12
2.3.1.6.Rosa miniatura.....	12
2.3.2.Diseño del control difuso de las variables	13
2.3.2.1.Variable humedad relativa y temperatura.....	14
2.3.2.2.Variable iluminación	16
2.3.2.3.Variable fertirriego	17
2.3.2.3.1.Frecuencia de riego.....	18
2.3.2.4.Variable nivel del recipiente	18
2.3.2.5.Lógica difusa	19
2.3.2.6.Controlador difuso utilizando Arduino	21
2.3.2.7.Controladores Arduino.....	25
2.3.2.8.Controlador Arduino Mega 2560 R3	25
2.3.2.9.Librería eFLL	27
2.3.3.Características de los sensores y actuadores elegidos	27
2.3.3.1.Concepto de sensor.....	27
2.3.3.2.Sensor para el control de humedad relativa y temperatura	27
2.3.3.3.Sensor para el control de la luminosidad	30
2.3.3.4.Sensor para el control de nivel del contenedor	30
2.3.3.5.Sensor para el fertirriego	31
2.3.3.6.Definición de actuador	32
2.3.3.7.Actuador para el control de humedad relativa y temperatura	32
2.3.3.8.Actuador para el control de luminosidad	33
2.3.3.9.Actuadores para el control de nivel.....	35
2.3.3.10.Actuador para el control del fertirriego.....	36
2.3.3.11.Goteros	36
2.3.3.12.Malla Raschel.....	37

2.3.3.13.Sistema Venturi	38
2.3.3.14.Fertilizante.....	39
CAPÍTULO III: DISEÑO Y SIMULACIÓN.....	40
3.1.Requerimientos generales	40
3.1.1.Requerimientos eléctricos y electrónicos de los sensores	40
3.1.2.Requerimientos eléctricos y electrónicos de los actuadores.....	40
3.1.3.Requerimientos de la estructura	40
3.2.Respuesta del requerimiento	41
3.3.Elementos a emplear	42
3.4.Software LabView	44
3.5.Control difuso de la temperatura y humedad relativa del aire	44
3.5.1.Ubicación de los componentes	46
3.5.2.Entradas y salidas del controlador difuso	46
3.5.3.Objetivos.....	47
3.5.4.Desarrollo	48
3.5.5.Resultados.....	48
3.5.6.Simulación final del controlador difuso en LabView.....	50
3.5.7.Conexiones electrónicas	54
3.6.Control de nivel del recipiente de irrigación	55
3.6.1.Ubicación de los componentes	55
3.6.2.Entradas y salidas del controlador difuso	56
3.6.3.Objetivos.....	56
3.6.4.Desarrollo	56
3.6.5.Resultados.....	60
3.6.6.Simulación final del controlador difuso en LabView.....	75
3.6.7.Conexiones electrónicas	77
3.7.Control difuso de la luminosidad.....	77

3.7.1. Ubicación de los componentes	78
3.7.2. Entradas y salidas del controlador difuso.....	78
3.7.3. Objetivos	78
3.7.4. Desarrollo	79
3.7.5. Resultados	79
3.7.6. Simulación final del controlador difuso en LabView	82
3.8. Control difuso del fertirriego	83
3.8.1. Ubicación de los componentes	83
3.8.2. Entradas y salidas del controlador difuso.....	84
3.8.3. Objetivos	84
3.8.4. Desarrollo	84
3.8.5. Resultados	87
3.8.6. Simulación final del controlador difuso en LabView	92
3.9. Muestreo de las variables que afectan al cultivo	93
3.10. Visualización de las variables a controlar.....	93
CAPITULO IV: IMPLEMENTACIÓN Y RESULTADOS	94
4.1 Proceso previo a la implementación	94
4.2 Conexiones generales	94
4.3 Control de la humedad relativa y temperatura.....	95
4.3.1. Prueba previa	95
4.3.2. Resultados de la prueba	96
4.4 Control de la luminosidad.....	97
4.4.1. Prueba previa	97
4.4.2. Resultados de la prueba	98
4.5 Control de nivel del recipiente.....	99
4.5.1. Prueba previa	99
4.5.2. Resultados de la prueba	99

4.6Control del fertirriego	103
4.6.1.Prueba previa	103
4.6.2.Resultados de la prueba	103
4.7Visualización de las variables en pantalla LCD	104
4.7.1.Prueba	104
4.7.2.Resultado de la prueba.....	104
4.8Programación del controlador difuso en Arduino.....	104
4.8.1.Descripción del problema	104
4.8.2.Algoritmo.....	105
4.8.3.Código fuente.....	105
4.9 Implementación final	108
4.9.1. Cultivo de la rosa miniatura	108
4.9.2. Estructura del invernadero	109
4.9.3. Circuito electrónico	111
4.9.4. Consideraciones sobre el circuito electrónico	112
4.9.5. Implementación del control de la humedad relativa y temperatura	112
4.9.6. Implementación del control de nivel del recipiente de irrigación	113
4.9.7. Implementación del control de luminosidad	120
4.9.8. Implementación del control del fertirriego.....	121
4.9.9. Implementación de la pantalla.....	125
4.10Resultados finales	128
4.11Costos.....	129
4.11.1.Costo equipamiento	129
4.11.2.Costo diseño.....	130
4.11.3.Costo energético	131
CONCLUSIONES	132
OBSERVACIONES.....	133

RECOMENDACIONES	134
REFERENCIAS BIBLIOGRÁFICAS.....	135
ANEXOS.....	142

ÍNDICE DE TABLAS

Tabla N° 1:	Variable independiente	8
Tabla N°2:	Variable dependiente	8
Tabla N° 3:	Valores climatológicos normales.....	10
Tabla N° 4:	Valor del SPV a una cierta temperatura.....	14
Tabla N°5:	Valores de la cantidad de lux en diferentes ambientes	17
Tabla N° 6:	Cuadro comparativo de controladores	26
Tabla N°7:	Cuadro comparativo de sensores	28
Tabla N°8:	Cuadro comparativo de actuadores.....	29
Tabla N° 9:	Características del ventilador SP06-08.....	33
Tabla N° 10:	Características del ventilador SYM BANG.....	33
Tabla N° 11:	Características de la bomba con filtro VENUSAQUA.....	37
Tabla N° 12:	Entradas y salidas del control de las variables temperatura y humedad relativa.....	46
Tabla N° 13:	Variación de la temperatura y humedad relativa con diferentes voltajes	47
Tabla N° 14:	Variación de la temperatura y humedad relativa con el voltaje máximo	47
Tabla N° 15:	Entradas y salidas del control del nivel del recipiente.....	56
Tabla N° 16:	Relación entre altura y volumen interno.....	60
Tabla N° 17:	Relación entre el volumen extraído por la bomba, la altura en centímetros y el tiempo transcurrido.	61
Tabla N° 18:	Altura vs segundos transcurridos.....	65
Tabla N° 19:	Caudal de la electroválvula 2.....	66
Tabla N° 20:	Tabla relacionando volumen, altura y tiempo de la electroválvula 2.	67
Tabla N° 21:	Relación entre la altura y segundos por transcurrir.	69
Tabla N° 22:	Características del control de la variable luminosidad	78

Tabla N° 23:	Características del control de la variable luminosidad	84
Tabla N° 24:	Definición de entradas y salidas del inyector	87
Tabla N° 25:	Recálculo de los segundos de las electroválvulas.....	102
Tabla N° 26:	Valores de salida del sensor MH	103
Tabla N° 27:	Materiales empleados en la construcción del invernadero	109
Tabla N° 28:	Cálculo del caudal aproximado de los goteros	123
Tabla N° 30:	Lista de materiales empleados en cada lazo de control y precios.....	129
Tabla N° 31:	Lista de materiales empleados en la estructura del invernadero y precios	130
Tabla N° 32:	Lista de materiales empleados en el método de cultivo y precios	130
Tabla N° 33:	Costo de diseño de ingeniería por horas	130
Tabla N° 34:	Costo de costo energético del sistema	131

ÍNDICE DE FIGURAS

Figura N° 1:	Estructura de la tesis	5
Figura N° 2:	Imagen de un invernadero.....	9
Figura N° 3:	Orientación de un invernadero.....	10
Figura N° 4:	Tipos de invernaderos.....	11
Figura N° 5:	Estructura del invernadero	11
Figura N° 6:	Ejemplo de rosa variedad miniatura	13
Figura N° 7:	Valores del DPV	15
Figura N° 8:	Relación entre temperatura y humedad relativa	16
Figura N° 9:	Elementos de un conjunto difuso.....	20
Figura N° 10:	Esquema general de un controlador difuso.....	22
Figura N° 11:	Esquema de funcionamiento de un sistema tipo Mamdani.....	22
Figura N° 12:	Forma de las reglas	23
Figura N° 13:	Funcionamiento de un sistema tipo Mamdani	25
Figura N° 14:	Arduino Mega 2560	27
Figura N° 15:	Comparación entre sensores DHT11 y DHT22.....	29
Figura N° 16:	Dispositivos usados para medir las variables	30
Figura N° 17:	Ecuación del módulo del sensor de humedad.....	32
Figura N° 18:	Ventilador SP06-08 y Ventilador SYM BANG	33
Figura N° 19:	Absorción de nutrientes por parte de las plantas	35
Figura N° 20:	Válvulas solenoides	35
Figura N° 21:	Ficha de la bomba y goteros regulables.....	36
Figura N° 22:	Malla Raschel de 80% de sombra.....	37
Figura N° 23:	Análisis de un inyector venturi	38
Figura N° 24:	Fertilizante para rosas	39
Figura N° 25:	Esquema del proceso de diseño	41
Figura N° 26:	Esquema del control del invernadero.....	42

Figura N° 27:	Bosquejo del invernadero experimental	43
Figura N° 28:	Vista isométrica del invernadero experimental	43
Figura N° 29:	Diagrama de control de la humedad relativa y temperatura	45
Figura N° 30:	Esquema de control y visualización de la humedad relativa y temperatura	45
Figura N° 31:	Componentes del control de humedad relativa y temperatura.....	46
Figura N° 32:	Conjuntos difusos del control de la RH y T.....	49
Figura N° 33:	Testeo del controlador difuso de la RH y T.....	50
Figura N° 34:	Reglas del controlador difuso del control de la RH y T	50
Figura N° 35:	Diagrama de bloques y panel frontal del subVI Ventilador	51
Figura N° 36:	Diagrama de bloques del control difuso de la RH y T.....	52
Figura N° 37:	Panel frontal del control difuso de la RH y T.....	53
Figura N° 38:	Pruebas de la simulación del control de la RH y T.....	54
Figura N° 39:	Diagrama de las conexiones del control de la humedad relativa y temperatura	55
Figura N° 40:	Componentes del control del nivel del recipiente.....	55
Figura N° 41:	Simulación de las piezas en el programa Solidworks.....	56
Figura N° 42:	Diagrama de flujo del control de nivel del recipiente.....	57
Figura N° 43:	Vista de la ventana del comando Combinar	58
Figura N° 44:	Simulación del volumen de agua	58
Figura N° 45:	Cálculo del volumen de agua.....	59
Figura N° 46:	Simulación de la bomba en Solidworks.....	60
Figura N° 47:	Altura vs Tiempo (bomba).....	61
Figura N° 48:	Relación entre la altura del recipiente y el volumen del recipiente.....	62
Figura N° 49:	Diagrama de bloques y el panel frontal del subVI de la función de extracción de la bomba	63

Figura N° 50:	Diagrama de bloques y panel frontal del subVI que relaciona el volumen y la altura	63
Figura N° 51:	Diagrama de bloques y panel frontal del subVI que relaciona el volumen y los segundos transcurridos.....	64
Figura N° 52:	Panel frontal mostrando los segundos transcurridos.....	65
Figura N° 53:	Altura vs Segundos	65
Figura N° 54:	Diagrama de bloques y panel frontal del subVI que relaciona los segundos transcurridos y la altura.....	66
Figura N° 55:	Relación entre la altura extraída y el tiempo	67
Figura N° 57:	Conjuntos difusos de la electroválvula 1 y testeo.....	70
Figura N° 58:	Conjuntos difusos de la electroválvula 2 y testeo.....	71
Figura N° 59:	Reglas de la electroválvula 2	71
Figura N° 60:	Panel frontal del control de nivel.....	72
Figura N° 61:	Parte I de diagrama de bloques del control de nivel.....	73
Figura N° 62:	Diagrama de bloques del control de nivel	74
Figura N° 63:	Simulación del control de nivel	75
Figura N° 64:	Pruebas de la simulación del control de nivel.....	76
Figura N° 65:	Diagrama de conexión del sensor ultrasónico	77
Figura N° 66:	Diagrama de conexión del sensor FS400A.....	77
Figura N° 67:	Componentes del control de luminosidad.....	78
Figura N° 68:	Diagrama de flujo del control de luminosidad.....	79
Figura N° 69:	Conjuntos difusos del control de la luminosidad y testeo	80
Figura N° 70:	Reglas del control difuso de la luminosidad.....	81
Figura N° 71:	Diagrama de bloques y el panel frontal del subVI del control de la luminosidad.....	81
Figura N° 72:	Prueba del control de la luminosidad.....	82
Figura N° 73:	Componentes para el control de fertirriego	83
Figura N° 74:	Recipiente para el fertilizante conectado a la bomba.	83

Figura N° 75:	Diagrama de flujo del control del fertirriego	85
Figura N° 76:	Análisis del inyector Venturi	86
Figura N° 77:	Gotero de 20 mL	87
Figura N° 78:	Gráfica humedad en la tierra vs segundos	88
Figura N° 79:	Diagrama de bloques y panel frontal de la función humedad de las macetas	89
Figura N° 80:	Reglas del control del fertirriego	89
Figura N° 81:	Conjuntos difusos del control del fertirriego y testeo.....	90
Figura N° 82:	Diagrama de bloques del control del fertirriego	91
Figura N° 83:	Panel frontal del control del fertirriego.....	92
Figura N° 84:	Simulación del control difuso del fertirriego.....	93
Figura N° 85:	Conexiones eléctricas generales para las pruebas.....	94
Figura N° 86:	Conexión para la prueba del ventilador	95
Figura N° 87:	Código para la prueba del ventilador	96
Figura N° 88:	Gráfica de la relación entre humedad relativa y temperatura	97
Figura N° 89:	Medidas de iluminancia con el sensor BH1750.....	98
Figura N° 90:	Prueba del sensor de caudal	99
Figura N° 91:	Código en Arduino del sensor FS400A G1.	100
Figura N° 92:	Valores del puerto serial del código del sensor de caudal.	100
Figura N° 93:	Lecturas de la distancia con el recipiente vacío y lleno.....	101
Figura N° 94:	Pruebas con las electroválvulas	102
Figura N° 95:	Prueba del sensor MH.....	103
Figura N° 96:	Prueba de la pantalla LCD 16x4.	104
Figura N° 97:	Diagrama de flujo para usar la librería eFLL.	106
Figura N° 98:	Declaración de las librerías de control fuzzy y del comando para crear un sistema difuso.	107
Figura N°99:	Creación de conjuntos difusos, asignación a variables de entrada o salida y creación de las reglas.....	107

Figura N° 100: Fuzzificación, desfuzzificación e impresión de datos.	108
Figura N° 101: Rosas miniatura.....	109
Figura N° 102: Estructura del invernadero	110
Figura N° 103: Estructura del invernadero con micas	110
Figura N° 104: Diagrama de flujo de la creación de la estructura.....	111
Figura N° 105: Diagrama de las conexiones de la implementación final.....	112
Figura N° 107: Conexiones de la implementación del control de nivel	113
Figura N° 108: Redefinición de los conjuntos difusos de la electroválvula 1	114
Figura N° 109: Redefinición de las reglas de la electroválvula 1	115
Figura N° 110: Testeo del controlador difuso de la electroválvula 1	115
Figura N° 111: Redefinición y testeo de los conjuntos difusos de la electroválvula 2 .	116
Figura N° 112: Redefinición de las reglas de la electroválvula 2.....	117
Figura N° 113: Bomba en funcionamiento	117
Figura N° 114: Conexiones eléctricas de las fuentes de alimentación	118
Figura N° 115: Conexiones eléctricas del módulo de relés	118
Figura N° 116: Lectura del sensor ultrasónico en el puerto serial.....	119
Figura N° 117: Inyector Venturi y bomba en funcionamiento	119
Figura N° 118: Sensor BH1750 colocado.....	120
Figura N° 119: Funcionamiento de los leds.....	120
Figura N° 120: Función analogWrite para el control de la luminosidad.....	121
Figura N° 121: Salida del sensor BH1750 empleando los leds	121
Figura N° 122: Redefinición de los conjuntos difusos del control del fertilizante	122
Figura N° 123: Redefinición de las reglas del control del fertilizante.....	122
Figura N° 124: Testeo del controlador difuso del fertilizante	122
Figura N° 125: Sensor MH colocado en la tierra.....	123
Figura N° 126: Prueba de los goteros	124

Figura N° 127: Funcionamiento de los goteros	124
Figura N° 128: Funcionamiento de un gotero	125
Figura N° 129: Conexiones de la pantalla	125
Figura N° 130: Prueba de las conexiones de la pantalla.....	126
Figura N° 131: Conexiones de la pantalla empleando los sensores	126
Figura N° 132: Plantaciones de rosa con oídio.....	127

ÍNDICE DE ANEXOS

Anexo I	Datasheet de controlador	142
Anexo II	Datos técnicos del DHT22	147
Anexo III	Especificaciones técnicas del sensor BH1750	148
Anexo IV	Especificaciones técnicas del sensor HC-SR04	149
Anexo V	Especificaciones técnicas del sensor FC-28	150
Anexo VI	Especificaciones técnicas de los módulos led	151
Anexo VII	Especificaciones técnicas de la electroválvula 1	152
Anexo VIII	Especificaciones técnicas de la malla Raschel	153
Anexo IX	Código para escribir valor analógico	154
Anexo X	Especificaciones técnicas del sensor de flujo FS400A-G1	156
Anexo XI	Código del sensor FS400A G1	157
Anexo XII	Código del sensor de humedad	159
Anexo XIII	Código de visualización de las variables	160
Anexo XIV	Código de visualización de la luminosidad	162
Anexo XV	Código simple del sensor ultrasónico	164
Anexo XVI	Código mejorado del sensor ultrasónico	166
Anexo XVII	Código del sensor de humedad del suelo	168
Anexo XVIII	Código final	169

RESUMEN

Esta tesis buscó demostrar que es posible diseñar e implementar invernadero automatizado de bajo costo para la producción de flores. Para ello aborda el diseño, simulación e implementación del controlador difuso de los lazos de control de las principales variables empleadas en la automatización de un mini invernadero de rosas miniatura, empleando el software Labview para la simulación y la plataforma Arduino para la implementación.

En primer lugar se realizó una recopilación de información. Se definieron los requerimientos para realizar los lazos de control, las cuales serían la humedad relativa y temperatura dentro del invernadero, el control de nivel del contenedor para el riego del cultivo, la luminosidad del invernadero y por último el fertirriego del cultivo. En base a ello se efectuó el planteamiento de la respuesta al requerimiento. Se realizó un algoritmo de control para cada variable y la simulación de cada uno fue realizada mediante el software LabView. Luego se realizó la prueba de cada lazo de control. Para la implementación se consideraron las características de la construcción del mini invernadero, con tal que pudiera albergar un pequeño cultivo de seis plantaciones de rosas y las características necesarias para realizar cada lazo de control. En segundo lugar se consideraron los resultados de la simulación para implementar cada uno.

Los resultados muestran que fue posible diseñar, simular e implementar un controlador difuso a bajo costo teniendo en cuenta las variables de humedad relativa y temperatura del aire, luminosidad, fertirriego y el control de nivel del recipiente usado para esto. Los sensores, actuadores y demás elementos usados fueron suficientes para lograr cada lazo de control. El costo energético total fue de 583.58 kw/h y el costo del controlador difuso implementado fue de S/ 465, pudiendo ser realizado con una inversión pequeña.

La tesis demostró que fue posible diseñar, simular e implementar el controlador difuso para un mini invernadero de rosas.

Palabras Claves: Arduino, Automatización, Control difuso, Invernadero

ABSTRACT

This thesis sought to demonstrate that it is possible to design and implement a low-cost automated greenhouse for the production of flowers. For this, it addresses the design, simulation, and implementation of the fuzzy controller of the control loops of the main variables used in the automation of a miniature mini roses greenhouse, using the Labview software for simulation and platform Arduino for the implementation.

In the first place there was a compilation of information. The requirements to perform the control loops were defined, which would be the relative humidity and temperature inside the greenhouse, the level control of the container for crop irrigation, the luminosity of the greenhouse and finally the fertirrigation of the crop. On this basis, the approach of the response to the request was reached. An algorithm of control for each variable was made and each simulation was performed using the LabVIEW software. The characteristics of the construction of the mini greenhouse were considered for the implementation, considering that it could hold a small crop of six plantations of roses and the necessary characteristics to perform each control loop. The simulation results were considered in second place to implement each one.

The results show that it was possible to design, simulate, and implement a low-cost fuzzy controller taking into account the variables of relative humidity and temperature of air, luminosity, fertigation and the level of the vessel used for this control. Sensors, actuators and other used elements were sufficient to achieve each control loop. The total energy cost was 583.58 kWh and cost of the diffuse driver implemented was s / 465, and can be done with a small investment

The thesis showed that it was possible to design, simulate and implement the controller for the mini rose greenhouse.

Keywords: Arduino, Automation, Fuzzy Control, Greenhouse.

INTRODUCCIÓN

La automatización de un invernadero es similar a la de cualquier proceso industrial. Cuando se trata de hacer tareas repetitivas o peligrosas para el ser humano, es lógico que se intente introducir algoritmos de control automatizado. El sector agrícola no es ajeno a este escenario, especialmente mediante la aplicación de sistemas de riego automatizados en cultivos al aire libre o en recintos cerrados. Estos últimos permiten el desarrollo de cultivos de una manera más óptima que los cultivos convencionales. Este tema es de interés debido al beneficio de emplear sistemas automatizados, los que permiten ahorrar tiempo y gastos inherentes a la operación de procesos de manera manual.

Sin embargo, a pesar de haberse desarrollado varios trabajos previos sobre automatización, la mayoría de éstos se enfocan en el empleo de algoritmos de control basados en el control PID o el ON/OFF. Últimamente se ha empezado a emplear la lógica difusa con éxito en diversos procesos, sean industriales o no. Sería provechoso evaluar su aplicación en la automatización de invernaderos.

Asimismo, Arduino es una plataforma de libre acceso eficiente para realizar muchas tareas automáticas de bajo costo. Además que la existencia de sensores y módulos especialmente diseñados para dichos controladores facilita bastante su uso. Existen numerosos trabajos que han empleado Arduino, así como la lógica difusa para diseñar e implementar diversos lazos de control con bastante éxito.

Por ello el presente proyecto tiene como objetivo demostrar la utilidad de la lógica difusa en la agricultura mediante la implementación de un prototipo experimental de un invernadero logrando el control adecuado del sistema, incluyendo su aplicación desde el riego y fertilización, el control de luminosidad, el control de nivel del contenedor del agua a emplear y el control de la humedad relativa y temperatura del aire dentro del invernadero. ¿Será posible diseñar, implementar y evaluar la automatización de un invernadero? Las consecuencias de este trabajo servirán a otras personas para hacer mejoras en dicha automatización, ya sea en el uso de mejores sensores, actuadores o mejoras a nivel de software.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1. Presentación de la problemática

La exportación de flores en el Perú tiene potencial para aumentar, sin embargo, “el Perú tiene solo 300 hectáreas cultivadas y solo exporta este tipo de productos por un valor de US\$10 millones”, resultados que son muy bajos comparados a otros países como Ecuador, país que cuenta con 2.500 hectáreas destinadas para el cultivo de flores y el valor de sus exportaciones es de aproximadamente US\$800 millones. Colombia posee 6.500 hectáreas cultivadas y exporta US\$1.400 millones en flores. (El Comercio, 2014). Una de las razones de ello es que no ha habido mucha inversión en la agro exportación de flores (Rojas, 2017).

Una manera de incrementar la producción es mediante el uso de invernaderos, el cual ha demostrado su utilidad al proteger las plantas del clima exterior y de los fenómenos climáticos que pudieran suceder. También, Reyna (2015) dice: “Facilita el control de plagas y enfermedades que afectan a los cultivos y que pueden ocasionar grandes pérdidas en la producción” (p.10). Asimismo, la automatización tiene varias ventajas como son el ahorro de tiempo y dinero para el usuario.

Por otra parte, el cultivo de rosas no es muy extendido debido a la falta de condiciones propicias para su desarrollo. En el Perú es difícil encontrar áreas ideales para su crecimiento (Miranda, O. ,2015).

Entonces sería bueno explorar si los agricultores y exportadores podrían incrementar su producción mediante la implementación de invernaderos automatizados de bajo costo.

1.2. Formulación del problema

La pregunta es: ¿Será posible diseñar e implementar la automatización de un mini invernadero de bajo costo para la producción de rosas?

1.2.1. Problema General

¿Cómo demostrar que es posible diseñar e implementar un invernadero automatizado de bajo costo para la producción de rosas?

1.2.2. Problemas Específicos

1. ¿Cómo lograr el control de las principales variables de un mini invernadero de rosas de bajo costo?
2. ¿Cómo verificar la correcta funcionalidad de los lazos de control diseñados?
3. ¿Cómo observar los resultados de los diferentes lazos de control del mini invernadero?

1.3. Objetivos

1.3.1. Objetivo general

Diseñar e implementar un mini invernadero automatizado de bajo costo para la producción de rosas.

1.3.2. Objetivos específicos

1. Diseñar el control de las principales variables de un invernadero basados en un controlador difuso utilizando Arduino.
2. Simular los diferentes lazos de control
3. Implementar los diferentes lazos de control para el cultivo de la rosa en un mini invernadero de bajo costo

1.4. Alcance y limitaciones de la investigación

El presente proyecto aborda el diseño, simulación e implementación de un controlador difuso utilizando Arduino para la automatización de un mini invernadero, por lo cual su alcance comprende:

- El diseño de los lazos de control de las principales variables que afectan en un invernadero.
- La simulación de cada lazo de control de los lazos de control de las principales variables que afectan en un invernadero
- La implementación de los lazos de control de las principales variables que afectan en un invernadero

Las limitaciones fueron las siguientes:

- Económica. Por ello se ha empleado el controlador Arduino en lugar de un PLC, el cual debido a su robustez haría frente sin problemas a condiciones climatológicas más diversas. De la misma manera para los sensores y actuadores utilizados, y la disponibilidad de ellos.
- Espacial. Está asociada al espacio disponible, por lo cual el invernadero experimental no puede ser de dimensiones grandes y el cultivo de prueba no puede ocupar mucho espacio.
- Existe una limitación en el diseño del control difuso, debido a que la librería de Arduino empleada para este fin sólo posee conjuntos difusos trapezoidales.

1.5. Justificación de la investigación

1.5.1. Justificación económica

La globalización y las altas exigencias de calidad de los productos agrícolas y de flores destinados a la exportación hacia mercados como el norteamericano, europeo y asiático; han creado la necesidad de buscar nuevas técnicas innovadoras, no solo para el mejoramiento de la calidad de los productos sino también de los procesos relacionados al tipo de cultivo; permitiendo tener un mejor monitoreo y control de las variables que afectan y/o se interrelacionan directamente con las plantaciones en cada uno de sus estados fenológicos, métodos de riego eficientes, tratamiento y manipulación tecnificada de acuerdo a cada especie.

1.5.2. Justificación técnica

La lógica difusa permite una forma de control basada en el conocimiento del experto en determinada área, sin necesidad de obtener un modelo matemático.

Arduino es una plataforma de software libre que permite crear desde robots hasta simples circuitos activadores de leds. Sus controladores no son tan robustos como un PLC, pero en su uso en la agricultura podría trabajar sin inconvenientes con una protección adicional.

LabView es una plataforma que permite diseñar sistemas inteligentes y maquinaria industrial. Su lenguaje de programación gráfico ayuda a crear y simular lazos de control rápidamente.

La rosa miniatura o rosal enano es un cultivo que se desarrolla en poco espacio, cuyo origen se deriva del Rosal de China Enano. Es resistente pero necesita un mayor control de plagas (Buczacki, 1996).

Es por ello que la presente tesis plantea la implementación de un controlador basado en lógica difusa implementado en Arduino y simulado en LabView para la automatización de un mini invernadero de rosas miniatura. Este proyecto beneficiará a los ingenieros agrónomos y agricultores al contar con una herramienta mecatrónica para tecnificar sus procesos: en la adquisición de datos y las diversas etapas relacionadas a la producción de flores y otras actividades agrícolas que requieran de este tipo de tecnología.

1.6. Estructura de la tesis

La tesis está dividida en Introducción, cuatro (04) capítulos, conclusiones, observaciones, recomendaciones, referencias bibliográficas y anexos. Véase figura N° 1.

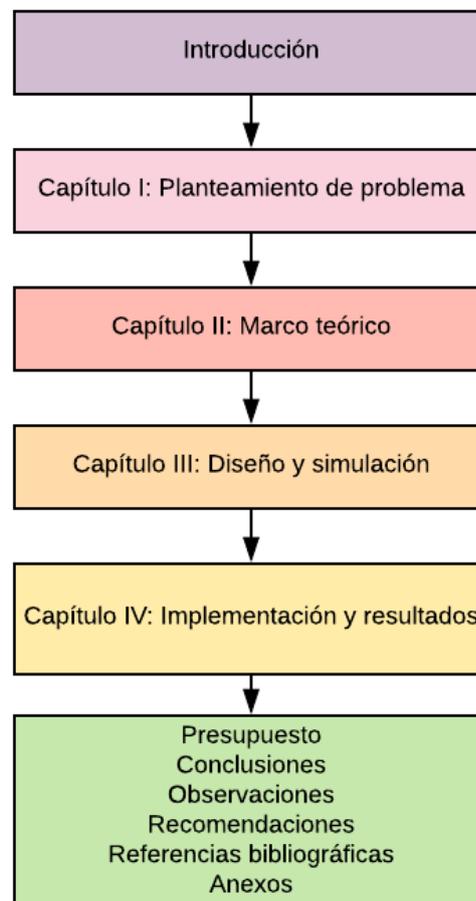


Figura N° 1: Estructura de la tesis

Fuente: Elaboración propia

Capítulo I. Planteamiento del problema

Este capítulo abarca la presentación de la problemática, la formulación del problema, los objetivos, las limitaciones y la estructura de la tesis.

Capítulo II. Marco teórico

En este capítulo se mencionan los antecedentes que sirvieron de guía para el presente trabajo, las variables y las bases teóricas en las que se apoya el proyecto.

Capítulo III. Diseño y simulación

En este capítulo se hace mención de los requerimientos del proyecto y el planteamiento de la solución.

Capítulo IV. Implementación y resultados

En este capítulo se describen la implementación de la solución y los resultados encontrados.

Presupuesto, Conclusiones, Observaciones, Recomendaciones, Bibliografía y Referencias y Anexos.

Se encuentran el presupuesto, las conclusiones, observaciones y recomendaciones de acuerdo al objetivo del proyecto. Por último la bibliografía y Referencias y los Anexos que sirven de apoyo a la tesis.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de la Investigación

A continuación se presentarán algunos de los antecedentes que sirvieron de apoyo para el presente proyecto de tesis.

2.1.1. Antecedentes nacionales

Reyna Huamán, C.E. (2015) de la Pontificia Universidad Católica del Perú, con su proyecto de tesis “Sistema automatizado para el monitoreo y control de humedad en un invernadero”, en el cual se automatizó el control de temperatura y humedad de un invernadero, concluyendo que es posible emplear un sensor DHT22 para un invernadero automatizado y que el sistema otorgaría cierta autonomía al operario.

Aliaga Mendoza, G. y Quispe Bolaños, P.R. (2015) de la Universidad Nacional de Huancavelica, con su proyecto de tesis “Sistema de control de la humedad Relativa para un invernadero utilizando El controlador lógico programable”, en el cual se realizó de humedad en un invernadero empleando un PLC, el cual llegó a la conclusión de que el control de humedad facilita el crecimiento biológico de la planta de tomate.

2.1.2. Antecedentes Internacionales

Se encuentran numerosos trabajos relacionados al tema, entre los más destacados se puede mencionar:

Ríos Suarez, E. R. (2015) de la Universidad Distrital Francisco José de Caldas, con su proyecto de tesis “Diseño e Implementación de un Sistema de Control de Humedad Relativa y absoluta en un Invernadero.”, en la que hace uso del control difuso de la variable humedad en un invernadero, así como la implementación del sistema, el cual concluyó que los sensores y actuadores empleados respondieron satisfactoriamente al control difuso.

Ramos-Fernández et al (2010) con su trabajo “Una estructura neurodifusa para modelar la evapotranspiración instantánea en invernaderos”, quienes emplearon un controlador difuso bajo el algoritmo Takagi-Sugeno en una planta, expresaron que mediante el cálculo del déficit de presión de vapor es posible controlar el microclima de un

invernadero y que es una variable valiosa para poder regular la cantidad de irrigación de un cultivo.

2.2. Selección de variables

Para responder a la pregunta planteada en la sección 1.2.1, se tienen las siguientes variables, descritas en la tabla N°1 y N° 2.

Tabla N° 1: Variable independiente

Variable	Dimensión	Indicador
Automatización de un mini invernadero de rosas	Diseño del control difuso de las variables	Características de la tarjeta del controlador
		Tipo de control
		VARIABLES A CONTROLAR
	Características de los sensores y actuadores elegidos	Características eléctricas
		Características electrónicas
		Precio

Fuente: Elaboración propia

Tabla N°2: Variable dependiente

Variable	Indicador
Controlador difuso utilizando Arduino	Consumo de energía en Kw/hr
	Funcionamiento de los lazos de control
	Precio total del controlador

Fuente: Elaboración propia

2.3. Bases teóricas

2.3.1. Características del invernadero y del cultivo

2.3.1.1. Concepto de invernadero

Un invernadero es un recinto cerrado acondicionado para proteger cultivos del clima externo. En el presente proyecto se tendrán en cuenta las características generales que debe tener el invernadero para albergar un cultivo de rosa miniatura en la ciudad de Lima, Perú. En la figura N° 2 se ilustra un invernadero de ejemplo.



Figura N° 2: Imagen de un invernadero

Fuente: <http://sistemaagricola.com.mx/blog/invernadero-exitoso-productividad-clave/>

Sobre su construcción, Espinosa, P. y Espinosa, L. (s.f.) comentan:

Un invernadero rústico no requiere de equipo sofisticado para su construcción. En un invernadero rústico se pueden cultivar flores, hortalizas y otras plantas, que le permitan al productor lograr un mayor ingreso familiar en la unidad de producción; es una alternativa que se puede realizar en pequeñas superficies dentro de su traspatio con buenos resultados.

2.3.1.2. Orientación del invernadero

Según Reyna (2015):

La radiación y la concentración de CO₂ influyen en el proceso de fotosíntesis. Más concentración de CO₂, proporciona más fotosíntesis; sin embargo, concentraciones superiores pueden resultar tóxicas para los cultivos. Por otro lado, la tasa de absorción de CO₂ es proporcional a la cantidad de luz recibida. A mayor luminosidad en el interior del invernadero se debe aumentar la temperatura, la humedad y el CO₂, para que la fotosíntesis sea máxima. Para mejorar la luminosidad natural se usan materiales de cubierta con buena transparencia, y que permitan recibir la mayor radiación solar posible. Los invernaderos se construyen mayormente en dirección norte- sur, de esta manera se tiene una mejor distribución de la luz en las plantas. (p.3)

Al seleccionar la mejor orientación del invernadero, en combinación con el material elegido para la cubierta, se toma en cuenta la mayor radiación solar posible que podría recibir el cultivo. En la figura N° 3 se ve la orientación de un invernadero.

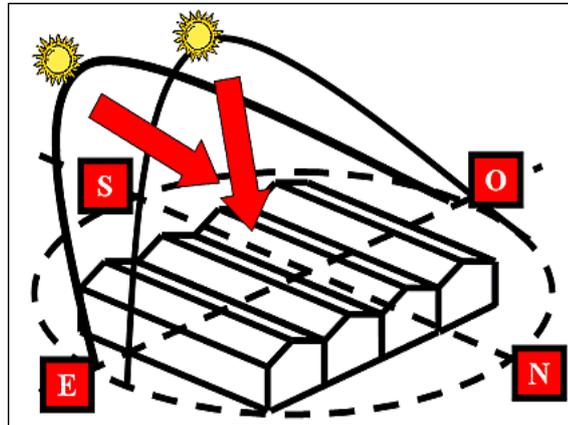


Figura N° 3: Orientación de un invernadero

Fuente: http://www.infoagro.com/industria_auxiliar/invernaderos_raspa_amagado.htm

Lima es la capital de la república del Perú, la cual tiene una probabilidad alta de presentar una temperatura máxima y mínima por encima del valor normal para el 2018. Asimismo posee una probabilidad de poseer lluvias normales para el mismo periodo. Sus valores climatológicos se describen en la tabla N°3.

Tabla N° 3: Valores climatológicos normales

Característica	Valor
Temperatura	24 °C
Humedad relativa	84%
Radiación ultravioleta	Extremadamente alta (14)

Fuente: SENAMHI

2.3.1.3. Estructura del invernadero

En general existen dos tipos básicos de invernaderos modulares que son: el de dos aguas y el de techumbre parabólica o de arco parabólico. En la figura N° 4 se ven algunos tipos de invernaderos más usados, como el parabólico tipo túnel, el parabólico tipo capilla y el de a dos aguas.

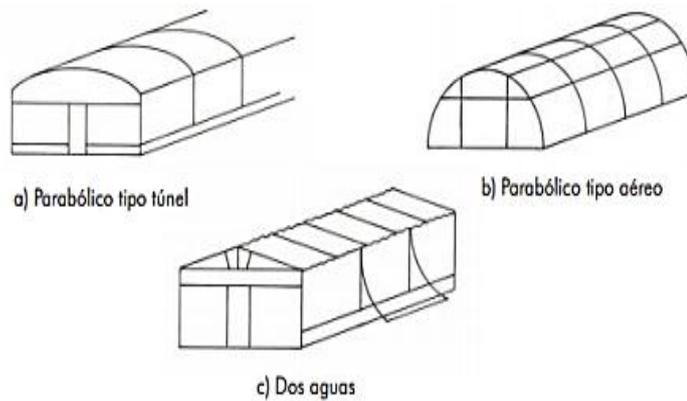


Figura N° 4: Tipos de invernaderos.

Fuente:

<http://www.sagarpa.gob.mx/desarrolloRural/Documents/fichasaapt/Invernadero%20R%C3%BAstico.pdf>.

En la presente tesis se ha considerado un invernadero rústico a dos aguas de similares características a la siguiente figura N°5.

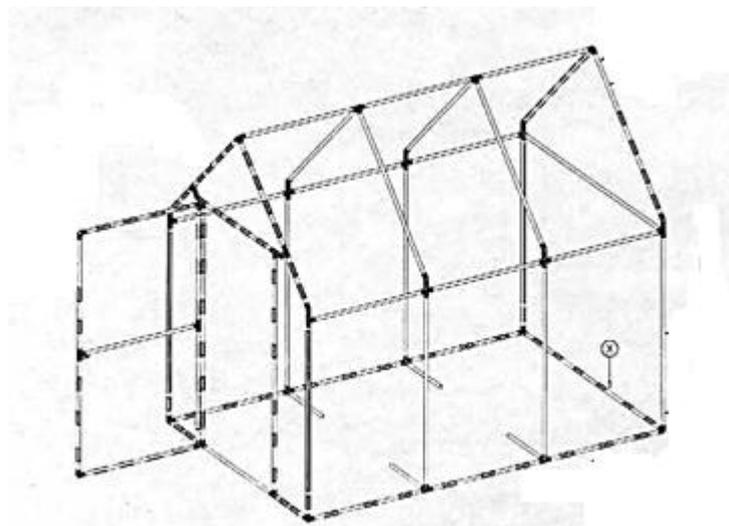


Figura N° 5: Estructura del invernadero

Fuente: <http://www.littlegreenhouse.com/greenhouse.shtml>

Elaborado con tubos y uniones PVC de 1 pulgada. Ciertas uniones tuvieron que ser creadas para poder conectar todas las tuberías correspondientes.

2.3.1.4. Material de la estructura

Los materiales más comunes que constituyen un invernadero son la madera y el hierro o acero, todos asentados en cuerpos de concreto o de ladrillo (Espinosa, P. y Espinosa,

L., s.f.). La madera es sencilla de trabajar y conseguir, mientras que el fierro tiene una resistencia y duración mayores.

2.3.1.5. La cubierta del invernadero

La cubierta de un invernadero es uno de los componentes más importantes a la hora de satisfacer las necesidades de cultivo. En ella factores como transparencia, retención de calor, el rendimiento térmico, flexibilidad, el envejecimiento o la resistencia al fuego son factores decisivos para escoger un buen material.

De acuerdo con Espinosa, P. y Espinosa, L (s.f.):

El material a utilizar deberá garantizar que proporcione el llamado "efecto de invernadero", retención de calor, rendimiento térmico, transparencia a la radiación solar, capacidad de retención a las radiaciones de onda larga emitidas por el suelo durante la noche y bajo costo. Los materiales que se pueden utilizar son fibra de vidrio y películas de plástico.

2.3.1.6. Rosa miniatura

a) Definición

De acuerdo con Esmeral (2011):

La rosa es una planta arbustiva, de porte abierto, ramas leñosas y espinosas. Generalmente, sus tallos son de color verde y a veces teñido de marrón rojizo con espinas más o menos desarrolladas según la variedad, pueden ser arqueadas, encorvadas, aladas, afiladas o cilíndricas (p.18).

El rosal miniatura es una variedad de rosa de maceta que se consigue por micro multiplicación. En la figura N° 6 se observa un rosal miniatura en maceta.



Figura N° 6: Ejemplo de rosa variedad miniatura

Fuente: <http://archivo.infojardin.com/tema/ficha-de-rosal-miniatura-o-rosal-mini.56219/>

b) Características del cuidado de la rosa miniatura

“La rosa miniatura debe ser ubicada en el interior. Su riego debe ser cuantioso y como abono debe emplearse un fertilizante en el agua de riego” (Infojardín®, 2007).

2.3.2. Diseño del control difuso de las variables

“En general se debe muestrear temperatura, humedad del aire y del suelo, nivel de CO₂, luminosidad, niveles de Ph y de conductividad eléctrica del suelo.” (Alvarado-Moya, P., González, A. y Villasenor-Gonzalez, L., 2008, p.2).

Según EasyVeep (2017):

Para la automatización de un invernadero se emplea un sensor para detectar si la luz natural es muy débil, dos sensores para detectar si la humedad relativa está por encima del valor máximo o mínimo, dos sensores para detectar si la humedad del suelo está por encima o por debajo del valor mínimo y tres sensores para detectar si la temperatura está por encima por mínimo, es óptima o encima del máximo.

En el caso del invernadero experimental, bastará con un sensor que mida cada variable. Se ha considerado realizar el control de la humedad relativa y la temperatura, el control de nivel del contenedor de agua, el control de la luminosidad y el control del fertirriego.

2.3.2.1. Variables humedad relativa y temperatura

“El déficit de presión de vapor puede utilizarse para evaluar la amenaza de enfermedades, el potencial de condensación y las necesidades de un cultivo bajo invernadero.” (Andrade. O., Castillo, R. y Laurentin, H.,s.f.).

DPV es la diferencia (déficit) entre la cantidad de agua en el aire (en forma de vapor) y la cantidad de humedad que puede acomodar cuando está saturado de agua (vapor). DPV funciona como un práctico indicador del potencial de condensación al cuantificar cuan cerca está el aire en el invernadero de su punto de saturación. (Andrade. O., Castillo, R. y Laurentin, H., s.f.).

La tabla N° 4 muestra los valores de la presión del vapor saturado a diferentes temperaturas.

Tabla N° 4: Valor del SPV a una cierta temperatura

Temperatura (°C)	SVP (Pa)	Temperatura (°C)	SVP (Pa)
0	611	21	2486
1	657	22	2643
2	706	23	2809
3	758	24	2983
4	813	25	3167
5	872	26	3361
6	935	27	3565
7	1002	28	3779
8	1073	29	4005
9	1148	30	4242
10	1228	31	4492
11	1312	32	4754
12	1402	33	5029
13	1497	34	5318
14	1598	35	5621
15	1705	36	5940
16	1818	37	6273
17	1937	38	6623
18	2064	39	6990
19	2197	40	7374
20	2338	41	7776

Fuente: Murray F. W.

El DPV es igual a la presión de vapor saturado (PVS) menos la presión actual de vapor (Vactual), siendo (SVP - Vactual), resultando la ecuación N° 1.

$$DPV = \left(1 - \frac{RH}{100}\right) * SVP \dots\dots\dots (1)$$

Donde RH es la humedad relativa, y SVP es la presión de vapor saturado. Los valores óptimos para una cierta humedad y temperatura se hallan en la zona verde de la figura N° 7.

TEMP		RELATIVE HUMIDITY													
°C	°F	100%	95%	90%	85%	80%	75%	70%	65%	60%	55%	50%	45%	40%	35%
15	59	0.0	0.8	1.7	2.5	3.4	4.2	5.1	5.9	6.8	7.6	8.5	9.4	10.2	11.1
16	61	0.0	0.9	1.8	2.8	3.7	4.6	5.5	6.4	7.3	8.2	9.1	10.0	10.9	11.8
17	63	0.0	1.0	2.0	2.9	3.9	4.9	5.8	6.8	7.8	8.8	9.7	10.6	11.6	12.6
18	64	0.0	1.0	2.0	3.1	4.1	5.1	6.2	7.2	8.2	9.3	10.3	11.3	12.4	13.4
19	66	0.0	1.1	2.2	3.3	4.4	5.5	6.6	7.7	8.8	9.9	11.0	12.1	13.2	14.3
20	68	0.0	1.2	2.4	3.5	4.7	5.9	7.0	8.2	9.4	10.6	11.7	12.8	14.0	15.2
21	70	0.0	1.2	2.4	3.7	4.9	6.2	7.4	8.6	9.9	11.1	12.4	13.7	14.9	16.1
22	72	0.0	1.3	2.6	3.9	5.3	6.6	7.9	9.2	10.5	11.9	13.2	14.5	15.8	17.2
23	73	0.0	1.4	2.8	4.2	5.6	7.0	8.5	9.9	11.3	12.7	14.1	15.4	16.8	18.2
24	75	0.0	1.5	3.0	4.5	5.9	7.4	8.9	10.4	11.9	13.4	14.9	16.4	17.9	19.4
25	77	0.0	1.6	3.2	4.8	6.4	8.0	9.5	11.1	12.7	14.3	15.9	17.4	19.0	20.5
26	79	0.0	1.7	3.4	5.1	6.7	8.4	10.1	11.8	13.4	15.1	16.8	18.4	20.1	21.8
27	81	0.0	1.8	3.5	5.3	7.1	8.9	10.7	12.4	14.2	16.0	17.8	19.6	21.3	23.1
28	82	0.0	1.9	3.8	5.7	7.6	9.5	11.4	13.3	15.1	17.0	18.9	20.7	22.6	24.5
29	84	0.0	2.0	4.0	6.0	8.0	10.0	12.0	14.0	16.0	18.0	20.0	22.1	24.1	26.1
30	86	0.0	2.1	4.2	6.4	8.5	10.6	12.7	14.8	17.0	19.1	21.2	23.3	25.4	27.5
31	88	0.0	2.2	4.5	6.7	9.0	11.2	13.4	15.7	17.9	20.2	22.4	24.6	26.9	29.1
32	90	0.0	2.4	4.7	7.1	9.5	11.9	14.2	16.6	19.0	21.3	23.7	26.1	28.4	30.8
33	91	0.0	2.5	5.0	7.5	10.0	12.5	15.0	17.6	20.1	22.6	25.1	27.6	30.1	32.6
34	93	0.0	2.7	5.3	8.0	10.6	13.3	15.9	18.6	21.2	23.9	26.5	29.2	31.8	34.5

Just4Growers.com

Figura N° 7: Valores del DPV

Fuente: <http://www.just4growers.com/stream/temperature-humidity-and-c02/vapor-pressure-deficit-the-hidden-force-on-your-plants.aspx>

Existe una relación entre la humedad relativa del aire y la temperatura. La temperatura que se mide con los termómetros, sean analógicos o digitales, se denomina temperatura de bulbo seco. Dicha temperatura tiene una relación inversamente proporcional con lo

que llamamos humedad relativa en el aire (Bijlo, 2014). En la figura N° 8 se aprecia gráficamente esta relación.

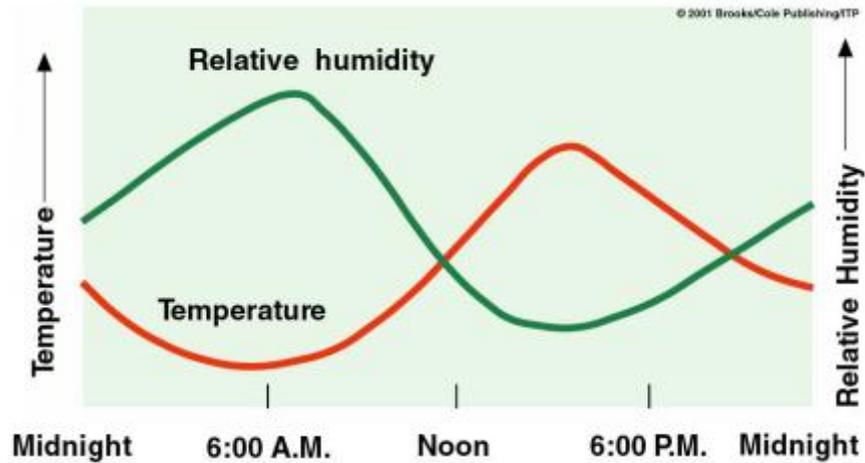


Figura N° 8: Relación entre temperatura y humedad relativa

Fuente: <http://meteo.fisica.edu.uy/Materias/TICA/TemperaturaYhumedad.pdf>

2.3.2.2. Variable iluminación

“La iluminación incide en el desarrollo de la planta y en la producción. A mayor iluminación se aumenta el follaje, se mejora la calidad, el color y el tamaño.” (Esmeral, 2011, p.20). En la tabla N° 5 se observan los valores de luminosidad en lux para diferentes ambientes.

Tabla N °5: Valores de la cantidad de lux en diferentes ambientes

Condición	Iluminación	
	FC	LUX
Luz solar	10,000	107,527
Luz del día	1,000	10,752.7
Día nublado	100	1,075.3
Día muy oscuro	10	107.53
Crepúsculo	1	10.75
Crepúsculo intenso	.1	1.08
Luna llena	.01	.108
Luna menguante	.001	.0108
Luz de las estrellas	.0001	.0011
Noche nublada	.00001	.0001

Fuente: Iluminación

2.3.2.3. Variable fertirriego

El sistema de riego será uno por goteo. Según DIY START (2018) comenta, el agua de un sistema de goteo es absorbida directamente por las raíces. Por lo tanto, las hojas no se mojen y por lo tanto el riesgo de daño y la enfermedad es considerablemente menor. Un sistema de riego por goteo puede utilizar cualquiera de los emisores o mangueras de remojo.

Existen dos tipos de riego por goteo.

a) Goteo emisor del sistema de riego

“El sistema emisor consta de una manguera de agua principal grande que tiene pequeños tubos que se extiende a las plantas que requieren riego. Las mangueras más pequeñas tienen boquillas que proporcionan el agua a la planta.”(DIY START, 2018).

b) Goteo de remojo del sistema de riego

Según DIY START (2018):

El sistema de remojo consiste en una manguera de agua principal que tiene agujeros pequeños que entregan el agua a la planta. Los agujeros son a lo largo de toda la manguera y están hechos de lona, plástico o

caucho negro. El tubo de goma negro tiene poros pequeños que permiten que el agua gotee lentamente a lo largo de la manguera. Las mangueras de remojo deben ser enterrados bajo tierra.

El gotero se caracteriza por brindar un mejor empleo del agua de riego o fertirriego. Existen los goteros autocompensantes y los no autocompensantes (Mundoriego®, 2018).

a) Goteros autocompensantes

Los goteros autocompensantes dan un caudal constante porque requieren de una presión mínima para funcionar. Se obturan con facilidad (Mundoriego®, 2018).

b) Goteros no autocompensantes

Los goteros no autocompensantes dan el caudal en función de la presión, por lo que aunque nos dará un caudal bastante inferior al que marca el propio gotero, nos va a permitir regar muy decentemente.(Mundoriego®, 2018)

La fertirrigación es el proceso de llevar a las plantas los nutrientes y el agua que necesitan para producir un cultivo de calidad.

La metodología a seguir es realmente lineal. Un sensor mide la humedad del suelo, si se vuelve muy seca, el cultivo es automáticamente irrigado.

2.3.2.3.1. Frecuencia de riego

“El riego deberá ser moderado, evitando que el agua sobrante se quede en un plato. El exceso de agua provoca la podredumbre de sus raíces.” (floresyplantas.net, 2014)

En el presente proyecto de tesis se ha considerado una programación de riego de dos veces por semana.

2.3.2.4. Variable nivel del recipiente

Se ha considerado realizar el control de nivel del recipiente que se empleará para realizar el fertirriego. La finalidad será mantener el recipiente lleno aun cuando la bomba esté activada y esté extrayendo el agua del recipiente.

2.3.2.5. Lógica difusa

“La lógica difusa es una forma matemática para la representación de definiciones vagas. Los conjuntos difusos son generalizaciones de la teoría de conjunto convencional”. (Kurokawa y Minussi, 1999, p.43)

Tradicionalmente los procesos han sido controlados utilizando técnicas de control convencional, pero la lógica difusa se ha empleado con éxito para reducir el esfuerzo para realizar control, haciendo que se pueda incrementar la complejidad de los sistemas, y por consiguiente de las aplicaciones. (Trillas, E. y Gutiérrez, J., 1992).

Existe una transición continua entre lo falso y lo verdadero que mediante principios matemáticos puede servir para modelar información basada en grados de membresía o pertenencia (Castro-Montoya, Vera-Monterrosas y Quintana-Silva, 2004).

a) Variables lingüísticas

Las variables lingüísticas son, en palabras de Zadeh, variables cuyos valores no son números sino palabras o frases en un lenguaje natural o artificial.

Un ejemplo de variable lingüística sería “Distancia”, la cual se descompone habitualmente en un conjunto de términos. Por ejemplo el mostrado en la ecuación N° 2.

$$Distancia = \{cerca, apropiada, lejos\} \dots\dots\dots (2)$$

Cada término tiene asociado un conjunto difuso. Por ejemplo el término *cerca* puede tener asociado el conjunto [1 cm, 3 cm].

b) Conjuntos difusos

Zadeh (1965) describió que “un conjunto difuso es un conjunto caracterizado por una función de pertenencia, la cual asigna a cada objeto un grado de pertenencia que varía entre cero y uno” (Citado en Pérez y León, 2007, p.15).

Según Castro-Montoya et al (2004) un conjunto difuso A en X se define como describe la expresión de la ecuación N° 3.

$$A = \{(x, \mu_A(x))\} | x \in X \dots\dots\dots (3)$$

Siendo $\mu_A(x)$ la función de pertenencia de X en A, la cual denota el grado en que x se encuentra en A. En la figura N° 9, se observa un conjunto difuso.

De acuerdo con Castro-Montoya et al (1999):

La función $\mu_A(x)$ asigna un valor real entre 0 y 1, que representa el grado de pertenencia del valor. Mientras esté más cerca al valor unitario, mayor es el grado de pertenencia. La forma de la función puede ser triangular, la más empleada, así como trapezoidal o sigmoideas, siendo estas funciones convexas. (p.16)

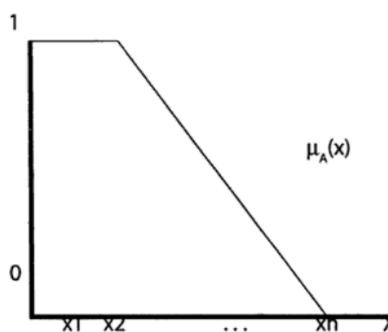


Figura N° 9: Elementos de un conjunto difuso

Fuente: Pérez y León (2004)

c) Operaciones con conjuntos difusos

Operaciones similares a AND, OR y NOT son algunos de los operadores más importantes de conjuntos difusos. Suponiendo que A y B son dos conjuntos difusos con las funciones de pertenencia designadas por $\mu_A(x)$ y $\mu_B(x)$, entonces:

- Operador AND: Está definida como indica la ecuación N° 4.
 - $\mu(x) = \min\{\mu_A(x), \mu_B(x)\}, x \in X \dots \dots \dots (4)$
- Operador OR: Está definida como indica la ecuación N° 5.
 - $\mu(x) = \max\{\mu_A(x), \mu_B(x)\}, x \in X \dots \dots \dots (5)$
- Operador NOT: Está definida como indica la ecuación N° 6.
 - $\mu(x) = 1 - \mu(x), x \in X \dots \dots \dots (6)$
- Relación difusa: La función difusa R de A en B satisface la siguiente regla de la composición y es caracterizado por la función $\mu_R(x, y)$. Está expresada en la ecuación N° 7.

$$\mu(x) = \max\{\min\{\mu_R(x, y), \mu_B(x)\}\}, x \in X \dots \dots \dots (7)$$

d) Implicaciones y composición de relaciones

Siendo A y B variables lingüísticas, las reglas de implicación sirven para trasladar una condición difusa, tal como “Si X es A entonces Y es B” a una relación difusa, que relaciona ambas variables lingüísticas.

De estas reglas de implicación, las más empleadas han sido las propuestas por Zadeh y Mamdani. Esta última constituye la más usada en aplicaciones de sistemas difusos. (Pérez y León, 2007).

$$p \rightarrow q = p \wedge q \dots\dots\dots(8)$$

La ecuación N° 8 ilustra esta regla de implicancia.

2.3.2.6. Controlador difuso utilizando Arduino

a) Controlador difuso

Es un sistema electrónico e informático que sirve de soporte físico para realizar inferencias con lógica difusa.

Según Pérez y León (2007), los controladores difusos están basados casi en su totalidad por modelos de inferencia max-min con la regla de implicación de Mamdani, mostrada en la ecuación N° 8. Su diseño depende mucho de las funciones de pertenencia. Sin embargo, es posible obtener una respuesta satisfactoria del modelo incluso en el primer diseño.

La arquitectura general de un Controlador basado en lógica difusa (CBLD) monolazo, para una sola entrada y una sola salida se aprecia en la figura N° 10. (Acosta, s.f.).

El CBLD es un caso particular de un controlador.

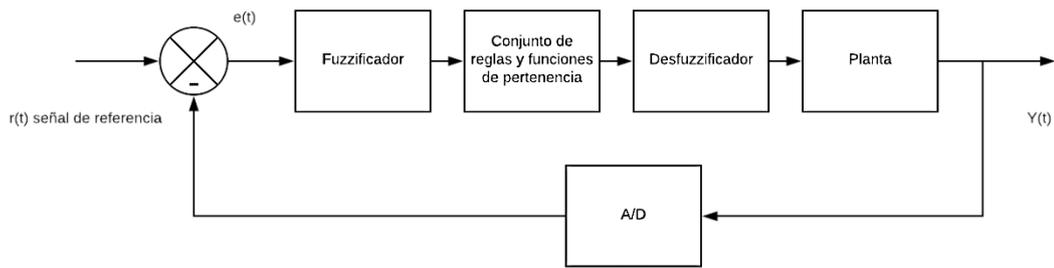


Figura N° 10: Esquema general de un controlador difuso

Fuente: Elaboración propia

De acuerdo con Castro-Montoya et al (2004), la gran ventaja del control difuso reside en el hecho de que no es necesario conocer el modelo de sistema. Es suficiente saber cómo el sistema se comporta. Conociendo esta información se elabora el conjunto de reglas.

b) Controlador difuso tipo Mamdani

Un controlador tipo Mamdani es aquel que cuenta con fuzzificador y defuzzificador. Para este trabajo se ha optado por emplear el algoritmo Mamdani, “por la fácil inclusión del conocimiento de expertos en forma de reglas lingüísticas para la manipulación” de las variables a controlar (Salazar, 2012, p. 59).

Las partes de un controlador difuso tipo Mamdani se muestran en la figura N° 11.

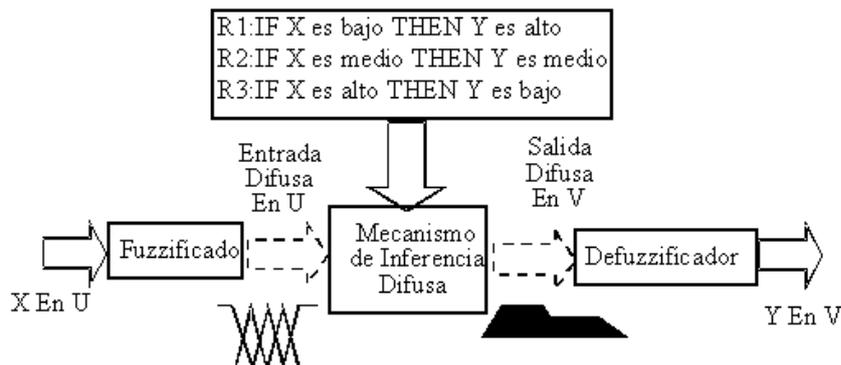


Figura N° 11: Esquema de funcionamiento de un sistema tipo Mamdani

Fuente: http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html

-Fuzzificador

Respecto al fuzzificador, de acuerdo con López (2003):

La entrada normalmente es un valor numérico proveniente de un sensor. La función del fuzzificador es convertir dicho valor numérico a un valor difuso para que pueda ser procesado por el sistema difuso. Estos valores difusos son los niveles de pertenencia de los valores de entrada a los diferentes conjuntos difusos en los cuales se ha dividido el universo de discurso de las diferentes variables de entrada al sistema.

-Mecanismo de inferencia difusa

Respecto al mecanismo de inferencia difusa, según López (2003):

Teniendo los diferentes niveles de pertenencia arrojados por el fuzzificador, los mismos deben ser procesados para generar una salida difusa. La tarea del sistema de inferencia es tomar los niveles de pertenencia y apoyado en la base de reglas generar la salida del sistema difuso.

-Base de reglas

El tercer componente, la base de reglas, es la manera que tiene el sistema difuso de guardar el conocimiento lingüístico que le permiten resolver el problema para el cual ha sido diseñado. Estas reglas son del tipo IF-THEN (López, 2003).

Una regla de la base de reglas o base de conocimiento tiene dos partes, el antecedente y la conclusión como se observa en la figura N° 12.



Figura N° 12: Forma de las reglas

Fuente: http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html

En un sistema difuso tipo Mamdani tanto el antecedente como el consecuente de las reglas están dados por expresiones lingüísticas.

-Desfuzzificador

Respecto al desfuzzificador, según López (2003):

La salida que genera el mecanismo de inferencia es una salida difusa, lo cual significa que no puede ser interpretada por un elemento externo que solo manipule información numérica. Para lograr que la salida del sistema difuso pueda ser interpretada por elementos que solo procesen información numérica, hay que convertir la salida difusa del mecanismo de inferencia; este proceso lo realiza el fuzzificador. La salida del mecanismo de inferencia es un conjunto difuso resultante, para generar la salida numérica a partir de este conjunto existen varias opciones como el Centro de Gravedad, los Centros Promediados entre otros.

La ecuación N° 9 muestra el cálculo siguiendo el método del Centro de Gravedad (López, 2003).

$$y = \frac{\sum_i b_i \int u(i)}{\sum_i \int u(i)} \dots\dots\dots (9)$$

La ecuación N° 10 muestra el cálculo siguiendo el método del Centro Promediados (López, 2003).

$$y = \frac{\sum_i b_i u_{premisas}(i)}{\sum_i u_{premisas}(i)} \dots\dots\dots (10)$$

En la figura N° 13 se muestra de manera más detallada como es el funcionamiento de un sistema difuso Mamdani. En control, es necesario que el sistema difuso entregue como salida un valor numérico, para calcularlos se usan las ecuaciones 9 y 10.

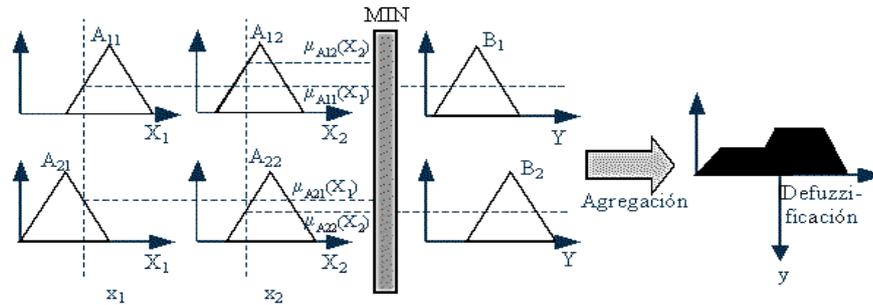


Figura N° 13: Funcionamiento de un sistema tipo Mamdani

Fuente: http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html.

-Funciones de pertenencia

Respecto de la forma que tendrán las funciones de pertenencia, no existen diferencias apreciables en los resultados de control entre las diferentes posibilidades (triangulares, trapezoidales, gausseanas, sigmoideas). Por ello la mayoría de los productos comerciales suelen ofrecer las trapezoidales, entendiendo la triangular como un caso particular, ya que son las de más fácil implementación (Acosta, s.f, p.10).

2.3.2.7. Controladores Arduino

“Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.” (Amangandi, 2012). En la tabla N° 6, se presenta un breve cuadro comparativo entre los principales controladores y el motivo por el cual se eligió el Arduino Mega 2560 R3. Más características del controlador pueden verse en el anexo I, correspondiente al datasheet del controlador.

2.3.2.8. Controlador Arduino Mega 2560 R3

Según arduino.cl® (2018):

Arduino Mega 2560 R3 es una tarjeta de desarrollo open-source construida con un microcontrolador modelo Atmega2560 que posee pines de entradas y salidas (E/S), analógicas y digitales. El Arduino Mega tiene 54 pines de entradas/salidas digitales (14 de las cuales pueden ser utilizadas como salidas PWM), 16 entradas analógicas, 4 UARTs

(puertos serial por hardware), cristal oscilador de 16MHz, conexión USB, jack de alimentación, conector ICSP y botón de reset. Arduino Mega incorpora todo lo necesario para que el microcontrolador trabaje; simplemente conéctalo a tu PC por medio de un cable USB o con una fuente de alimentación externa (9 hasta 12VDC). El Arduino Mega es compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO.

Tabla N° 6: Cuadro comparativo de controladores

CUADRO COMPARATIVO DE CONTROLADORES				
	ARDUINO UNO REV3	ARDUINO MEGA 2560 REV3	ARDUINO NANO	Elección
Características principales	Microcontrolador: ATmega328p Led: Pin 13 Conexión USB y por jack Cristal de 16 MHz	Microcontrolador: ATmega2560 Led: Pin 13 Conexión USB y por jack Cristal de 16 MHz	Microcontrolador: ATmega328 Arquitectura: AVR Led: Pin 13 Cristal de 16 MHz	<i>ARDUINO MEGA 2560 REV3: Porque tiene mayor cantidad de pines y tiene conexión por jack y usb.</i>
Pines	Digitales Entrada/Salida: 14 PWM Entrada/ Salida: 6 Analógicos: 6 Corriente en los pines: 20 mA Corriente en el pin de 3.3 V: 50mA	Digitales Entrada/Salida: 54 PWM Entrada/ Salida: 15 Analógicos: 16 Corriente en los pines: 20 mA Corriente en el pin de 3.3 V: 50mA	Digitales Entrada/Salida: 22 PWM Entrada/ Salida: 6 Corriente en los pines: 40 mA	
Alimentación	Voltaje de operación: 5V Voltaje de entrada recomendado: 7-12 V Voltaje de entrada límite: 6-20 V	Voltaje de operación: 5V Voltaje de entrada recomendado: 7-12 V Voltaje de entrada límite: 6-20 V	Voltaje de operación: 5V Voltaje de entrada recomendado: 7-12 V	
Memoria	Memoria flash: 32 KB SRAM: 2KB EEPROM: 1 KB	Memoria flash: 256 KB SRAM: 8 KB EEPROM: 4 KB	Memoria flash: 32 KB SRAM: 2KB EEPROM: 1 KB	
Tamaño y peso	Longitud: 68.6 mm Ancho: 53.4 mm Peso: 25 g	Longitud: 101.52 mm Ancho: 53.3 mm Peso: 37 g	Longitud: 45 mm Ancho: 18mm Peso: 7g	

Fuente: <https://store.arduino.cc/usa/arduino-nano>, <https://store.arduino.cc/usa/arduino-uno-rev3>, y <https://store.arduino.cc/usa/arduino-mega-2560-rev3>

En la figura N° 14 se observa un Arduino Mega 2560.



Figura N° 14: Arduino Mega 2560

Fuente: <http://arduino.cl/arduino-mega-2560/>

2.3.2.9. Librería eFLL

La librería eFLL fue desarrollada por un grupo de investigación en robótica de la Universidad Estatal de Piau (UESPI-Teresina). Fue escrita en C++/C para cualquier sistema embebido. Su finalidad es poder implementar el control difuso en las placas de Arduino. Utiliza el método Mamdani, con un método de inferencia de máximo y mínimo, y una desfuzzificación de centro de área (Blog Zerokol, 2012).

2.3.3. Características de los sensores y actuadores elegidos

En esta sección se hace una breve explicación sobre el motivo de la elección de los sensores y actuadores. En la tabla N° 7 se describen los sensores y en la tabla N° 8 los actuadores.

2.3.3.1. Concepto de sensor

Un sensor es un “dispositivo que da una señal de salida transducible que es un función de la variable medida” (Pallás, 2003, p.3).

2.3.3.2. Sensor para el control de humedad relativa y temperatura

Para el siguiente proyecto se ha optado por emplear el DHT22, el cual es un sensor de bajo costo y cuyas características son suficientes para este tipo de aplicación. En la figura N° 15, se observa una tabla con una comparación general entre los sensores DHT11 y DHT22, respecto al rango de medición y la precisión. Más características se pueden ver en el anexo II.

Usa un sensor capacitivo y un termistor para medir el aire circundante, y da una señal de salida digital. No son necesarios pines analógicos. Requiere ser cuidadoso con el manejo de los datos. (adafruit® ,2018). En el ítem 1 de la figura N° 16 se observa un sensor DHT22, el cual cuenta con un tamaño de 14 x 18 x 5.5mm.

Tabla N °7: Cuadro comparativo de sensores

CUADRO COMPARATIVO DE SENSORES			
	Opción 1	Opción 2	Elección
Humedad relativa y temperatura	LM355, TMP35	DHT22	DHT22
	Salida analógica. Tienen un voltaje proporcional a la temperatura. Rango de medición: -40 °C a +125 °C Precisión: Entre ±1°C y ±2°C	Salida digital a través de un solo pin. Rango de medición: -40 a 80 °C(T), 0-100% (HR) Precisión: Entre 2 y 5% para RH. ±0.5 °C para T	<i>El sensor DHT22 presenta más precisión y cumple con los requisitos necesarios, al sensar tanto temperatura como humedad relativa.</i>
Nivel	HC-SR04	IR FC-51	HC-SR04
	Rango de medición: 2cm - 400cm Tipo de salida: Analógica	Rango de medición: 2- 30cm Tipo de salida: Digital	<i>El sensor ultrasónico tiene salida analógica.</i>
Luminosidad	LDR	BH1750	BH1750
	Genera una corriente proporcional a la cantidad de luz. Requiere circuito Temperatura de trabajo:-60 a +75 °C Rango de medición: De 10 a 1000 lux	Emplea el protocolo de comunicación i2c. Envía salida digital (cantidad de lux). No necesita componentes externos Temperatura de trabajo:-40 a +85 °C Rango de medición: 1 - 65535 lx	<i>El módulo BH1750 presenta mayor rango de medición y no requiere componentes externos</i>
Fertirriego	FC-28	DFRobot capacitivo	FC-28
	Tipo de salida: Analógica y digital. Señal de salida: 0 a 4.2 V Resistente a corrosión:No Precio: S/20	Tipo de salida: digital Señal de salida: 0 a 3 VDC Resistente a corrosión: Sí Precio: S/100	<i>A pesar de ser menos resistente, el sensor FC-28 fue más sencillo de adquirir y más económico</i>

Fuente: Desai, <http://www.micropik.com/PDF/HCSR04.pdf>, Kühnel, <http://qqtrading.com.my/ir-infrared-obstacle-detection-sensor-module-fc-5y> y <https://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf>

Tabla N °8: Cuadro comparativo de actuadores

CUADRO COMPARATIVO DE ACTUADORES			
	Opción 1	Opción 2	Elección
Humedad relativa y temperatura	Ventilador	Ventana cenital automatizada	Ventilador
	Permite bajar la temperatura y aumentar la RH Simple y rápido	Permite bajar la temperatura y aumentar la RH Difícil construcción y diseño	<i>El ventilador es más sencillo y económico</i>
	Precio: S/50	Precio: S/200	
Nivel	Electroválvula de 12 VDC	Electroválvula de baja presión	Electroválvula de 12 VDC
	Presión de operación: 0.02 MPa a 0.8 MPa Alimentación: 12 VDC, 1 A Precio: S/30	Presión de operación: 0,05-1 atm Alimentación: 12 VDC Precio: S/200	<i>La electroválvula de 12 VDC cumple con los requisitos y es más económica</i>
Luminosidad	Tira de leds	Módulos Led	Módulos Led
	Disponible en color rojo y azul	Disponible en color rojo y azul	<i>No requiere diseñar los soportes</i>
	Requiere diseñar los soportes Voltaje de alimentación: 12 VDC	Viene listo para pegar Voltaje de alimentación: 12 VDC	
Fertirriego	Bomba+Inyector Venturi	Bomba con inyector venturi	Bomba con inyector venturi
	Necesidad de diseñar el inyector Caudal: Hasta 600 lt/hr Alimentación: 12 VDC, 1.5 A Precio: S/300	Simple y rápido Caudal: 500 lt/hr Alimentación: 12 VDC, 1 A Precio: S/50	<i>Cumple con los requisitos y es más económico</i>

Fuente: <http://tienda.agrologica.es/sistemas-riego-agricola/820-electrovalvula-baja-presion.html> y <https://www.totcampingcanet.com/es/accesorios-camping/material-de-agua/bombas-de-agua-y-accesorios/bomba-sumergible-elegant.aspx>

Modelo	DHT11	DHT22
Rango de medición de humedad	20-90 % HR	0-100 % HR
Rango de medición de temperatura	0 hasta 50 °C	-40 hasta 80 °C
Precisión de temperatura	±2 °C	±0.5 °C
Precisión de humedad	±5 % HR	±2 % HR

Figura N° 15: Comparación entre sensores DHT11 y DHT22

Fuente: <http://panamahitek.com/acerca-de-panamahitek/>

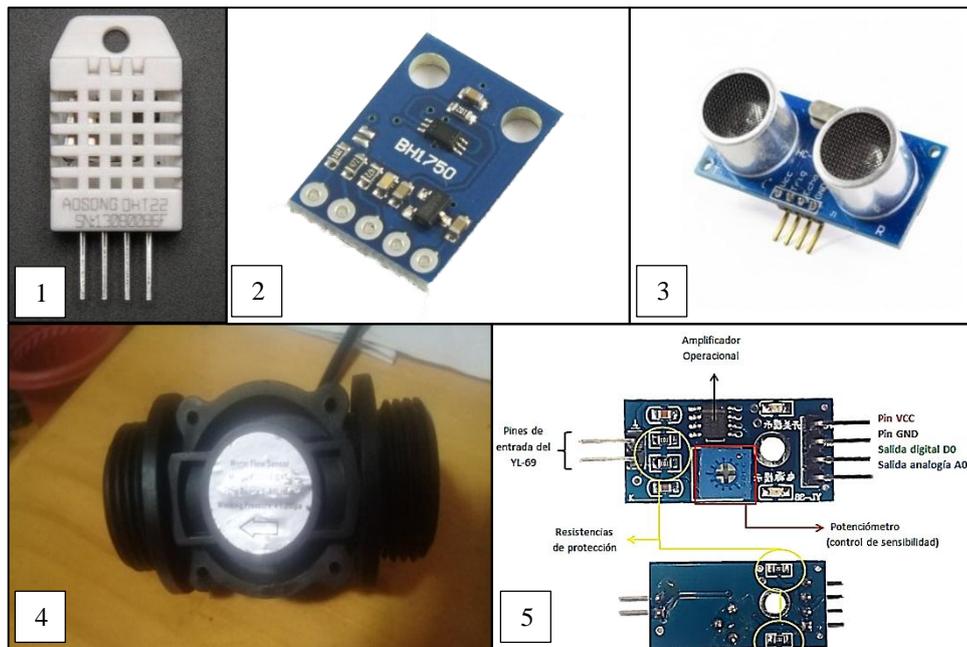


Figura N° 16: Dispositivos usados para medir las variables

Fuentes: <https://www.adafruit.com/product/385>, <https://www.mysensors.org/build/light-bh1750>, <https://naylampmechatronics.com/sensores-luz-y-sonido/76-modulo-sensor-de-luz-digital-bh1750.html>, Elaboración propia y <https://myrobotself.com/2016/07/23/sensor-de-humedad-del-suelo-modulos-yl-69-y-yl-38/#more-172>

2.3.3.3. Sensor para el control de la luminosidad

Según Naylamp Mechatronics® (2018a):

El módulo BH1750 es un sensor de iluminación digital para medición de flujo luminoso (iluminancia) de la empresa Rohm Semiconductor. Posee un conversor interno de 16-bit, por lo que entrega una salida digital en formato I2C. Su desempeño es mejor al de un Foto-Resistor (LDR), pues no es necesario realizar conversiones de voltaje para obtener datos interpretables.

En el ítem 2 de la figura N° 16 se observa un módulo BH1750, el cual se conecta a través de tres pines. En el anexo III se tienen sus características.

2.3.3.4. Sensor para el control de nivel del contenedor

Se ha empleado el sensor ultrasónico HC-SR04, el cual se puede observar en el ítem 3 de la figura N° 16. Sus características se pueden ver en el anexo IV.

ElecFreaks® (s.f.) describe:

Se usa el disparador IO para la señal de alto nivel. El Módulo envía automáticamente ocho 40kHz y detecta si hay una señal de pulso recibida. Si hay una señal recibida, se calcula la distancia a través del tiempo que duró la señal de alto nivel. (p.1)

En la ecuación N° 11 se muestra la ecuación para sacar la distancia medida con el sensor.

$$Distancia = \frac{Tiempo\ en\ nivel\ alto \times 340 \frac{m}{s}}{2} \dots\dots\dots (11)$$

Asimismo para hallar el caudal del suministro de agua se ha empleado el sensor FS400A G1, que se puede apreciar en el ítem 4 de la figura N° 16. Sus características se detallan en el anexo XX.

Según Naylamp Mechatronics ® (2018b), la salida de pulsos es una onda cuadrada cuya frecuencia es proporcional al caudal. El factor de conversión de frecuencia (Hz) a caudal (L/min) varía entre modelos y depende de la presión, densidad e incluso del mismo caudal. Para el modelo del sensor empleado el factor es de 3.5.

El fabricante da una fórmula, expresada en la ecuación N° 12.

$$f(Hz) = 3.5 \times Q \dots\dots\dots (12)$$

Donde el caudal (Q) está expresado en litros por minuto. Trabajar con estos valores sirve para aplicaciones simples.

2.3.3.5. Sensor para el fertirriego

El FC-28, en adelante también llamado sensor MH, es un sensor sencillo que mide la humedad del suelo por la variación de su conductividad. Sus características se detallan en el anexo V.

Según García González, A. (2014):

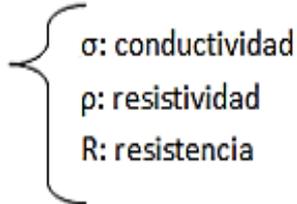
El módulo de este sensor consiste en dos placas recubiertas de un material conductor separadas entre sí por una distancia determinada. Si existe humedad en el suelo se creará un puente entre una punta y otra, lo que será detectado por un circuito de control con un amplificador

operacional que será el encargado de transformar la conductividad registrada a un valor analógico que podrá ser leído por Arduino.

En el ítem 5 de la figura N° 16 se ve el circuito de control.

El sensor MH puede dar lecturas en valor analógico o digital. Para un mejor control se ha optado por la salida analógica.

En la figura N° 17 se muestra la ecuación del módulo de sensor de humedad.

$$R \propto \rho \quad ; \quad \rho = \frac{1}{\sigma}$$


σ : conductividad
 ρ : resistividad
R: resistencia

Figura N° 17: Ecuación del módulo del sensor de humedad

Fuente: Sensor de humedad del suelo: módulos yl-69 e yl-38

2.3.3.6. Definición de actuador

Un actuador es, como afirma Vildósola (s.f):

Es un dispositivo inherentemente mecánico cuya función es proporcionar fuerza para mover o actuar otro dispositivo mecánico. La fuerza que provoca el actuador proviene de tres fuentes posibles: Presión neumática, presión hidráulica, y fuerza motriz eléctrica (motor eléctrico o solenoide). Dependiendo del origen de la fuerza el actuador se denomina “neumático”, “hidráulico” o “eléctrico”.

2.3.3.7. Actuador para el control de humedad relativa y temperatura

De acuerdo con Reyna (2015), las formas para disminuir la humedad son la ventilación, calefacción y las pantallas térmicas.

De acuerdo con Reyna (2015), las formas para aumentar la humedad son el humidificador, la pared húmeda, microaspersores, hidrofanes y ventiladores.

En el presente proyecto, para fines prácticos se reemplazan las ventanas por un ventilador accionado por el Arduino. En la parte izquierda de la figura N° 18 se muestra

el ventilador empleado para las pruebas previas, y en la derecha el ventilador usado en la implementación final.



Figura N° 18: Ventilador SP06-08 y Ventilador SYM BANG

Fuente: <http://lonen-ledlighting.com/Index.php/product/detail/token/SP06-08.html> y https://articulo.mercadolibre.com.pe/MPE-430700493-ventilador-para-gabinetes-de-12-x-12-x-38-mm-_JM#redirectedFromSimilar

Las características del ventilador SP06-08 empleado para las pruebas fueron: las mostradas en la tabla N° 9.

Tabla N° 9: Características del ventilador SP06-08

Ventilador	
Voltaje	DC 12V
RPM	3500-3800 RPM
Potencia	7W pot max.

Fuente: Datasheet del producto.

Las características de este ventilador son las mostradas en la tabla N°10.

Tabla N° 10: Características del ventilador SYM BANG

Ventilador	
Voltaje	220/240 VAC
Potencia	18/19 W

Fuente: Datasheet del producto

2.3.3.8. Actuador para el control de luminosidad

Para el presente proyecto se ha considerado una iluminación LED ya que presenta los siguientes beneficios: Fomenta la fotosíntesis y el crecimiento, produce un mayor

ahorro de energía, adelanta la producción y logra plantas más fuertes. (El Mercurio Campo, 2015). Las características de estos actuadores se resumen en el cuadro del anexo VI.

Infoagro Systems(s.f.) sostiene que:

El índice de crecimiento para la mayoría de los cultivares de rosa sigue la curva total de luz a lo largo del año. Así, en los meses de verano, cuando prevalecen elevadas intensidades luminosas y larga duración del día, la producción de flores es más alta que durante los meses de invierno. Una práctica muy utilizada en Holanda consiste en una irradiación durante 16 horas, con un nivel de iluminación de hasta 3.000 lux (lámparas de vapor de sodio), pues de este modo se mejora la producción invernal en calidad y cantidad...Es necesario el sombreo u oscurecimiento durante el verano e incluso la primavera y el otoño, dependiendo de la climatología del lugar, ya que elevadas intensidades luminosas van acompañadas de un calor intenso. La primera aplicación del oscurecimiento deberá ser ligera, de modo que el cambio de la intensidad luminosa sea progresivo. Se ha comprobado que en lugares con días nublados y nevadas durante el invierno, podría ser ventajosa la iluminación artificial de las rosas, debido a un aumento de la producción, aunque siempre hay que estudiar los aspectos económicos para determinar la rentabilidad.

Tobajas, J. (2016) sostiene que los LEDs se pueden ajustar para emitir luz en partes muy específicas del espectro. En la figura N°19 se observa un gráfico que muestran los nutrientes absorbidos según el tipo de luz.

Asimismo, Tobajas, 2016 también comenta:

Las clorofilas, biomoléculas esenciales en el proceso de fotosíntesis, tienen típicamente dos tipos de absorción en el espectro visible, uno en el entorno de la luz azul (400-500 nm de longitud de onda), y otro en la zona roja del espectro (600-700 nm); sin embargo reflejan la parte media del espectro, la más nutrida y correspondiente al color verde (500-600 nm).

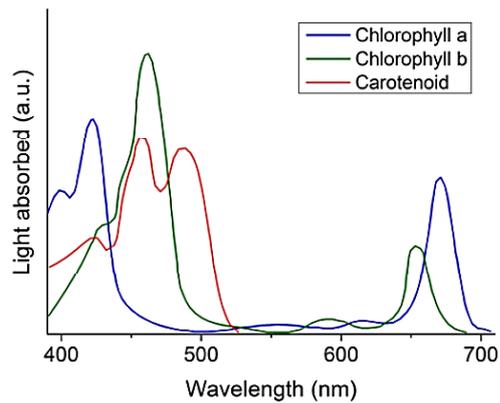


Figura N° 19: Absorción de nutrientes por parte de las plantas

Fuente: <https://www.linkedin.com/pulse/iluminaci%C3%B3n-led-para-invernaderos-eficientes-jos%C3%A9-tobajas-povar/>

2.3.3.9. Actuadores para el control de nivel

Los actuadores utilizados fueron electroválvulas de 12V, que fueron accionadas por Arduino mediante el módulo de relés. Para el llenado del recipiente fue usada una electroválvula tal como se observa en la figura N° 20 (izquierda). Permiten o no el paso del flujo de agua. Dichas electroválvulas son ideales para este tipo de aplicaciones. En el anexo VII se detallan las características de la electroválvula 1.

Para el vaciado del recipiente fue empleada una electroválvula normalmente cerrada, que depende de una mínima presión para su funcionamiento, como la mostrada en la figura N° 20 (derecha).



Figura N° 20: Válvulas solenoides

Fuente: https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/ROB-10456_Web.pdf y https://guide.alibaba.com/shop/super-bargain-new-model-new-1-4-air-train-horn-electric-solenoid-valve-heavy-duty-air-valve-12v-24v_10010061.html

2.3.3.10. Actuador para el control del fertirriego

El actuador para este control es una bomba de fertirrigación con inyector venturi. Después de que el agua pasa por el filtro, es extraída por la bomba del mismo recipiente y redirigida hacia la boquilla con un caudal de 500 lt/hr.

En la figura N° 21 (izquierda) se muestra la ficha de la bomba empleada. En la tabla N°11 se muestran las características principales de la bomba.

2.3.3.11. Goteros

Los goteros seleccionados fueron regulables de la marca PLASTICA ALFA para que no dependan de una presión mínima de riego. Se pueden observar en la figura N°21 (derecha).

“Es un gotero ajustable de alta calidad con un flujo de hasta 70 LPH; con el cual se puede regular según la cantidad de clicks la cantidad de riego deseado y así aumentar o disminuir la cantidad de riego según el ciclo de vida o el tamaño de la planta.”(Insumos de Riego ®,2018)



Figura N° 21: Ficha de la bomba y goteros regulables

Fuente: <http://www.hasakvaryum.com/detay.asp?id=6580> y

http://www.insumosderiego.com.ar/riego_por_goteo_goteros_insertables_regulables.php?titulo=gotero-regulable-para-riego-0-70-lh-gotero-regulable-para-riego-70&ELID=1583

Las ecuación N° 13 muestra el cálculo del caudal máximo de riego de los goteros empleados. La ecuación N° 14 el valor calculado final.

$$\text{Caudal máximo de riego} = 70 \frac{\text{lt}}{\text{hr}} \times 6 \dots\dots\dots(13)$$

$$\text{Caudal máximo de riego} = 420 \text{ lt/hr} \dots\dots\dots(14)$$

Tabla N° 11: Características de la bomba con filtro VENUSAQUA

Bomba	
Voltaje de alimentación	220 V, 50Hz
Potencia	6 W
Material del motor	Bronce
Caudal máxima	500 L/H

Fuente: Datasheet del producto

2.3.3.12.Malla Raschel

La malla Raschel es una red tejida de polietileno de Alta Densidad. Produce la sombra necesaria para un adecuado desarrollo del cultivo, además de protegerlo del viento (Polytex.cl®, s.f.).

La empleada es una malla Raschel que contiene “aditivos que le brindan resistencia a la radiación solar” (Litec®, 2012). En la figura N° 22 se observa la malla, y en el anexo VIII se muestran sus especificaciones técnicas.



Figura N° 22: Malla Raschel de 80% de sombra

Fuente: <http://www.litecperu.com/productos/malla-raschel/>

Con un 80% de sombra, al interior quedan aproximadamente 2150 lux, como muestra la ecuación N° 15.

$$20\% \text{ de } 10,752.7 = 2150.54 \text{ lux} \dots \dots \dots (15)$$

2.3.3.13.Sistema Venturi

García et al (2016) mencionan que Mataix (2005) definió al inyector tipo Venturi como: “un dispositivo con una sección convergente, seguida de una garganta y de una sección divergente, que gradualmente vuelve al diámetro inicial”. (p.29). En la figura N° 23 se aprecia el diagrama de un inyector Venturi.

García et al (2016) también expresa:

Un inyector del tipo Venturi genera succión debido al diferencial de presión negativa que se produce entre su entrada y su salida al reducir su diámetro de garganta; la velocidad del fluido se acelera debido a este decremento en los diámetros. La presión de entrada de energía se traduce en energía cinética cuando el agua fluye a través de la garganta del inyector. A medida que aumenta la presión de entrada y la velocidad de descarga, aumenta la energía cinética en la garganta. Cuando la energía cinética en la garganta alcanza un cierto valor, la energía (presión) en la garganta desaparece por completo y una presión negativa se produce. (p.30).

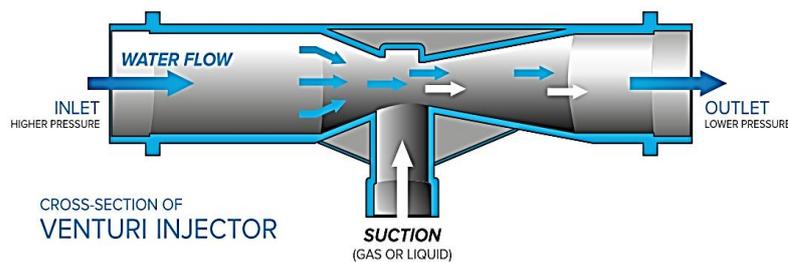


Figura N° 23: Análisis de un inyector venturi
Fuente: <https://www.wassertec.co.za/venturi-aeration/>

2.3.3.14.Fertilizante

El fertilizante empleado fue un líquido creado especialmente para los cultivos de rosas, el cual contiene Nitrógeno 6%, Fósforo 7% y Potasio 6%, principalmente. En la figura N° 24 se observa el fertilizante líquido empleado.



Figura N° 24: Fertilizante para rosas

Fuente: <http://www.sodimac.com.pe/sodimac-pe/product/2495694/> Fertilizante-liquido-para-rosas-y-flores-450ml

CAPÍTULO III: DISEÑO Y SIMULACIÓN

3.1.Requerimientos generales

Tanto los sensores, actuadores y los materiales a emplear en la estructura deben ajustarse a los siguientes requerimientos:

1. Ser de bajo costo
2. Ser de tamaño compacto

3.1.1.Requerimientos eléctricos y electrónicos de los sensores

1. Ser de implementación rápida con la plataforma Arduino
2. Voltaje de alimentación: 5 V, 500 mA
3. Precisión: Hasta 10%
4. Temperatura de operación: Hasta 40 °C
5. Humedad de operación: Hasta 100% RH
6. Envío de datos: A través del puerto serial

3.1.2.Requerimientos eléctricos y electrónicos de los actuadores

1. Voltaje de alimentación: 12 V, hasta 1 A o 220 VAC.
2. Temperatura de operación: Hasta 40 °C
3. Humedad de operación: Hasta 100% RH
4. Manejo de datos: Pin del controlador

3.1.3. Requerimientos de la estructura

La estructura debió ser de dimensiones reducidas debido a las limitaciones descritas en la sección 1.4 del Capítulo 1. Asimismo la ubicación corresponde al domicilio de la tesista.

1. Dimensiones: Base: 1 metro de largo x ½ metro de ancho
Altura: 1metro
2. Ubicación: Santiago de Surco, Lima, Perú

3.2.Respuesta del requerimiento

El proceso de diseño se encuentra explicado en la figura N° 25. El primer paso es la definición del problema, el cual está detallado en el apartado 1.2.1 de la presente tesis. Luego, a partir de la recopilación de la información encontrada en la literatura se definieron los requerimientos para cada lazo de control. Se planteó una solución y se realizaron pruebas. Finalmente se hizo la prueba final, lista a ser implementada en el mini invernadero.

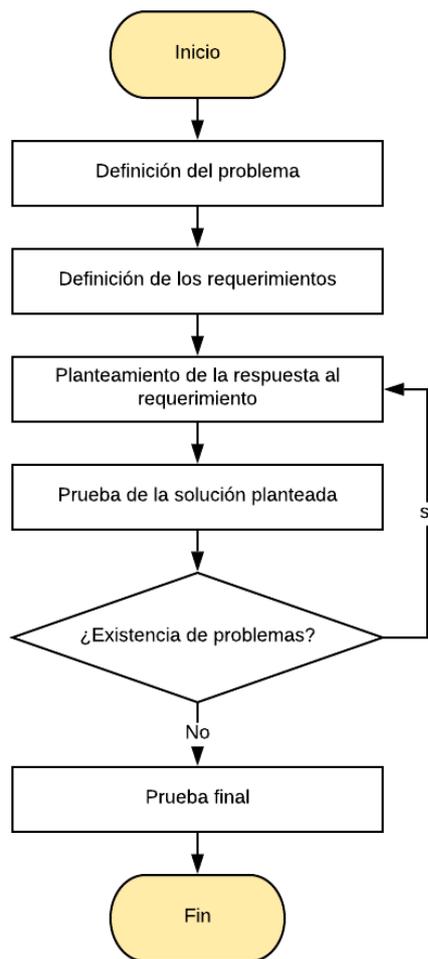


Figura N° 25: Esquema del proceso de diseño

Fuente: Elaboración propia

Teniendo en cuenta las variables que afectan al cultivo, en general el esquema del controlador difuso sería como el ilustrado en la figura N° 26.

El controlador difuso dispone de un fuzzificador, el cual mediante el mecanismo de inferencia y la base de conocimiento (reglas), obtiene las salidas y las desfuzzifica con tal de que el programa produzca las salidas de voltaje necesarias. De esa manera acciona los actuadores de la planta (invernadero).

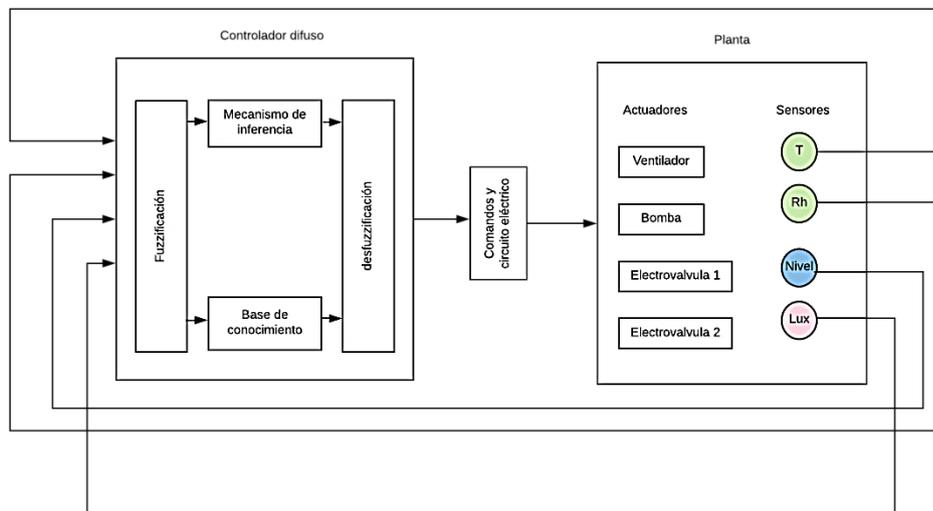


Figura N° 26: Esquema del control del invernadero

Fuente: Elaboración propia

3.3.Elementos a emplear

En el presente proyecto se elabora un invernadero experimental. Los componentes a emplear son los siguientes, que pueden visualizarse en la figura N° 27 y N°28.

a) Sensores y actuadores

Sensores: Sensor BH1750, para la luminosidad. Sensor DHT22, para la humedad relativa y temperatura del aire. Sensor HC-SR04, para el nivel (distancia)

Actuadores: Ventilador, Electroválvula 1, Electroválvula 2.

b) Controlador: Arduino Mega 2560

c) Elementos de visualización: Pantalla LCD 16x4

d) Estructura del mini invernadero: La construcción del invernadero experimental se basó en tuberías y uniones PVC de ½” formando un invernadero a dos aguas.

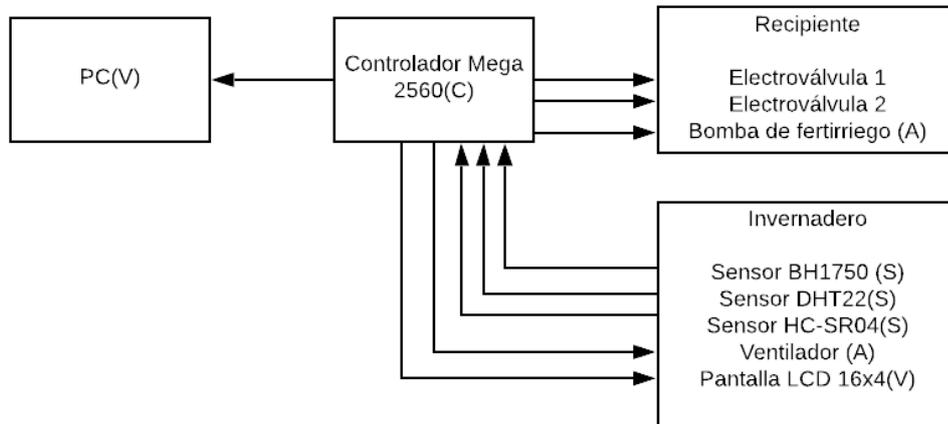


Figura N° 27: Bosquejo del invernadero experimental

Fuente: Elaboración propia

e) Cultivo

Como muestra del cultivo se han considerado seis (06) plantas de rosa miniatura.

f) Software LabView

El software LabView permitió simular los lazos de control difuso para el mini invernadero.

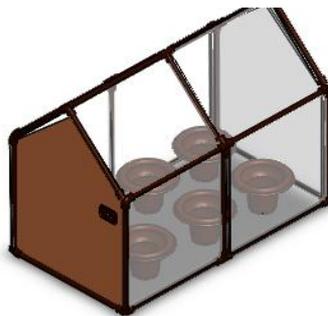


Figura N° 28: Vista isométrica del invernadero experimental

Fuente: Elaboración propia

g) Software Proteus Professional

El software Proteus Professional permitió diagramar las conexiones eléctricas y electrónicas de los componentes.

h) Software Arduino

El software Arduino permitió programar el controlador difuso.

i) Software Solidworks

El software Solidworks permitió simular la estructura mecánica del mini invernadero del invernadero, según véase en la figura N° 28, así como analizar el inyector Venturi que vino con la bomba.

3.4. Software LabView

A continuación se detallarán algunos conceptos sobre el software LabView (Corcuera, s.f).

- VI(Virtual Instrument) : Programa en LabView
- Fuzzy System Designer: Herramienta del software que permite crear sistemas difusos.
- Control: Entradas de un VI
- Indicador: Salidas de un VI
- Panel frontal: Interfaz entre el programa y el usuario.
- Diagrama de bloques: Código del programa.

3.5. Control difuso de la temperatura y humedad relativa del aire

La temperatura y la humedad pueden interactuar en un solo control, ya que una está en función de otra. En el proyecto se empleó un sensor de humedad y temperatura del aire del invernadero, el DHT22.

Se realizó un esquema del sistema de control del invernadero, el cual se visualiza en la figura N° 29. Inicia con la medición de las variables deseadas, y se quedará sólo midiéndolas siempre que estén en los valores adecuados. Si dejan de estar en los valores adecuados activan el actuador. Luego de unos minutos el ciclo vuelve a repetirse.

El sensor medirá la señal a controlar, la cual pasa a ser digitalizada por el conversor A/D, para ser transmitida al controlador. Véase figura N° 30.

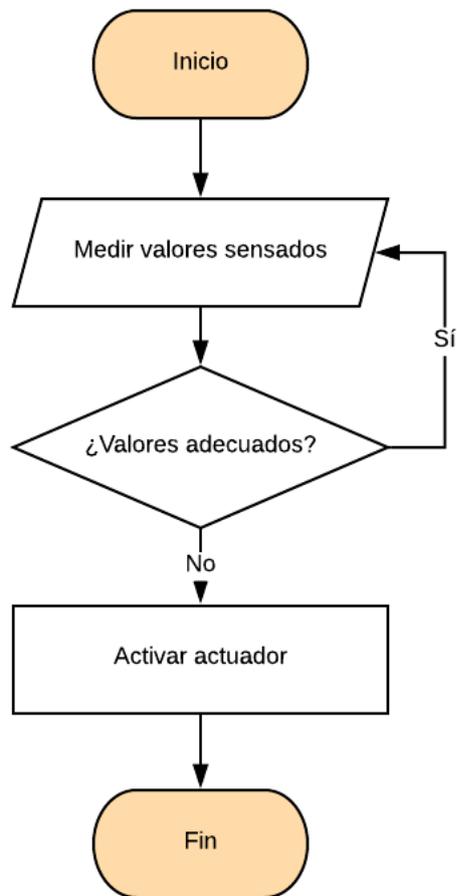


Figura N° 29: Diagrama de control de la humedad relativa y temperatura

Fuente: Elaboración propia

El actuador a emplear será un ventilador, el cual bajará la temperatura y aumentará la humedad relativa del invernadero. Primero se utilizará un ventilador de prueba y luego un ventilador de más potencia.

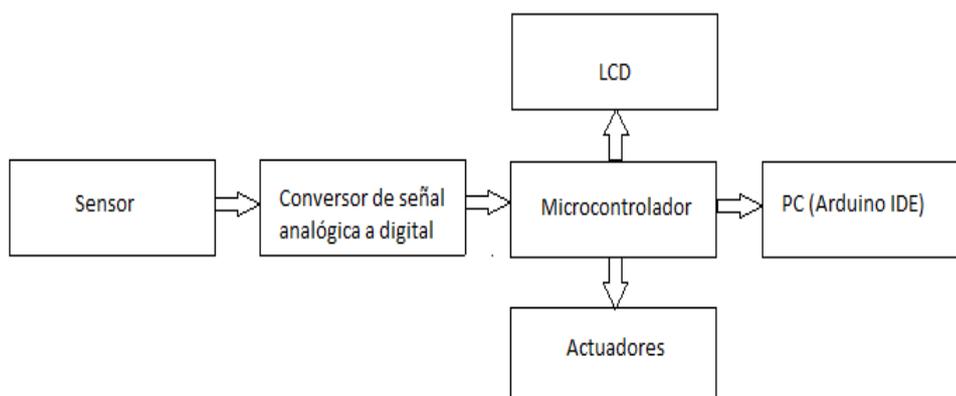


Figura N° 30: Esquema de control y visualización de la humedad relativa y temperatura

Fuente: Elaboración propia

3.5.1. Ubicación de los componentes

El sensor DHT22 se ubicó en el centro del invernadero experimental, conectado mediante cables estañados al controlador Arduino.

El actuador se ubicó en la parte trasera del invernadero, el cual también estuvo conectado mediante cables estañados al controlador Arduino.

En la figura N° 31 se tiene una vista de los componentes a emplear en este lazo de control, desarrollado en el software Solidworks.



Figura N° 31: Componentes del control de humedad relativa y temperatura

Fuente: Elaboración propia

3.5.2. Entradas y salidas del controlador difuso

Para realizar el control de la humedad y temperatura se ha colocado las entradas y salidas según muestra la tabla N° 12.

Tabla N° 12: Entradas y salidas del control de las variables temperatura y humedad relativa

Variable	Temperatura y humedad relativa del aire dentro del invernadero
Entrada(s)	Cantidad de T y RH
Salida(s)	Voltaje del ventilador

Fuente: Elaboración propia

a) Cantidad de T y RH

Los rangos se elaboraron de acuerdo a los valores presentados en la figura N° 7.

Estos son:

Humedad relativa: 35 a 40%, 41 a 55%, 56% a 80%, 81% a 100%.

Temperatura: 15 a 19 °C, 20 a 24°C, 25 a 31 °C, 32 a 35°.

b) Voltaje del ventilador

Para sacar el voltaje que se mandó al ventilador se hicieron pruebas sobre las variaciones que éste producía en humedad y temperatura, según puede observarse en la tabla N° 9. Para realizar las siguientes pruebas el código empleado fue el ubicado en el anexo IX, en el que se observa en el puerto serial del Arduino IDE la temperatura y humedad relativas. Se colocó el valor del voltaje del ventilador deseado. El resultado obtenido se expresa mediante la tabla N° 13. En la figura N° 86 se muestra el cableado que se realizó.

Tabla N° 13: Variación de la temperatura y humedad relativa con diferentes voltajes

Voltaje	Temperatura inicial	Humedad Relativa inicial	Δ Temperatura (-)	Δ HR (+)
255 (Prueba 1)	24	79.9	0.3	1.6
255 (Prueba 2)	29.1	83.8	4.1	6.2
200	24.1	79.5	0.4	1.6
150	24	77	0.3	1.6
100	24.2	78.2	0.1	0.8

Fuente: Elaboración propia

No existe mucha variación de humedad relativa y temperatura con 200, 150 y 100. Por ello sólo se han tomado en cuenta los valores de 255. Asimismo se han empleado estos valores, expresados en la tabla N° 14, para realizar la función de simulación del efecto del ventilador.

Tabla N° 14: Variación de la temperatura y humedad relativa con el voltaje máximo

Voltaje	Temperatura inicial	Humedad Relativa inicial	Δ Temperatura	Δ HR (+)
255 (1)	24	79.9	0.3	1.6
255 (2)	29.1	83.8	4.1	6.2

Fuente: Elaboración propia

3.5.3. Objetivos

1. Lograr que tanto la temperatura como la humedad relativa estén dentro de un margen adecuado para las plantas, es decir, que tengan un DPV adecuado, según la tabla mostrada en la figura N°7.

2. Configurar las entradas y salidas.
3. Armar los conjuntos difusos para la humedad relativa.
4. Armar los conjuntos difusos para la temperatura.
5. Definir las reglas.

3.5.4. Desarrollo

1. El control fue establecido de acuerdo a la figura N°7, considerando una RH que va desde 35 a 100% y una temperatura que va desde 15 hasta 35 °C.
2. Se definieron los nombres de las entradas y salidas.
3. Se armaron los conjuntos difusos para la humedad relativa de acuerdo al cuadro considerando los rangos [35-40], [40-55], [55-80] y [80-100].
4. Se armaron los conjuntos difusos para la temperatura de acuerdo al cuadro considerando los rangos [15-19], [20-24], [25-31] y [32-35].
5. Se definieron las reglas.

3.5.5. Resultados

En la figura N° 32 se tiene una vista de la ventana del Fuzzy System Designer de LabView, el cual muestra los conjuntos difusos de la variable humedad relativa (arriba) y temperatura(abajo) y la variable voltaje.

El Fuzzy System Designer también tiene una ventana para visualizar la salida a partir de una entrada determinada. Asimismo se observan las reglas que se cumplen para dicha entrada ingresada. En la figura N° 33, se ve la ventana cuando la humedad relativa es 100 y la temperatura 15(arriba) y cuando la humedad relativa es 40 y la temperatura 15(abajo). Vemos que se cumple según la tabla. En la figura N° 34, se aprecian las reglas creadas para este control difuso.

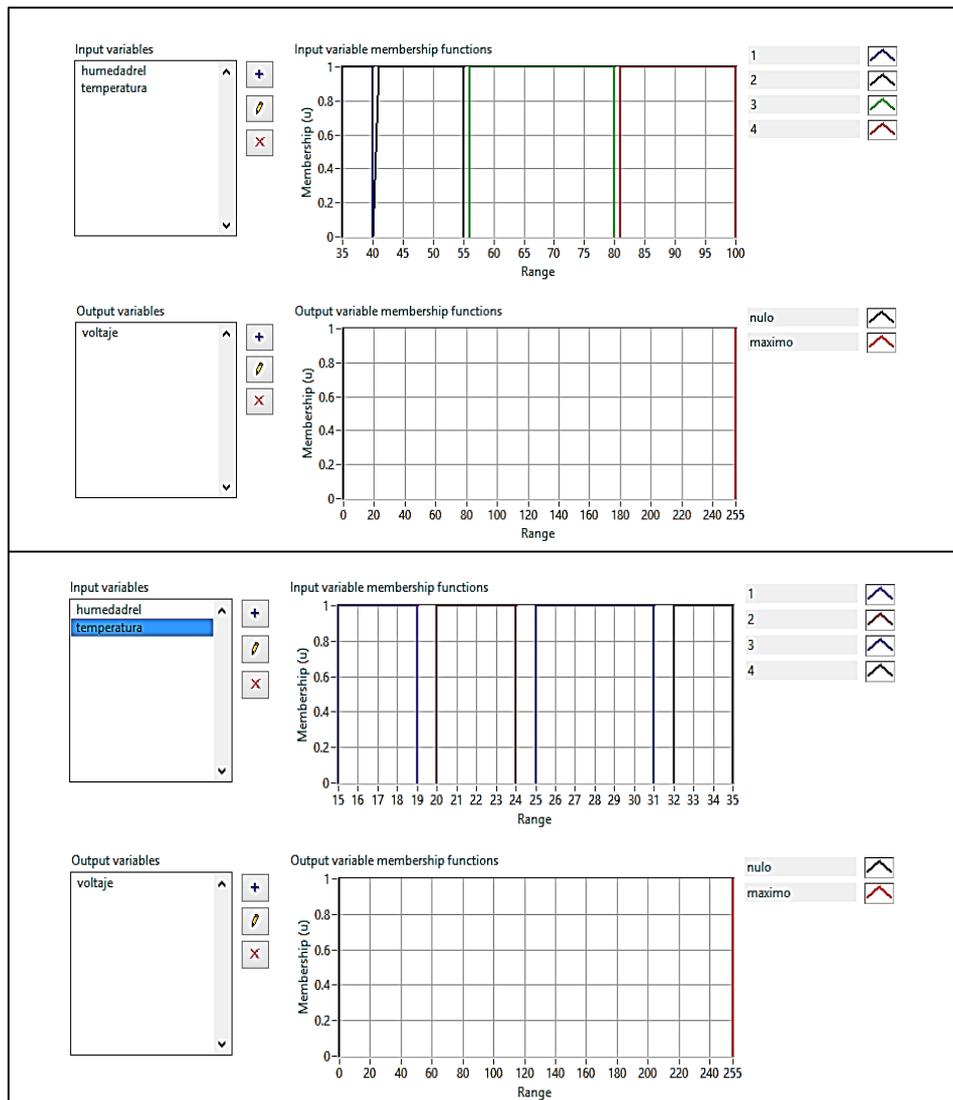


Figura N° 32: Conjuntos difusos del control de la RH y T.

Fuente: Elaboración propia

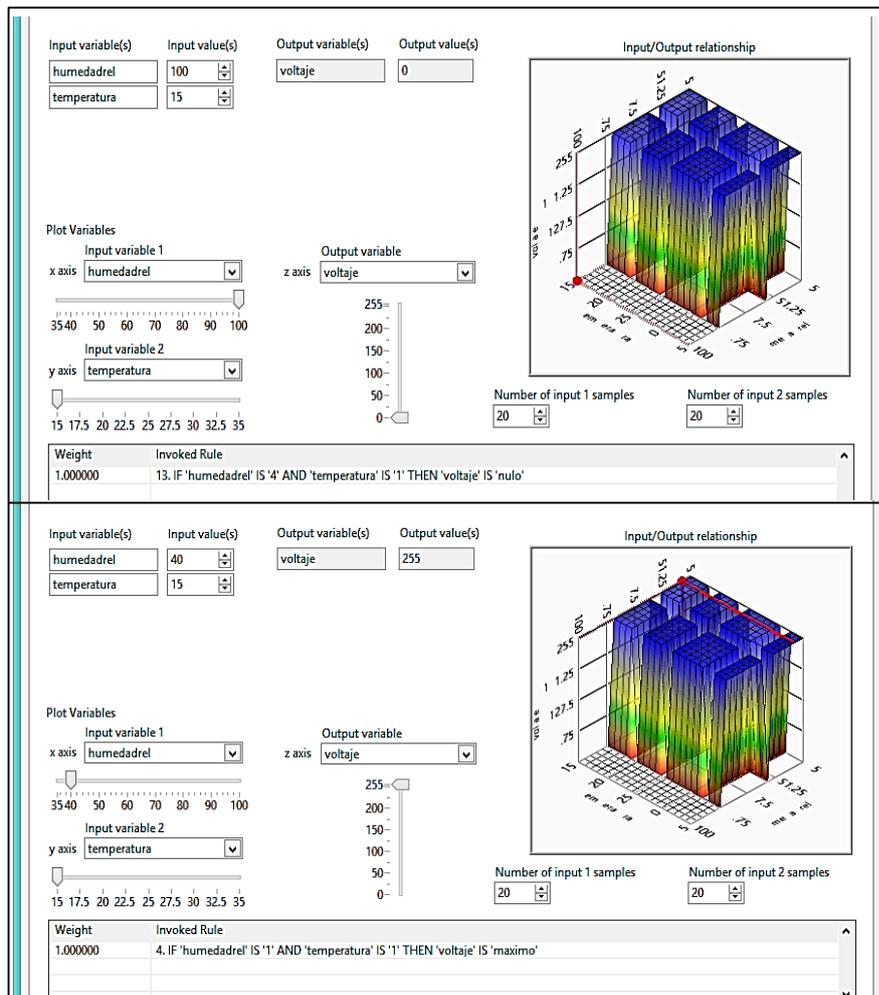


Figura N° 33: Testeo del controlador difuso de la RH y T

Fuente: Elaboración propia

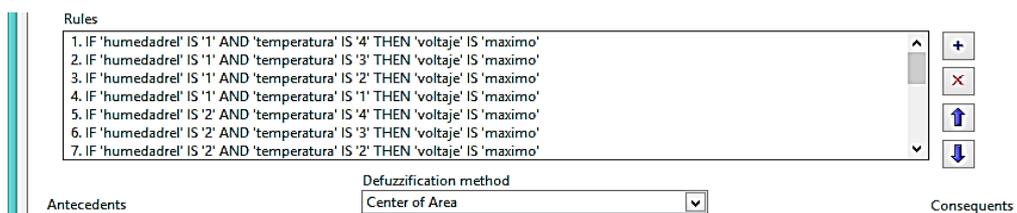


Figura N° 34: Reglas del controlador difuso del control de la RH y T

Fuente: Elaboración propia

3.5.6. Simulación final del controlador difuso en LabView

a) Desarrollo

Se crearon ecuaciones empleando los datos de la tabla N° 14. La ecuación para humedad relativa se observa en la ecuación N° 16, donde X es la humedad relativa anterior. La

ecuación para la temperatura se observa en la ecuación N° 17, donde X es la temperatura anterior.

$$y = 0.902 * X - 20.047 \dots\dots\dots (16)$$

$$y = 0.7451 * X - 17.582 \dots\dots\dots (17)$$

Con dichas ecuaciones se creó un subVI en Labview, las cuales pueden observarse en la figura N° 35. Se han creado restricciones dentro de las ecuaciones para cuando el valor de la humedad relativa sea mayor a 100.

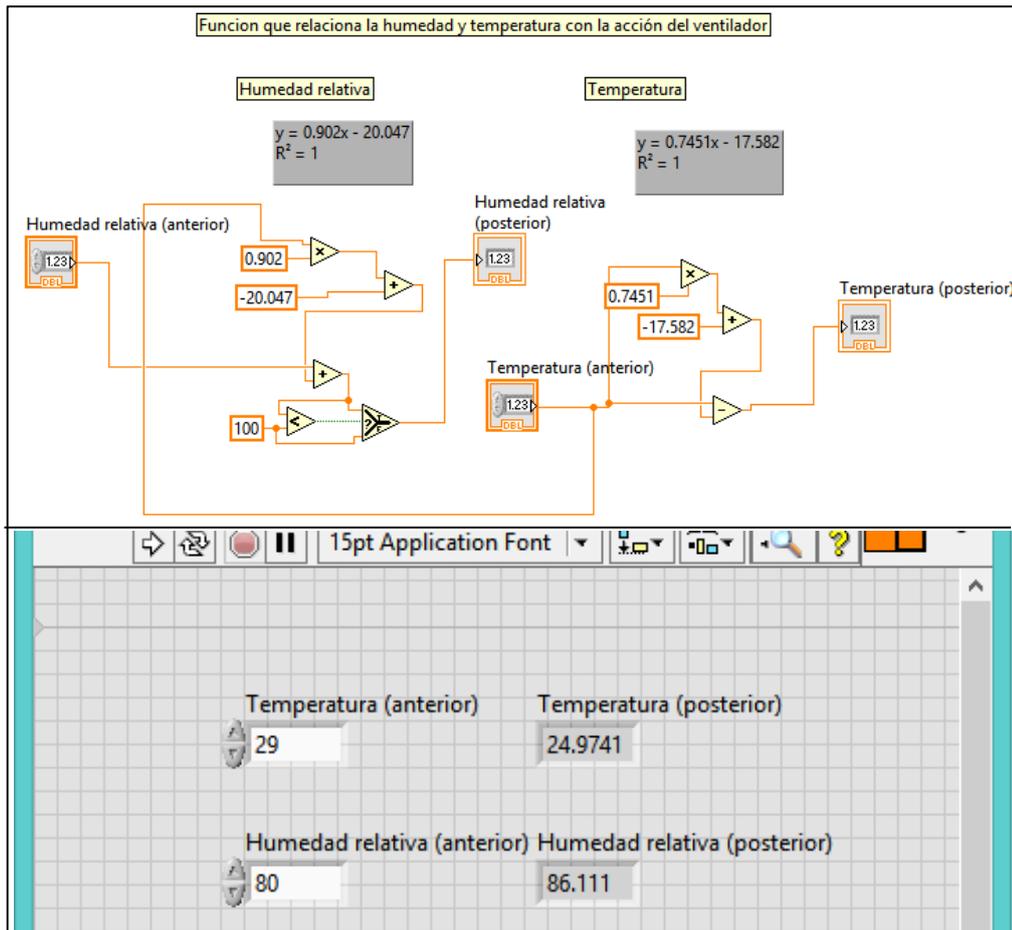


Figura N° 35: Diagrama de bloques y panel frontal del subVI Ventilador.

Fuente: Elaboración propia

Empleando el subVI del ventilador se ha creado el VI correspondiente al control de la temperatura y humedad relativa, cuyo diagrama de bloques se ve en la figura N° 36 y panel frontal en la figura N° 37.

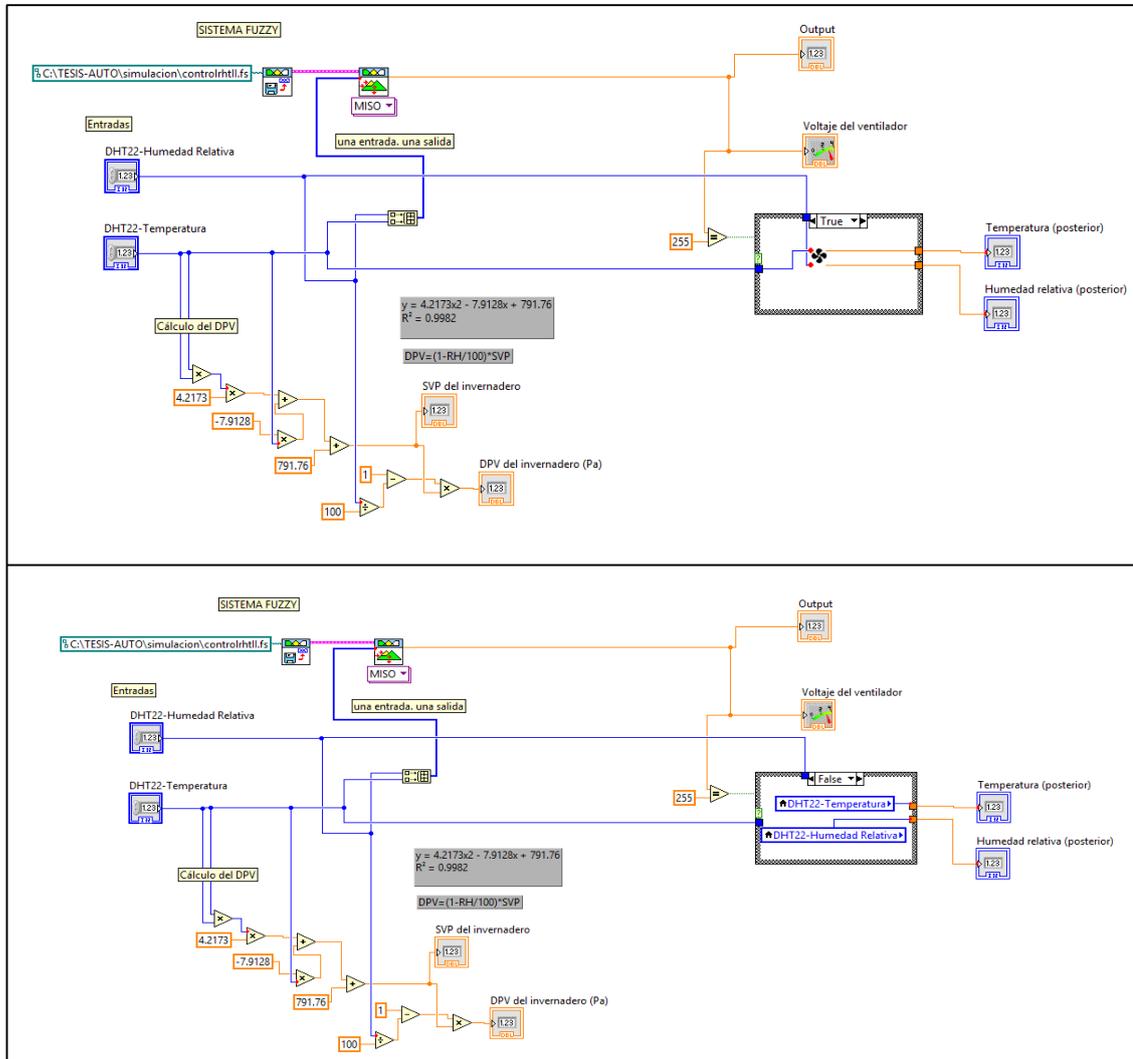


Figura N° 36: Diagrama de bloques del control difuso de la RH y T

Fuente: Elaboración propia

En la simulación se consideró el cálculo del DPV del invernadero, según se mostró en la ecuación N° 1. El SVP fue calculado a partir de una ecuación realizada con Microsoft Excel a partir de los valores de SVP dados, la cual se expresa en la ecuación N° 18. Se usó un condicional para cuando se activa o no el subVI ventilador.

$$SVP = 4.2173 * X^2 - 7.9128 * X + 791.76 \dots \dots \dots (18)$$

En la figura N° 37 se observa el panel frontal del control difuso de la humedad relativa y temperatura.

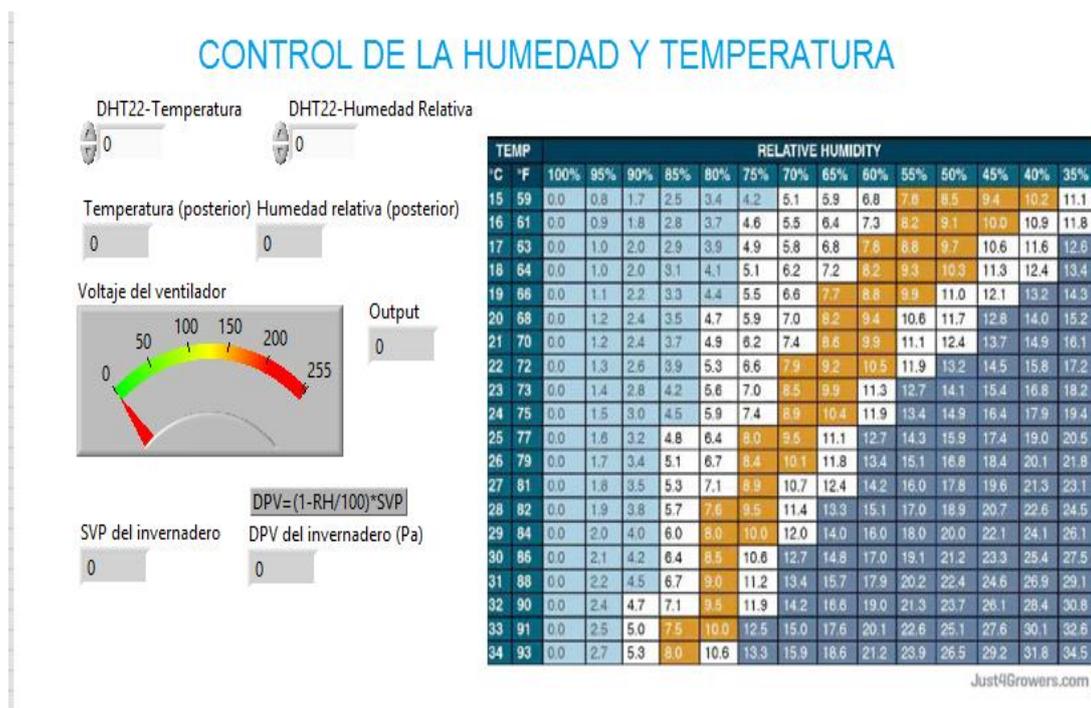


Figura N° 37: Panel frontal del control difuso de la RH y T

Fuente: Elaboración propia

El panel frontal muestra los siguientes controles e indicadores. Entre paréntesis los tipos de datos.

Controles: DHT22-Temperatura (numérico), DHT22 humedad relativa numérico).

Indicadores: Temperatura (posterior) (numérico), humedad relativa (posterior) (numérico), output (numérico), SVP del invernadero (numérico), DVP del invernadero (Pa) (numérico), voltaje del ventilador (numérico)

Imágenes: Gráfico de la tabla

En la parte superior izquierda se muestran los valores capturados por el sensor DHT22. Debajo de ellos se muestran los valores posteriores a la activación o no activación del ventilador. En la parte derecha se muestra la tabla de la figura N° 7.

b) Características del controlador difuso

Se trata de un sistema MISO (Multiple Input, Single Output), con dos entradas (Temperatura y Humedad relativa) y una salida (Voltaje del ventilador).

Colocó la salida al máximo (255) cuando las condiciones no sean las óptimas, debido a la limitación del ventilador.

c) Resultados de la simulación

En la figura N° 38 se observan dos pruebas de la simulación. En la primera imagen, para cuando la humedad relativa es 100 y la temperatura 15, el ventilador se mantiene desactivado. En la segunda imagen, para cuando la humedad relativa es 70 y la temperatura 26, el ventilador es activado. También se mostró el cálculo del DPV en Pascales para cada caso.

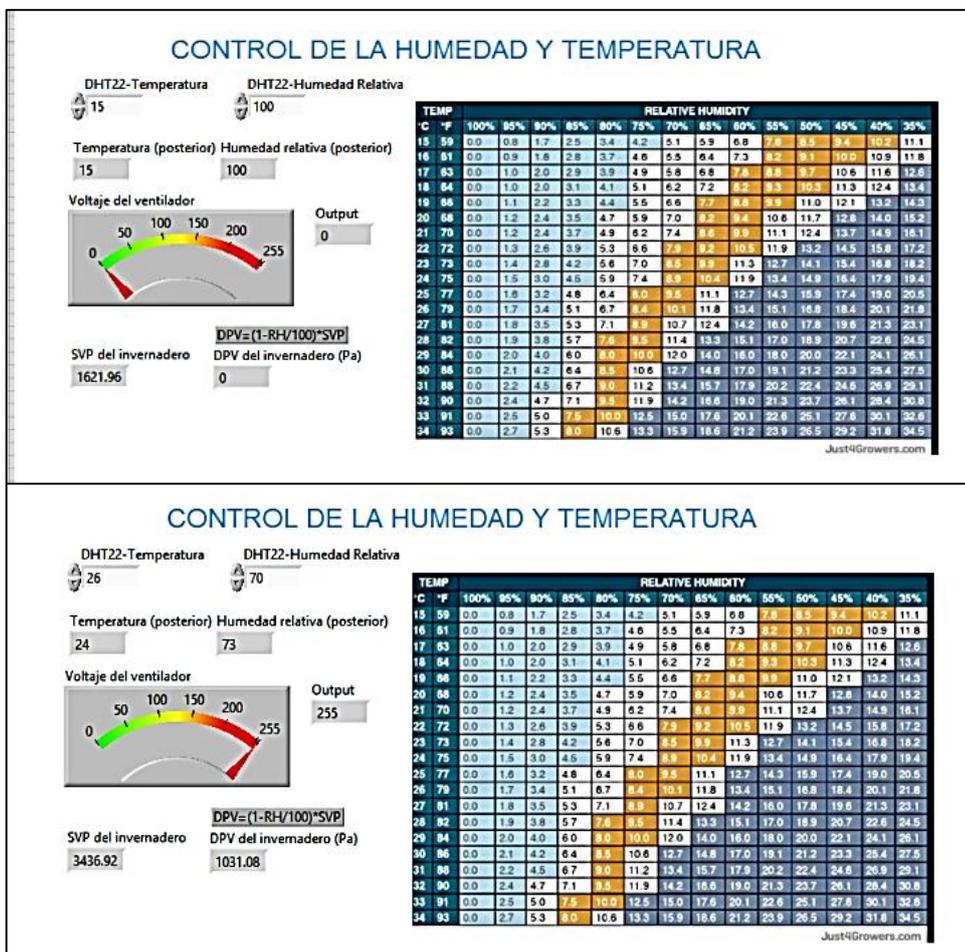


Figura N° 38: Pruebas de la simulación del control de la RH y T

Fuente: Elaboración propia

3.5.7. Conexiones electrónicas

En la figura N° 39 se observan las conexiones realizadas para el sensor y el actuador.

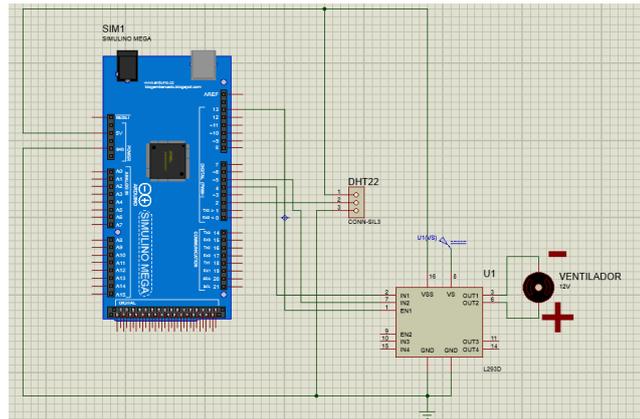


Figura N° 39: Diagrama de las conexiones del control de la humedad relativa y temperatura

Fuente: Elaboración propia

3.6. Control de nivel del recipiente de irrigación

3.6.1. Ubicación de los componentes

En la figura N° 40 se puede apreciar la simulación de los componentes empleados para este lazo de control, realizada en el software Solidworks. En el ítem 3 de la figura N° 41 se ve la simulación del recipiente.



Figura N° 40: Componentes del control del nivel del recipiente

Fuente: Elaboración propia

- La electroválvula 2, cuya imagen se puede ver en el ítem 1 de la figura N° 41, se colocó en la parte inferior del recipiente, a unos 10 cm de la base.
- La electroválvula 1 se ubicó sobre un soporte. Se puede ver en el ítem 2 de la figura N° 41.
- El sensor ultrasónico HC-SR04 se ubicó en la parte superior del recipiente.

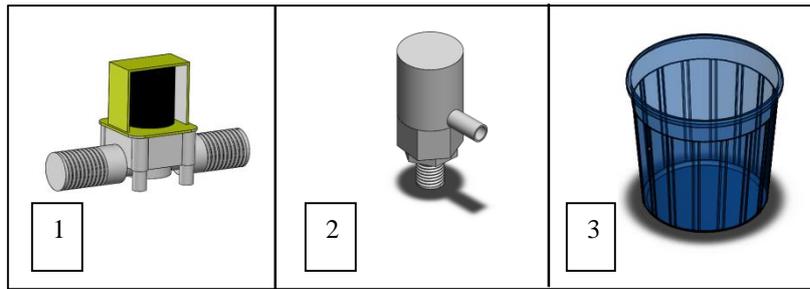


Figura N° 41: Simulación de las piezas en el programa Solidworks

Fuente: Elaboración propia

3.6.2. Entradas y salidas del controlador difuso

El controlador difuso actuará en función a la distancia medida por el sensor ultrasónico, la cual está expresada por la diferencia entre la ubicación del sensor y la base del recipiente. Las salidas serán los tiempos en segundos que estarán abiertas las electroválvulas, ya sea para el llenado o para el vaciado del recipiente. En la tabla N° 15 se observan las entradas y salidas de este control.

Tabla N° 15: Entradas y salidas del control del nivel del recipiente

Variable	Nivel del agua del recipiente
Entrada(s)	Distancia entre el sensor y el nivel del agua
Salida(s)	Tiempo de activación de la electroválvula 1
	Tiempo de activación de la electroválvula 2

Fuente: Elaboración propia

3.6.3. Objetivos

1. Mantener el nivel del recipiente adecuado.
2. Configurar las entradas y salidas.
3. Armar los conjuntos difusos.
4. Definir las reglas.

3.6.4. Desarrollo

1. Se diseñó el control de nivel del recipiente mediante la distancia medida con el sensor ultrasónico y la activación de las electroválvulas. Para realizar este control se

realizaron los procesos ilustrados en el diagrama de flujo ilustrado en la figura N° 42. Primero se realiza la medición con el sensor ultrasónico, después se examina si el valor obtenido es mayor, menor o igual a 6. Si la distancia obtenida es mayor a 6, se apertura la electroválvula 1 durante el tiempo T1, Si es menor se apertura la electroválvula 2 durante el tiempo T2. Se ha optado por ubicar el SET POINT a una distancia de 6 cm del sensor ultrasónico debido a que dicho nivel la bomba puede extraer el agua sin ningún inconveniente.

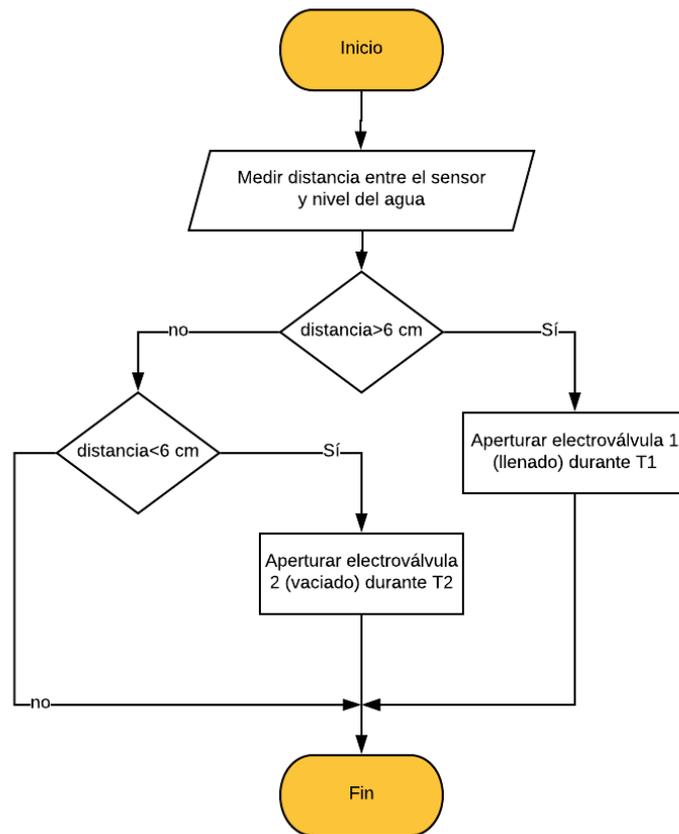


Figura N° 42: Diagrama de flujo del control de nivel del recipiente

Fuente: Elaboración propia

A partir de la distancia medida por el sensor ultrasónico, se realizaron cálculos para relacionarlo con la cantidad de volumen de agua contenido en el recipiente. Posteriormente se calculó el tiempo en llegar a dicha cantidad de volumen de agua.

a) Cálculo del caudal del suministro

Se empleó el sensor FS400A G1, cuyas características se resumen en la tabla del anexo X. Se creó el código del anexo XI y con las conexiones visualizadas en la figura N° 66.

b) Cálculo del volumen de agua en el recipiente

La cantidad de volumen de agua se calculó mediante el software Solidworks, realizando la simulación con alturas de 10, 20 y 30 centímetros. Se ha empleado el comando Combinar del programa Solidworks, según se aprecia en la figura N° 43.

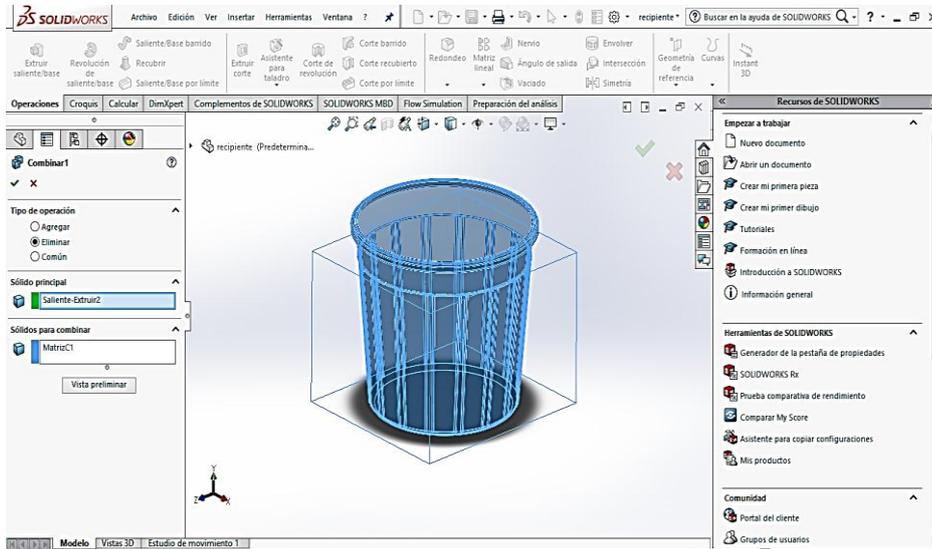


Figura N° 43: Vista de la ventana del comando Combinar

Fuente: Elaboración propia

De esta manera se ha podido simular el volumen de agua a diferentes alturas (Véase figura N° 44) y posteriormente hallado el volumen interior correspondiente al agua para diferentes alturas en centímetros. Según se muestra en la figura N° 45, de acuerdo a lo calculado por el programa bajo la opción de “Propiedades Físicas”, el ítem 1 corresponde a la simulación para 10 centímetros, el ítem 2 para 20 centímetros y el ítem 3 para 30 centímetros.

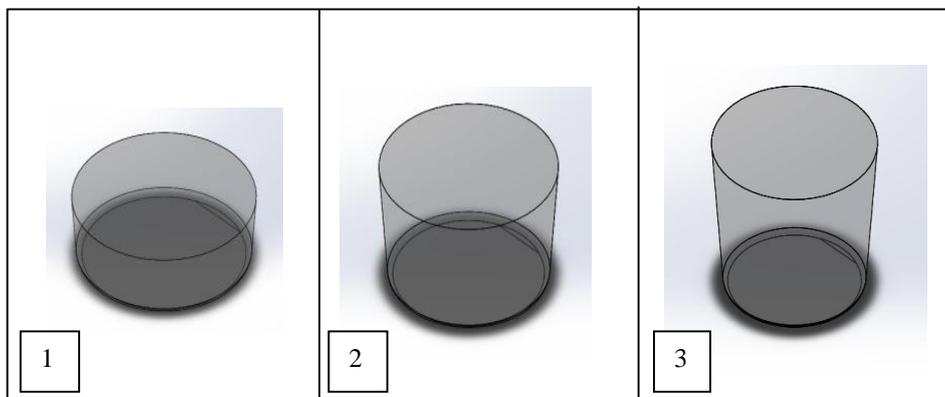


Figura N° 44: Simulación del volumen de agua

Fuente: Elaboración propia

El programa ha calculado el volumen de agua para cada altura en milímetros cúbicos. Posteriormente se ha convertido dicho volumen a metros cúbicos.

<p>Propiedades de masa de volumen Configuración: Predeterminado Sistema de coordenadas: -- predeterminado --</p> <p>Densidad = 0.00 gramos por milímetro cúbico</p> <p>Masa = 7837.24 gramos</p> <p>Volumen = 7324527.61 milímetros cúbicos</p> <p>Área de superficie = 240406.58 milímetros cuadrados</p> <p>Centro de masa: (milímetros)</p>
<p>Propiedades de masa de volumen Configuración: Predeterminado Sistema de coordenadas: -- predeterminado --</p> <p>Densidad = 0.00 gramos por milímetro cúbico</p> <p>Masa = 16610.83 gramos</p> <p>Volumen = 15524140.86 milímetros cúbicos</p> <p>Área de superficie = 350713.61 milímetros cuadrados</p> <p>Centro de masa: (milímetros)</p>
<p>Propiedades de masa de volumen Configuración: Predeterminado Sistema de coordenadas: -- predeterminado --</p> <p>Densidad = 0.00 gramos por milímetro cúbico</p> <p>Masa = 26312.76 gramos</p> <p>Volumen = 24591367.38 milímetros cúbicos</p> <p>Área de superficie = 466710.96 milímetros cuadrados</p> <p>Centro de masa: (milímetros) X = 0.00 Y = -201.45 Z = -180.11</p>

Figura N° 45: Cálculo del volumen de agua

Fuente: Elaboración propia

c) Cálculo del volumen de agua extraído por la bomba

Cuando la bomba extrae agua con el caudal uniforme de 500 Lt/hr, según su hoja técnica, extrae determinado volumen de agua según la cantidad de segundos transcurridos. En la figura N° 46 se aprecia la simulación de la bomba en el software Solidworks.



Figura N° 46: Simulación de la bomba en Solidworks.

Fuente: Elaboración propia

d) Cálculo del tiempo de apertura de las electroválvulas

Los valores que se han calculado corresponden a los tiempos que se tarda cada válvula en llenar y vaciar el recipiente, respectivamente.

2. Se configuraron las entradas y salidas.
3. Se armaron los conjuntos difusos.
4. Se definieron las reglas.

3.6.5. Resultados

a) Cálculo del caudal del suministro

Como resultado sale el caudal según se muestra en la ecuación N° 19.

$$Q = 7.6474375 \text{ (lt/min) } \dots\dots\dots(19)$$

b) Cálculo del volumen de agua en el recipiente

A partir de los datos de la figura N° 45 se ha elaborado la tabla N° 16.

Tabla N° 16: Relación entre altura y volumen interno

Altura (Centímetros)	Volumen(Metros cúbicos)
10	0.007324528
20	0.015524141
30	0.024591367

Fuente: Elaboración propia

c) Cálculo del volumen de agua extraído por la bomba

Se comparó el caudal extraído por la bomba transcurridos determinados segundos y la altura del recipiente para tal volumen de agua. Se obtuvo la tabla N° 17.

Tabla N° 17: Relación entre el volumen extraído por la bomba, la altura en centímetros y el tiempo transcurrido.

Volumen (m3)	Altura (centímetros)	Tiempo (segundos)
0.003	5	21.6
0.0075	10	54
0.012	15	86.4
0.0165	20	118.8
0.021	25	151.2
0.0255	30	183.6
0.03	35	216

Fuente: Elaboración propia

A partir de estos datos se saca otra función relacionando la altura con el tiempo. Se elaboró un subVI con la función de extracción de la bomba. Véase figura N° 47.

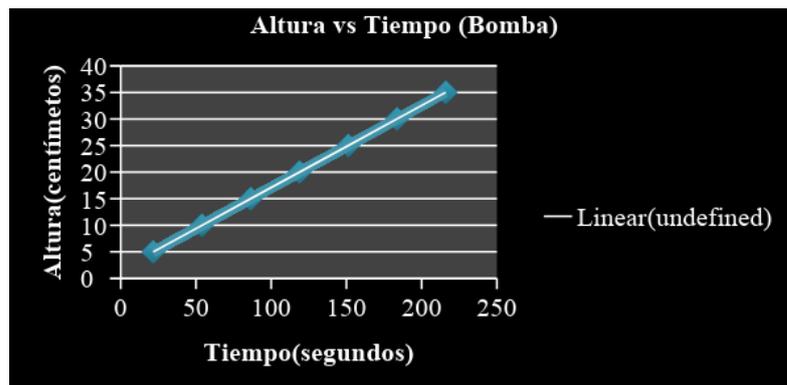


Figura N° 47: Altura vs Tiempo (bomba)

Fuente: Elaboración propia

La ecuación N° 20 relaciona dichas variables.

$$y = 0.1543 * X + 16667 \dots\dots\dots(20)$$

Se han llevado los datos de la tabla N° 17 a Microsoft EXCEL y se ha creado la gráfica mostrada en la figura N° 48, con la correspondiente ecuación (ecuación N° 21) de relación de ambas variables.

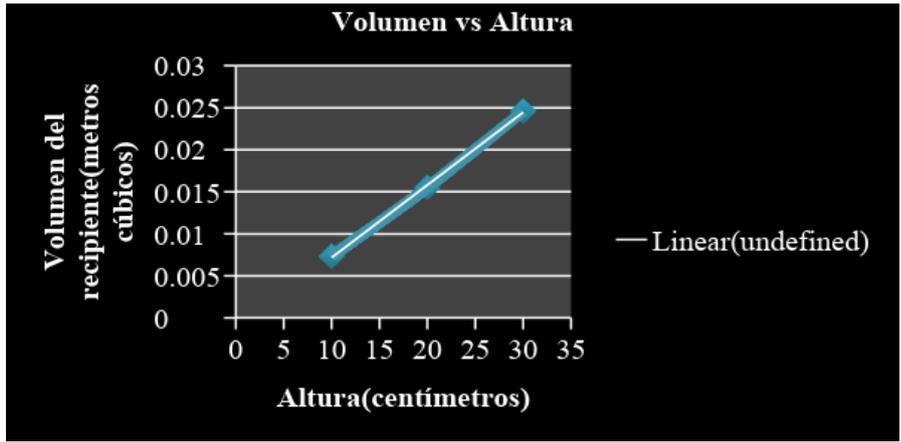


Figura N° 48: Relación entre la altura del recipiente y el volumen del recipiente
Fuente: Elaboración propia

$$y = 0.0009 * X - 0.015 \dots \dots \dots (21)$$

Donde X es la altura en centímetros y Y es el volumen de agua en metros cúbicos.

Teniendo la ecuación N° 20 se armó el subVI correspondiente que corresponde a esta función, en Labview, según se observa en la figura N° 49. Se creó una restricción en la altura, para que cuando el valor sea menor a 1 y mayor que 35, arroje el valor nulo (0). Esto es para que los resultados de la ecuación tengan sentido.

Con este subVI, se elaboró otro subVI con la ecuación N° 21 para calcular los segundos que deben transcurrir para llenar el volumen del recipiente. Véase en la figura N° 50 (arriba) el diagrama de bloques y en la figura N° 50 (abajo) el panel frontal.

Mediante pruebas con diferentes valores se sacaron las ecuaciones correspondientes al volumen y el caudal que va a emplearse en llenar y vaciar el recipiente.

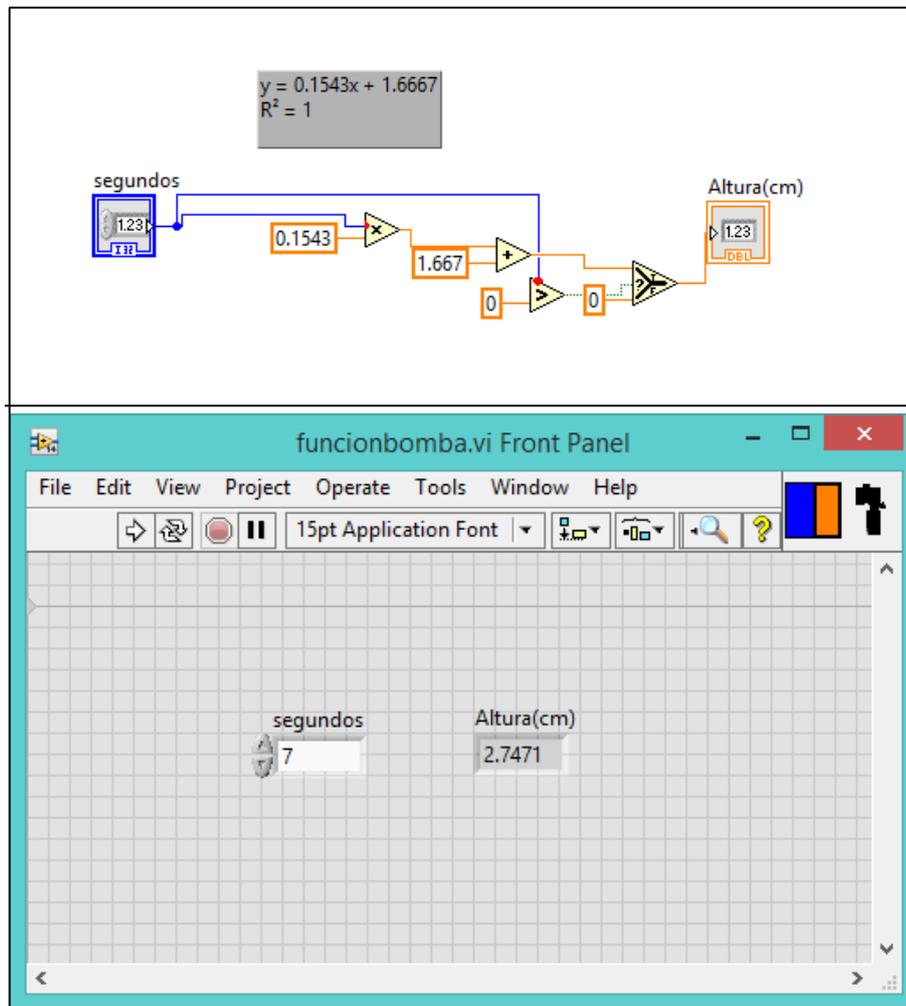


Figura N° 49: Diagrama de bloques y el panel frontal del subVI de la función de extracción de la bomba

Fuente: Elaboración propia

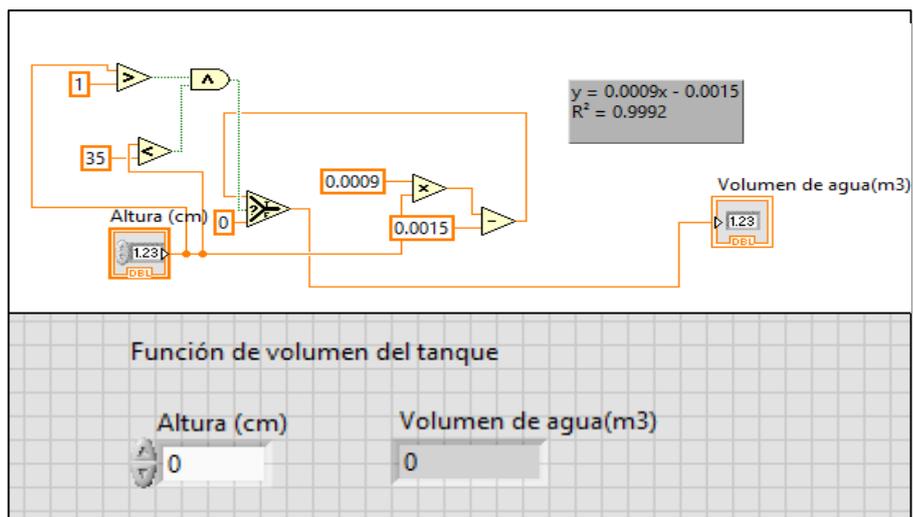


Figura N° 50: Diagrama de bloques y panel frontal del subVI que relaciona el volumen y la altura

Fuente: Elaboración propia

Del caudal expresado en la ecuación N° 12, al trasladarse a metros cúbicos se obtiene la ecuación N° 22.

$$Q = 0.000127457 \text{ m}^3/\text{s} \dots \dots \dots (22)$$

Dicho valor es multiplicado por los segundos transcurridos, dentro de un bucle For y se lo compara con el valor obtenido por el subVI que relaciona la altura con el volumen. De esa manera se obtienen los segundos que deben transcurrir para llegar a determinado volumen. Dicho subVI creado se aprecia en la figura N° 51. Una vista de la ejecución de este subVI puede observarse en la figura N° 52.

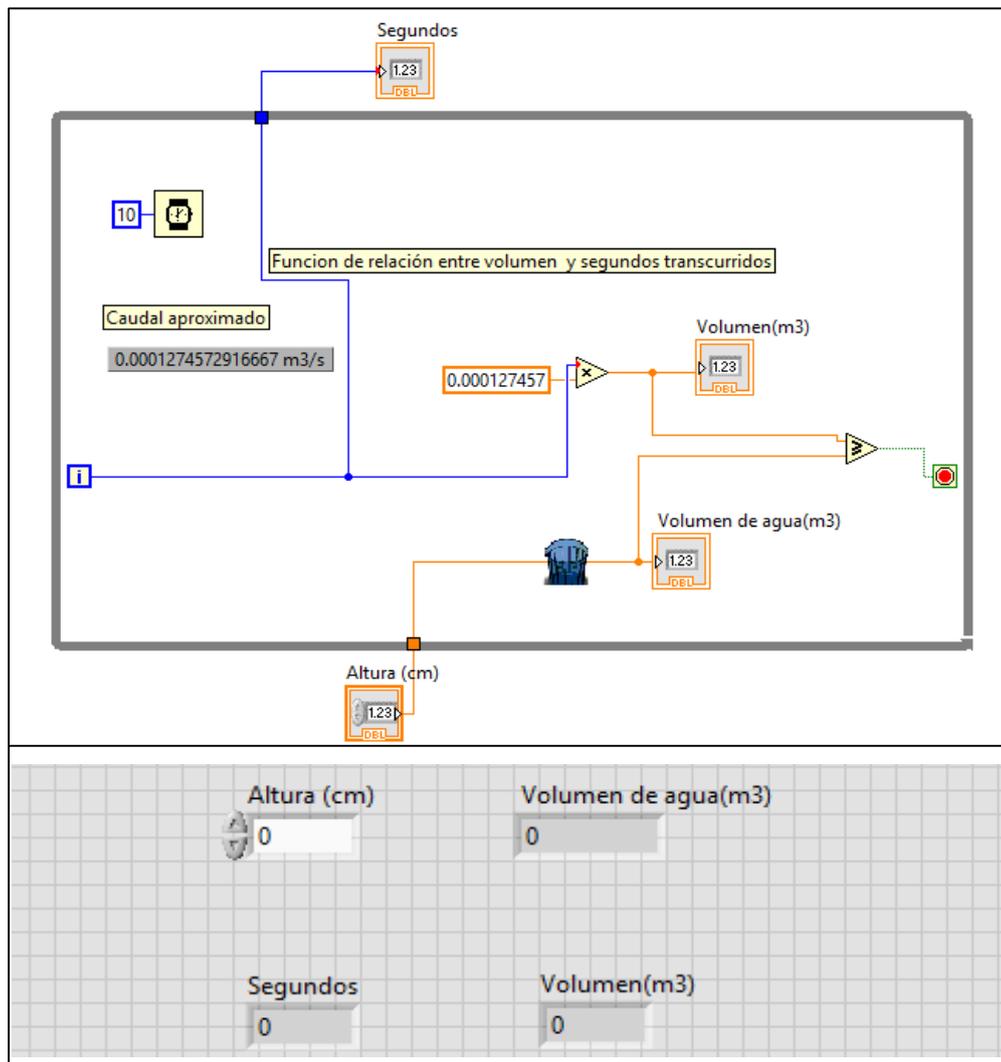


Figura N° 51: Diagrama de bloques y panel frontal del subVI que relaciona el volumen y los segundos transcurridos.

Fuente: Elaboración propia



Figura N° 52: Panel frontal mostrando los segundos transcurridos

Fuente: Elaboración propia

A continuación se ha elaborado la tabla N° 18 para hallar los segundos que deben transcurrir para llenar el recipiente a una determinada altura, hallándose la función respectiva, expresada en la ecuación N° 23. Dicha gráfica realizada en Excel puede observarse en la figura N° 53.

Tabla N° 18: Altura vs segundos transcurridos

Altura(cm)	Segundos	Altura(cm)	Segundos	Altura(cm)	Segundos	Altura(cm)	Segundos
2	3	11	66	20	130	29	194
3	10	12	73	21	137	30	201
4	17	13	81	22	144	31	208
5	24	14	88	23	151	32	215
6	31	15	95	24	158	33	222
7	38	16	102	25	165	34	229
8	45	17	109	26	172		
9	52	18	116	27	179		
10	59	19	123	28	186		

Fuente: Elaboración propia

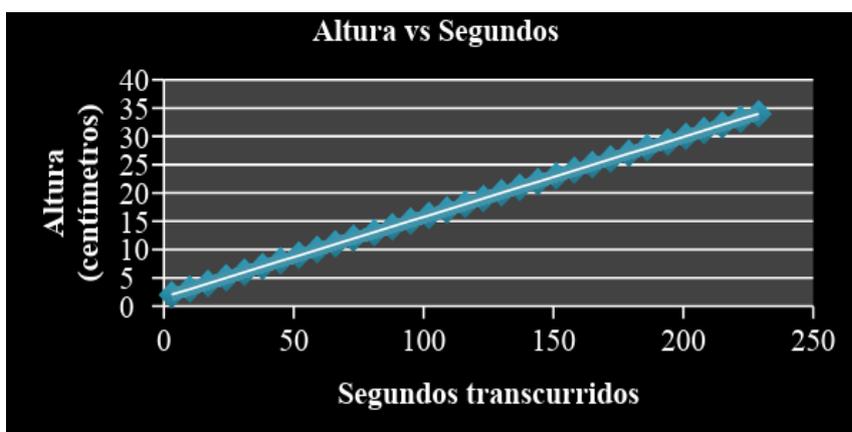


Figura N° 53: Altura vs Segundos

Fuente: Elaboración propia

$$y = 0.1415 * X + 1.6086 \dots \dots \dots (23)$$

Donde X son los segundos transcurridos. Con estos datos se creó otro subVI, con su diagrama de bloques según véase figura N° 54 (arriba) y su panel frontal (abajo).

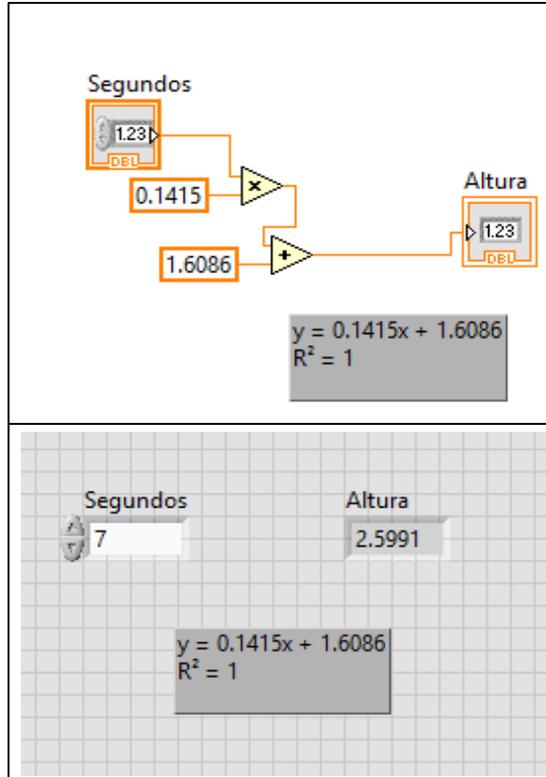


Figura N° 54: Diagrama de bloques y panel frontal del subVI que relaciona los segundos transcurridos y la altura

Fuente: Elaboración propia

Para el caso de la electroválvula 2, se ha obtenido el caudal de extracción de manera experimental. Está expresado en la tabla N° 19.

Tabla N° 19: Caudal de la electroválvula 2

Tiempo(segundos)	Volumen (mL)	Caudal (mL/s)	Caudal (m3/s)
25	100	4	0.000004

Fuente: Elaboración propia

Con el caudal obtenido se ha hallado el tiempo que tomaría vaciar el recipiente, el cual se observa en la tabla N° 20. Se observa la gráfica en la figura N° 55 y la ecuación N° 17 relacionando ambas variables.

Tabla N° 20: Tabla relacionando volumen, altura y tiempo de la electroválvula 2.

Volumen(m3)	Altura (centímetros)	Tiempo (segundos)
0.003	5	750
0.0075	10	1875
0.012	15	3000
0.0165	20	4125
0.021	25	5250
0.0255	30	6375
0.03	35	7500

Fuente: Elaboración propia

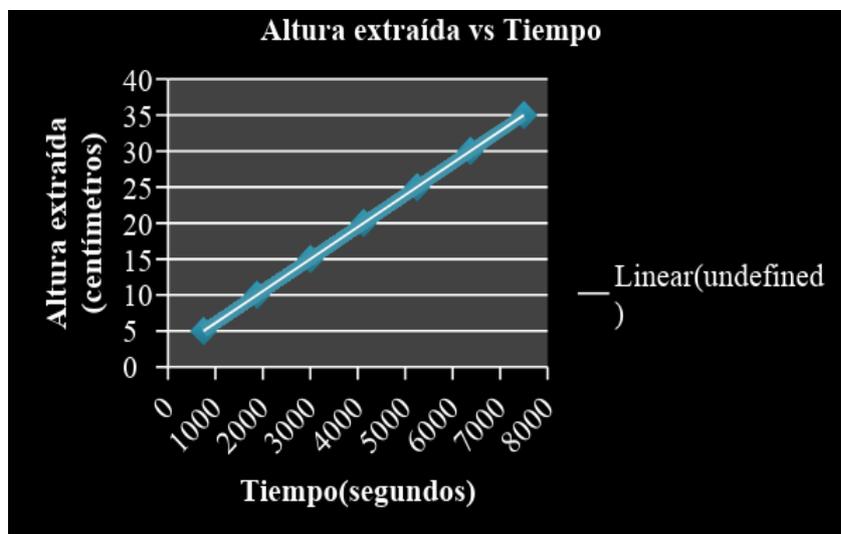


Figura N° 55: Relación entre la altura extraída y el tiempo

Fuente: Elaboración propia

$$y = 0.004 * X + 1.6667 \dots \dots \dots (17)$$

Con la ecuación N° 17 se realizó un subVI. Refieráse a la figura N° 56.

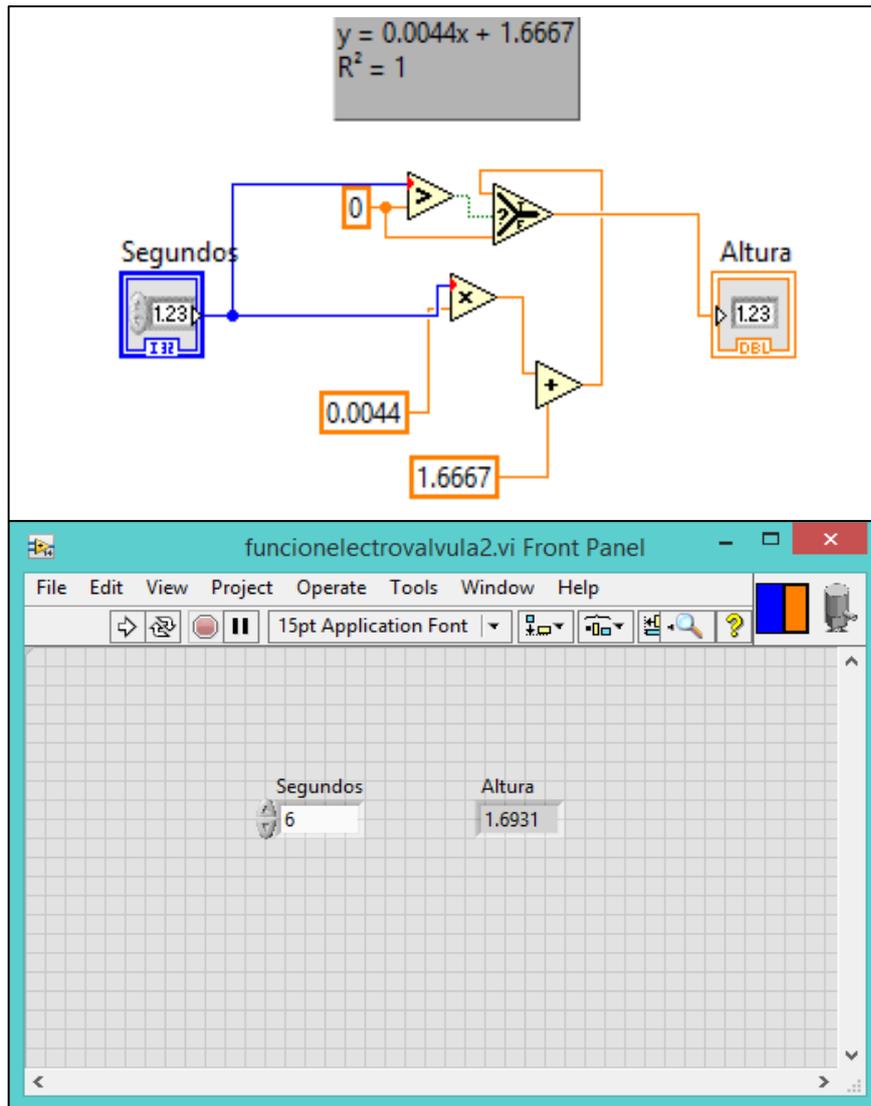


Figura N° 56: Diagrama de bloques y el panel frontal del subVI de la electroválvula 2

Fuente: Elaboración propia

La tabla N° 21 muestra los segundos que tarda dicha electroválvula en vaciar el recipiente a determinada altura.

Con la información de la tabla N° 21 se elaboraron los conjuntos difusos de la entrada, según se ve en las figuras N° 57 (arriba) y N° 57 (abajo), correspondientes al Fuzzy System Designer de Labview.

Tabla N° 21: Relación entre la altura y segundos por transcurrir.

Altura(cm)	Segundos	Segundos	Segundos (arreglado)
2	3		
3	10		
4	17		
5	24		
6	31		
7	38	7	7
8	45	14	14
9	52	21	21
10	59	28	28
11	66	35	35
12	73	42	42
13	81	50	49
14	88	57	56
15	95	64	63
16	102	71	70
17	109	78	77
18	116	85	84
19	123	92	91
20	130	99	98
21	137	106	105
22	144	113	112
23	151	120	119
24	158	127	126
25	165	134	133
26	172	141	140
27	179	148	147
28	186	186	154
29	194	163	161
30	201	198	168
31	208	177	175
32	215	198	182
33	222	191	189
34	229	198	196

Fuente: Elaboración propia

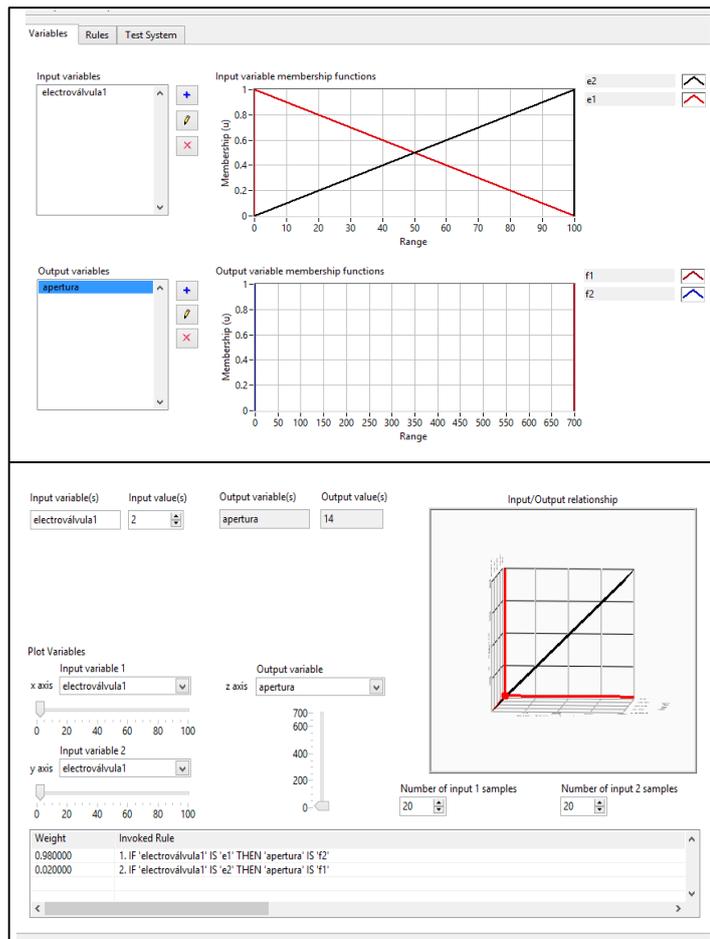


Figura N° 57: Conjuntos difusos de la electroválvula 1 y testeo

Fuente: Elaboración propia

Con la información de la tabla N° 20, se elaboraron los conjuntos difusos para la electroválvula 2(vaciado). En la figura N° 58 se observan dichos conjuntos difusos, así como su testeo.

Debido a que la ecuación obtenida que relaciona la altura con el volumen no es del todo exacta, se formula una salida para cuando la entrada es de 1 (10 segundos), eso se logra colocando una función de pertenencia Singleton en la entrada y la salida.

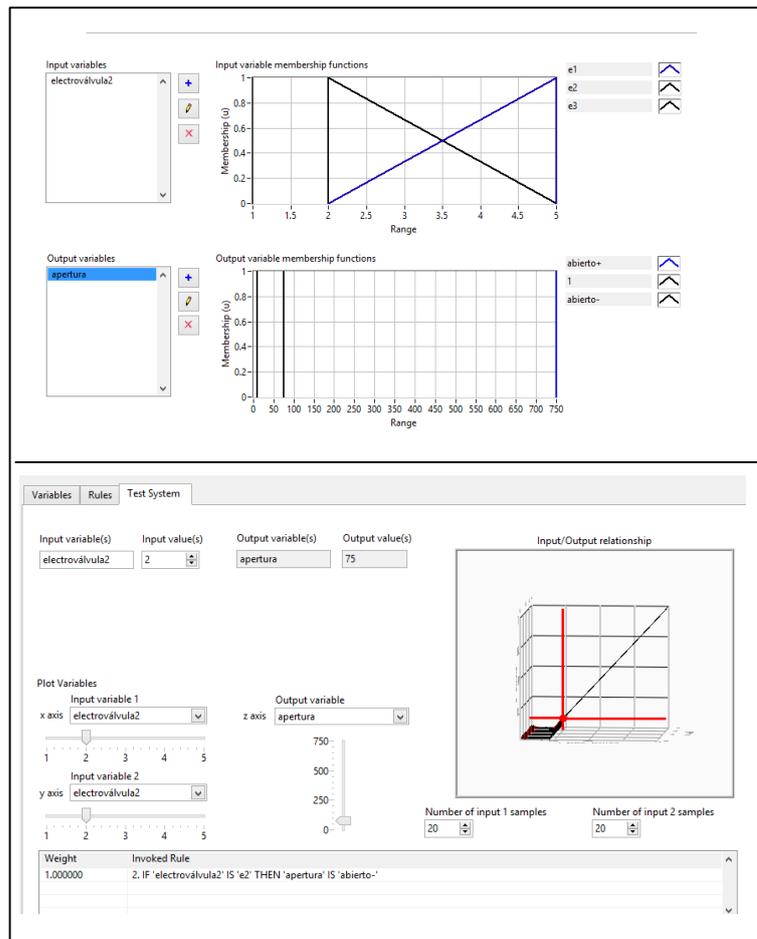


Figura N° 58: Conjuntos difusos de la electroválvula 2 y testeo

Fuente: Elaboración propia

En la figura N° 59 se observan las reglas creadas para este controlador, las que fueron sólo tres.

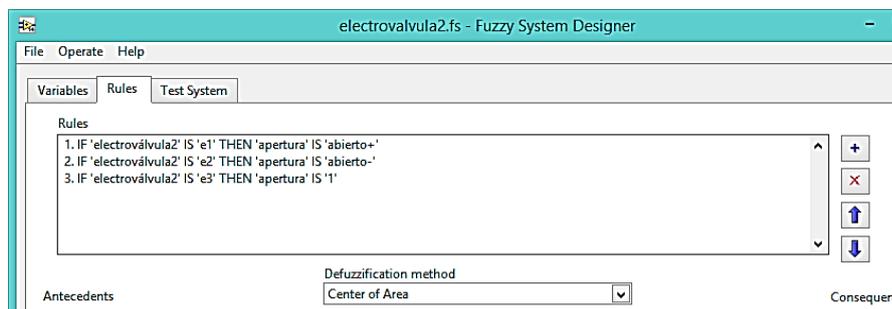


Figura N° 59: Reglas de la electroválvula 2

Fuente: Elaboración propia

Una vez creadas estas funciones se ha procedido a realizar la simulación del control de nivel. En la figura N° 60 se observa la imagen del panel frontal del control de nivel.



Figura N° 60: Panel frontal del control de nivel

Fuente: Elaboración propia

El panel frontal muestra los siguientes controladores e indicadores. Entre paréntesis se describe el tipo de componente.

Controladores: distancia hasta el Set Point (numérico), botón para la bomba (booleano),

Indicadores: tiempo de activación de la electroválvula 1 (numérico), tiempo de activación de la electroválvula 2 (numérico), gráfico del lazo de control de nivel, tanque para visualizar el nivel del recipiente (numérico), tanque para visualizar para el nivel del residuo (numérico), led para la electroválvula 1 (booleano) y led para la electroválvula 2 (booleano).

En la figura N° 61 se muestra el diagrama de bloques de la función extracción de la bomba. Contiene un bloque For, accionado por un control booleano correspondiente a la bomba. Se acciona durante 30 segundos (El delay es de 1000 ms). El valor de la altura del recipiente sirve de entrada a la función. Existe una restricción para se detenga cuando es menor a 0. Asimismo, el valor del sensor cambiará según la altura del recipiente varíe. Cuando el control booleano no está activado, la función no operará.

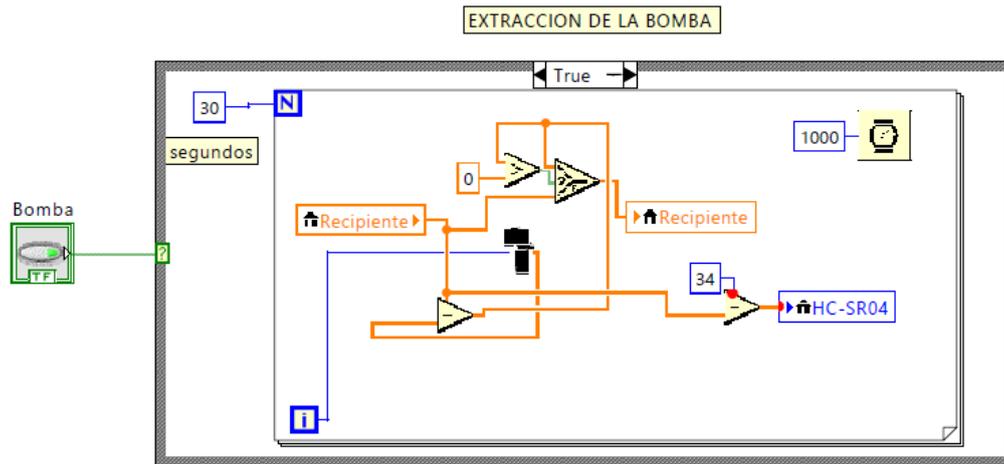


Figura N° 61: Parte I de diagrama de bloques del control de nivel

Fuente: Elaboración propia

En la figura N° 62, se observa el diagrama de bloques completo. Se hacen de las funciones:

- Extracción de la bomba
- Electroválvula 1
- Electroválvula 2

Se establecen condiciones iniciales en las salidas. Se emplean dos bucle For, uno para cada electroválvula. El primer bucle aumentará el nivel del recipiente de acuerdo al tiempo de activación de la electroválvula 1, en tanto que el segundo bucle disminuirá el nivel del recipiente y aumentará el nivel del recipiente residuo. Asimismo, para cada bucle se manda la información a los indicadores correspondientes.

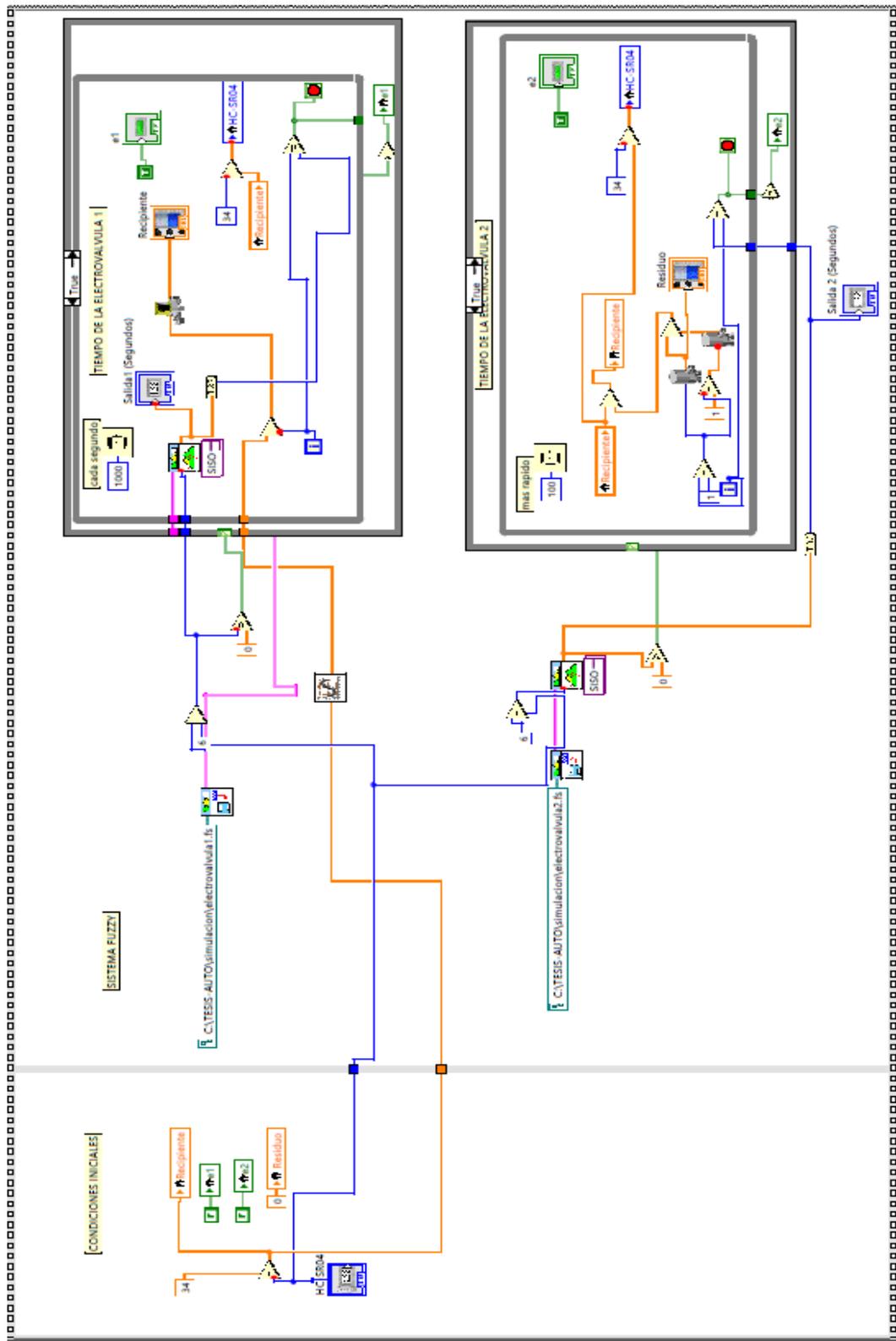


Figura N° 62: Diagrama de bloques del control de nivel

Fuente: Elaboración propia

3.6.6. Simulación final del controlador difuso en LabView

a) Características del controlador difuso

Se trata de dos sistemas SISO (Single Input, Single Output), uno para el control de cada electroválvula.

Relacionó la distancia medida con el sensor con los tiempos de activación de cada electroválvula.

b) Resultados de la simulación del controlador difuso

Se colocó una distancia de 30 centímetros hasta el fondo del recipiente. En la figura N° 63 se observa la simulación.



Figura N°63: Simulación del control de nivel

Fuente: Elaboración propia

Se activa la salida 1 (electroválvula 1) que cuenta el tiempo hasta llegar al S.P. En cuanto llegó se detuvo (Véase ítem 1 y 2 de la figura N° 64).

Como segunda prueba (Véase ítem 3,4 y 4 de la figura N° 64), se colocó una distancia de 2 centímetros hasta el fondo del recipiente, como condición inicial. Se activó la salida 2 (electroválvula 2) que contará el tiempo hasta llegar al S.P. En cuanto llegó se detuvo.

Como conclusión de esta simulación, se debió aumentar el caudal de entrada (suministro) para que el recipiente esté en el Set Point a todo momento. Se tuvo que modificar los conjuntos difusos de la electroválvula 1.

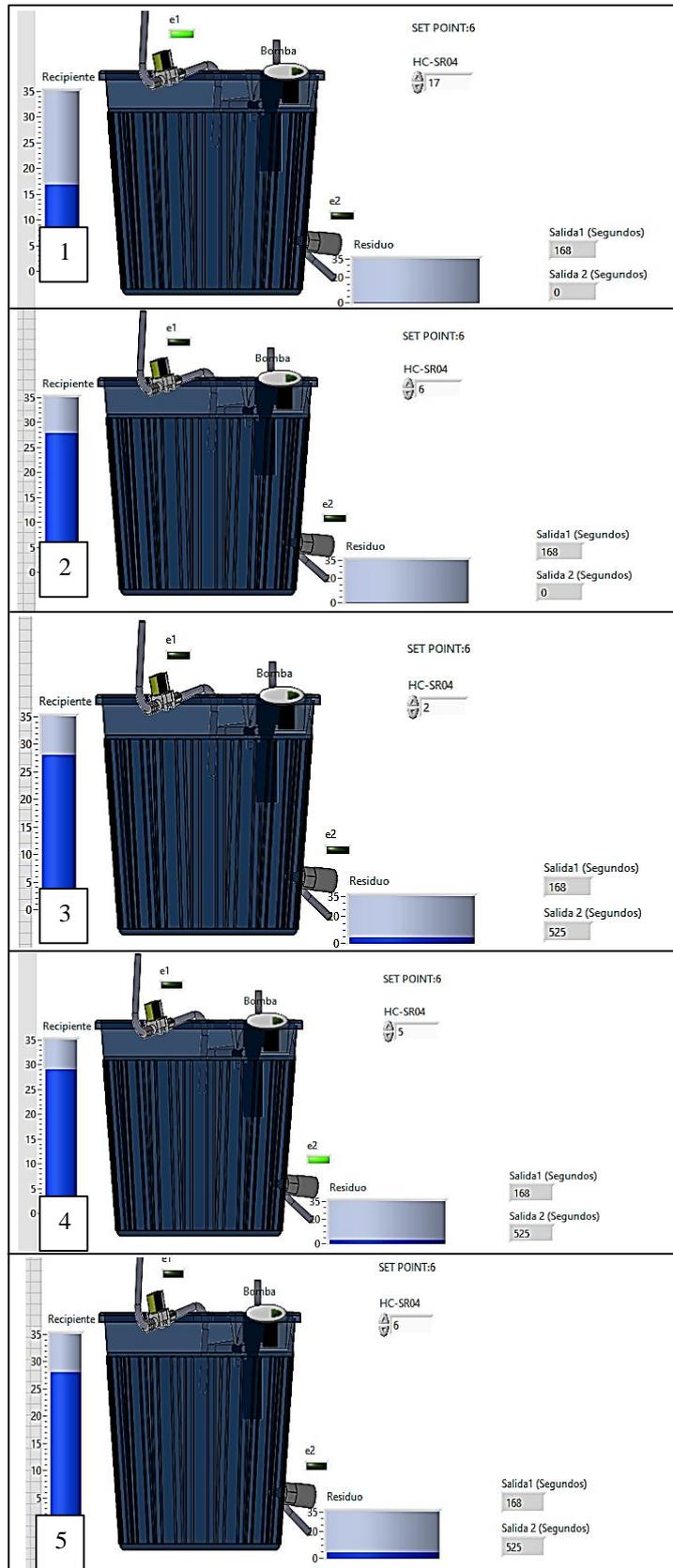


Figura N° 64: Pruebas de la simulación del control de nivel

Fuente: Elaboración propia

3.6.7. Conexiones electrónicas

En las figuras N° 65 y N° 66 se observan los diagramas de las conexiones electrónicas del sensor ultrasónico y el sensor de caudal.

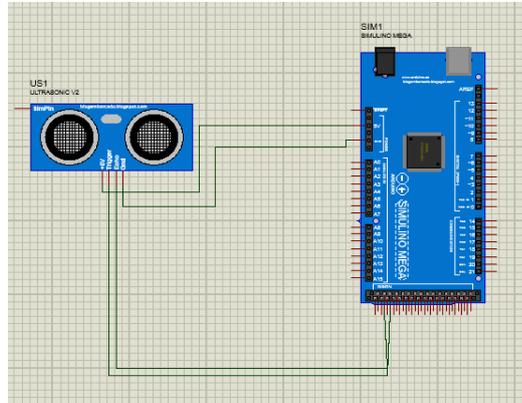


Figura N° 65: Diagrama de conexión del sensor ultrasónico

Fuente: Elaboración propia

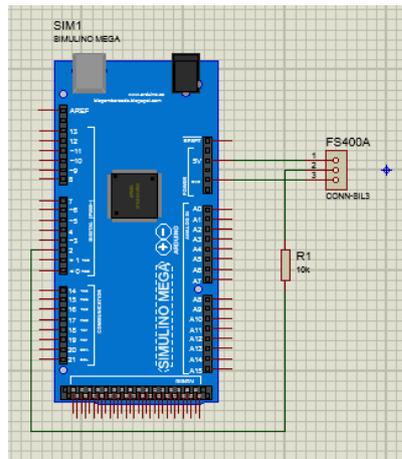


Figura N° 66: Diagrama de conexión del sensor FS400A

Fuente: Elaboración propia

3.7. Control difuso de la luminosidad

Para que exista una explotación adecuada de la energía solar, la producción agrícola se ha visto abocada a la utilización de ambientes cerrados. Cuando exista un déficit de ella, se realiza un control tal que permita suplir los niveles de energía con métodos artificiales. Para poder medir la cantidad de lux de manera más adecuada se ha empleado un sensor BH1750, el cual mide la iluminancia en cantidad de lux.

3.7.1. Ubicación de los componentes

Los módulos leds se ubicaron en la parte superior del mini invernadero.

El módulo BH1750 se ubicó en la parte inferior del mini invernadero, hacia la parte derecha. En la figura N° 67 se tiene una vista de los componentes a emplear en este lazo de control, desarrollado en el software Solidworks.

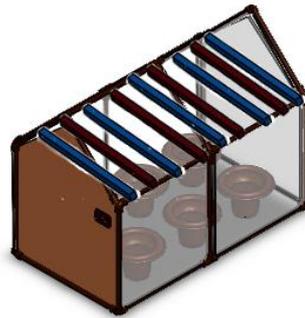


Figura N° 67: Componentes del control de luminosidad

Fuente: Elaboración propia

3.7.2. Entradas y salidas del controlador difuso

En la tabla N° 22 se muestra la entrada y salida del controlador difuso.

Tabla N° 22: Características del control de la variable luminosidad

Variable	Luminosidad
Entrada(s)	Cantidad de lux
Salida(s)	Voltaje de los módulos LED

Fuente: Elaboración propia

3.7.3. Objetivos

1. Crear el algoritmo para el control de la luminosidad.
2. Configurar las entradas y salidas.
3. Armar los conjuntos difusos.
4. Definir las reglas.

3.7.4. Desarrollo

1. Se creó el algoritmo para el control de la luminosidad.

Se diseñó el control de la luminosidad según el diagrama de flujo de la figura N° 68. Se ha investigado la cantidad de lux adecuada para las plantas (Rosas), el cual es un valor alrededor de 3000 lux.

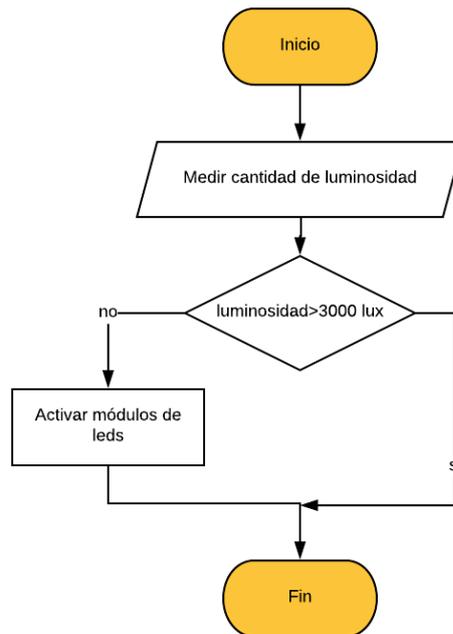


Figura N° 68: Diagrama de flujo del control de luminosidad

Fuente: Elaboración propia

Asimismo la cantidad de lux será visualizada en la pantalla LCD 16x4.

2. Se configuraron las entradas y salidas.
3. Se armaron los conjuntos difusos.
4. Se definieron las reglas.

3.7.5. Resultados

Se definieron la entrada y la salida y sus respectivos conjuntos difusos. En la figura N° 69.

- 1: El conjunto difuso con valor de 0.
- 2: El conjunto difuso con valores de entre 50 y 3000.

3: El conjunto difuso con valores de entre 3000 y 3100.

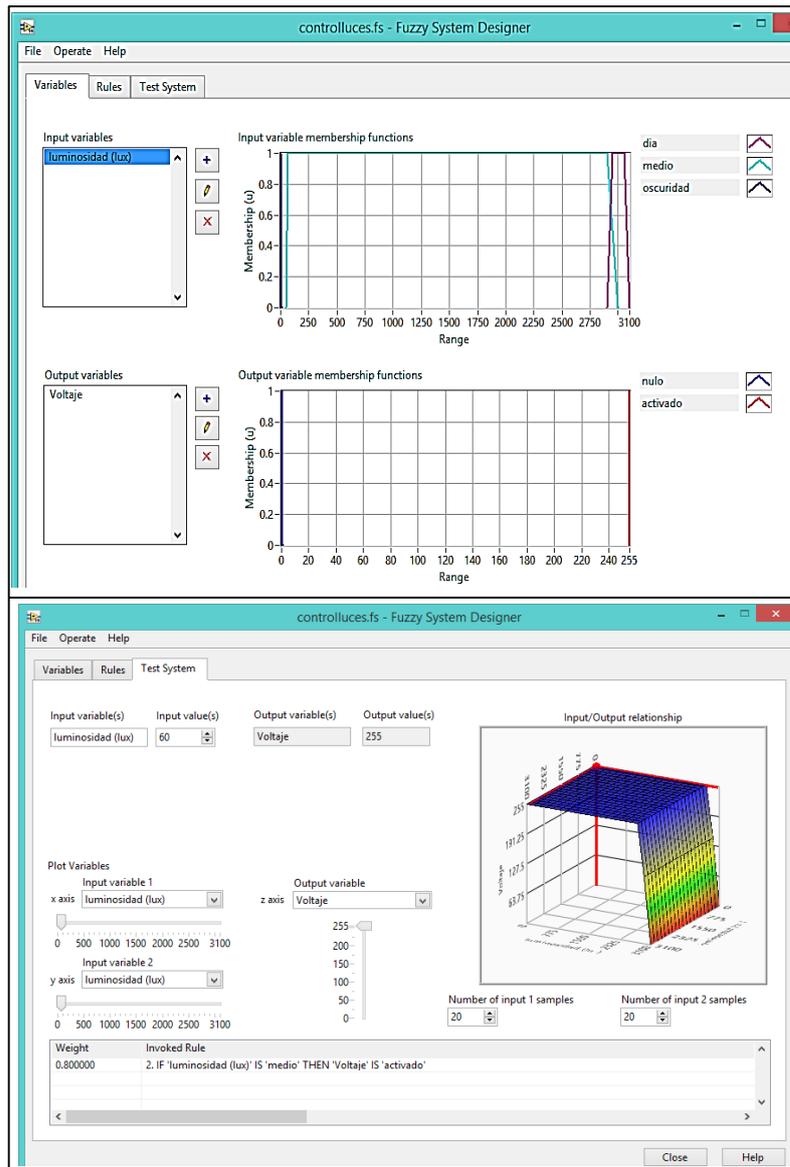


Figura N° 69: Conjuntos difusos del control de la luminosidad y testeo

Fuente: Elaboración propia

Las funciones de pertenencia de la entrada tienen la forma trapezoidal porque así se logra que un conjunto de valores de entrada tenga el mismo grado de pertenencia a un valor de salida. En este caso, se coloca un rango de valores cercano al ideal (3000 lux) y se lo crea como el conjunto “día”. El valor de 0 lux corresponde a una función de un solo valor (0) que corresponde al conjunto “noche”. Y el rango que está entre ambos valores se le da el nombre de “medio”. Las funciones de pertenencia de la salida tienen la forma Singleton porque el rango de valores que adopta la salida sólo es 0 (desactivado) y 255 (activado). En la figura N° 70 se aprecian las reglas creadas.

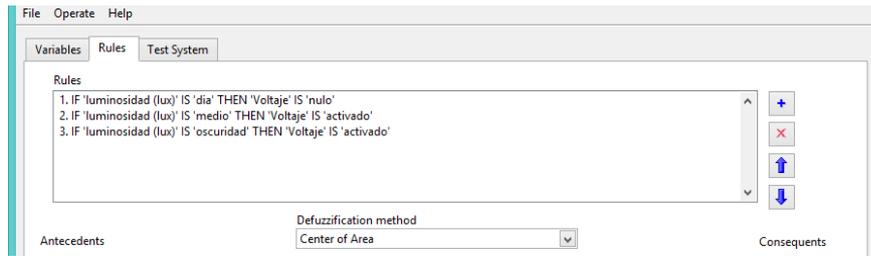


Figura N° 70: Reglas del control difuso de la luminosidad.

Fuente: Elaboración propia

La creación de reglas en este caso fue sencilla. Relacionó la activación de los leds (conjunto “activado”), con un valor menor al del conjunto “día”, para de esta manera activar los leds. El diagrama de bloques, según se ve en la figura N° 71(arriba), activó indicadores booleanos cuando la salida del controlador (voltaje) era de 255.

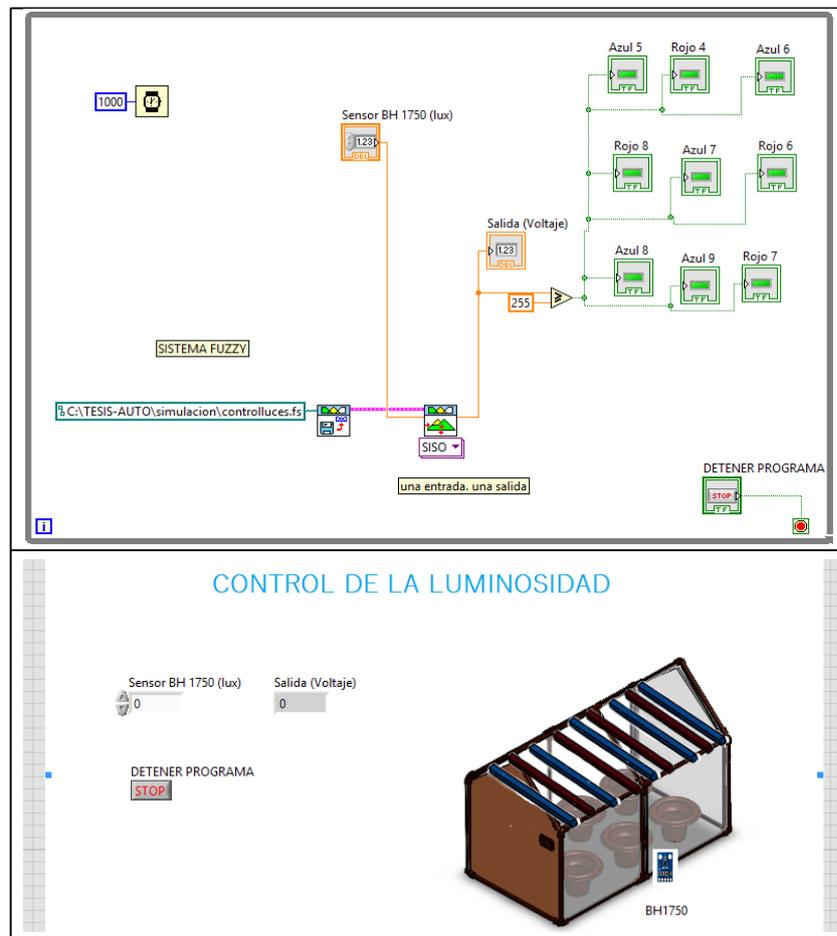


Figura N° 71: Diagrama de bloques y el panel frontal del subVI del control de la luminosidad

Fuente: Elaboración propia

En la figura N° 71 (abajo) se aprecia el panel frontal, el cual muestra los siguientes indicadores y controles.

Controles: la salida a los módulos led (numérico), botón para detener el programa (booleano).

Indicadores: valor medido por el sensor BH1750 (numérico), cinco leds para los módulos leds (booleano).

Imágenes: gráfico del mini invernadero, gráfico del mini invernadero.

3.7.6. Simulación final del controlador difuso en LabView

a) Características del controlador difuso

Se trata de un sistema SISO (Single Input, Single Output), para la activación de los módulos led.

b) Resultados de la simulación del controlador difuso

Se simuló el sistema colocando diferentes valores en la entrada. En la figura N° 72, se aprecia la prueba para 100 lux (arriba) y 2000 lux (abajo), viendo que los leds se prenden en ambos casos.

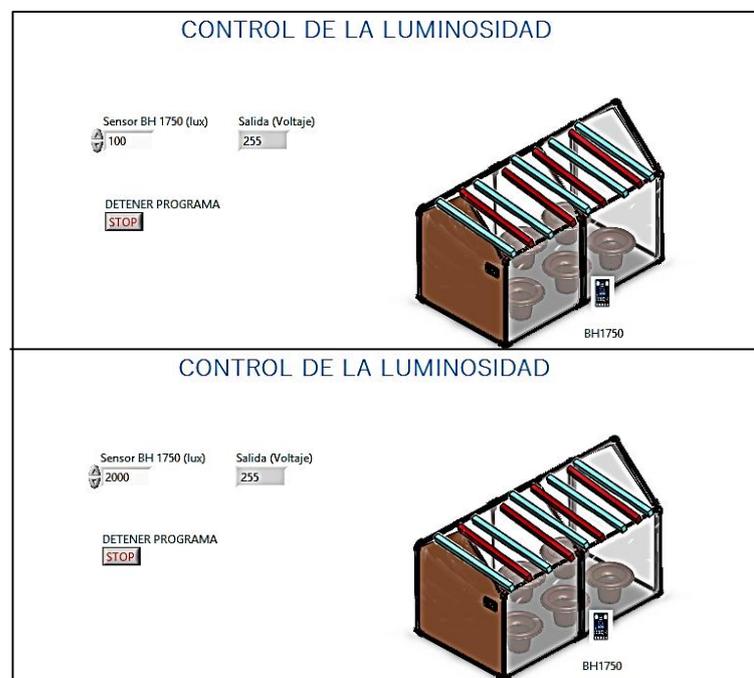


Figura N° 72: Prueba del control de la luminosidad

Fuente: Elaboración propia

3.8. Control difuso del fertirriego

3.8.1. Ubicación de los componentes

La bomba para el fertirriego se ubicó dentro de recipiente, según se detalló en el apartado 3.5.1.

Se han empleado dos mangueras de plástico de un metro de longitud, en la que se han colocado los goteros a distancias equidistantes, tal que estén encima de cada maceta. Véase figura N° 73.

El recipiente del fertilizante se ubicó cerca de la bomba, conectada hacia ella mediante una pequeña manguera flexible de ½ centímetro de diámetro, la cual vino con la bomba. Véase figura N° 74.



Figura N° 73: Componentes para el control de fertirriego

Fuente: Elaboración propia

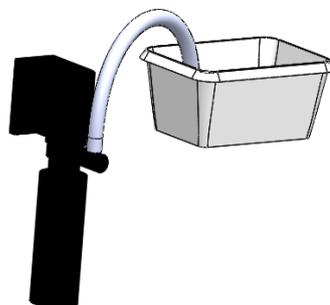


Figura N° 74: Recipiente para el fertilizante conectado a la bomba.

Fuente: Elaboración propia

3.8.2. Entradas y salidas del controlador difuso

La tabla N° 23 muestra la variable del controlador difuso, así como la entrada y la salida.

Tabla N° 23: Características del control de la variable luminosidad

Variable	Humedad en la tierra de las macetas
Entrada(s)	Cantidad de humedad en la tierra de las macetas
Salida(s)	Voltaje de la bomba

Fuente: Elaboración propia

3.8.3. Objetivos

1. Crear el algoritmo para el control del fertirriego.
2. Configurar las entradas y salidas.
3. Armar los conjuntos difusos.
4. Definir las reglas.

3.8.4. Desarrollo

Para crear el algoritmo para el control del fertirriego se tomó en consideración la cantidad de humedad presente en la tierra de las macetas y la frecuencia de riego recomendada para las plantas. Véase figura N° 75.

El control del fertirriego se da a partir del cronograma de riego establecido para la planta (dos veces por semana) y la humedad que se halla en la tierra de las macetas.

El control difuso activa el actuador (bomba) de acuerdo a las reglas implementadas en el controlador, ya que solo se dará la activación o disparo (ON/OFF) de la electroválvula al cumplir las condiciones.

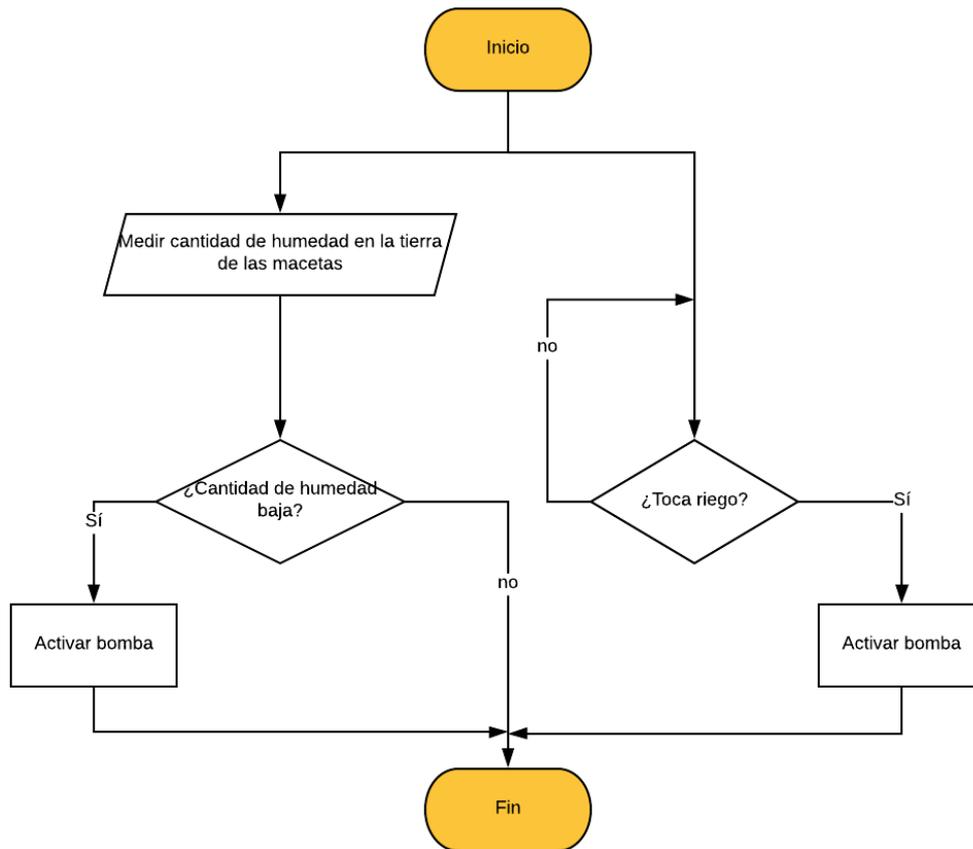


Figura N° 75: Diagrama de flujo del control del fertirriego

Fuente: Elaboración propia

Para configurar las entradas y salidas fue necesario realizar los siguientes pasos:

a) Análisis del Inyector Venturi presente en la bomba

Para analizar las condiciones del fertirriego (inyección del fertilizante) se ha empleado el complemento Flow Simulation de Solidworks, el cual permite calcular la velocidad y la presión en el inyector que viene integrado con la bomba. Una vez que se establece el dominio computacional del objeto a analizar, asegurándose de que tanto entradas como salidas estén cerradas, se definen las condiciones en cada una de éstas, estableciendo las condiciones de entorno (Boundary Conditions). Véase ítem 1 de la figura N° 76. Tal como la velocidad o el volumen de flujo en la entrada, y la presión atmosférica a la salida. Véase ítem 2 de la figura N° 76.

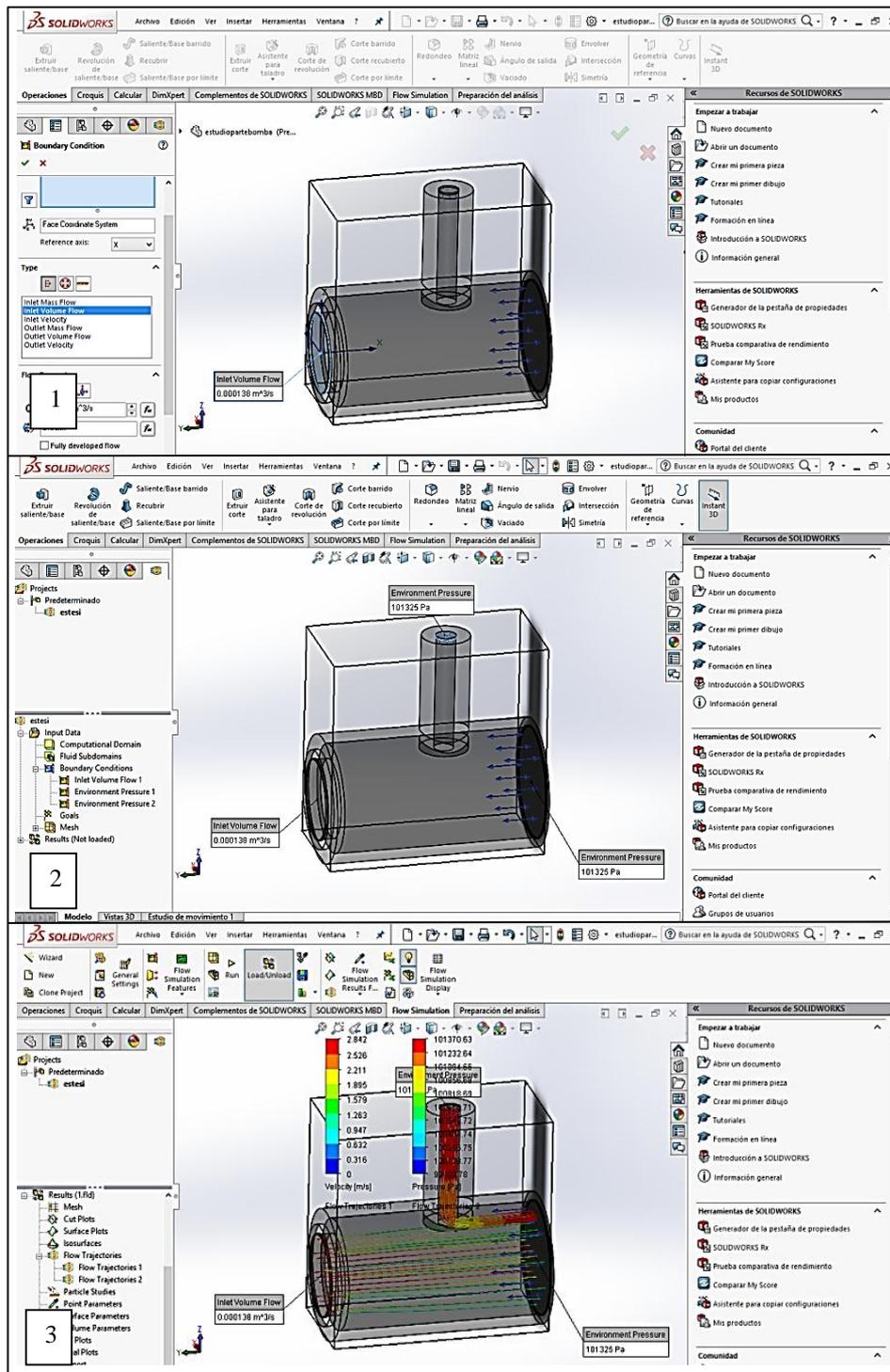


Figura N° 76: Análisis del inyector Venturi

Fuente: Elaboración propia

Para este caso se han establecido los datos de la tabla N° 24.

Tabla N° 24: Definición de entradas y salidas del inyector

Entrada 1	Salida1	Salida2
0.000138 m ³ /s	Presión atmosférica (101325 Pa)	Presión atmosférica (101325 Pa)

Fuente: Elaboración propia

La entrada es designada de acuerdo al caudal de la bomba elegida (500 lt/h).

Con estos datos el programa permite calcular la velocidad y/o presión en las salidas establecidas, así como otros parámetros, y dibujarlas con curvas de colores. Véase ítem 3 de la figura N° 76.

b) Cálculo de la cantidad de agua para cada maceta

Para realizar la simulación primero se procedió a hacer pruebas con la cantidad de agua que se debe despedir en cada maceta.

Con la ayuda de un gotero de 20 mL (véase figura N° 77) se hicieron pruebas con el código del sensor de humedad (anexo XII).



Figura N° 77: Gotero de 20 mL

Fuente: <https://www.medicalcenter.com.mx/frasco-gotero-20-ml>

3.8.5. Resultados

a) Análisis del Inyector Venturi presente en la bomba

En el ítem 3 de la figura N° 76 se tiene una vista de los resultados del Solver del Flow Simulation. En conclusión, la velocidad en la salida 2 se mantiene según la de la entrada. Se crea una presión de vacío debido a la cavidad más delgada (efecto Venturi).

b) Cálculo de la cantidad de agua para cada maceta

En aproximadamente 30 segundos la maceta llega a humedecerse a un valor correspondiente a muy húmedo, empleando 100 mL de agua. De acuerdo a la función

que se obtuvo, según se ve en la figura N° 78, se estableció la ecuación para realizar la simulación en Labview, que se puede ver en la figura N°79.

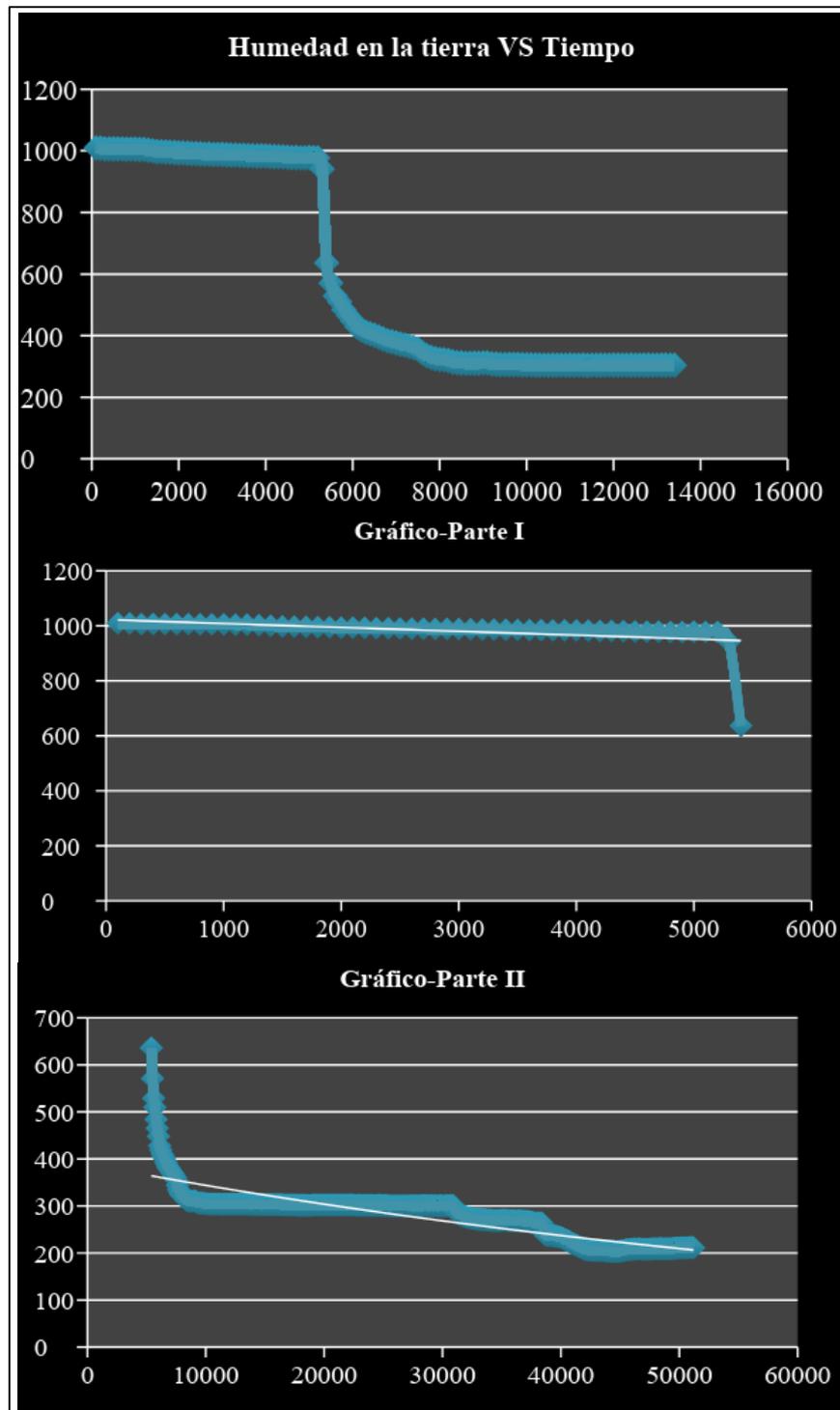


Figura N° 78: Gráfica humedad en la tierra vs segundos

Fuente: Elaboración propia

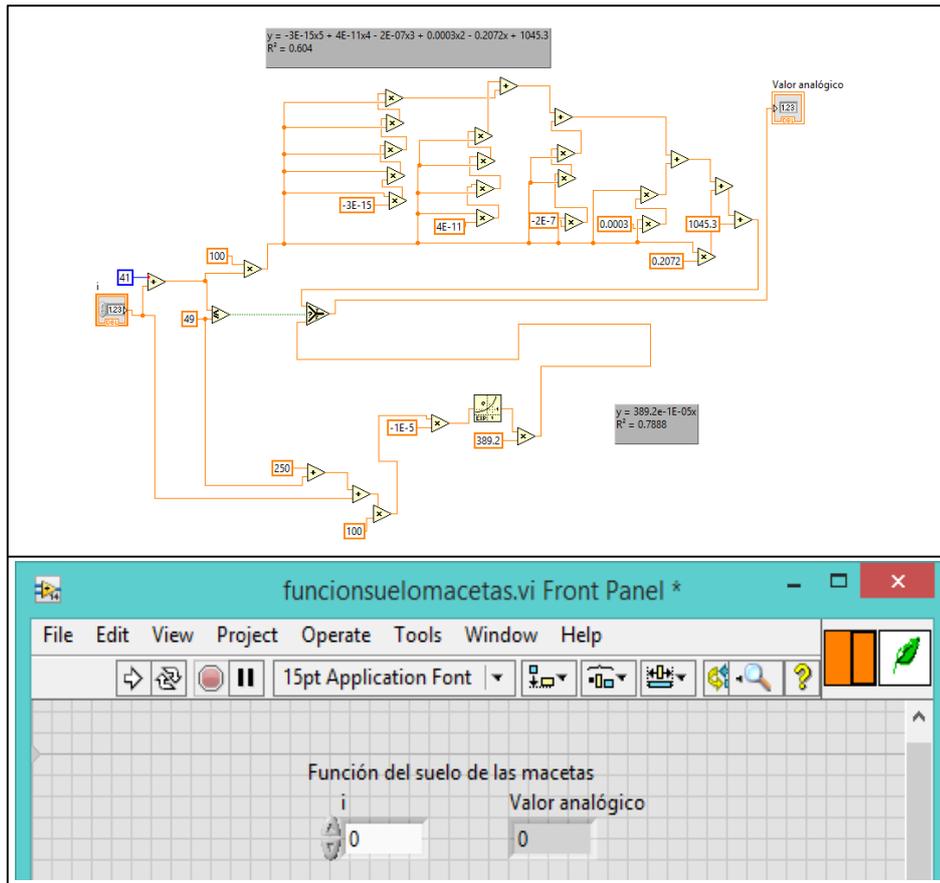


Figura N° 79: Diagrama de bloques y panel frontal de la función humedad de las macetas
Fuente: Elaboración propia

Con este subVI se procede a crear el VI principal. La función de este subVI era mostrar el cambio del valor. Cada 84 horas se realiza el fertirriego. La cantidad de tiempo que se accionará la bomba dependerá del valor de humedad de las macetas de ese momento. En la figura N° 80 se observan las reglas creadas para este controlador difuso.

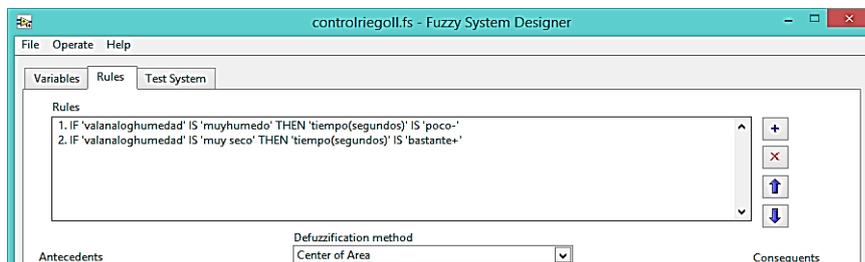


Figura N° 80: Reglas del control del fertirriego
Fuente: Elaboración propia

La figura N° 81 (arriba) muestra las variables de entrada y salida y sus conjuntos difusos en el Fuzzy System Designer, así como un testeo de los mismos (abajo). Se empleó una forma gausseana porque de esta manera se acomoda más a la forma en que cambia el valor correspondiente a la humedad de la tierra (cambios abruptos).

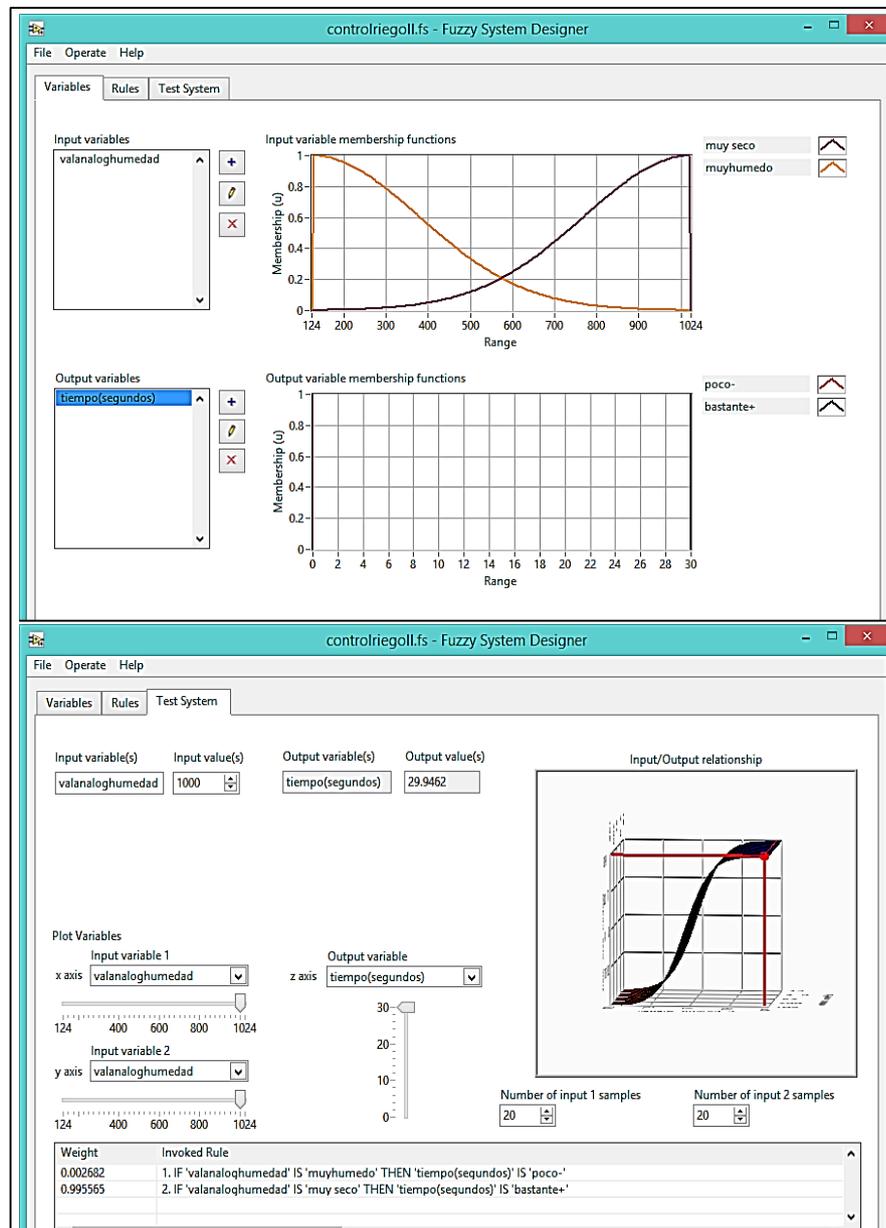


Figura N° 81: Conjuntos difusos del control del fertirriego y testeo

Fuente: Elaboración propia

En la figura N° 82 se muestra el diagrama de bloques del control difuso del fertirriego. A la izquierda se muestra la programación cuando el condicional es verdadero, y a la derecha cuando es falso.

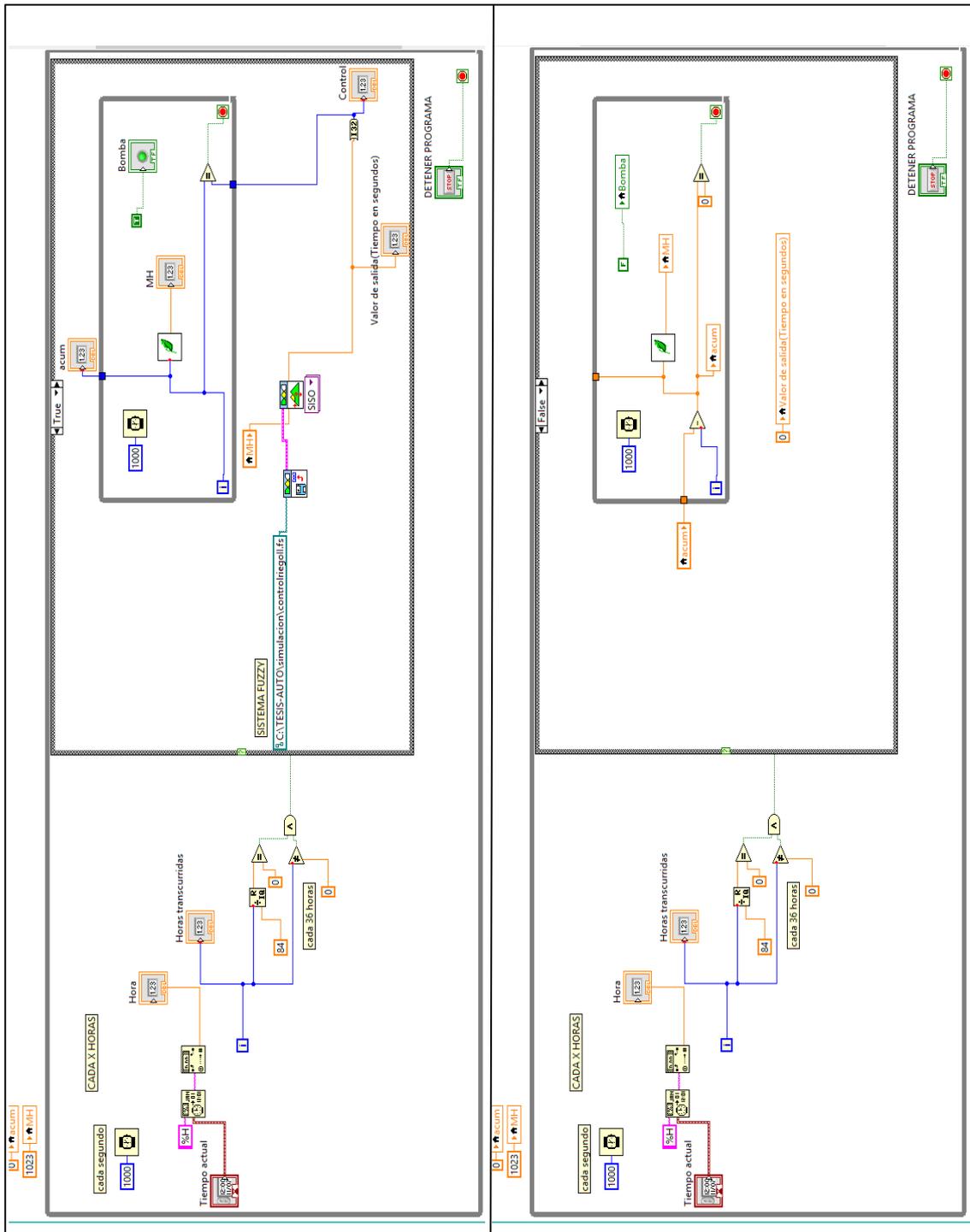


Figura N° 82: Diagrama de bloques del control del fertirriego

Fuente: Elaboración propia

En la figura N° 83 se tiene una vista de panel frontal, el cual contiene:

Controles: tiempo actual (numérico), hora (numérico), horas transcurridas (numérico), valor del sensor MH (numérico), un botón para detener el programa (booleano).

Indicadores: tiempo de activación de la bomba (numérico)

Imágenes: Gráfico del conjunto de componentes, gráfico del sensor MH



Figura N° 83: Panel frontal del control del fertirriego

Fuente: Elaboración propia

3.8.6. Simulación final del controlador difuso en LabView

a) Características del controlador difuso

Se trata de un sistema SISO (Single Input, Single Output), para la activación de la bomba de fertirriego.

Activó la bomba según la programación de riego (cada 84 horas) y si la humedad en la tierra de las plantas era muy baja.

b) Resultados de la simulación del controlador difuso

Se hizo una simulación del control difuso del fertirriego, el cual se activaba cada vez que se transcurrían 84 horas, durante el tiempo mostrado en el valor de salida, el cual es la salida del controlador difuso para determinado valor de humedad. En la figura N° 84 se tiene una visualización de este procedimiento.



Figura N° 84: Simulación del control difuso del fertirriego

Fuente: Elaboración propia

3.9. Muestreo de las variables que afectan al cultivo

“El nivel de respuesta (dinamismo) requerido por el sistema a diseñar no es muy complejo debido a que la variación de temperatura ambiental interna en el invernadero es lenta.” (López, 2008, p.30). El número de sensores a emplear dependerá del volumen del invernadero al cual se desea implementar el sistema a diseñar. Es necesario tomar un promedio de los valores sensados (López, 2008).

- Para el caso de control de nivel del recipiente, se ha tomado el promedio de los valores sensados para evitar el efecto del rebote.
- Para el control de la humedad y temperatura, se tomará el muestreo cada diez minutos.
- Para el control de la luminosidad, este será cada diez minutos.
- Para el control del fertirriego, éste será cada 84 horas (dos veces por semana), y se dará cuando inicie el día.

3.10. Visualización de las variables a controlar

Se ha considerado el empleo de una pantalla LCD de 16 x 4 para visualizar el estado de las variables humedad relativa y temperatura del aire dentro del invernadero y la cantidad de luminosidad existente, medidos con los sensores DHT22 y BH1750. El código se muestra en el anexo XIII.

CAPITULO IV: IMPLEMENTACIÓN Y RESULTADOS

4.1 Proceso previo a la implementación

El proceso fue realizado de acuerdo a los sistemas que lo componen. Siendo el orden:

- Control de la humedad relativa y temperatura
- Control de la luminosidad
- Control de nivel del recipiente
- Control del fertirriego

En este capítulo se describen las pruebas previas realizadas para cada lazo de control, así como los resultados de cada una. Y finalmente la implementación final de cada una.

4.2 Conexiones generales

Para realizar las pruebas con cada lazo de control se ha empleado el Arduino Mega 2560, según las conexiones que se ven en la figura N° 85.

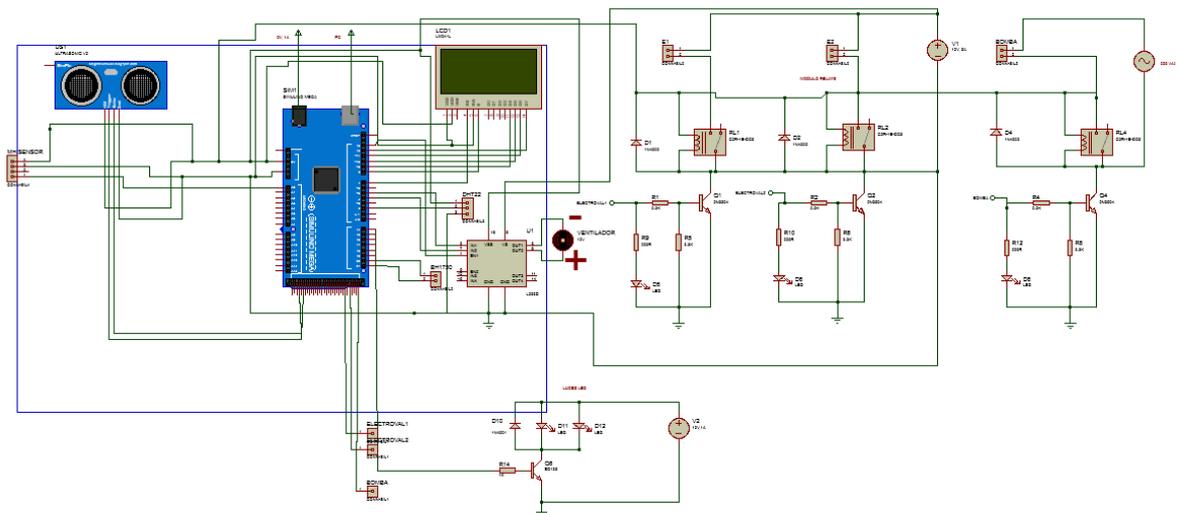


Figura N° 85: Conexiones eléctricas generales para las pruebas

Fuente: Elaboración propia

Las fuentes de alimentación empleadas son:

- Un transformador de voltaje de 220 VAC 50/60 Hz a 12 V DC, 5 A, para las electroválvulas.

- Un transformador de voltaje de 220 VAC 50/60 Hz a 12 V DC, 1 A, para las luces
- Cable de datos de la computadora portátil, para el controlador Arduino.
- La corriente eléctrica de 220 VAC, para alimentar los transformadores de voltaje y la bomba.

4.3 Control de la humedad relativa y temperatura

4.3.1. Prueba previa

Para realizar la prueba se realizaron los siguientes pasos:

- Se creó el código del anexo IX.
- Se realizaron las conexiones eléctricas y electrónicas correspondientes según se observa en la Figura N° 85.
- Se alimentó el Ventilador SP06-08
- Se alimentó el sensor DHT22.
- Se conectó el controlador a la PC

En la figura N° 86 se pueden ver las conexiones realizadas.

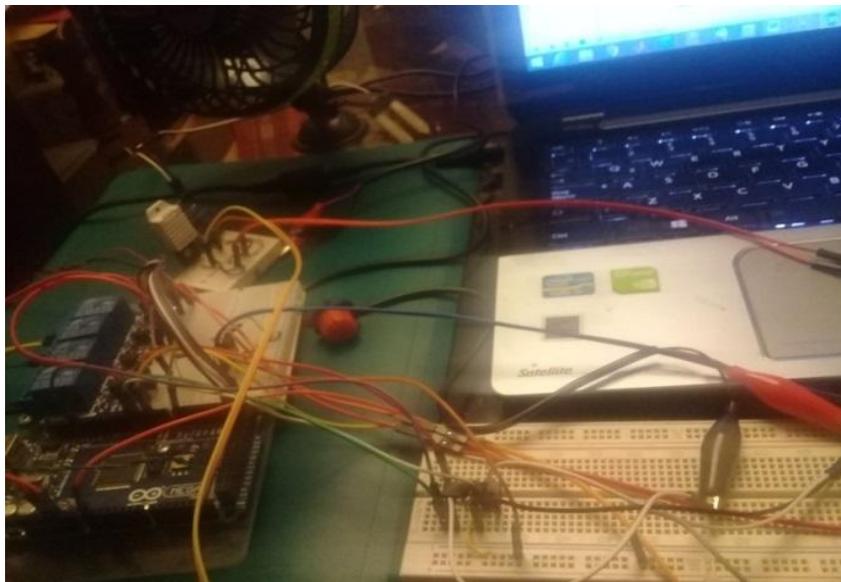


Figura N° 86: Conexión para la prueba del ventilador

Fuente: Elaboración propia

En la figura N° 87 (izquierda) se puede apreciar una vista del código para la prueba del ventilador.

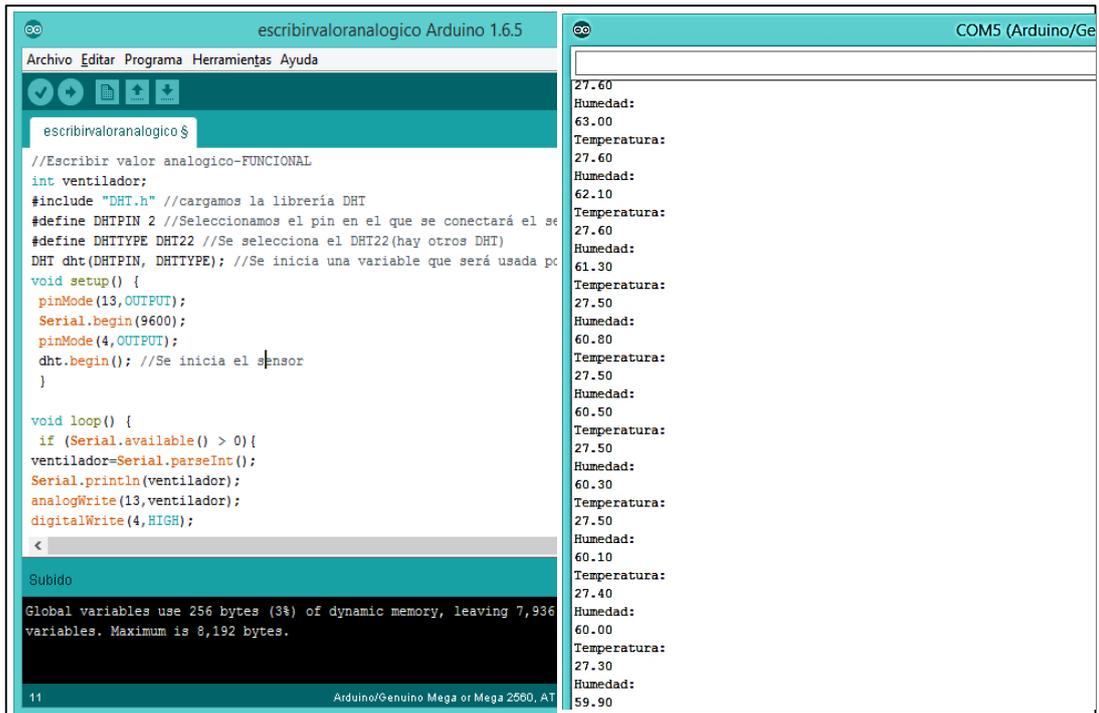


Figura N° 87: Código para la prueba del ventilador

Fuente: Elaboración propia

4.3.2. Resultados de la prueba

La prueba sirvió para ver cómo el ventilador de la figura afectaba a las variables medidas con el sensor DHT22, empleando el código del anexo IX.

A partir del código, en el puerto serial COM5 se apreciaron los valores sensados con el sensor DHT22, tal como aparece en la figura N° 87 (derecha).

Se trasladaron los resultados a Microsoft Excel y se obtuvo la gráfica mostrada en la figura N° 88.

Los valores obtenidos están expresados en la tabla N° 13.

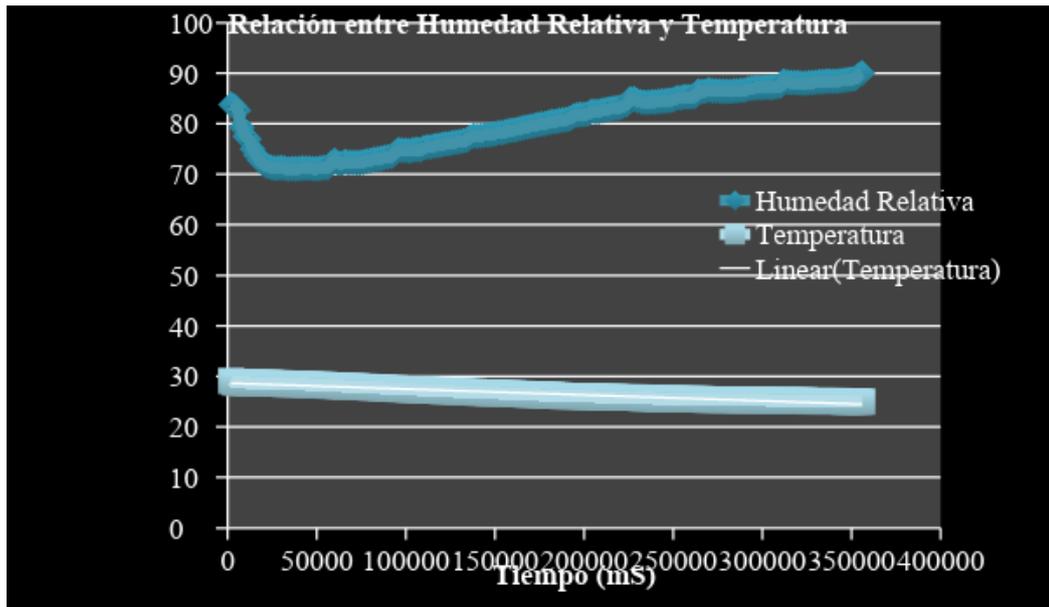


Figura N° 88: Gráfica de la relación entre humedad relativa y temperatura

Fuente: Elaboración propia

Aquí se ve que la humedad relativa y temperatura al principio descienden, pero luego la humedad relativa aumenta inversamente proporcional a la temperatura.

Se decidió reemplazar el ventilador por uno de más potencia, como es el SYM BANG modelo A12038V2HSL-C, el cual funciona a 220 VAC, cuyas características se mencionan en la tabla N° 10 para poder apreciar mejor los cambios de temperatura y humedad relativa.

4.4 Control de la luminosidad

4.4.1. Prueba previa

Para realizar esta prueba se siguieron los siguientes pasos:

- Se creó el código del anexo XIV.
- Se realizaron las conexiones eléctricas y electrónicas correspondientes según se observa en la Figura N° 85.
- Se alimentó el sensor BH1750.
- Se alimentó la iluminación LED.
- Se conectó el controlador a la PC.

4.4.2. Resultados de la prueba

Empleando el sensor BH1750 con el código en el anexo XIV, se tomó las medidas de luminosidad con y sin la malla Raschel al mediodía.

En la figura N° 89 (arriba) se observa la iluminancia bajo el sol a mediodía y en la parte de abajo se observa la iluminancia bajo la malla.

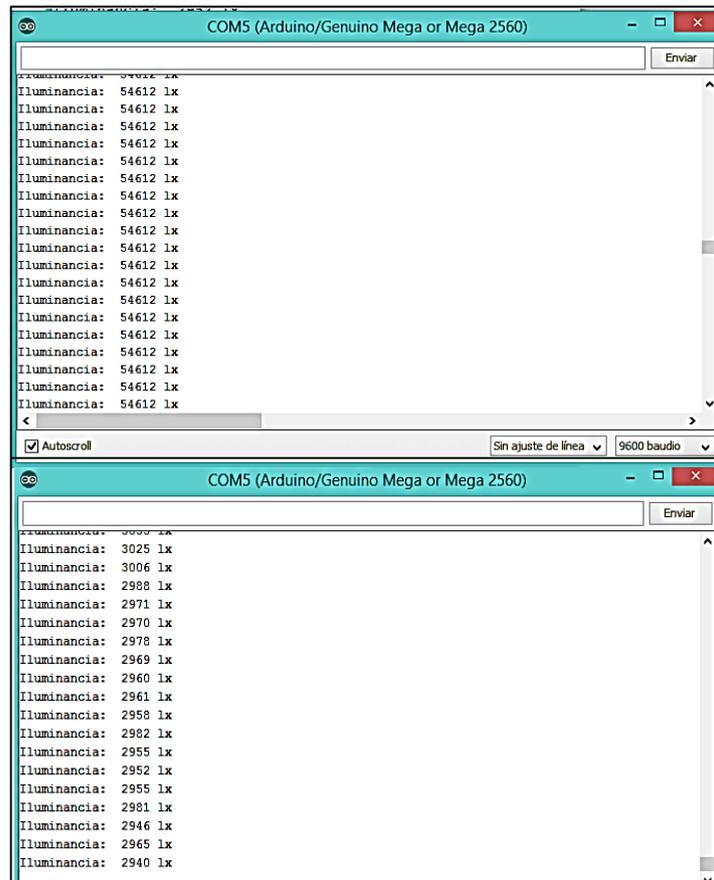


Figura N° 89: Medidas de iluminancia con el sensor BH1750

Fuente: Elaboración propia

Siendo el valor de 2900 lux, un valor adecuado para las plantas. De noche la iluminancia producida por los leds es de 30 lux aproximadamente (Véase figura N° 121).

4.5 Control de nivel del recipiente

4.5.1. Prueba previa

Para realizar la prueba se realizaron los siguientes pasos:

- Se creó el código del anexo XI.
- Se realizaron las conexiones eléctricas y electrónicas correspondientes según se observa en la Figura N° 85.
- Se realizaron las conexiones mecánicas correspondientes.
- Se conectó el controlador a la PC.

La conexión puede visualizarse en la figura N° 90 y el código en la figura N° 91.

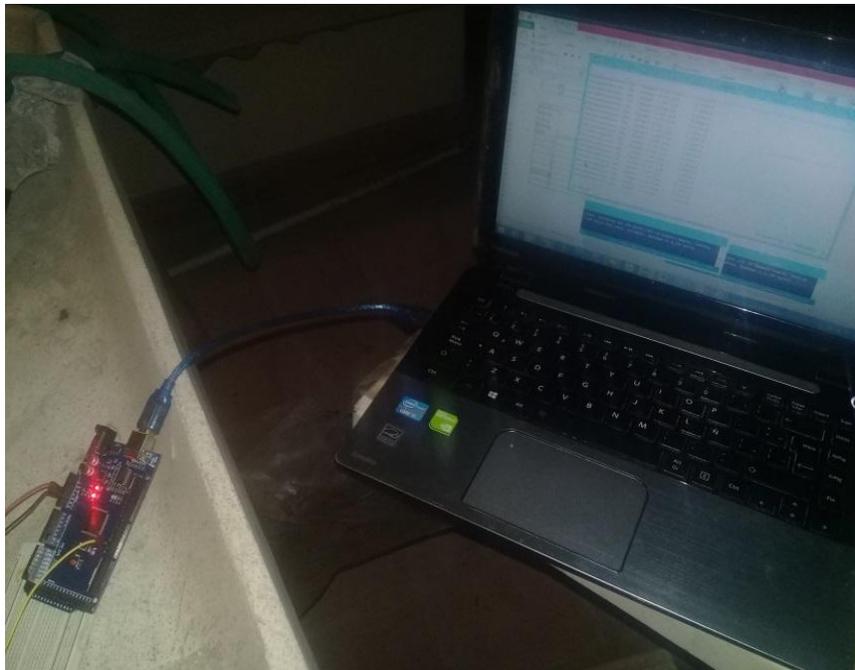


Figura N° 90: Prueba del sensor de caudal

Fuente: Elaboración propia

4.5.2. Resultados de la prueba

En la figura N° 92 pueden visualizarse los valores del caudal medidos con el sensor FS400A-G1.

```

sketch_jan04a Arduino 1.6.5
Archivo Editar Programa Herramientas Ayuda
sketch_jan04a
volatile int NumPulsos; //variable para la cantidad de pulsos recibidos
int PinSensor = 2; //Sensor conectado en el pin 2
float factor_conversion=3.5; //para convertir de frecuencia a caudal

//---Función que se ejecuta en interrupción-----
void ContarPulsos ()
{
  NumPulsos++; //incrementamos la variable de pulsos
}

//---Función para obtener frecuencia de los pulsos-----
int ObtenerFrecuencia()
{
  int frecuencia;
  NumPulsos = 0; //Ponemos a 0 el número de pulsos
  interrupts(); //Habilitamos las interrupciones
  delay(1000); //muestra de 1 segundo
  noInterrupts(); //Desabilitamos las interrupciones
  frecuencia=NumPulsos;
}
13 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM5

```

Figura N° 91: Código en Arduino del sensor FS400A G1.

Fuente: Elaboración propia

```

COM3
Enviar
FrecuenciaPulsos: 30Hz Caudal: 6.571 L/m 514.200L/h
FrecuenciaPulsos: 34Hz Caudal: 9.714 L/m 582.857L/h
FrecuenciaPulsos: 34Hz Caudal: 9.714 L/m 582.857L/h
FrecuenciaPulsos: 33Hz Caudal: 9.429 L/m 565.714L/h
FrecuenciaPulsos: 31Hz Caudal: 8.857 L/m 531.429L/h
FrecuenciaPulsos: 33Hz Caudal: 9.429 L/m 565.714L/h
FrecuenciaPulsos: 35Hz Caudal: 10.000 L/m 600.000L/h
FrecuenciaPulsos: 33Hz Caudal: 9.429 L/m 565.714L/h
FrecuenciaPulsos: 33Hz Caudal: 9.429 L/m 565.714L/h
FrecuenciaPulsos: 33Hz Caudal: 9.429 L/m 565.714L/h
FrecuenciaPulsos: 34Hz Caudal: 9.714 L/m 582.857L/h
FrecuenciaPulsos: 34Hz Caudal: 9.714 L/m 582.857L/h
FrecuenciaPulsos: 34Hz Caudal: 9.714 L/m 582.857L/h
FrecuenciaPulsos: 24Hz Caudal: 6.857 L/m 411.429L/h
FrecuenciaPulsos: 0Hz Caudal: 0.000 L/m 0.000L/h

```

Figura N° 92: Valores del puerto serial del código del sensor de caudal.

Fuente: Elaboración propia

El sensor ultrasónico tendía a dar lecturas erróneas con el código provisto, por lo que se limpió la señal de salida a través de código, según véase anexo XVI, en el que se promedian 5 valores de entrada y se ignoran las lecturas por encima de 200 cm, el cual se eligió al ser un valor que nunca llegaría a medir el sensor en el sistema.

A continuación se visualizan los valores obtenidos con el código corregido. En la figura N° 93 (arriba) se observan las lecturas con el recipiente vacío y en la figura N° 93 (abajo) las lecturas con el recipiente lleno.

Se vio que el sensor no medía los 34 centímetros del recipiente, sino máximo 13 centímetros. Debido a ello se reacomodó los valores del valor sensado y los segundos para llenar el recipiente.

Se tomaron las alturas experimentalmente. Se calcularon las diferencias como las diferencias entre el tiempo para llegar a X altura y a los 6 centímetros. Véase tabla N° 25.

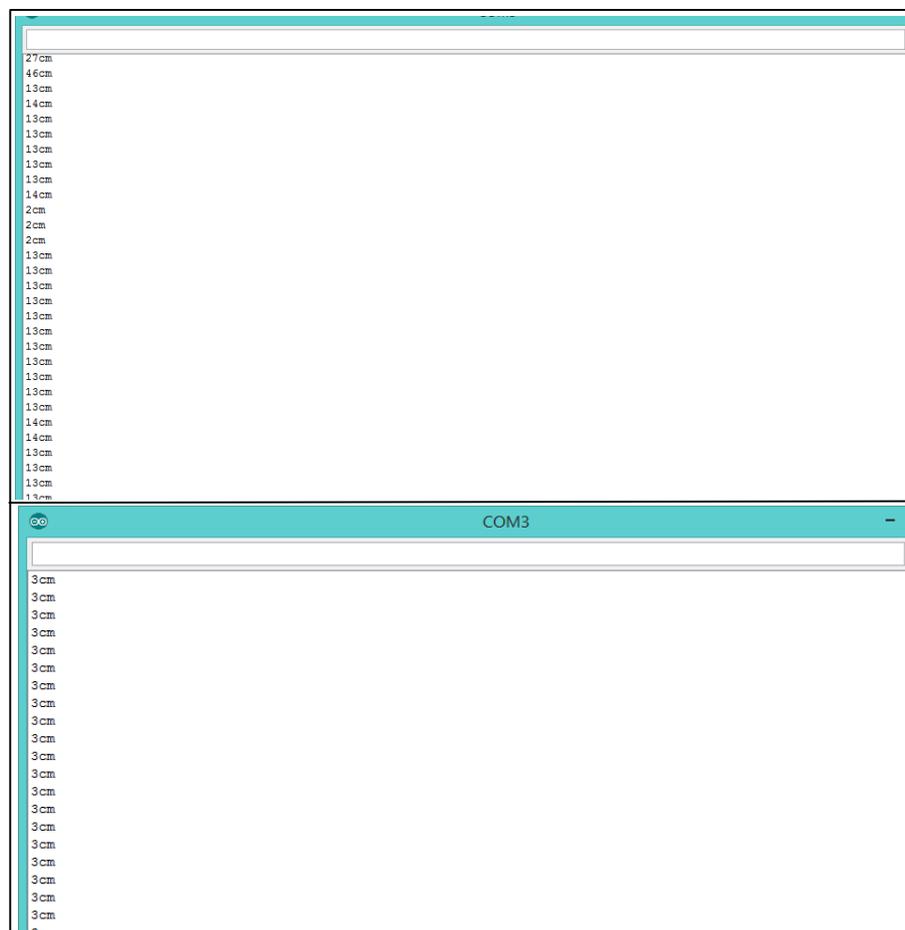


Figura N° 93: Lecturas de la distancia con el recipiente vacío y lleno.

Fuente: Elaboración propia

Tabla N° 25: Recálculo de los segundos de las electroválvulas.

DISTANCIA MEDIDA CON EL SENSOR (CM)	ALTURA REAL (MM)	ELECTROVÁLVULA 1		ELECTROVÁLVULA 2	
		Tiempo	Diferencia	Tiempo	Diferencia
13	0	0	95.1317		
12	15	1	94.1317		
11	40	14.3874	80.7443		
10	60	26.8096	68.3221		
9	80	39.2318	55.8999		
8	110	57.8651	37.2666		
7	140	76.4984	18.6333		
6	170	95.1317			
5	195	110.65945		187.5	
4	200			300	300
3	225			750	750
2	250			1425	

Fuente: Elaboración propia

Asimismo se probó el funcionamiento de las electroválvulas. La electroválvula 2 deja pasar el fluido cuando la señal está en 0 lógico, mientras que la electroválvula 1 lo contrario. Véase figura N° 94.



Figura N° 94: Pruebas con las electroválvulas

Fuente: Elaboración propia

4.6 Control del fertirriego

4.6.1. Prueba previa

Para realizar la prueba se realizaron los siguientes pasos:

- Se creó el código del anexo XVII.
- Se realizaron las conexiones electrónicas correspondientes según la figura N° 85.
- Se llenó con tierra una de las macetas.
- Se ubicó el sensor MH dentro de la tierra.
- Se conectó el controlador a la PC

La conexión del sensor de humedad se visualiza en la figura N° 95.



Figura N° 95: Prueba del sensor MH

Fuente: Elaboración propia

4.6.2. Resultados de la prueba

A partir de pruebas en diferentes medios se pudo concluir lo de tabla N° 26.

Tabla N° 26: Valores de salida del sensor MH

Valor analógico	Medio
240	Agua
230~260	Suelo muy húmedo
1023	Aire/Suelo muy seco

Fuente: Elaboración propia

4.7 Visualización de las variables en pantalla LCD

4.7.1. Prueba

Para la visualización de las principales variables en la pantalla LCD se siguieron los siguientes pasos:

- Se creó el código del anexo XIII.
- Se realizaron las conexiones eléctricas y electrónicas correspondientes según se observa en la figura N° 85.
- Se alimentó la pantalla LCD
- Se alimentaron los sensores DHT22, BH1750, HC-SR04 y MH.
- Se conectó el controlador a la PC

4.7.2. Resultado de la prueba

La pantalla mostró adecuadamente los valores medidos con los sensores, como se puede apreciar en la figura N° 96.



Figura N° 96: Prueba de la pantalla LCD 16x4.

Fuente: Elaboración propia

4.8 Programación del controlador difuso en Arduino

4.8.1. Descripción del problema

El problema a resolver consistió en crear el programa para realizar los lazos de control planteados en la sección 4.1 del presente capítulo.

4.8.2. Algoritmo

Se empleó la librería eFLL para implementar los conjuntos difusos en el controlador Arduino. El algoritmo del programa para realizar el control con dicha librería es el mostrado en la figura N° 97, considerando las partes que posee un programa en Arduino, las funciones Void Setup y Void Loop.

4.8.3. Código fuente

La librería eFLL cuenta con la limitación de poseer sólo conjuntos difusos trapezoidales. Por ello se ha tenido que modificar algunos conjuntos difusos mostrados en el Capítulo 3. Para explicar el código fuente se ha dividido la sección en Void Setup y Void Loop.

a) Void Setup

Dentro del Void Setup se realizaron los siguientes pasos:

Declarar las librerías de control fuzzy, tal como se observa en la figura N° 98 (arriba).

Crear el sistema fuzzy, tal como se observa en la figura N°98 (abajo).

Definir entradas y salidas digitales y analógicas.

Declarar los conjuntos difusos de entrada y salida, tal como se observa en la figura N° 99 (arriba).

Crear las reglas, tal como se observa en la figura N° 99(abajo).

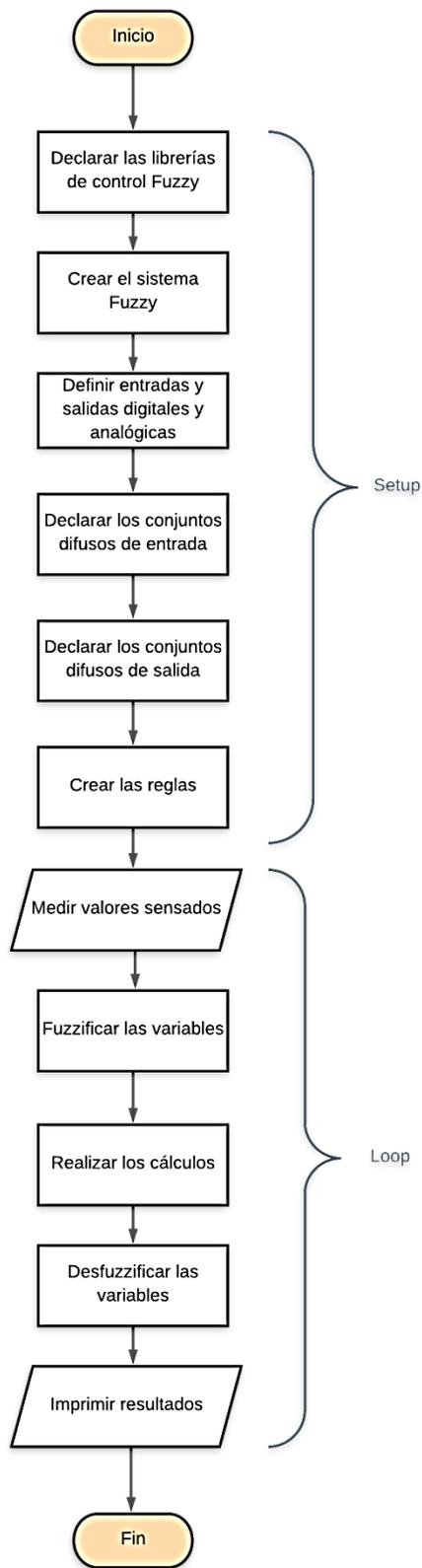


Figura N° 97: Diagrama de flujo para usar la librería eFLL.

Fuente: Elaboración propia

```

//FUZZY
#include <FuzzyRule.h>
#include <FuzzyComposition.h>
#include <Fuzzy.h>
#include <FuzzyRuleConsequent.h>
#include <FuzzyOutput.h>
#include <FuzzyInput.h>
#include <FuzzyIO.h>
#include <FuzzySet.h>
#include <FuzzyRuleAntecedent.h>

Fuzzy* fuzzy = new Fuzzy();

```

Figura N° 98: Declaración de las librerías de control fuzzy y del comando para crear un sistema difuso.

Fuente: Elaboración propia

```

// Criando o FuzzyInput distancia
FuzzyInput* distance = new FuzzyInput(1);
// Criando os FuzzySet que compoem o FuzzyInput distancia
FuzzySet* small = new FuzzySet(0, 20, 20, 40); // Distancia pequena
distance->addFuzzySet(small); // Adicionando o FuzzySet small em distance
FuzzySet* safe = new FuzzySet(30, 50, 50, 70); // Distancia segura
distance->addFuzzySet(safe); // Adicionando o FuzzySet safe em distance
FuzzySet* big = new FuzzySet(60, 80, 80, 80); // Distancia grande
distance->addFuzzySet(big); // Adicionando o FuzzySet big em distance

fuzzy->addFuzzyInput(distance); // Adicionando o FuzzyInput no objeto Fuzzy

// Criando o FuzzyOutput velocidade
FuzzyOutput* velocity = new FuzzyOutput(1);
// Criando os FuzzySet que compoem o FuzzyOutput velocidade
FuzzySet* slow = new FuzzySet(0, 10, 10, 20); // Velocidade lenta
velocity->addFuzzySet(slow); // Adicionando o FuzzySet slow em velocity
FuzzySet* average = new FuzzySet(10, 20, 30, 40); // Velocidade normal
velocity->addFuzzySet(average); // Adicionando o FuzzySet average em velocity
FuzzySet* fast = new FuzzySet(30, 40, 40, 50); // Velocidade alta
velocity->addFuzzySet(fast); // Adicionando o FuzzySet fast em velocity

fuzzy->addFuzzyOutput(velocity); // Adicionando o FuzzyOutput no objeto Fuzzy

FuzzyRuleAntecedent* ifDistanceSmall = new FuzzyRuleAntecedent(); // Instanciando um Antecedente para a expresso
ifDistanceSmall->joinSingle(small); // Adicionando o FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* thenVelocitySlow = new FuzzyRuleConsequent(); // Instancinado um Consequente para a expressao
thenVelocitySlow->addOutput(slow); // Adicionando o FuzzySet correspondente ao objeto Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule01 = new FuzzyRule(1, ifDistanceSmall, thenVelocitySlow); // Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule01); // Adicionando o FuzzyRule ao objeto Fuzzy

// FuzzyRule "SE distancia = segura ENTIAO velocidade = normal"
FuzzyRuleAntecedent* ifDistanceSafe = new FuzzyRuleAntecedent(); // Instanciando um Antecedente para a expresso
ifDistanceSafe->joinSingle(safe); // Adicionando o FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* thenVelocityAverage = new FuzzyRuleConsequent(); // Instancinado um Consequente para a expressao
thenVelocityAverage->addOutput(average); // Adicionando o FuzzySet correspondente ao objeto Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule02 = new FuzzyRule(2, ifDistanceSafe, thenVelocityAverage); // Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule02); // Adicionando o FuzzyRule ao objeto Fuzzy

// FuzzyRule "SE distancia = grande ENTIAO velocidade = alta"
FuzzyRuleAntecedent* ifDistanceBig = new FuzzyRuleAntecedent(); // Instanciando um Antecedente para a expresso
ifDistanceBig->joinSingle(big); // Adicionando o FuzzySet correspondente ao objeto Antecedente
FuzzyRuleConsequent* thenVelocityFast = new FuzzyRuleConsequent(); // Instancinado um Consequente para a expressao
thenVelocityFast->addOutput(fast); // Adicionando o FuzzySet correspondente ao objeto Consequente
// Instanciando um objeto FuzzyRule
FuzzyRule* fuzzyRule03 = new FuzzyRule(3, ifDistanceBig, thenVelocityFast); // Passando o Antecedente e o Consequente da expressao
fuzzy->addFuzzyRule(fuzzyRule03); // Adicionando o FuzzyRule ao objeto Fuzzy

```

Figura N°99: Creación de conjuntos difusos, asignación a variables de entrada o salida y creación de las reglas.

Fuente: Blog Zerokol

b) Void Loop

Dentro del Void Setup se realizaron los siguientes pasos:

Medir los valores sensados.

Fuzzificar las variables.

Realizar los cálculos.

Desfuzzificar las variables.

Imprimir los resultados.

En la figura N° 100 se muestra el código fuente que comprende estos pasos.

```
fuzzy->setInput(1, averageDistance);  
  
fuzzy->fuzzify();  
  
float output = fuzzy->defuzzify(1);  
  
Serial.print("Distancia: ");  
Serial.print(averageDistance);  
Serial.print("Velocidade: ");  
Serial.println(output);  
// wait 100 milli seconds before looping again  
delay(100);
```

Figura N° 100: Fuzzificación, desfuzzificación e impresión de datos.

Fuente: Blog Zerokol

4.9 Implementación final

La implementación final siguió el mismo orden de la mencionada en la sección 4.1 del presente capítulo.

En la sección 4.9.1 se detalla acerca del cultivo de la rosa miniatura utilizado.

4.9.1. Cultivo de la rosa miniatura

En la figura N° 101 (izquierda) se observa uno de los ejemplares de rosa miniatura empleados. En la parte derecha todos los ejemplares.



Figura N° 101: Rosas miniatura

Fuente: Elaboración propia

Fueron considerados seis ejemplares de rosa miniatura. Las macetas empleadas fueron de plástico con protección UV, de 16 cm aproximadamente de diámetro. La tierra a emplear no debió ser muy arenosa, debe ser capaz de retener regular cantidad de humedad.

4.9.2. Estructura del invernadero

a) Materiales

Los materiales utilizados se mencionan en la tabla N° 27. Para unir las tuberías con las uniones fue empleado el pegamento para tuberías PVC azul. Para crear las uniones necesarias fue empleada la macilla epóxica. Finalmente se le dio un acabado metálico con spray y se le colocó deslizantes para muebles para poder mover la estructura. En la figura N° 102 se observa esta estructura.

Tabla N° 27: Materiales empleados en la construcción del invernadero

Ítem	Unidades
Tuberías de medio metro y 1/2"	15
Codos 1/2"	18
Uniones T 1/2"	4
Pegamento para tuberías PVC azul	1
Macilla epóxica Moldimix	5
Deslizantes para muebles	4
Spray de acabado metálico	2

Fuente: Elaboración propia



Figura N° 102: Estructura del invernadero

Fuente: Elaboración propia

Para aislar el invernadero fueron empleadas micas transparentes de plástico. Se tuvo en consideración la orientación del invernadero recomendada, tal como se muestra en la figura N° 103. En la figura N° 104 se muestra el diagrama del flujo de la creación de esta estructura, la cual sigue un proceso lineal y empieza con la creación de las uniones con masilla epóxica y termina con la colocación de las micas.



Figura N° 103: Estructura del invernadero con micas

Fuente: Elaboración propia

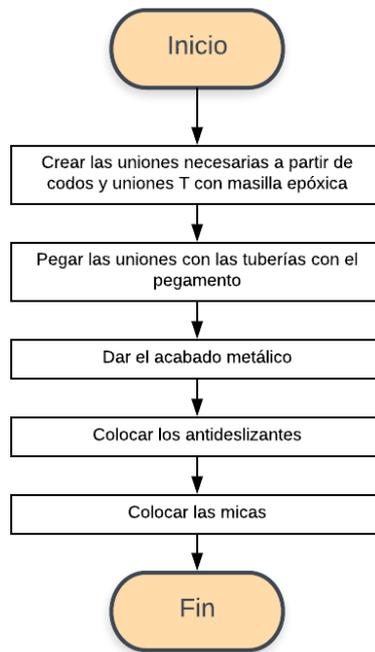


Figura N° 104: Diagrama de flujo de la creación de la estructura

Fuente: Elaboración propia

4.9.3. Circuito electrónico

El circuito considera, como se había anticipado en la figura N° 27:

- a) Sensores
 - Sensor BH1750
 - Sensor DHT22
 - Sensor HC-SR04
- b) Actuadores
 - Electroválvula 1
 - Electroválvula 2
 - Bomba para fertirriego
 - El ventilador SYM BANG
 - Módulo de 4 relés.
- c) Elementos para visualización
 - Pantalla LCD 16x4
 - Potenciómetro de 10 K Ω

Se utilizó todas las salidas del módulo de relés.

En la figura N° 105 se observa el diagrama del circuito en Professional Proteus.

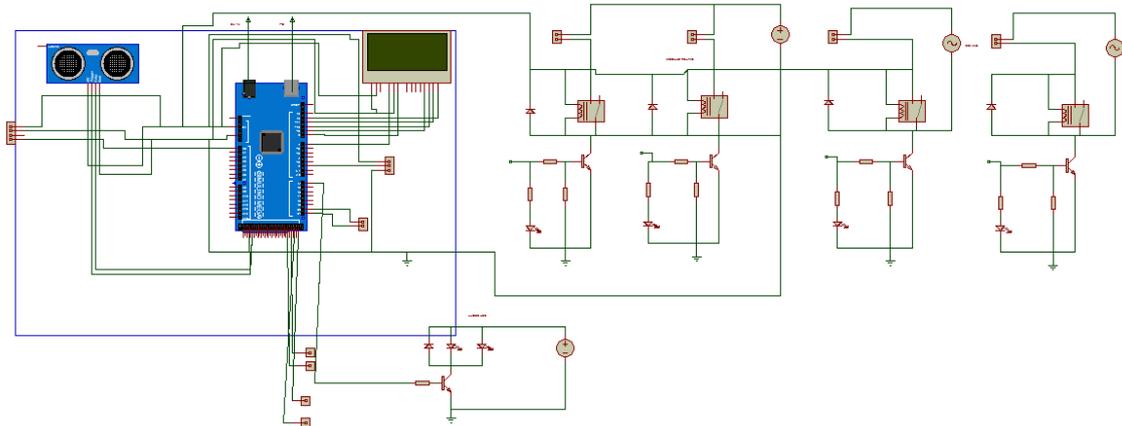


Figura N° 105: Diagrama de las conexiones de la implementación final

Fuente: Elaboración propia

4.9.4. Consideraciones sobre el circuito electrónico

- Fue necesario proteger los cables con un tubo corrugado de 1/2" para protegerlos del agua.
- Fue necesario proteger los sensores con micas para proteger de condiciones ambientales adversas.

4.9.5. Implementación del control de la humedad relativa y temperatura

No fue necesario modificar los conjuntos difusos. Por lo que simplemente se realizaron las conexiones respectivas según la figura N° 106 y se escribió su parte del código del anexo XVIII.

El sensor DHT22 fue protegido para evitar que sea muy afectado por la condensación en el ambiente. Los cables son estañados porque permiten una mejor transmisión de la señal, a comparación de los cables prefabricados para Protoboard.



Figura N° 106: Conexión de la implementación del sensor DHT22

Fuente: Elaboración propia

4.9.6. Implementación del control de nivel del recipiente de irrigación

Para el control de nivel fue necesario modificar los conjuntos difusos. La implementación contempla los componentes mencionados en el apartado 3.5 y que se visualizan en la figura N° 107.



Figura N° 107: Conexiones de la implementación del control de nivel

Fuente: Elaboración propia

1. Llenado del recipiente

Como se mencionó anteriormente, fue necesario rediseñar el controlador difuso de las electroválvulas por dos razones. Siendo la primera, la limitación de la librería difusa para realizar la desfuzzificación del centro de área y la segunda, que el caudal del suministro debía ser mayor.

Para la entrada, se formaron conjuntos difusos triangulares por cada centímetro de altura del recipiente, uniformemente separados.

Para la salida, se formaron conjuntos difusos triangulares estrechos por cada tiempo en segundos, para que de esta manera el sistema difuso pueda integrar el área bajo la función triangular. En la figura N° 108 se aprecian los conjuntos difusos.

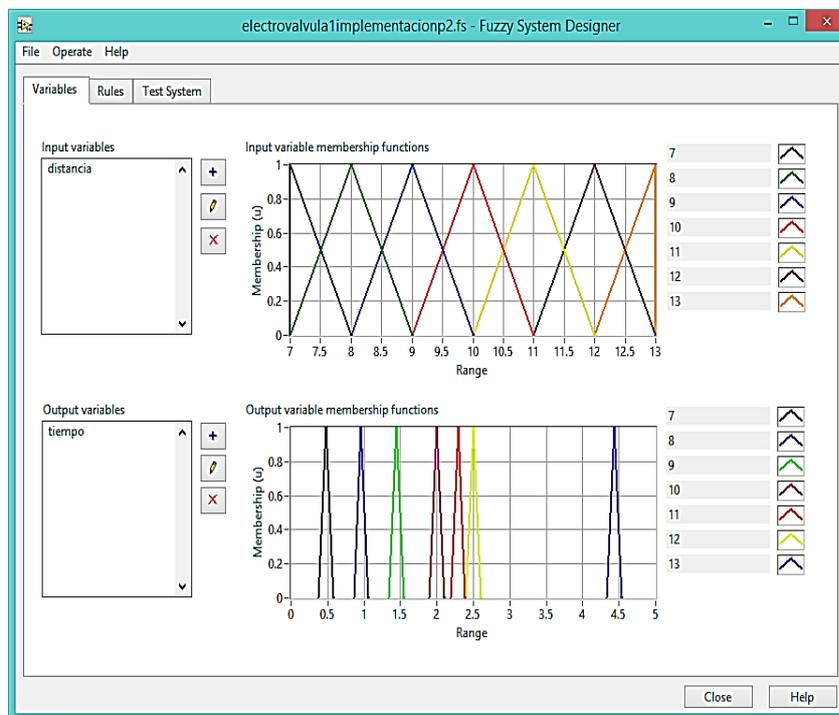


Figura N° 108: Redefinición de los conjuntos difusos de la electroválvula 1

Fuente: Elaboración propia

En la figura N° 109 se observan las nuevas reglas que relacionan estos conjuntos difusos, siguiendo la misma lógica que se menciona en la sección 4.5.

En la figura N° 110 se observa el testeo del nuevo controlador creado. Para cuando la distancia al SP es 0, el tiempo de activación será 0.

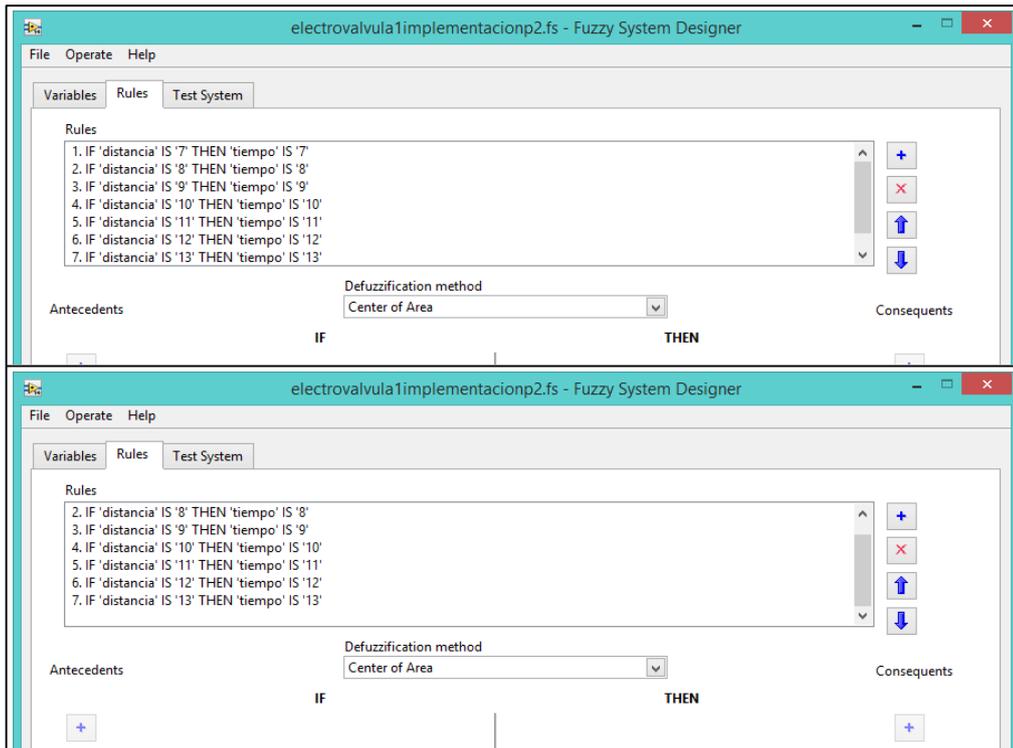


Figura N° 109: Redefinición de las reglas de la electroválvula 1

Fuente: Elaboración propia

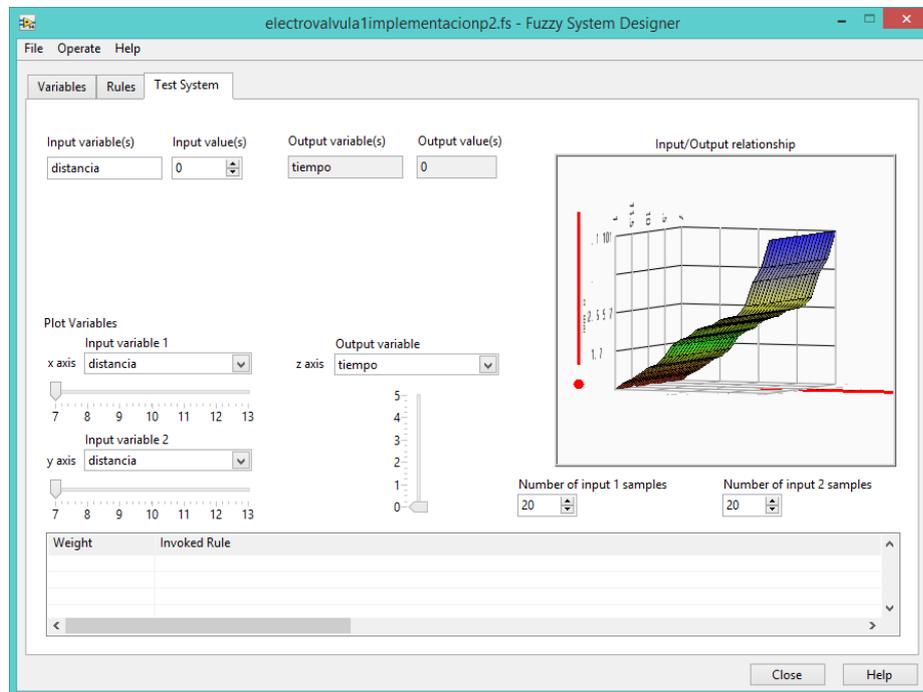


Figura N° 110: Testeo del controlador difuso de la electroválvula 1

Fuente: Elaboración propia

2. Vaciado del recipiente

En el caso de la electroválvula 2 (vaciado) también fue necesario modificar las salidas y entradas por la limitación de la librería eFLL, para obtener las respuestas deseadas. En la figura N° 111 (arriba) se aprecian los nuevos conjuntos difusos.

En la figura N° 112 se observan las reglas creadas y en la figura N° 111 (abajo) se observa el testeo del controlador difuso.

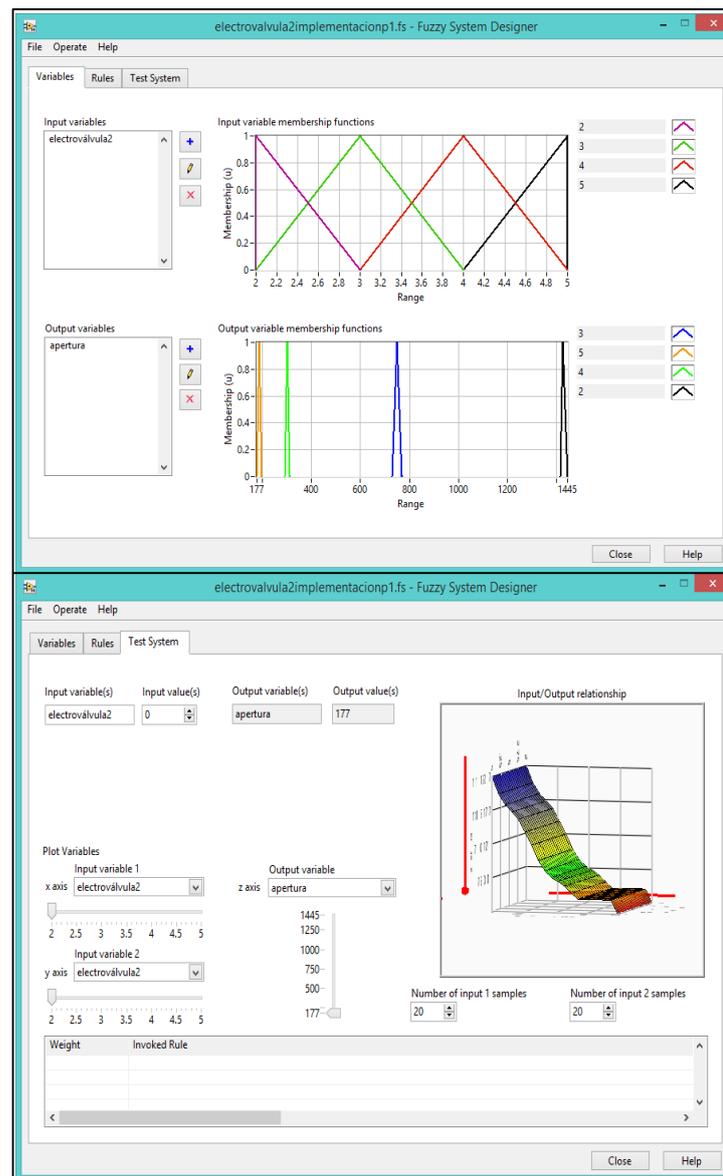


Figura N° 111: Redefinición y testeo de los conjuntos difusos de la electroválvula 2

Fuente: Elaboración propia

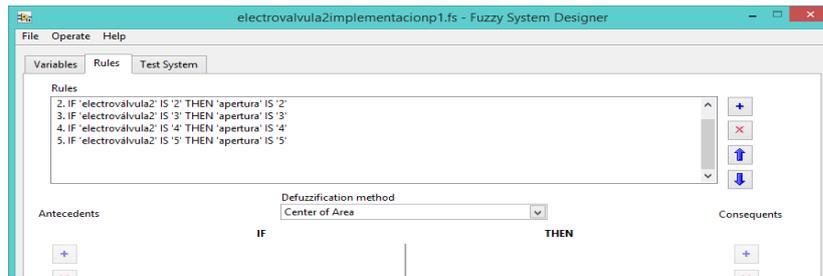


Figura N° 112: Redefinición de las reglas de la electroválvula 2

Fuente: Elaboración propia

Luego se accionó el lazo de control. En la figura N° 113 se aprecia la bomba en funcionamiento, habiendo conectado la manguera flexible al inyector Venturi.



Figura N° 113: Bomba en funcionamiento

Fuente: Elaboración propia

Para accionar la bomba se empleó como fuente de alimentación la línea de 220 VAC de un interruptor, la cual comparte con el ventilador. Véase figura N° 114.



Figura N° 114: Conexiones eléctricas de las fuentes de alimentación
Fuente: Elaboración propia

En la figura N° 115 se observa las conexiones del módulo de relés.

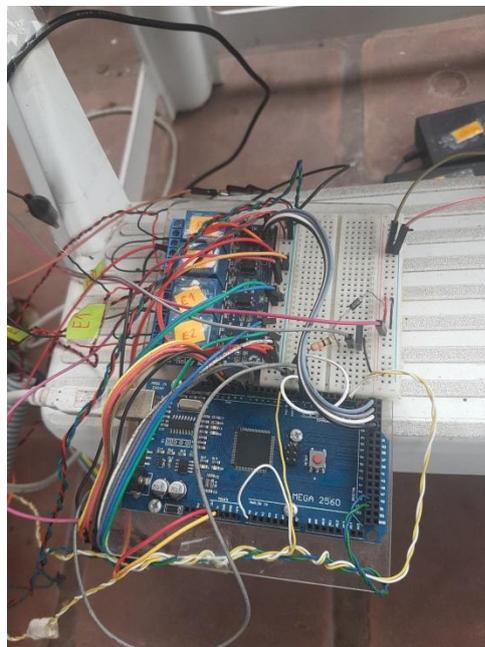


Figura N° 115: Conexiones eléctricas del módulo de relés
Fuente: Elaboración propia

Asimismo el código incluido en el anexo XVIII incluye una vista en el puerto serial de la lectura del sensor ultrasónico. Una muestra de cómo está conectada la PC se aprecia en la figura N° 116.

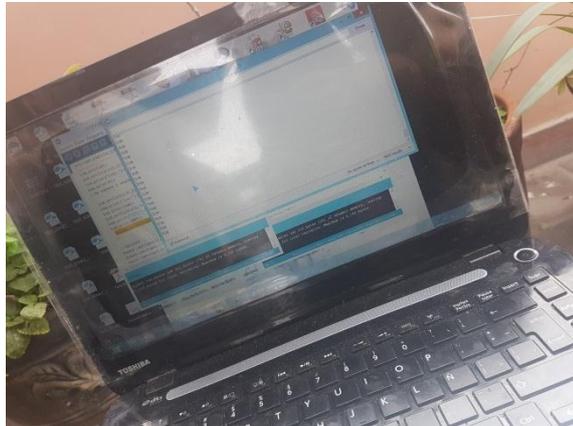


Figura N° 116: Lectura del sensor ultrasónico en el puerto serial

Fuente: Elaboración propia

En la figura N° 117 se ve cómo la bomba succiona el agua (la manguera está retirada) del recipiente del fertilizante y el inyector Venturi está en funcionamiento.



Figura N° 117: Inyector Venturi y bomba en funcionamiento

Fuente: Elaboración propia

4.9.7. Implementación del control de luminosidad

El sensor BH1750 fue colocado en medio del mini invernadero. Fue colocado dentro de una mica para protegerlo del polvo y la temperatura del medio ambiente. Véase figura N° 118.



Figura N° 118: Sensor BH1750 colocado

Fuente: Elaboración propia

Fueron colocados módulos LED rojos y azules, los cuales se alimentan con una fuente de 12 VDC, según véase anexo VI. El funcionamiento de los leds puede apreciarse en la figura N° 119.



Figura N° 119: Funcionamiento de los leds

Fuente: Elaboración propia

La salida en este caso se dio por un pin digital del controlador Arduino, a través de la función `analogWrite`. Véase figura N° 120.

```
//LUCES  
analogWrite(luces, outputluxes);
```

Figura N° 120: Función `analogWrite` para el control de la luminosidad

Fuente: Elaboración propia

En esta oportunidad la cantidad de luminosidad dada por los leds fue de 20 lux, como se mostró cuando se probó la pantalla. Véase figura N° 121.

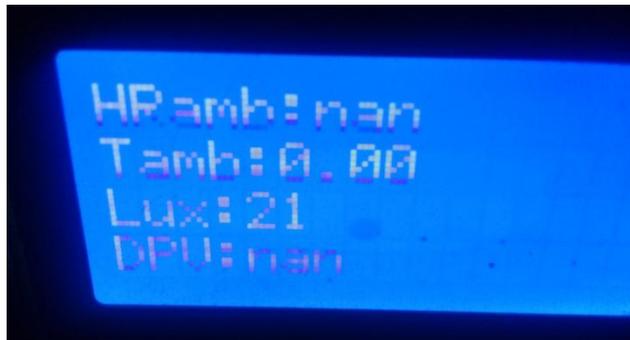


Figura N° 121: Salida del sensor BH1750 empleando los leds

Fuente: Elaboración propia

Por lo que se realizó una variación en los puntos de los conjuntos difusos de la entrada, en el código final. Véase anexo XVIII.

4.9.8. Implementación del control del fertirriego

Como la librería `eFLL` no cuenta con conjuntos difusos de forma gaussiana entonces fue necesaria reajustarlos para que la función de salida sea de similar salida.

Se crearon dos conjuntos difusos triangulares para la entrada y para la salida, de manera que la salida fuera lo más similar posible a la obtenida en el apartado 3.7.5. Véase figuras N° 122 y N° 123.

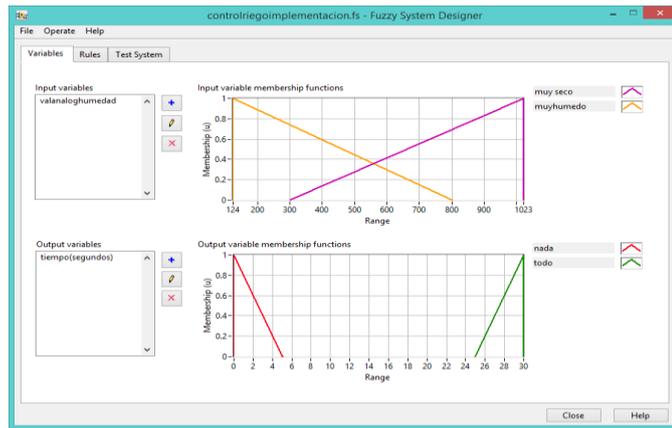


Figura N° 122: Redefinición de los conjuntos difusos del control del fertirriego

Fuente: Elaboración propia

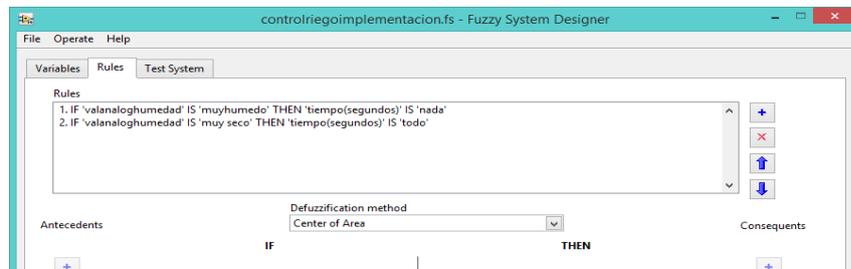


Figura N° 123: Redefinición de las reglas del control del fertirriego

Fuente: Elaboración propia

En la figura N° 124 se observa una prueba con estos conjuntos y reglas redefinidos. El gráfico de salida es similar al del apartado 3.7.5.

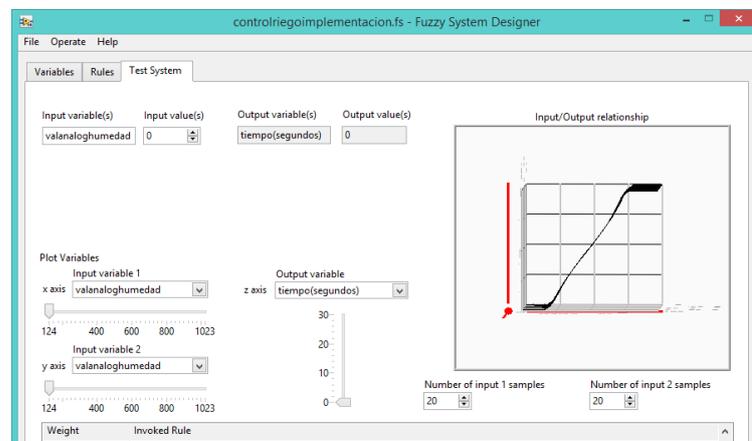


Figura N° 124: Testeo del controlador difuso del fertirriego

Fuente: Elaboración propia

Se ubicó el sensor FC-28 en una de las macetas, según se ve en la figura N° 125.



Figura N° 125: Sensor MH colocado en la tierra

Fuente: Elaboración propia

-Cálculo del volumen de agua en cada maceta y el caudal de los goteros.

El volumen de agua en cada maceta fue el mostrado en la tabla N° 28.

Tabla N° 28: Cálculo del caudal aproximado de los goteros

Volumen de agua (mL)	Tiempo (segundos)	Caudal (mL/s)
120	75	1.6
48	30	1.6

Fuente: Elaboración propia

Teniendo el suministro de agua un caudal aproximado de 562 Lt/hr. La segunda fila corresponde a la prueba realizada con uno de los goteros en un recipiente graduado, como se ve en la figura N° 126.



Figura N° 126: Prueba de los goteros

Fuente: Elaboración propia

Se pueden ver en la figura N° 127 los goteros cuando la bomba está en funcionamiento. Hay que tener en cuenta el desborde de agua producido cuando hay un exceso de tiempo de regado. En la figura N° 128, se ve un solo gotero.



Figura N° 127: Funcionamiento de los goteros

Fuente: Elaboración propia



Figura N° 128: Funcionamiento de un gotero

Fuente: Elaboración propia

4.9.9. Implementación de la pantalla

Las conexiones de la pantalla fueron sencillas, según las conexiones mostradas en la figura N° 129. Se adicionó un potenciómetro para modificar el contraste de las letras.

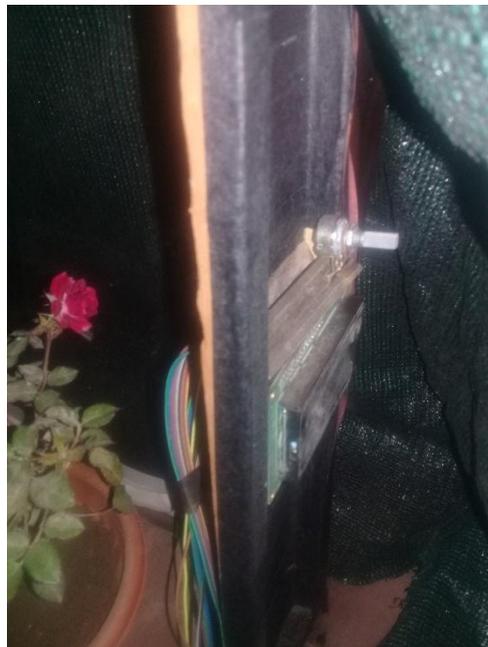


Figura N° 129: Conexiones de la pantalla

Fuente: Elaboración propia

Las variables a visualizar en la pantalla fueron:

- La humedad relativa en el aire dentro del invernadero, expresado en %.
- La temperatura en el aire dentro del invernadero, expresado en °C.
- La luminosidad, expresada en lux.
- El DPV del invernadero, expresado en Pascales.

En las figura N° 130 se aprecia la prueba de la visualización de las variables y en la figura N° 131 la pantalla una vez conectados los sensores.



Figura N° 130: Prueba de las conexiones de la pantalla

Fuente: Elaboración propia



Figura N° 131: Conexiones de la pantalla empleando los sensores

Fuente: Elaboración propia

Hay que tener en consideración el clima externo para el cuidado de las rosas. Cuando el clima externo es muy húmedo, es altamente probable que se desarrolle oídio,

enfermedad que provoca que las hojas se tornen amarillas y se sequen, como se puede apreciar en la figura N° 132.

Por otro lado, no todas las plantaciones de rosa reaccionaron de igual manera. Hubo algunas que resistieron de mejor manera el clima, en tanto que otras se mostraron débiles.



Figura N° 132: Plantaciones de rosa con oídio

Fuente: Elaboración propia

4.10 Resultados finales

En la tabla N° 29, se resumen los resultados de este proyecto. En la sección de Costos se encuentran detallados el consumo de energía y los precios de los sensores, actuadores y otros elementos necesarios.

Tabla N°29: Resumen de resultados finales

Variable	Indicador	Resultado
Controlador difuso utilizando Arduino de bajo costo	Consumo de energía	583.58 Watts/hora en un día
	Funcionamiento de los lazos de control	Fue posible implementar un control difuso para la humedad relativa y temperatura mediante el cálculo de DPV Fue posible implementar un controlador difuso para la luminosidad teniendo en cuenta el umbral ideal para el cultivo de la rosa Fue posible implementar un controlador difuso para el control de nivel del recipiente teniendo en cuenta el cálculo de los tiempos de llenado y vaciado. Fue posible implementar un controlador difuso para el fertirriego teniendo en cuenta la programación de riego y el tiempo necesario para regar las macetas.
	Precio total del controlador	S/ 465.7

Fuente: Elaboración propia

4.11 Costos

4.11.1. Costo equipamiento

Tabla N° 30: Lista de materiales empleados en cada lazo de control y precios

Item	Unidades	Precio unitario	Subtotal (S/)
1. Iluminación			
1.1 Cintas led rojo y azul	4 m	7	28
1.2. Batería 12 V, 5A	1	15	15
1.3. LDR	2	0.5	1
1.4 Resistencias de 330,410 Ω	4	0.1	0.4
1.5 Led azul	2	0.3	0.6
1.6 Galleta 72 x 47 mm	2	1	2
1.7 Clips medianos negros y rojos	12	0.5	6
1.8 Módulo arduino relays 4 canales	1	13.5	13.5
1.9 Alambre estañado	10 m	0.5	5
1.10 Acrílico	---	---	40
1.11 Transistores BD 137	2	0.5	1
1.12 Módulos	16	1	16
Subtotal			128.5
2. Fertirriego /Nivel			
2.1 Sensor de flujo efecto Hall	1	33.8	33.8
2.2 Módulo arduino relays 4 canales	1	13.5	13.5
2.3 Válvula solenoide 1	1	30	30
2.4 Válvula solenoide 2	1	21	21
2.5 Inyector venturi	1	40	40
2.6 Fertilizante	1	16.9	16.9
2.7 Protoboard	1	5	5
2.8 Manguera lisa transparente	2m	2.5	5
2.9 Recipiente 35 lt	1	16.9	5
2.10 Bomba de agua con filtro	1	25	25
2.11 Moldimix	5	5	25
SubTotal			220.2
3. Humedad relativa y temperatura			
3.1 Ventilador SP06-08	1	29.9	29,9
3.2 Ventilador SYM BANG	1	50	50
3.3 DHT22	2	30	60
3.4 FC-28/MH	1	7	7
SubTotal			117
Total			465.7

Fuente: Elaboración propia

Tabla N° 31: Lista de materiales empleados en la estructura del invernadero y precios

Item	Unidades	Precio unitario	Subtotal (S/)
Pegamento 200 gr	5	1.20	6.00
Spray metálico	2	8.00	16.00
Trupan	1	34.00	34.00
Tuberías PVC de 1/2"	13	3.80	49.40
Codos PVC de 1/2"	15	1.00	15.00
Uniones T (Tee) PVC de 1/2"	10	1.40	14.00
Pegamento PVC	1	13.00	13.00
Total			147.40

Fuente: Elaboración propia

Tabla N° 32: Lista de materiales empleados en el método de cultivo y precios

Item	Unidades	Precio unitario	Subtotal (S/)
Plantación de rosa miniatura roja	10	5	50
Macetas	6	6	36
Total			86

Fuente: Elaboración propia

4.11.2. Costo diseño

Tabla N° 33: Costo de diseño de ingeniería por horas

Item	Unidades	Precio unitario	Subtotal (S/)
Diseño de ingeniería	24h	12.50	300.00
Total			300.00

Fuente: Elaboración propia

4.11.3. Costo energético

Tabla N° 34: Costo energético del sistema

Consumo energético del sistema por día (24 horas)				
N°	Item	Potencia (Watts)	Tiempo encendido (horas por día)	Consumo (Watts hora por día)
1	Electroválvula 1	3.84	0.03	0.10
2	Electroválvula 2	3.00	0.40	1.19
3	Bomba	6.00	0.01	0.05
4	Luces (16 módulos)	11.52	12.00	138.24
5	Ventilador	19.00	12.00	228.00
6	Arduino Mega 2560	9.00	24.00	216.00
Total				583.58

Fuente: Elaboración propia

CONCLUSIONES

La tesis tuvo como objetivo diseñar e implementar invernadero automatizado de bajo costo para la producción de rosas, de lo cual se extrae lo siguiente:

1. Fue posible diseñar el control de las principales variables del invernadero, estimado mediante cálculos: el tiempo necesario para el llenado y descarga de agua del recipiente en función al caudal de suministro, el volumen extraído por la bomba y los parámetros establecidos en el controlador de riego, los cuales se ajustaron al tipo de cultivo y cuya operatividad pudo comprobarse en las etapas de simulación e implementación del proyecto.
2. Mediante la simulación virtual con LabView de los procesos diseñados, pudo comprobarse la respuesta del controlador difuso frente a la interacción con las variables temperatura y humedad relativa del aire; asimismo, se comprobó el efecto esperado de este método de control sobre los actuadores en las etapas de control de nivel del recipiente de irrigación, control de luminosidad y control de fertirriego, los cuales fueron demostrados en los conjuntos difusos de control y testeo.
3. Fue posible implementar el sistema de control difuso mediante una unidad de hardware con arduino, capaz de monitorear las variables propuestas en la tesis. Esto fue posible implementar a un bajo costo: el controlador (S/. 689.71) y la estructura (S/. 147.40), creando una ventaja para los que están interesados en el empleo de controladores difusos en la industria de la agricultura

OBSERVACIONES

1. Es importante considerar el tipo de tierra para realizar la plantación del cultivo de la rosa miniatura.
2. Es importante considerar las condiciones climáticas en las que será implementado el invernadero. Con climas especialmente lluviosos hay que tomar precauciones para proteger los sensores.
3. Es importante considerar las plagas que afectan al cultivo de la rosa en ambientes con mucha humedad relativa en el aire.
4. Es importante considerar el tipo de cableado a emplear, así como su protección ante posibles fugas de agua o lluvias.
5. El sensor HC-SR04 es sensible a las perturbaciones y a las condiciones de luminosidad en el ambiente.
6. En contraste con el control ON-OFF, el control de nivel fue más laborioso empleando la lógica difusa.

RECOMENDACIONES

1. Usar otro sensor ultrasónico o de tecnología similar para evitar el problema del rebote.
2. Usar un módulo inalámbrico para evitar el problema del mojado del cableado.
3. Utilizar el módulo RTC para una mejor autonomía del arduino y menor consumo de energía.
4. Utilizar leds más potentes para aumentar la cantidad de iluminancia que llegue a las plantas.
5. Emplear un humidificador para realizar un mejor control de la humedad relativa y la temperatura.

REFERENCIAS BIBLIOGRÁFICAS

- Acosta, G. (s.f) *Taller de Controlador Basado en Lógica Difusa*. Argentina, Facultad de Ingeniería-UNCPBA.
- AgriFarming (2017). *Red Rose Variety in Greenhouse*. [Figura] Recuperado de: <http://www.agrifarming.in/rose-cultivation/>
- Agrologica ® (s.f) *Electroválvula RPE para baja presión*. España: tienda.agrologica.es
Recuperado de:
<http://tienda.agrologica.es/sistemas-riego-agricola/820-electrovalvula-baja-presion.html>
- Alvarado-Moya, P. y González, A. y Villasenor-Gonzalez, L. (2008). *Propuesta de aplicación de redes de sensores en el modelado de cultivos protegidos y en campo*. p.2
- Amangandi , J. (2012). *Arduino*. Perú: jamangandi2012.blogspot.pe. Recuperado de: <http://jamangandi2012.blogspot.pe/2012/10/que-es-arduino-te-lo-mostramos-en-un.html>
- Andrade. O., Castillo, R. y Laurentin, H. (s.f.) *Utiliza el Deficit de Presion del Vapor en invernadero*. Venezuela: universidadagricola.com Recuperado de <http://universidadagricola.com/utiliza-el-deficit-de-presion-del-vapor-en-invernadero/>
- Arévalo Hernández, Y.R. (2011). *Evaluación del efecto de la aplicación de diferentes láminas de riego en el cultivo de rosa (Rosa sp) cv. Freedom, bajo invernadero en la sabana de Bogotá*. (Tesis de maestría). Universidad Nacional de Colombia, Bogotá D.C.p.27
- Arévalo Hernández, Y.R. (2011). Principales plagas que atacan a la rosa [Tabla]. *Evaluación del efecto de la aplicación de diferentes láminas de riego en el cultivo de rosa (Rosa sp) cv. Freedom, bajo invernadero en la sabana de Bogotá*. (Tesis de maestría). Universidad Nacional de Colombia, Bogotá D.C.p.27

- Arévalo Hernández, Y.R. (2011). Principales enfermedades del cultivo de la Rosa [Tabla]. *Evaluación del efecto de la aplicación de diferentes láminas de riego en el cultivo de rosa (Rosa sp) cv. Freedom, bajo invernadero en la sabana de Bogotá*. (Tesis de maestría). Universidad Nacional de Colombia, Bogotá D.C.p.27
- Buczacki, S. (1996). *Rosales*, Madrid, España: Hermann Blume ediciones
- Bijlo, R. (2014). Relación entre temperatura y humedad relativa. España: Meteobenidorm.com. Recuperado de:
<http://meteobenidorm.blogspot.pe/2014/03/relacion-entre-temperatura-yhumedad.html>
- Blog Zerokol (28 de setiembre de 2012). eFLL - A Fuzzy Library for Arduino and Embeded Systems [Mensaje en un blog]. Blog Zerokol. Recuperado de <https://blog.zerokol.com/2012/09/arduinofuzzy-fuzzy-library-for-arduino.html>
- Castro-Montoya, A.J. Vera-Monterrosas F. y Quintana-Silva, J.A (2004). *Control Difuso de Flujo de fluidos en una Estación de Laboratorio*. Información tecnológica. *Volumen 15. pp. 46*
- Corcuera, P. (s.f). *Introducción a LabVIEW* [Presentación de power point]. Departamento de Matemática Aplicada y Ciencias de la Computación, Universidad de Cantabria, España. Recuperado de:
http://personales.unican.es/corcuerp/LabView/Slides/Intro_LabView.pdf
- Desai, P., (2015). *Python Programming for Arduino*. Birmingham, Reino Unido: Packt Publishing, Ltd. pp.34, 117.
- DIY START (2018). *Sistema De Riego Por Goteo Rosas*. Estados unidos: mustknowhow.com. Recuperado de:
<https://es.mustknowhow.com/tag/sistema-de-riego-por-goteo-rosas>
- EasyVeep (2017). *Greenhouse*. Recuperado de:
http://easyveep.com/modules.php?akt_modul=12&akt_Lang=2

- El Comercio (23 de setiembre de 2014). El Perú tiene gran potencial para ser un exportador de flores. El Comercio. Recuperado de:
<https://elcomercio.pe/economia/peru/peru-gran-potencial-exportador-flores-177520>
- El Mercurio Campo. (2015). *Las ventajas de la iluminación LED en invernaderos*. Chile: elmercurio.com. Recuperado de:
<http://www.elmercurio.com/Campo/Noticias/Noticias/2013/10/04/Las-ventajas-de-la-iluminacion-LED-en-los-invernaderos.aspx>
- ElecFreaks® (s.f). *Ultrasonic Ranging Module HC - SR04*. Recuperado de
<http://www.micropik.com/PDF/HCSR04.pdf>
- Esmeral Vargas, Y. R. (2011). *Análisis de la evapotranspiración real en el cultivo de la rosa*. (Tesis de maestría). Universidad Nacional de Colombia, Bogotá.
- Espinosa Robles, P. y Espinosa Mendoza, L. (s.f.). *Invernaderos rústicos*. México: Secretaría de agricultura, ganadería, desarrollo rural pesca y alimentación. Recuperado de <http://www.sagarpa.gob.mx/desarrolloRural/Documents/fichasaapt/Invernadero%20R%C3%BAstico.pdf>
- Falkvidd, M. (2016). *Light Level Sensor - BH1750* [Figura] Recuperado de:
<https://www.mysensors.org/build/light-bh1750>
- floresyplantas.net (25 de febrero de 2014). Cuidados del rosal mini [Mensaje en un blog]. flores y plantas.net Recuperado de:
<https://www.floresyplantas.net/cuidados-del-rosal-mini/>
- Food and Agriculture Organization of the United Stations (2017). *Capítulo 4: control del medio ambiente*. Recuperado de:
<http://www.fao.org/docrep/005/S8630S/s8630s06.htm>
- fritzing® (2017) *HC-SR04 Arduino*. Recuperado de: <http://fritzing.org/projects/hc-sr04-arduino>
- García González, A. (2014). Módulo HL-69: Un sensor de humedad de suelo. Panamá: Panamahitek.com. Recuperado de
<http://panamahitek.com/modulo-hl-69-un-sensor-de-humedad-de-suelo/>

- García González, A. (2014). *DHT22: Sensor de humedad/temperatura de precisión para Arduino*. Panamá: Panamahitek.com. Recuperado de:
<http://panamahitek.com/acerca-de-panamahitek/>
- García Saldaña, A., Landeros Sánchez, C., Carrillo Ávila, E., Castañeda Chávez, M., Pérez Vázquez, A. y Martínez Dávila, J. (2016) *Hacia un manejo más eficiente de fertilizantes en sistemas de riego de baja presión. Instituciones de Educación Superior. La labor investigadora e innovadora en México*. Cheyenne, Estados Unidos de América: 2016 Science Associated Editors, L. L. C
- Guangzhou ZIC International Co © (2014). *12V VEHICLE FAN-SP06-08*. [Figura] .
Recuperado de
<http://lonen-ledlighting.com/Index.php/product/detail /token/SP06-08.html>
- HAS Akvaryum © (2012). Venus Aqua VS-220F İç Filtre. [Figura]. Recuperado de
<http://www.hasakvaryum.com/detay.asp?id=6580>
- Iluminación (s.f.). [Tabla] Recuperado de
<http://apuntescientificos.org/tablas-iluminacion.html>
- Infojardín® (2007). *Ficha de rosal miniatura o rosal mini*. España: infojardin.com .
Recuperado de
<http://archivo.infojardin.com/tema/ficha-de-rosal-miniatura-o-rosal-mini.56219/>
- Infoagro Systems, S.L (2017). *Invernaderos de raspa y amagado*. España: infoAgro.com. Recuperado de:
http://www.infoagro.com/industria_auxiliar/invernaderos_raspa_amagado.htm
- Infoagro Systems(s.f.). El Cultivo de las rosas para corte (1era parte). España: InfoAgro.com. Recuperado de:
<http://www.infoagro.com/flores/flores/rosas.htm>
- Insumos de Riego® (2018). *Gotero regulable para riego 0-70 l/h*. Argentina:
<http://www.insumosderiego.com.ar>. Recuperado de:

http://www.insumosderiego.com.ar/riego_por_goteo_goteros_insertables_regulables.php?titulo=gotero-regulable-para-riego-0-70-lh-gotero-regulable-para-riego-70&ELID=1583

Jiménez Garrido, D. A. (2011). *La importancia de la automatización en invernaderos para el cultivo de jitomate en el estado de Morelos* (Tesis de pregrado). Universidad Nacional Autónoma de México, México.

Kühnel, C., (2015). *Arduino for the Cloud. Arduino Yún and Dragino Yún Shield*, Florida, Estados Unidos: Universal Publishers. pp.43

Kurokawa, S. y Minussi, C.R. (1999). *Control Automático de un Sistema Eléctrico Mediante Lógica Difusa*. Información tecnológica. Volumen 10, pp. 42-43

Lopez Vargas, L.A. (2008). *Diseño de un sistema de control de temperatura On/off para aplicaciones en invernadero utilizando Energía solar y gas natural* (Tesis de pregrado). Pontificia Universidad Católica del Perú, Lima.

Miranda, O. (2015). El negocio de las rosas. Lima, Perú: Diario La República.
Recuperado de:
<http://larepublica.pe/archivo/875775-el-negocio-de-las-rosas>.

Mundoriego®. *Tipos de goteros para sistemas de riego por goteo*. España: mundoriego.es. Recuperado de:
https://www.mundoriego.es/blog-sobre-riego-agricola/23_Gotero-sistema-de-riego-por-goteo.html

Murray, F. W (1967) On the computation of saturation vapor pressure. *Journal of applied Meteorology and Climatology*. Volumen 6. pp. 203-204.

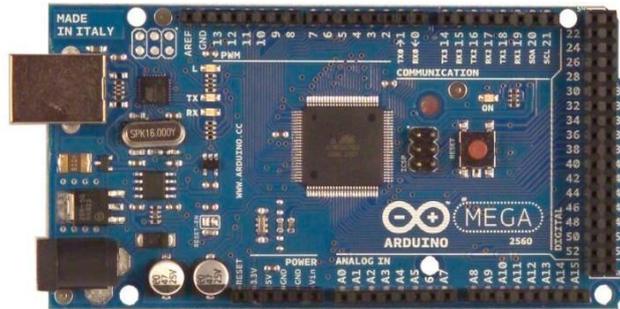
Naylamp Mechatronics® (2018a). *Módulo Sensor de Luz digital BH1750*. Perú: naylampmechatronics.com. Recuperado de:
<https://naylampmechatronics.com/sensores-luz-y-sonido/76-modulo-sensor-de-luz-digital-bh1750.html>

- Naylamp Mechatronics® (2018b). *Sensor Ultrasonido HC-SR04*. Perú: naylampmechatronics.com. Recuperado de:
<http://www.naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- Naylamp Mechatronics® (2018b). *Sensor Ultrasonido HC-SR04*. [Figura] Recuperado de:
<http://www.naylampmechatronics.com/sensores-proximidad/10-sensor-ultrasonido-hc-sr04.html>
- Rojas, N. (2017). *Las floricultura sigue siendo la “cenicienta” de la agroexportación*. Perú: agronegociosperu.org. Recuperado de
<https://agronegociosperu.org/2017/07/04/las-floricultura-sigue-siendo-la-cenicienta-de-la-agroexportacion/>
- Pallango Fernández, O. E. (2002). *Monitoreo de radiación solar y control de iluminación dentro de un invernadero con la utilización de instrumentación virtual*. (Tesis de pregrado). Escuela politécnica nacional, Ecuador.
- Pallás Areny, R. (2003). *Sensores y acondicionadores de señal*. Barcelona, España, Editorial Marcombo.
- Pérez, I. y León B. (2007). *Lógica difusa para principiantes: Teoría y práctica*. Caracas, Venezuela, Universidad Valle del Momboy.
- Pérez, I. y León B. (2004). Elementos de un conjunto difuso [Figura]. *Lógica difusa para principiantes: Teoría y práctica*. Caracas, Venezuela, Universidad Valle del Momboy. pp.16
- Polytex.cl® (s.f.) *Malla Raschel*. Chile: www.polytex.cl. Recuperado de
<http://www.polytex.cl/en/PDF%20Fichas%20Polytex/malla%20raschel.pdf>
- QQ Online Trading (s.f). IR Infrared 2 - 30cm Obstacle Detaction Sensor Module FC-51. Recuperado de:
<http://qqtrading.com.my/ir-infrared-obstacle-detaction-sensor-module-fc-5>

- Ramos–Fernández, J.C., López–Morales V., Lafont F., Enea G. y Duplaix J. (2010). Una estructura neurodifusa para modelar la evapotranspiración instantánea en invernaderos. *Ingeniería, investigación y tecnología*, 11(2): 127-139
- Revista Mundo HVACR. (2017). Sistemas de climatización en invernaderos. México: Mundo Hvacr. Recuperado de:
<https://www.mundohvacr.com.mx/mundo/2014/05/sistemas-de-climatizacion-en-invernaderos/>
- Reyna, C. (2015). *Sistema automatizado para el monitoreo y control de humedad en un invernadero* (Tesis de pregrado). Pontificia Universidad Católica del Perú, Lima.
- SENAMHI (2017). *Pronóstico y temperatura actual para Lima Metropolitana*. Perú: Senamhi.gob.pe. Recuperado de:
http://www.senamhi.gob.pe/?p=pronostico_lima
- sparkfun.com (17 de agosto de 2016). *12V Solenoid Valve - 3/4"*. [Figura]. Recuperado de https://media.digikey.com/pdf/Data%20Sheets/Sparkfun%20PDFs/ROB-10456_Web.pdf
- Trillas,E. y Gutierrez Rios, J. (1992). Aplicaciones de la lógica borrosa. Madrid, España, Consejo superior de investigaciones científicas.
- Vildósola C, E. (s.f). *ACTUADORES*. Recuperado de:
<http://www.aie.cl/files/file/comites/ca/abc/actuadores.pdf>
- Wikipedia® (2018) *Miniatura (rosa)*. Recuperado de:
[https://es.wikipedia.org/wiki/Miniatura_\(rosa\)](https://es.wikipedia.org/wiki/Miniatura_(rosa))

ANEXOS

ANEXO I-DATASHEET DE CONTROLADOR



Overview

The Arduino Mega 2560 is a microcontroller board based on the ATmega2560 ([datasheet](#)). It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega is compatible with most shields designed for the Arduino Duemilanove or Diecimila.

The Mega 2560 is an update to the [Arduino Mega](#), which it replaces.

Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-9V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB (8 KB used by bootloader)
SRAM	8 KB
EEPROM	4 KB (ATmega328)
Clock Speed	16 MHz

Schematic & Reference Design

EAGLE files: [arduino-mega2560-reference-design.zip](#)

Schematic: [arduino-mega2560-schematic.pdf](#)

Power

The Arduino Mega can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The Mega2560 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega8U2 programmed as a USB-to-serial converter.

The power pins are as follows:

- **VIN.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3.** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND.** Ground pins.

Memory

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the [EEPROM library](#)).

Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- **Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.
- **PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

- **SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I²C: 20 (SDA) and 21 (SCL).** Support I²C (TWI) communication using the [Wire library](#) (documentation on the Wiring website). Note that these pins are not in the same location as the I²C pins on the Duemilanove or Diecimila.

The Mega2560 has 16 analog inputs, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and `analogReference()` function.

There are a couple of other pins on the board:

- **AREF.** Reference voltage for the analog inputs. Used with [analogReference\(\)](#).
- **Reset.** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Mega2560 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega8U2 on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically). The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A [SoftwareSerial library](#) allows for serial communication on any of the Mega2560's digital pins.

The ATmega2560 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the [documentation on the Wiring website](#) for details. For SPI communication, use the [SPI library](#).

Programming

The Arduino Mega can be programmed with the Arduino software ([download](#)). For details, see the [reference](#) and [tutorials](#).

The ATmega2560 on the Arduino Mega comes preburned with a [bootloader](#) that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol ([reference](#), [C header files](#)).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see [these instructions](#) for details.

The ATmega8U2 firmware source code is available [in the Arduino repository](#). The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2. You can then use [Atmel's FLIP software](#) (Windows) or the [DFU programmer](#) (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See [this user-contributed tutorial](#) for more information.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Mega2560 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega2560 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Mega2560 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Mega2560. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Mega2560 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labelled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see [this forum thread](#) for details.

USB Overcurrent Protection

The Arduino Mega2560 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

The Mega2560 is designed to be compatible with most shields designed for the Uno, Diecimila or Duemilanove. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Further the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on both the Mega2560 and Duemilanove / Diecimila. *Please note that I²C is not located on the same pins on the Mega (20 and 21) as the Duemilanove / Diecimila (analog inputs 4 and 5)*

ANEXO II-DATOS TÉCNICOS DEL DHT22

Model	DHT22
Power supply	3.3-6V DC
Output signal	digital signal via single-bus
Sensing element	Polymer capacitor
Operating range	humidity 0-100%RH; temperature -40~80Celsius
Accuracy	humidity +-2%RH(Max +-5%RH); temperature <+-0.5Celsius
Resolution or sensitivity	humidity 0.1%RH; temperature 0.1Celsius
Repeatability	humidity +-1%RH; temperature +-0.2Celsius
Humidity hysteresis	+/-0.3%RH
Long-term Stability	+/-0.5%RH/year
Sensing period	Average: 2s
Interchangeability	fully interchangeable
Dimensions	small size 14*18*5.5mm; big size 22*28*5mm

Fuente: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>

ANEXO III - ESPECIFICACIONES TÉCNICAS DEL SENSOR BH1750

● Absolute Maximum Ratings

Parameter	Symbol	Ratings	Units
Supply Voltage	Vmax	4.5	V
Operating Temperature	Topr	-40~85	°C
Storage Temperature	Tstg	-40~100	°C
SDA Sink Current	I _{max}	7	mA
Power Dissipation	Pd	260*	mW

* 70mm × 70mm × 1.6mm glass epoxy board. Derating in done at 3.47mW/°C for operating above Ta=25°C.

● Operating Conditions

Parameter	Symbol	Ratings			Units
		Min.	Typ.	Max.	
Vcc Voltage	Vcc	2.4	3.0	3.6	V
I ² C Reference Voltage	V _{DVI}	1.65	-	Vcc	V

Fuente: <http://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf>

ANEXO IV- ESPECIFICACIONES TÉCNICAS DEL SENSOR HC-SR04

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

Fuente: <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

ANEXO V - ESPECIFICACIONES TÉCNICAS DEL SENSOR FC-28

Item	Condition	Min	Typical	Max	Unit
Voltage	-	3.3	/	5	V
Current	-	0	/	35	mA
Output Value	Sensor in dry soil	0	~	300	/
	Sensor in humid soil	300	~	700	/
	Sensor in water	700	~	950	/

Fuente: https://www.mouser.com/ds/2/744/Seeed_101020008-1217463.pdf

ANEXO VI- ESPECIFICACIONES TÉCNICAS DE LOS MÓDULOS LED

Longitud del Módulo: 68mm.

Ancho del Módulo: 19.5mm.

Alto del Módulo: 6mm.

Distancia Entre los Módulos: 77mm.

Tipo de LED Implementado: SMD 5050.

Cantidad de LEDs: 3 LEDs.

Flujo Lumínico por LED: R: 5lm, G: 8lm, B: 8lm, Y: 5lm, W: 16-18lm.

Ángulo de Iluminación: 120°.

Voltaje de Operación: 12VDC.

Consumo en Corriente: 60mA.

Consumo en Potencia: 0.72W.

Colores disponibles:

Nivel de Sellado: IP65 (Uso en exteriores).

Fuente: <http://www.ledsinternational.com/es/modulo-de-leds-3-x-smd-5050-ip65/>

ANEXO VII - ESPECIFICACIONES TÉCNICAS DE LA ELECTROVÁLVULA

1

TECHNICAL DETAILS

- 1/2" Nominal NPS
- Working Pressure: 0.02 Mpa - 0.8 Mpa
- Working Temperature: 1 °C - 75 °C
- Response time (open): ≤ 0.15 sec
- Response time (close): ≤ 0.3 sec
- Actuating voltage: 12VDC (but we found it would work down to 6V)
- Actuating life: ≥ 50 million cycles
- Weight: 4.3 oz
- Dimensions: 3.3" x 1.69" x 2.24"

Fuente: <https://www.verical.com/datasheet/adafruit-i-o-modules-997-4370495.pdf>

ANEXO VIII-ESPECIFICACIONES TÉCNICAS DE LA MALLA RASCHEL

Sombra	Trama	Ancho	Largo	Garantía UV	Gramaje	Origen
80%	90%	4.2 m	100 m	3 años	105 gr/m ²	Importada

Fuente: <http://www.liteperu.com/productos/malla-raschel/>

ANEXO IX- CÓDIGO PARA ESCRIBIR VALOR ANALÓGICO

```
int ventilador;

#include "DHT.h";

#define DHTPIN 2 ;

#define DHTTYPE DHT22,

DHT dht(DHTPIN, DHTTYPE);

void setup() {

  pinMode(13,OUTPUT);

  Serial.begin(9600);

  pinMode(4,OUTPUT);

  dht.begin(); //Se inicia el sensor

}

void loop() {

  if (Serial.available() > 0){

  ventilador=Serial.parseInt();

  Serial.println(ventilador);

  analogWrite(13,ventilador);

  digitalWrite(4,HIGH);

}

float h = dht.readHumidity(); //Se lee la humedad

float t = dht.readTemperature(); //Se lee la temperatura

//Se imprimen las variables

Serial.println("Humedad: ");

Serial.println(h);

Serial.println("Temperatura: ");
```

```
Serial.println(t);  
  
delay(2000); //Se esperan 2 segundos para seguir leyendo //datos  
  
}
```

Fuente: Elaboración propia

**ANEXO X-ESPECIFICACIONES TÉCNICAS DEL SENSOR DE FLUJO
FS400A-G1**

Specification

Mini Working Voltage	DC 4.5V
Max. Working Current	15mA(DC 5V)
Working Voltage	5V~24V
Flow Rate Range	1~60L/min
Load Capacity	≤10mA(DC 5V)
Operating Temperature	≤80°C
Liquid Temperature	≤120°C
Operating Humidity	35%~90%RH
Water Pressure	≤1.75MPa
Storage Temperature	-25°C~+80°C
Storage Humidity	25%~95%RH

Fuente: https://www.seeedstudio.com/g34-water-flow-sensor-p-1083.html?cPath=144_151

ANEXO XI- CÓDIGO DEL SENSOR FS400A G1

```
volatile int NumPulsos; //variable para la cantidad de pulsos recibidos

int PinSensor = 2; //Sensor conectado en el pin 2

float factor_conversion=3.5; //para convertir de frecuencia a caudal

//---Función que se ejecuta en interrupción-----

void ContarPulsos ()

{

    NumPulsos++; //incrementamos la variable de pulsos

}

//---Función para obtener frecuencia de los pulsos-----

int ObtenerFrecuencia()

{

    int frecuencia;

    NumPulsos = 0; //Ponemos a 0 el número de pulsos

    interrupts(); //Habilitamos las interrupciones

    delay(1000); //muestra de 1 segundo

    noInterrupts(); //Desabilitamos las interrupciones

    frecuencia=NumPulsos; //Hz(pulsos por segundo)

    return frecuencia;

}

void setup()

{

    Serial.begin(9600);
```

```

pinMode(PinSensor, INPUT);

attachInterrupt(0,ContarPulsos,RISING); //(Interrupcion 0(Pin2),funcion,Flanco de subida)

}

void loop ()

{

float frecuencia=ObtenerFrecuencia(); //obtenemos la Frecuencia de los pulsos en Hz

float caudal_L_m=frecuencia/factor_conversion; //calculamos el caudal en L/m

float caudal_L_h=caudal_L_m*60; //calculamos el caudal en L/h

//----Enviamos por el puerto serie-----

Serial.print ("FrecuenciaPulsos: ");

Serial.print (frecuencia,0);

Serial.print ("Hz\tCaudal: ");

Serial.print (caudal_L_m,3);

Serial.print (" L/m\t");

Serial.print (caudal_L_h,3);

Serial.println ("L/h");

}

```

Fuente: Elaboración propia

ANEXO XII-CÓDIGO DEL SENSOR DE HUMEDAD

```
//Conectar el sensor MH

void setup() {

  Serial.begin(9600);

  //pinMode(10, INPUT);

}

void loop() {

  if (Serial.available() > 0){

    Serial.println(analogRead(A0)); //lectura analógica

    delay(100); //cada medio segundo

  }
}
```

Fuente: Elaboración propia

ANEXO XIII-CÓDIGO DE VISUALIZACIÓN DE LAS VARIABLES

```
#include <LiquidCrystal.h>

#include <Wire.h>

#include <BH1750.h>

#include "DHT.h" //cargamos la librería DHT

#define DHTPIN 2 //Seleccionamos el pin en el que se conectará el sensor

#define DHTTYPE DHT22 //Se selecciona el DHT22(hay otros DHT)

DHT dht(DHTPIN, DHTTYPE);

BH1750 luxometro;

LiquidCrystal lcd(7,8,9,10,11,12);

void setup() {

  lcd.begin(16,4);

  lcd.print("Bienvenido");

  dht.begin(); //detector de humedad y temperatura

  Wire.begin(); //luxometro

  luxometro.begin(BH1750::ONE_TIME_HIGH_RES_MODE);

  Serial.begin(9600);

  //Serial.println(F("Inicializando sensor..."));

}

void loop() {

  //humedad y temperatura
```

```
float h = dht.readHumidity(); //Se lee la humedad

float t = dht.readTemperature(); //Se lee la temperatura

//Se imprimen las variables

lcd.setCursor(0,1);

lcd.print("HRamb:");

lcd.print(h);

lcd.setCursor(0,2);

lcd.print("Tamb:");

lcd.print(t);

//Se espera 2 segundos para seguir leyendo //datos

uint16_t lux = luxometro.readLightLevel(); // Lectura del BH1750

lcd.setCursor(0,3);

lcd.print("lux");

lcd.print(lux);

delay(2000);

}
```

Fuente: Elaboración propia

ANEXO XIV-CÓDIGO DE VISUALIZACIÓN DE LA LUMINOSIDAD

```
#include <LiquidCrystal.h>

#include <Wire.h>

#include <BH1750.h>

BH1750 luxometro;

LiquidCrystal lcd(7,8,9,10,11,12);

// BH1750_CONTINUOUS_HIGH_RES_MODE
// BH1750_CONTINUOUS_HIGH_RES_MODE_2
// BH1750_CONTINUOUS_LOW_RES_MODE
// BH1750_ONE_TIME_HIGH_RES_MODE
// BH1750_ONE_TIME_HIGH_RES_MODE_2
// BH1750_ONE_TIME_LOW_RES_MODE

void setup() {

  lcd.begin(16,4);

  lcd.print("Bienvenido");

  Wire.begin();

  luxometro.begin(BH1750::ONE_TIME_HIGH_RES_MODE);

  Serial.begin(9600);

  // Serial.println(F("Inicializando sensor..."));

}

void loop() {

  uint16_t lux = luxometro.readLightLevel(); // Lectura del BH1750

  //lcd.setCursor(0, 8);

  //lcd.print("Cantidad de lux");
```

```
Serial.print(lux);  
Serial.println(" lx");  
delay(500);  
}
```

Fuente: Elaboración propia

ANEXO XV-CÓDIGO SIMPLE DEL SENSOR ULTRASÓNICO

```
const int trigPin = 48;

const int echoPin = 50;

// defines variables

long duration;

int distance;

const int numOfReadings = 5;

int readings[numOfReadings];

int arrayIndex = 0;

int sumreadings=0;

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

digitalWrite(trigPin, LOW);

}

void loop() {

Serial.begin(9600); // Starts the serial communication

for (int thisReading = 0; thisReading < numOfReadings; thisReading++) {

readings[thisReading] = 0;

digitalWrite(trigPin, HIGH); // genera el pulso de trigger por 10ms

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

// Calculating the distance
```

```
distance= duration/59; //distancia en centimetros

if (distance>200)

{ distance=0;}

readings[arrayIndex] = distance;

sumreadings=distance+readings[arrayIndex];

delay(100);}

// Serial.print("Distance: ");

//Serial.print(distance);

//Serial.println("cm");

int averagereading= sumreadings/numOfReadings;

Serial.print(averagereading);

Serial.println("cm");

delay(200);

}
```

Fuente: Naylamp Mechatronics ®

ANEXO XVI-CÓDIGO MEJORADO DEL SENSOR ULTRASÓNICO

```
const int trigPin = 48;

const int echoPin = 50;

// defines variables

long duration;

int distance;

const int numOfReadings = 5;

int readings[numOfReadings];

int arrayIndex = 0;

int sumreadings=0;

void setup() {

pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output

pinMode(echoPin, INPUT); // Sets the echoPin as an Input

digitalWrite(trigPin, LOW);

}

void loop() {

Serial.begin(9600); // Starts the serial communication

for (int thisReading = 0; thisReading < numOfReadings; thisReading++) {

readings[thisReading] = 0;

digitalWrite(trigPin, HIGH); // genera el pulso de trigger por 10ms

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

// Calculating the distance
```

```
distance= duration/59; //distancia en centimetros

if (distance>200)//eliminar picos

{ distance=0;}

readings[arrayIndex] = distance;

sumreadings=distance+readings[arrayIndex];

delay(100);}

// Serial.print("Distance: ");

//Serial.print(distance);

//Serial.println("cm");

int averagereading= sumreadings/numOfReadings;

Serial.print(averagereading);

Serial.println("cm");

delay(200);//recomendacion fabricante mayor a 60 ms

}
```

Fuente: Elaboración propia

ANEXO XVII-CÓDIGO DEL SENSOR DE HUMEDAD DEL SUELO

```
void setup() {  
  
  Serial.begin(9600);  
  
  //pinMode(10, INPUT);  
  
}  
  
void loop() {  
  
  if (Serial.available() > 0){  
  
    Serial.println(analogRead(A0)); //lectura analógica  
  
    delay(100); //cada medio segundo  
  
  }  
}
```

Fuente: Elaboración propia

ANEXO XVIII-CÓDIGO FINAL

```
//MOSTRAR TODAS LAS VARIABLES EN LA PANTALLA 16X4

//CALCULO DEL DPV DEL INVERNADERO

#include <Time.h>

#include <TimeAlarms.h>

//FUZZY

#include <FuzzyRule.h>

#include <FuzzyComposition.h>

#include <Fuzzy.h>

#include <FuzzyRuleConsequent.h>

#include <FuzzyOutput.h>

#include <FuzzyInput.h>

#include <FuzzyIO.h>

#include <FuzzySet.h>

#include <FuzzyRuleAntecedent.h>

//--TODAS LAS LIBRERIAS--

#include <LiquidCrystal.h>

#include <Wire.h>

#include <BH1750.h>

#include "DHT.h" //cargamos la librería DHT

#define DHTPIN 2 //Seleccionamos el pin en el que se conectará el sensor

#define DHTTYPE DHT22 //Se selecciona el DHT22(hay otros DHT)

DHT dht(DHTPIN, DHTTYPE);

BH1750 luxometro;

const int ventilador=24;

const int bomba= 22;

const int electrovalvula1=26; //CONSTANTE DE LA E1

const int electrovalvula2=28; //CONSTANTE DE LA E2
```

```

const int luces= 14;

//CONSTANTES DEL SENSOR ULTRASONICO

const int trigPin = 48;

const int echoPin = 50;

//Definición de variables

long duration;

int distance;

const int numOfReadings = 5;

int readings[numOfReadings];

int arrayIndex = 0;

int sumreadings=0;

LiquidCrystal lcd(7,8,9,10,11,12);

Fuzzy* fuzzy = new Fuzzy();

//

void setup() {

  lcd.begin(16,4);

  lcd.print("Bienvenido...");

  dht.begin(); //detector de humedad y temperatura

  Wire.begin(); //luxometro

  luxometro.begin(BH1750::ONE_TIME_HIGH_RES_MODE);// resolución 1 lux

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  digitalWrite(trigPin, LOW);

  pinMode(electrovalvula2,OUTPUT);

  pinMode(electrovalvula1,OUTPUT);

  pinMode(luces,OUTPUT);

  Serial.begin(9600);

//ENTRADAS

//HUMEDAD Y TEMPERATURA

```

```

//HUMEDAD RELATIVA

FuzzySet*htuno = new FuzzySet(35,35,40,40);

FuzzySet*htdos = new FuzzySet(41,41,55,55);

FuzzySet*httres = new FuzzySet(56,56,80,80);

FuzzySet*htcuatro = new FuzzySet(81,81,100,100);

//TEMPERATURA

FuzzySet*htunodos = new FuzzySet(15,15,19,19);

FuzzySet*htdosdos= new FuzzySet(20,20,24,24);

FuzzySet*httresdos = new FuzzySet(25,25,31,31);

FuzzySet*htcuatrosdos= new FuzzySet(32,32,35,35);

//LUCES

FuzzySet*dia = new FuzzySet(2900,2950,3050,3100);

FuzzySet*medio = new FuzzySet(10,62.5,2800,2900);

FuzzySet*oscuridad = new FuzzySet(-1,2.5,2.5,5);

//LLENADO Y VACIADO

FuzzySet* dos= new FuzzySet(2, 2, 2, 3);

FuzzySet* tres= new FuzzySet(2, 3, 3, 4);

FuzzySet* cuatro= new FuzzySet(3, 4, 4, 5);

FuzzySet* cinco= new FuzzySet(4, 5, 5, 5);

//

FuzzySet* siete= new FuzzySet(7, 7, 7, 8);

FuzzySet* ocho= new FuzzySet(7, 8, 8, 9);

FuzzySet* nueve= new FuzzySet(8, 9, 9, 10);

FuzzySet* diez= new FuzzySet(9, 10, 10, 11);

FuzzySet* once= new FuzzySet(10, 11, 11, 12);

FuzzySet* doce= new FuzzySet(11, 12, 12, 13);

FuzzySet* trece= new FuzzySet(12, 13, 13, 13);

//FERTIRRIEGO

FuzzySet* muyhumedo = new FuzzySet(124, 124, 124, 800);

```

```

FuzzySet* muyseco = new FuzzySet(300, 1023, 1023, 1023);

//SALIDAS

//VOLTAJE-LUCES/ HUMEDAD Y TEMPERATURA

FuzzySet* nulo = new FuzzySet(0,0,0,0);

FuzzySet* activado = new FuzzySet(254,255,255,256);

//TIEMPOS LLENADO Y VACIADO

//

FuzzySet* tdos = new FuzzySet(1415, 1425, 1425, 1445);

FuzzySet* ttres = new FuzzySet(730, 750, 750, 770);

FuzzySet* tcuatro = new FuzzySet(290, 300, 300, 310);

FuzzySet* tcinco = new FuzzySet(177, 187, 187, 197);

//

FuzzySet* tsiete = new FuzzySet(18.4333, 18.6333, 18.6333, 18.8333);

FuzzySet* tocho = new FuzzySet(37.0666, 37.2666, 37.2666, 37.4666);

FuzzySet* tnueve = new FuzzySet(55.6999, 55.8999, 55.8999, 56.0999);

FuzzySet* tdiez = new FuzzySet(68.1211, 68.3221, 68.3211, 68.5211);

FuzzySet* tonce = new FuzzySet(80.5443, 80.7443, 80.7443, 80.9443);

FuzzySet* tdoce = new FuzzySet(93.9317, 94.1317, 94.1317, 94.3337);

FuzzySet* ttrece = new FuzzySet(94.9317, 95.1317 , 95.1317, 95.3317);

//FERTIRRIEGO

FuzzySet* nada = new FuzzySet(0, 0, 0, 5);

FuzzySet* todo = new FuzzySet(25, 30, 30, 30);

//ASIGNACION DE CONJUNTOS DIFUSOS

//HUMEDAD RELATIVA Y TEMPERATURA

FuzzyInput* humedadrelativa = new FuzzyInput(1);

humedadrelativa->addFuzzySet(htuno);

humedadrelativa->addFuzzySet(htdos);

humedadrelativa->addFuzzySet(httres);

humedadrelativa->addFuzzySet(htcuatro);

```

```
fuzzy->addFuzzyInput(humedadrelativa);
```

```
FuzzyInput*temperatura = new FuzzyInput(2);
```

```
temperatura->addFuzzySet(htunodos);
```

```
temperatura->addFuzzySet(htdosdos);
```

```
temperatura->addFuzzySet(httresdos);
```

```
temperatura->addFuzzySet(htcuatrodos);
```

```
fuzzy->addFuzzyInput(temperatura);
```

```
FuzzyOutput*voltaje = new FuzzyOutput(1);
```

```
voltaje->addFuzzySet(nulo);
```

```
voltaje->addFuzzySet(activado);
```

```
fuzzy->addFuzzyOutput(voltaje);
```

```
//REGLAS
```

```
FuzzyRuleAntecedent* ifrh1andt4 = new FuzzyRuleAntecedent();
```

```
ifrh1andt4->joinWithAND(htuno,htcuatrodos);
```

```
FuzzyRuleConsequent* thenvoltajeactivado = new FuzzyRuleConsequent();
```

```
thenvoltajeactivado->addOutput(activado);
```

```
FuzzyRule*fuzzyRule01 = new FuzzyRule(1, ifrh1andt4,thenvoltajeactivado);
```

```
fuzzy->addFuzzyRule(fuzzyRule01);
```

```
FuzzyRuleAntecedent* ifrh1andt3 = new FuzzyRuleAntecedent();
```

```
ifrh1andt3->joinWithAND(htuno,httresdos);
```

```
FuzzyRule*fuzzyRule02 = new FuzzyRule(2, ifrh1andt3,thenvoltajeactivado);
```

```
fuzzy->addFuzzyRule(fuzzyRule02);
```

```
FuzzyRuleAntecedent* ifrh1andt2 = new FuzzyRuleAntecedent();  
ifrh1andt2->joinWithAND(htuno,htdosdos);  
FuzzyRule*fuzzyRule03 = new FuzzyRule(3, ifrh1andt2,thenvoltajeactivado);  
fuzzy->addFuzzyRule(fuzzyRule03);
```

```
FuzzyRuleAntecedent* ifrh1andt1 = new FuzzyRuleAntecedent();  
ifrh1andt1->joinWithAND(htuno,htunodos);  
FuzzyRule*fuzzyRule04 = new FuzzyRule(4, ifrh1andt1,thenvoltajeactivado);  
fuzzy->addFuzzyRule(fuzzyRule04);
```

```
//
```

```
FuzzyRuleAntecedent* ifrh2andt4 = new FuzzyRuleAntecedent();  
ifrh2andt4->joinWithAND(htdos,htcuatrodos);  
FuzzyRule*fuzzyRule05 = new FuzzyRule(5, ifrh2andt4,thenvoltajeactivado);  
fuzzy->addFuzzyRule(fuzzyRule05);
```

```
FuzzyRuleAntecedent* ifrh2andt3 = new FuzzyRuleAntecedent();  
ifrh2andt3->joinWithAND(htdos,httresdos);  
FuzzyRule*fuzzyRule06 = new FuzzyRule(6, ifrh2andt3,thenvoltajeactivado);  
fuzzy->addFuzzyRule(fuzzyRule06);
```

```
FuzzyRuleAntecedent* ifrh2andt2 = new FuzzyRuleAntecedent();  
ifrh2andt2->joinWithAND(htdos,htdosdos);  
FuzzyRule*fuzzyRule07 = new FuzzyRule(7, ifrh2andt2,thenvoltajeactivado);  
fuzzy->addFuzzyRule(fuzzyRule07);
```

```
FuzzyRuleAntecedent* ifrh2andt1 = new FuzzyRuleAntecedent();  
ifrh2andt1->joinWithAND(htdos,htunodos);  
FuzzyRule*fuzzyRule08 = new FuzzyRule(8, ifrh2andt1,thenvoltajeactivado);  
fuzzy->addFuzzyRule(fuzzyRule08);
```

```
//
```

```

FuzzyRuleAntecedent* ifrh3andt4 = new FuzzyRuleAntecedent();
ifrh3andt4->joinWithAND(httres,htcuatrodos);
FuzzyRule*fuzzyRule09 = new FuzzyRule(9, ifrh3andt4,thenvoltajeactivado);
fuzzy->addFuzzyRule(fuzzyRule09);

FuzzyRuleAntecedent* ifrh3andt3 = new FuzzyRuleAntecedent();
ifrh3andt3->joinWithAND(httres,httresdos);
FuzzyRule*fuzzyRule10 = new FuzzyRule(10, ifrh3andt3,thenvoltajeactivado);
fuzzy->addFuzzyRule(fuzzyRule10);

FuzzyRuleAntecedent* ifrh3andt2 = new FuzzyRuleAntecedent();
ifrh3andt2->joinWithAND(httres,htdosdos);
FuzzyRule*fuzzyRule11 = new FuzzyRule(11, ifrh3andt2,thenvoltajeactivado);
fuzzy->addFuzzyRule(fuzzyRule11);

FuzzyRuleAntecedent* ifrh3andt1 = new FuzzyRuleAntecedent();
ifrh3andt1->joinWithAND(httres,htunodos);
FuzzyRule*fuzzyRule12 = new FuzzyRule(12, ifrh3andt1,thenvoltajeactivado);
fuzzy->addFuzzyRule(fuzzyRule12);
//
FuzzyRuleAntecedent* ifrh4andt4 = new FuzzyRuleAntecedent();
ifrh4andt4->joinWithAND(htcuatro,htcuatrodos);
FuzzyRuleConsequent* thenvoltajenulo = new FuzzyRuleConsequent();
thenvoltajenulo->addOutput(nulo);
FuzzyRule*fuzzyRule13 = new FuzzyRule(13, ifrh4andt4,thenvoltajenulo);
fuzzy->addFuzzyRule(fuzzyRule13);

FuzzyRuleAntecedent* ifrh4andt3 = new FuzzyRuleAntecedent();
ifrh4andt3->joinWithAND(htcuatro,httresdos);
FuzzyRule*fuzzyRule14 = new FuzzyRule(14, ifrh4andt3,thenvoltajenulo);

```

```

fuzzy->addFuzzyRule(fuzzyRule14);

FuzzyRuleAntecedent* ifrh4andt2 = new FuzzyRuleAntecedent();
ifrh4andt2->joinWithAND(htcuatro,htdosdos);
FuzzyRule*fuzzyRule15 = new FuzzyRule(15, ifrh4andt2,thenvoltajenulo);
fuzzy->addFuzzyRule(fuzzyRule15);

FuzzyRuleAntecedent* ifrh4andt1 = new FuzzyRuleAntecedent();
ifrh4andt1->joinWithAND(htcuatro,htunodos);
FuzzyRule*fuzzyRule16 = new FuzzyRule(16, ifrh4andt1,thenvoltajenulo);
fuzzy->addFuzzyRule(fuzzyRule16);

/--LUMINOSIDAD
FuzzyInput*luminosidad = new FuzzyInput(2);
luminosidad->addFuzzySet(dia); // Adicionando o FuzzySet slow em velocity
luminosidad->addFuzzySet(medio); // Adicionando o FuzzySet average em velocity
luminosidad->addFuzzySet(oscuridad); // Adicionando o FuzzySet fast em velocity

fuzzy->addFuzzyInput(luminosidad); // Adicionando o FuzzyOutput no objeto Fuzzy

//creando salida. voltaje
FuzzyOutput*voltajeluces = new FuzzyOutput(2);

voltajeluces->addFuzzySet(nulo);
voltajeluces->addFuzzySet(activado);

fuzzy->addFuzzyOutput(voltajeluces);

//
FuzzyRuleAntecedent* ifcantidadluxdia = new FuzzyRuleAntecedent();
ifcantidadluxdia->joinSingle(dia);
FuzzyRuleConsequent* thenvoltajelucesnulo = new FuzzyRuleConsequent();

```

```

thenvoltajelucesnulo->addOutput(nulo);

FuzzyRule*fuzzyRule17 = new FuzzyRule(17, ifcantidadluxdia,thenvoltajelucesnulo);

fuzzy->addFuzzyRule(fuzzyRule17);

FuzzyRuleAntecedent*ifcantidadluxmedio = new FuzzyRuleAntecedent();

ifcantidadluxmedio->joinSingle(medio);

FuzzyRuleConsequent*thenvoltajelucesactivado = new FuzzyRuleConsequent();

thenvoltajelucesactivado->addOutput(activado);

FuzzyRule*fuzzyRule18 = new FuzzyRule(18, ifcantidadluxmedio,thenvoltajelucesactivado);

fuzzy->addFuzzyRule(fuzzyRule18);

FuzzyRuleAntecedent* ifcantidadluxoscuridad = new FuzzyRuleAntecedent();

ifcantidadluxoscuridad->joinSingle(oscuridad);

FuzzyRule*fuzzyRule19 = new FuzzyRule(19,ifcantidadluxoscuridad,thenvoltajelucesactivado);

fuzzy->addFuzzyRule(fuzzyRule19);

//NIVEL

FuzzyInput*electrovalvula = new FuzzyInput(3);

electrovalvula->addFuzzySet(dos);

electrovalvula->addFuzzySet(tres);

electrovalvula->addFuzzySet(cuatro);

electrovalvula->addFuzzySet(cinco);

electrovalvula->addFuzzySet(siete);

electrovalvula->addFuzzySet(ocho);

electrovalvula->addFuzzySet(nueve);

electrovalvula->addFuzzySet(diez);

electrovalvula->addFuzzySet(once);

electrovalvula->addFuzzySet(doce);

electrovalvula->addFuzzySet(trece);

```

```

fuzzy->addFuzzyInput(electrovalvula);

FuzzyOutput*tiempoapertura = new FuzzyOutput(3);

tiempoapertura->addFuzzySet(tdos);

tiempoapertura->addFuzzySet(ttres);

tiempoapertura->addFuzzySet(tcuatro);

tiempoapertura->addFuzzySet(tcinco);

tiempoapertura->addFuzzySet(tsiete);

tiempoapertura->addFuzzySet(tocho);

tiempoapertura->addFuzzySet(tnueve);

tiempoapertura->addFuzzySet(tdiez);

tiempoapertura->addFuzzySet(tonce);

tiempoapertura->addFuzzySet(tdoce);

tiempoapertura->addFuzzySet(ttrece);

fuzzy->addFuzzyOutput(tiempoapertura);

//CREACION DE REGLAS

FuzzyRuleAntecedent* ifdos = new FuzzyRuleAntecedent();

ifdos->joinSingle(dos);

FuzzyRuleConsequent* thentempodos = new FuzzyRuleConsequent();

thentempodos->addOutput(tdos);

FuzzyRule*fuzzyRule20 = new FuzzyRule(20, ifdos,thentempodos);

fuzzy->addFuzzyRule(fuzzyRule20);

FuzzyRuleAntecedent* iftres = new FuzzyRuleAntecedent();

iftres->joinSingle(tres);

FuzzyRuleConsequent* thentempotres = new FuzzyRuleConsequent();

thentempotres->addOutput(ttres);

FuzzyRule*fuzzyRule21 = new FuzzyRule(21, iftres,thentempotres);

fuzzy->addFuzzyRule(fuzzyRule21);

```

```
FuzzyRuleAntecedent* ifcuatro = new FuzzyRuleAntecedent();  
ifcuatro->joinSingle(cuatro);  
FuzzyRuleConsequent* thentempocuatro = new FuzzyRuleConsequent();  
thentempocuatro->addOutput(tcuatro);  
FuzzyRule*fuzzyRule22 = new FuzzyRule(22, ifcuatro,thentempocuatro);  
fuzzy->addFuzzyRule(fuzzyRule22);
```

```
FuzzyRuleAntecedent* ifcinco = new FuzzyRuleAntecedent();  
ifcinco->joinSingle(cinco);  
FuzzyRuleConsequent* thentempocinco = new FuzzyRuleConsequent();  
thentempocinco->addOutput(tcinco);  
FuzzyRule*fuzzyRule23 = new FuzzyRule(23, ifcinco,thentempocinco);  
fuzzy->addFuzzyRule(fuzzyRule23);
```

```
FuzzyRuleAntecedent* ifsiete = new FuzzyRuleAntecedent();  
ifdos->joinSingle(siete);  
FuzzyRuleConsequent* thentemposiete = new FuzzyRuleConsequent();  
thentemposiete->addOutput(tsiete);  
FuzzyRule*fuzzyRule24 = new FuzzyRule(24, ifsiete,thentemposiete);  
fuzzy->addFuzzyRule(fuzzyRule24);
```

```
FuzzyRuleAntecedent* ifocho = new FuzzyRuleAntecedent();  
ifocho->joinSingle(ocho);  
FuzzyRuleConsequent* thentempoocho = new FuzzyRuleConsequent();  
thentempoocho->addOutput(tocho);  
FuzzyRule*fuzzyRule25 = new FuzzyRule(25, ifocho,thentempoocho);  
fuzzy->addFuzzyRule(fuzzyRule25);
```

```
FuzzyRuleAntecedent* ifnueve = new FuzzyRuleAntecedent();  
ifnueve->joinSingle(nueve);
```

```
FuzzyRuleConsequent* thentiemponueve = new FuzzyRuleConsequent();
thentiempos->addOutput(tdos);
FuzzyRule*fuzzyRule26 = new FuzzyRule(26, ifnueve,thentiemponueve);
fuzzy->addFuzzyRule(fuzzyRule26);
```

```
FuzzyRuleAntecedent* ifdiez = new FuzzyRuleAntecedent();
ifdiez->joinSingle(diez);
FuzzyRuleConsequent* thentiempodiez = new FuzzyRuleConsequent();
thentiempodiez->addOutput(tdiez);
FuzzyRule*fuzzyRule27 = new FuzzyRule(27, ifdiez,thentiempodiez);
fuzzy->addFuzzyRule(fuzzyRule27);
```

```
FuzzyRuleAntecedent* ifonce = new FuzzyRuleAntecedent();
ifonce->joinSingle(once);
FuzzyRuleConsequent* thentiempoonce = new FuzzyRuleConsequent();
thentiempoonce->addOutput(tonce);
FuzzyRule*fuzzyRule28 = new FuzzyRule(28, ifonce,thentiempoonce);
fuzzy->addFuzzyRule(fuzzyRule28);
```

```
FuzzyRuleAntecedent* ifdoce = new FuzzyRuleAntecedent();
ifdoce->joinSingle(doce);
FuzzyRuleConsequent* thentiempodoce = new FuzzyRuleConsequent();
thentiempodoce->addOutput(tdoce);
FuzzyRule*fuzzyRule29 = new FuzzyRule(29, ifdoce,thentiempodoce);
fuzzy->addFuzzyRule(fuzzyRule29);
```

```
FuzzyRuleAntecedent* iftrece = new FuzzyRuleAntecedent();
iftrece->joinSingle(trece);
FuzzyRuleConsequent* thentiempotrece = new FuzzyRuleConsequent();
thentiempotrece->addOutput(ttrece);
```

```

FuzzyRule*fuzzyRule30 = new FuzzyRule(30, ifrece,thentimpotrece);
fuzzy->addFuzzyRule(fuzzyRule30);
//
FuzzyInput*humedadtierra = new FuzzyInput(1);
humedadtierra->addFuzzySet(muyseco);
humedadtierra->addFuzzySet(muyhumedo);

fuzzy->addFuzzyInput(humedadtierra);

FuzzyOutput*tiempo = new FuzzyOutput(1);
tiempo->addFuzzySet(nada);
tiempo->addFuzzySet(todo);

fuzzy->addFuzzyOutput(tiempo);

FuzzyRuleAntecedent* ifhumedadmuyseco = new FuzzyRuleAntecedent();
ifhumedadmuyseco->joinSingle(muyseco);
FuzzyRuleConsequent* thentimpotodo = new FuzzyRuleConsequent();
thentimpotodo->addOutput(todo);
FuzzyRule*fuzzyRule31 = new FuzzyRule(31, ifhumedadmuyseco,thentimpotodo);
fuzzy->addFuzzyRule(fuzzyRule31);

FuzzyRuleAntecedent* ifhumedadmuyhumedo = new FuzzyRuleAntecedent();
ifhumedadmuyhumedo->joinSingle(muyhumedo);
FuzzyRuleConsequent* thentiemponada = new FuzzyRuleConsequent();
thentiemponada->addOutput(nada);
FuzzyRule*fuzzyRule32 = new FuzzyRule(32,ifhumedadmuyhumedo,thentiemponada);
fuzzy->addFuzzyRule(fuzzyRule32);

setTime(12,0,0, 11, 4, 18);

```

```

}

void loop() {
  //humedad y temperatura
  for (int thisReading = 0; thisReading < numOfReadings; thisReading++) {
    readings[thisReading] = 0;
    digitalWrite(trigPin, HIGH); // genera el pulso de trigger por 10ms
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    // Calculating the distance
    distance= duration/59; //distancia en centimetros
    if (distance>200)//eliminar picos
    { distance=0;}
    readings[arrayIndex] = distance;
    sumreadings=distance+readings[arrayIndex];
    delay(100);}
    int averagereading= sumreadings/numOfReadings;
    delay(200);//recomendacion fabricante mayor a 60 ms

    float h = dht.readHumidity(); //Se lee la humedad
    float t = dht.readTemperature(); //Se lee la temperatura
    float SPV= 4.2173*t*t-7.9128*t+791.76;
    float DPV= (1-h/100)*SPV;
    uint16_t lux = luxometro.readLightLevel(); // Lectura del BH1750

    //Se imprimen las variables
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("HRamb:");

```

```

lcd.print(h);

lcd.setCursor(0,1);

lcd.print("Tamb:");

lcd.print(t);

//Se espera 2 segundos para seguir leyendo //datos

lcd.setCursor(0,2);

lcd.print("Lux:");

lcd.print(lux);

lcd.setCursor(0,3);

lcd.print("DPV:");

lcd.print(DPV);

Serial.println(averagereading);

//

//SALIDAS

fuzzy->setInput(1,h); //humedad relativa

fuzzy->setInput(2,t); //temperatura

fuzzy->setInput(3,lux); // cantidad de luxes

fuzzy->setInput(4,averagereading);

//

fuzzy->fuzzify();

float outputrht= fuzzy->defuzzify(1);

float outputluxes= fuzzy->defuzzify(2);

float outputfertirriego= fuzzy->defuzzify(3);

float tiempoe= fuzzy->defuzzify(4);

//NIVEL

if (averagereading>0){

```

```

if (averagereading<6) {
digitalWrite(electrovalvula1,LOW);
digitalWrite(electrovalvula2,LOW); //bomba activada X segundos
delay(tiempoe*1000); //bomba activada X segundos
digitalWrite(electrovalvula2,HIGH);
delay(1000000); //recomendacion fabricante mayor a 60 ms
}
if (averagereading>6){
digitalWrite(electrovalvula2,LOW);
digitalWrite(electrovalvula1,LOW); //bomba activada X segundos
delay(tiempoe*1000); //bomba activada X segundos
digitalWrite(electrovalvula1,HIGH);
delay(1000000); //recomendacion fabricante mayor a 60 ms
}}

```

```
//FERTIRRIEGO
```

```
Alarm.alarmRepeat(24,0,0,riego);
```

```
//VENTILADOR
```

```
if (outputrht==255)
```

```
{ digitalWrite(ventilador, LOW);}
```

```
//LUCES
```

```
else
```

```
{ digitalWrite(ventilador, HIGH);}
```

```
analogWrite(luces,outputluxes);
```

```
Alarm.delay(1000);
```

```
delay(600000); //diez minutos
```

```
}
```

```
void riego()
```

```
{  
  
  int valormh = analogRead(A0); //lectura analógica  
  
  fuzzy->setInput(5, valormh);  
  
  fuzzy->fuzzify();  
  
  float tiempobomba= fuzzy->defuzzify(5);  
  
  digitalWrite(bomba, LOW);  
  
  delay(tiempobomba*1000);  
  
  digitalWrite(bomba, HIGH);  
  
  delay(600000);  
  
}
```

Fuente: Elaboración propia