

**UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
SEDE CONCEPCIÓN - REY BALDUINO DE BÉLGICA**

**“Control Automatizado de Temperatura, Humedad, Riego por
nebulización para cultivo a escala de lechugas”**

Trabajo de Titulación para optar al Título de Técnico
Universitario en **Automatización y Control**.

Alumno:

Felipe A. Yevilao Cuevas.

Profesor Guía:

Helmut Contreras Novoa.

Dedicatoria.

"Agradezco a Dios, a mis padres, hermanos, novia y familia en general por el apoyo constante en estos largos años de universidad, por la paciencia y entrega hacia mí, que gracias a ello he cumplido con una etapa más de mi vida"

RESUMEN

Hoy en día se ha masificado la creación de invernaderos caseros, esto por demanda de productos orgánicos.

Uno de los requerimientos principales es como aplicar un regadío eficiente, ya que a la hora de cultivar el agua es un elemento indispensable.

Para esto existen múltiples sistemas, destacando entre ellos el sistema de regadío por nebulización, porque no solo permite regar, sino también, controlar más factores de crecimiento como lo son la humedad, la temperatura del recinto y también ahorrar hasta un 25 % del agua en comparación a un sistema de regadío tradicional. Ya que el sistema permite crear una capa fina de agua sobre las plantas las que bajan hasta sus raíces.

Por ello, el interés de crear un proyecto de un sistema automatizado que permita controlar el regadío, humedad relativa, humedad de suelo y la temperatura de un recinto a menor escala destinado al cultivo. Ya que se busca la mayor eficiencia posible.

Dicho proyecto escrito quedará a disposición de los alumnos de la Universidad, con el fin de su posible implementación y/o mejoras

ÍNDICE.

INTRODUCCIÓN 1

DESCRIPCIÓN DEL PROBLEMA: 2

OBJETIVO GENERAL..... 2

OBJETIVOS ESPECÍFICOS. 2

ALCANCES..... 3

METODOLOGÍA..... 4

CAPÍTULO 1 . TEORÍA Y TIPOS DE INVERNADEROS. 5

 1.1 TEORÍA DE LOS INVERNADEROS..... 6

 1.1.1 Tipos de invernaderos..... 6

CAPÍTULO 2. TEORÍA DEL CONTROL AUTOMÁTICO. 7

 2.1 DEFINICIÓN. 8

 2.4 CLASIFICACIÓN DE LOS SISTEMAS DE CONTROL..... 10

 2.4.1 Sistema de control de lazo abierto..... 10

 2.4.2 Sistema de control de lazo cerrado. 11

CAPÍTULO 3. MICROCONTROLADORES. 12

 3.1 ¿Qué es un microcontrolador?..... 13

 3.1.1 Arquitectura. 13

 3.2. Microcontroladores AVR. 14

 3.2.2 Grupos..... 14

CAPÍTULO 4. INTERNET DE LAS COSAS..... 16

 4.2 Aplicaciones de consumo..... 18

CAPÍTULO 5. PROTOCOLO DE COMUNICACIÓN I2C..... 19

 5.1 Historia protocolo I2C. 20

 5.2 I2C protocolo de comunicación serial. 20

 5.3 ¿Cómo funciona el protocolo I2C? 21

5.3.1	Descripción de trama de datos.	22
5.4	Conceptos del I2C.	22
CAPÍTULO 6. TEMPERATURA, HUMEDAD RELATIVA; MÉTODOS Y PRINCIPIOS DE MEDICIÓN.		24
6.1	TEMPERATURA.	25
6.1.1	Definición de variable.	25
6.1.2	Métodos y principios de medición.	25
6.2	HUMEDAD RELATIVA.	26
6.2.1	Definición.	26
6.2.2	MÉTODOS Y PRINCIPIOS DE MEDICIÓN.	26
6.2.2.1	Higrómetro.	26
6.2.2.1.1	Mecánico.	26
6.2.2.1.2	Bulbo húmedo y seco (sicrómetro).	27
6.2.2.2	Impedancia eléctrica.	27
6.2.2.2.1	Sensores capacitivos.	27
6.2.2.2.2	Sensores resistivos.	27
CAPÍTULO 7. ELECCIÓN DE HORTALIZA A CULTIVAR.		28
7.1	ELECCIÓN DE HORTALIZA.	29
7.2.	Requerimientos Edafoclimáticos.	30
7.2.1.	Temperatura.	30
7.2.2	Humedad relativa.	30
7.2.3	Riego.	30
CAPÍTULO 8. ELECCIÓN DEL MICROCONTROLADOR.		31
8.1	ENTRADAS Y SALIDAS ESTIMADAS.	32
8.1.1	Tabla de entradas estimadas.	32
8.1.2	Tabla de salidas estimadas.	33
8.2	ELECCIÓN DEL MICROCONTROLADOR.	33

8.2.1	MICROCONTROLADORES PIC.	33
8.2.1.1	estructura central.	33
8.2.1.2	Programación.	34
8.2.1.3	Placa de desarrollo para PIC.	35
8.2.1.4	Costo.	35
8.2.1.5	Compatibilidad con módulos externos (sensores, actuadores, relés etc.)	35
8.2.2	ARDUINO.	36
8.2.2.1	Código abierto y hardware extensible.	36
8.2.2.2	Programación.	37
8.2.2.3	Código abierto y software extensible.	37
8.2.2.4	Costo.	37
8.2.2.5	Multiplataforma.	37
8.2.3	RASPBERRY PI.	38
8.2.3.1	Hardware.	38
8.2.3.2	Compatibilidad con elementos de sensado y actuadores.	39
8.2.3.3	Costo.	39
8.2.3.4	Software y programación.	39
8.3	ELECCIÓN FINAL DEL CONTROLADOR.	40
8.4	ELECCIÓN MODELO ARDUINO.	41
8.5	ARDUINO UNO R3.	42
8.6	ARDUINO CONECTADO A INTERNET.	43
8.6.1	Modelo TCP/IP.	43
8.6.2	HTTP.	44
8.6.2.1	HTTP Request.	44
8.6.2.2	HTTP Response.	45
8.6.3	Protocolo MQTT.	45

8.7	Lenguaje y programación software.	47
8.7.1	Variables.	48
8.7.2	Configuración de pines.	49
8.7.3	Comunicación puerto serial.	50
8.7.4	Bloques condicionales.	51
8.7.5	Control de flujo.	51
8.7.5.1	if.	52
8.7.5.2	if...else.	52
8.7.5.3	for.	52
8.7.5.4	while.	53
8.7.5.5	do...while.	53
CAPÍTULO 9. DESARROLLO DEL PROYECTO, DEFINICIÓN DE SENSORES, ACTUADORES Y CONDICIONES DE TRABAJO.		54
9.1	CONTROL DE TEMPERATURA.	55
9.1.1	Sensor.	56
9.1.1.1	Conexión con Arduino.	58
9.1.1.2	Comunicación con Arduino.	59
9.2	CONTROL DE HUMEDAD.	60
9.2.1	Condiciones de trabajo.	60
9.3	Actuadores.	62
9.3.1	Extractor de aire.	62
9.3.2	Calefactor.	64
9.4	Modulo relé 4 canales optoacoplado.	66
9.5	CONTROL RIEGO POR NEBULIZACIÓN.	68
9.5.1	Sensor FC-28.	69
9.5.1.1	Conexión con Arduino.	71
9.5.2	SISTEMA DE BOMBEO.	72

9.5.2.1	Bomba.	72
9.5.2.1.1	Características técnicas.	73
9.5.3	Sistema de Nebulización.....	73
9.5.3.1	Tuberías.	73
9.5.3.2	Boquillas de asiento.	74
9.5.3.3	Boquillas de nebulización.	74
9.5.4	VISUALIZACIÓN DE VARIABLES.	75
9.5.4.1	Pantalla LCD 16x2.	75
9.5.4.2	Pin Out adaptador I2C.....	76
9.5.4.3	Conexión con Arduino.	77
CAPÍTULO 10. INTEGRACIÓN INTERNET DE LAS COSAS		78
10.1	ESTABLECER COMUNICACIÓN DE ARDUINO A INTERNET.	79
10.1.2	ESP8266 ESP-01.	79
10.1.2.1	Pin Out.	80
10.1.2.2	Conexión con Arduino.....	81
10.1.2.3	Convertidor de voltaje.	82
10.1.2.3.1	Especificaciones técnicas conversor DC/DC.....	83
10.1.2.4	Programación Comandos AT.....	83
CAPÍTULO 11. SISTEMA DE ALIMENTACIÓN Y TABLERO DE CONTROL.		87
11.1	ALIMENTACIÓN SISTEMA DE CONTROL.	88
11.1.1	Potencia de suministro necesaria.	89
11.1.2	Corriente de suministro necesaria.	90
11.2	Tipo de fuente.....	91
11.2.1	Fuentes Lineales.....	91
11.2.2	Fuentes conmutadas.	92
11.3	ELECCIÓN FINAL FUENTE ALIMENTACIÓN.	93
11.3.1	Fuente conmutada montaje riel din.	93

11.4	Fuente conmutada montaje en placa.....	94
11.4.1	Datos técnicos.....	95
11.4.2	Cálculo factor de rizado.	95
11.5	TABLERO CONTROL.	96
11.6	GABINETE.	98
11.6.1	Protección IP.....	98
11.6.1.1	Nomenclatura IEC 60529.	98
11.6.2	Tamaño.....	101
11.7	PROTECCIONES.	103
11.8	CANALIZACIÓN.	106
11.8.1	Canalización interna.....	106
11.8.2	Canalización externa.	106
11.9	CONDUCTORES.	108
11.9.1	Dimensionamiento cables alimentación.....	108
11.9.2	Cable de control.	110
11.9.3	Cable instrumentación.....	111
	CONCLUSIONES.	113

ÍNDICE DE FIGURAS.

Capítulo 2.

Figura 2- 2 diagrama sistema de control lazo abierto. 10

Figura 2- 3 diagrama sistema control lazo cerrado. 11

Capítulo 3

Figura 3- 1 Microcontrolador ATmega, ATXmega y ATtiny. 14

Figura 3- 2 Microcontrolador AT90S2333. 14

Capítulo 4.

Figura 4- 1 Diagrama representativo IOT. 17

Capítulo 5.

Figura 5- 1 vías o cables de comunicación protocolo I2C..... 20

Figura 5- 2 Trama de datos protocolo de comunicación I2C.**iError! Marcador no definido.**

Capítulo 8.

Figura 8- 1 Placa de desarrollo PIC. 35

Figura 8- 2 Logo Arduino. 36

Figura 8- 3 logo RaspBerry Pi 38

Figura 8- 4 Arduino UNO R3. 42

Figura 8- 5 TCP/IP a OSI. 43

Figura 8- 6 HTTP Request. 44

Figura 8- 7 HTTP Response. 45

Figura 8- 8 Diagrama estructural Protocolo MQTT 46

Figura 8- 9 Estructura jerárquica protocolo MQTT 46

Capítulo 9.

Figura 9- 1 Sensor DHT22..... 56

Figura 9- 2 Dimensiones DHT22. 57

Figura 9- 3 PIN OUT DHT22 58

Figura 9- 4 Extractor axial. 63

Figura 9- 5 Módulo relés optoacoplados 66

Figura 9- 6 Pin-Out Módulo relé 4 canales 68

Figura 9- 7 Sensor de Humedad FC-28. 69

Figura 9- 8 Pin Out FC-28 70

Figura 9- 9 Conexión FC-28 con Arduino..... 71

Figura 9- 10 Bomba nebulizadora. 72

Figura 9- 11 Tuberías nebulización 74

Figura 9- 12 Boquillas de asiento nebulización..... 74

Figura 9- 13 Boquilla de nebulización 74

Figura 9- 14 Pantalla LCD 16x2 con módulo I2C. 75

Figura 9- 15 Pin-Out adaptador I2C para pantalla LCD 16x2. 76

Figura 9- 16 Conexión pantalla LCD con Arduino..... 77

Capítulo 10.

Figura 10- 1 Módulo WIFI ESP-01. 79

Figura 10- 2 Pin Out ESP-01..... 80

Figura 10- 3 Conexión módulo wi-fi ESP8266 ESP-01. 81

Figura 10- 4 Fuente de poder para protoboard 3.3 [V]-5 [V]. 81

Figura 10- 5 Convertidor DC/DC. 82

Capítulo 11.

Figura 11- 1 Fuentes lineales. 91

Figura 11- 2 Fuentes conmutadas. 92

Figura 11- 3 Fuente conmutada montaje en placa..... 94

Figura 11- 4 Módulo adaptador Arduino UNO R3 montaje Riel DIN. 96

Figura 11- 5 Módulo montado sobre Riel DIN. 97

Figura 11- 6 Separador roscado hexagonal M3. 97

Figura 11- 7 Canaleta ranurada. 106

Figura 11- 8 Conduit hermético 1". 107

Figura 11- 9 Cordón alimentación. 109

Figura 11- 10 Cable control..... 110

Figura 11- 11 Cordón multipar apantallado. 111

ÍNDICE DE TABLAS.

Capítulo 3.

Tabla 3- 1 Datos técnicos ATmega 328. 15

Capítulo 5.

Tabla 5- 1 Especificaciones protocolo I2C..... 21

Tabla 5- 2 Trama de datos protocolo de comunicación I2C. 21

Capítulo 7.

Tabla 7- 1 Condiciones climáticas óptimas para vegetales. 29

Tabla 7- 2 Temperaturas cultivo lechuga 30

Capítulo 8.

Tabla 8- 1 número de entradas físicas, análogas y discretas estimadas. 32

Tabla 8- 2 Entradas estimadas 33

Tabla 8- 3 Elección microcontrolador. 40

Tabla 8- 4 Ficha técnica Arduino UNO R3. 42

Capítulo 9.

Tabla 9- 1 Especificaciones técnicas sensor DHT22. 57

Tabla 9- 2 PinOut sensor DHT22. 58

Tabla 9- 3 Conexión DHT22 con Arduino. 58

Tabla 9- 4 Características módulo relé 4 canales..... 67

Tabla 9- 5 Características técnicas FC-28..... 70

Tabla 9- 6 Datos técnicos bomba nebulizadora. 73

Tabla 9- 7 Datos técnicos pantalla LCD 16x2. 76

Tabla 9- 8 Pin Out Adaptador I2C. 76

Capítulo 10.

Tabla 10- 1 Especificaciones Módulo WIFI ESP-01. 80

Tabla 10- 2 Datos técnicos conversor DC/DC..... 83

Tabla 10- 3 Comandos AT. 86

Capítulo 11.

Tabla 11- 1 Tabulación potencias. 89

Tabla 11- 2 Tabla consumo de corrientes. 90

Tabla 11- 3 Datos técnicos fuente conmutada montaje en placa 95

Tabla 11- 4 Datos técnicos módulo Arduino riel DIN. 97

Tabla 11- 5 Niveles protección contra sólidos. 99

Tabla 11- 6 Niveles protección contra líquidos. 101

Tabla 11- 7 Ficha técnica tablero control..... 102

Tabla 11- 8 Valor protecciones. 104

Tabla 11- 9 Conversión calibre AWG. 110

SIGLAS Y SIMBOLOGÍAS.

Siglas.

µC: Micro Controlador.

CPU: Unidad Central de Procesamiento.

E/S: Entrada / Salida.

AVR: Automatic voltage regulator (Regulador Automático de Voltaje).

DAC: Digital to Analog Conversion (Conversión digital a análogo).

USB: Universal Serial Bus.

LCD: Liquid cristal display (representación por cristal líquido).

RAM: Random Access Memory (Memoria de Acceso Aleatorio).

EEPROM: Electrically Erasable Programmable Read-Only Memory (ROM programable borrrable).

UART: Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal).

USART: Universal Synchronous and Asynchronous Receiver-Transmitter (Transmisor-Receptor Asincrónico y Síncrono Universal).

PWM: Pulse Width Modulation (Modulación por Ancho de Pulso).

SDA: Serial Data (Datos serial).

SCL: Serial Clock (Reloj serial).

IOT: Internet Of Things (Internet de las cosas).

I2C: Inter-Integrate Circuit (Circuito Inter-integrado).

ACK: Acknowledgement (Reconocimiento).

NACK: Negative Acknowledgement (Reconocimiento negativo).

FEM: Fuerza Electro Motriz.

ETC: Etcétera.

PSI: pounds-force per square inch (Libra por pulgada cuadrada).

PIC: Programmable Interrupt Controller (Controlador programable de interrupciones).

ICSP: In-Circuit Serial Programming (Programación Serial en Circuito).

LVP: Low Voltage Programming (Programación en Bajo Voltaje).

GPU: Graphics Processing Unit (Unidad de Procesamiento Gráfico).

GND: Ground (Tierra).

HDMI: High-Definition Multimedia Interface (Interfaz Multimedia de Alta Definición).

GPIO: General Purpose Input/Output (Entrada/Salida de propósito general).

AI: Analog Input (Entrada Análoga).

AO: Analog Output (Salida Análoga).

DI: Digital Input (Entrada Digital).

DO: Digital Output (Salida Digital).

IP: Internet Protocol (Protocolo de Internet).

TCP: Transmission Control Protocol (Protocolo de Control de Transmisión).

HTTP: Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto).

MQTT: Message Queuing Telemetry Transport (Mensaje de Telemetría en cola de espera).

TTL: Transistor-Transistor Logic (Lógica Transistor-Transistor).

IEC: International Electrotechnical Commission (Comisión Electrotécnica Internacional).

AWG: American Wire Gauge (Calibre de Cable Americano).

NCh: Norma Chilena.

ISO: International Organization for Standardization. (Organización Internacional de Normalización).

Simbología.

Ω : Ohmio.

K Ω : Kilo Ohmio.

V: Voltaje.

mV: Milivoltios.

AC:

DC:

A: Amperes.

mA: Miliamperes.

W: Watt.

mW: MiliWatt.

Kw: Kilowatt.

m: Metro.

m²: Metro cuadrado.

m³: Metro cúbico.

cm: Centímetro.

mm: Milímetro.

mm²: Milímetro cuadrado.

h: Hora.

s: segundo.

min: minuto.

Kcal: Kilo caloría.

L: Litro.

KB: Kilo Byte.

Kbps: Kilo bits por segundo.

Mbps: Mega bits por segundo.

°C: Grados Celsius.

Hz: Hertz.

INTRODUCCIÓN.

El siguiente trabajo tiene como finalidad la creación de un proyecto teórico de un sistema de control de regadío, humedad y temperatura adecuado para un recinto invernadero a escala, esto mediante un estudio de comparación y justificación de tecnologías y elementos correspondientes al área de automatización. Los cuales se determinan mediante criterios de robustez y costo. Para así crear un sistema pequeño y accesible monetariamente.

El proyecto se basa en un controlador tipo placa de desarrollo llamado Arduino el cual permite una amplia gama de sensores para la toma de los datos necesarios para el control.

Destacar que consta de elementos de hardware dejando la programación de software, el cual se asume como conocimiento previo, de quien implemente de manera física el proyecto.

Para una mayor comprensión y facilidad de instalación se deja disposición los esquemas de conexionado y la distribución de los componentes dentro del tablero a instalar.

DESCRIPCIÓN DEL PROBLEMA:

Falta de un sistema eficiente, y con aprovechamiento hídrico óptimo, para el proceso de germinación y crecimiento de diferentes tipos de lechugas.

OBJETIVO GENERAL.

Diseñar un sistema automatizado para el control de temperatura humedad y riego de alta eficiencia del recurso hídrico.

OBJETIVOS ESPECÍFICOS.

Con el fin de lograr estos objetivos, se realizarán las siguientes actividades:

- ✓ Diseño teórico del sistema.
- ✓ Diseño de sistema de control automatizado de las variables (controlador, sensores y actuadores).
- ✓ Análisis de posibles mejoras mediante del denominado "Concepto del internet de cosas".
- ✓ Determinar costo total del proyecto.

ALCANCES.

- ✓ Se busca definir un sistema prototipo, en base a equipos disponibles en el mercado, el cual cumpla con una eficiencia hídrica óptima.
- ✓ Se busca establecer el costo del sistema y con una robustez física y tecnológica acorde.
- ✓ Proponer mejora a través de plataforma compatible con IOT.
- ✓ Estudio escrito.

METODOLOGÍA.

Metodología mediante desarrollo de proyecto, y por estudio comparado.

CAPÍTULO 1 . TEORÍA Y TIPOS DE INVERNADEROS.

1.1 TEORÍA DE LOS INVERNADEROS.

Un invernadero o invernáculo, es un lugar cerrado, estático y accesible a pie, que se destina a la producción de cultivos, dotado habitualmente de una cubierta exterior translúcida de vidrio o plástico, que permite el control de la temperatura, la humedad y otros factores ambientales para favorecer el desarrollo de las plantas.

El invernadero, aprovecha el efecto producido por la radiación solar y que, al atravesar un vidrio u otro material translucido, calienta los objetos que hay adentro; estos, a su vez, emiten radiación infrarroja, con una longitud de onda mayor que la solar, por lo cual no pueden atravesar los vidrios a su regreso quedando atrapados y produciendo el calentamiento.

El cristal o plástico usado para un invernadero trabaja como medio selectivo de la transmisión para diversas frecuencias espectrales, y su efecto es atrapar energía dentro del invernadero, que calienta el ambiente interior. También sirve para evitar la pérdida de calor por convección. Esto puede ser demostrado abriendo una ventana pequeña cerca de la azotea de un invernadero: la temperatura cae considerablemente. Este principio es la base del sistema de enfriamiento automático auto ventilación.

1.1.1 Tipos de invernaderos.

Pueden clasificarse según diferentes criterios por ej. Materiales para la construcción, tipo de material de cobertura, características de la techumbre, etc. no obstante, se prefiere enumerar los más importantes obviando algunas características para su clasificación. Dentro de los tipos de invernaderos más comunes, se encuentran:

- ✓ Invernadero Tipo Túnel.
- ✓ Invernadero Tipo Capilla o Dos Aguas.
- ✓ Invernadero Tipo Dientes de Sierra.
- ✓ Invernadero Tipo Capilla Modificado.
- ✓ Invernadero con Techumbre Curva.
- ✓ Invernadero Tipo Venlo

CAPÍTULO 2. TEORÍA DEL CONTROL AUTOMÁTICO.

2.1 DEFINICIÓN.

El control automático es el mantenimiento de un valor deseado dentro de una cantidad o condición, midiendo el valor existente, comparándolo con el valor deseado, y utilizando la diferencia para proceder a reducirla. En consecuencia, el control automático exige un lazo cerrado de acción y reacción que funcione sin intervención humana.

El concepto de la realimentación no es nuevo, el primer lazo de realimentación fue usado en 1774 por James Watt para el control de la velocidad de cualquier máquina de vapor. A pesar de conocerse el concepto del funcionamiento, los lazos se desarrollaron lentamente hasta que los primeros sistemas de transmisión neumática comenzaron a volverse comunes en los años 1940, los años pasados han visto un extenso estudio y desarrollo en la teoría y aplicación de los lazos realimentados de Control.

En la actualidad los lazos de control son un elemento esencial para la manufactura económica y prospera de virtualmente cualquier producto, desde el acero hasta los productos alimenticios. A pesar de todo, este lazo de control que es tan importante para la industria está basado en algunos principios fácilmente entendibles y fáciles. Este artículo trata este lazo de control, sus elementos básicos, y los principios básicos de su aplicación.

2.3 FASES DE ESTUDIO EN LA ELABORACIÓN DE AUTOMATISMO.

Para el desarrollo y elaboración correcta de un automatismo, es necesario conocer Previamente los datos siguientes:

- a) Las especificaciones técnicas del sistema o proceso a automatizar y su correcta interpretación.
- b) La parte económica asignada para no caer en el error de elaborar una buena opción desde el punto de vista técnico, pero inviable económicamente.
- c) Los materiales, aparatos etc., existentes en el mercado que se van a utilizar para diseñar el automatismo. En este apartado es importante conocer también:
 - ✓ Calidad de la información técnica de los equipos.
 - ✓ Disponibilidad y rapidez en cuanto a recambio y asistencia pública.

2.4 CLASIFICACIÓN DE LOS SISTEMAS DE CONTROL.

2.4.1 Sistema de control de lazo abierto.

Es aquel sistema en que solo actúa el proceso sobre la señal de entrada y da como resultado una señal de salida independiente a la señal de entrada, pero basada en la primera. Esto significa que no hay retroalimentación hacia el controlador para que este pueda ajustar la acción de control. Es decir, la señal de salida no se convierte en señal de entrada para el controlador. La Figura 2-2, Representa de mejor manera este sistema.

Estos sistemas se caracterizan por:

- ✓ Ser sencillos y de fácil concepto.
- ✓ NO asegura su estabilidad ante una perturbación.
- ✓ La salida no se compara con la entrada.
- ✓ Ser afectado por las perturbaciones. Estas pueden ser tangibles o intangibles.
- ✓ La precisión depende de la previa calibración del sistema.

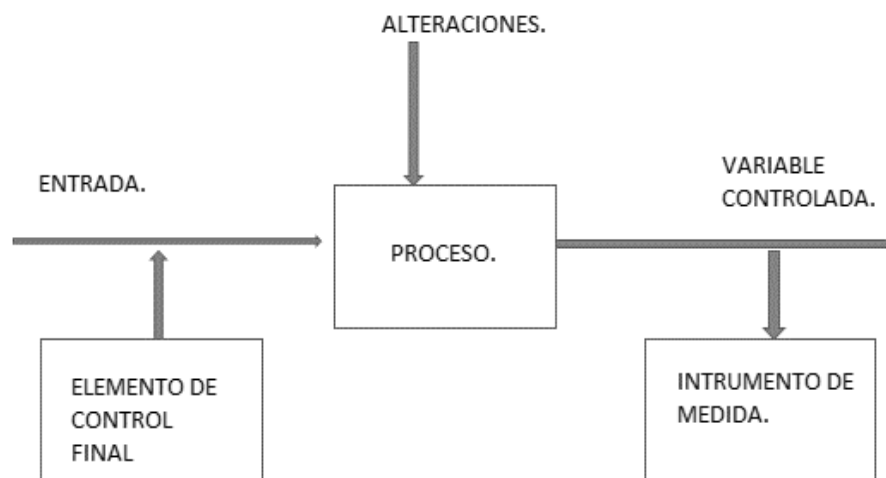


Figura 2- 1 diagrama sistema de control lazo abierto.

2.4.2 Sistema de control de lazo cerrado.

Son los sistemas en los que la acción de control está en función de la señal de salida. Los sistemas de circuito cerrado usan la retroalimentación desde un resultado final para ajustar la acción de control en consecuencia, tal como se muestra la Figura 2-4.

El control en lazo cerrado es imprescindible cuando se da alguna de las siguientes circunstancias:

- ✓ Cuando un proceso no es posible de regular por el hombre.
- ✓ Una producción a gran escala que exige grandes instalaciones y el hombre no es capaz de manejar.
- ✓ Vigilar un proceso es especialmente difícil en algunos casos y requiere una atención que el hombre puede perder fácilmente por cansancio o despiste, con los consiguientes riesgos que ello pueda ocasionar al trabajador y al proceso.

Sus características son:

- ✓ Ser complejos, pero amplios en cantidad de parámetros.
- ✓ La salida se compara con la entrada y le afecta para el control del sistema.
- ✓ Su propiedad de retroalimentación.
- ✓ Ser más estable a perturbaciones y variaciones internas.

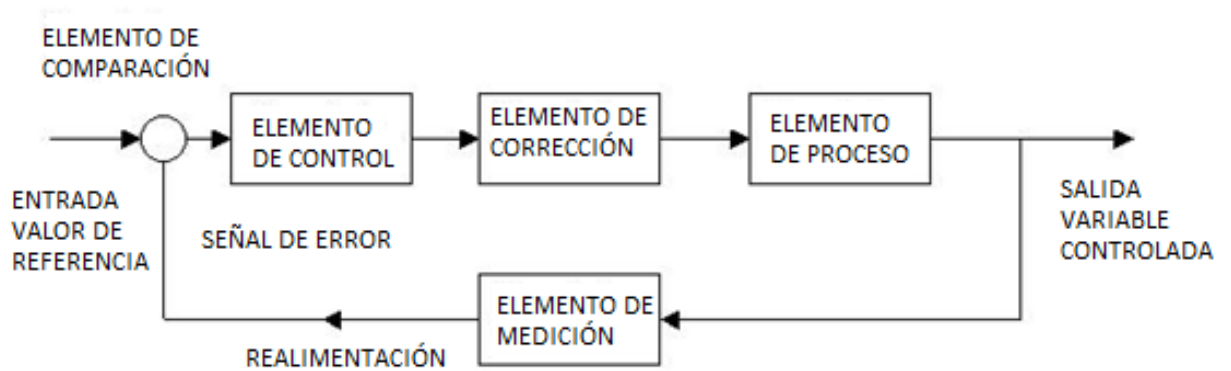


Figura 2- 2 diagrama sistema control lazo cerrado.

CAPÍTULO 3. MICROCONTROLADORES.

3.1 ¿Qué es un microcontrolador?

Un microcontrolador (abreviado μC , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida.

3.1.1 Arquitectura.

1. **CPU (Unidad Central de Proceso):** es la parte encargada de ejecutar cada instrucción y de controlar que dicha ejecución se realice correctamente. Normalmente, estas instrucciones hacen uso de datos disponibles previamente como lo son los datos de entrada, y generan como resultado otros datos diferentes como lo es los datos de salida, que podrán ser utilizados (o no) por la siguiente instrucción.
2. **Diferentes tipos de memorias:** son en general las encargadas de alojar tanto las instrucciones como los diferentes datos que estas necesitan. De esta manera posibilitan que toda esta información (instrucciones y datos) esté siempre disponible para que la CPU pueda acceder y trabajar con ella en cualquier momento. Generalmente se encontrarán dos tipos de memorias: las que su contenido se almacena de forma permanente incluso tras cortes de alimentación eléctrica (llamadas persistentes), y las que su contenido se pierde al dejar de recibir alimentación (llamadas volátiles). Según las características de la información a guardar, esta se grabará en un tipo u otro de memoria de forma automática.
3. **Entradas y salidas (E / S):** son las encargadas de comunicar el microcontrolador con el exterior. En los pines de entrada del microcontrolador se puede conectar sensores para que este pueda recibir datos provenientes de su entorno, y en sus pines de salida se conectarán actuadores para que el microcontrolador pueda enviarles órdenes y así interactuar con el medio físico. De todas formas, muchos pines de la mayoría de microcontroladores no son exclusivamente de entrada o de salida, sino que pueden ser utilizados indistintamente para ambos propósitos (de ahí el nombre de E/S)

3.2. Microcontroladores AVR.

Los AVR son una familia de microcontroladores RISC del fabricante estadounidense Atmel. La arquitectura de los AVR fue concebida por dos estudiantes en el Norwegian Institute of Technology, y posteriormente refinada y desarrollada en Atmel Norway, la empresa subsidiaria de Atmel, fundada por los dos arquitectos del chip.

3.2.2 Grupos.

Cuenta con bastantes aficionados debido a su diseño simple y la facilidad de programación. Se pueden dividir en los siguientes grupos:

- **ATxmega:** procesadores muy potentes con 16 a 384 kB de memoria flash programable, encapsulados de 44, 64 y 100 pines (A4, A3, A1), capacidad de DMA, eventos, criptografía y amplio conjunto de periféricos con DACs.
- **ATmega:** microcontroladores AVR grandes con 4 a 256 kB de memoria flash programable, encapsulados de 28 a 100 pines, conjunto de instrucciones extendido (multiplicación y direccionamiento de programas mayores) y amplio conjunto de periféricos.
- **ATtiny:** pequeños microcontroladores AVR con 0,5 a 8 kB de memoria flash programable, encapsulados de 6 a 20 pines y un limitado set de periféricos.
- **AT90USB:** ATmega integrado con controlador USB
- **AT90CAN:** ATmega con controlador de BUS CAN.
- Tipos especiales: algunos modelos especiales, por ejemplo, para el control de los cargadores de baterías, pantallas LCD y los controles de los motores o la iluminación.
- **AT90S:** tipos obsoletos, los AVR clásicos.



Figura 3- 2 Microcontrolador AT90S2333.



Figura 3- 1 Microcontrolador ATmega, ATxmega y ATtiny.

Unos de los microcontroladores más conocidos es el ATmega 328, el cual es encontrado en una de las placas de desarrollo más usadas "ARDUINO UNO".
 Dentro de las características principales que posee este microcontrolador son:

DATOS TÉCNICOS.	
Tipo de memoria de programa.	Flash.
Memoria de programa (KB).	32.
Velocidad CPU (MIPS).	20.
Bytes RAM	2,048.
Datos EEPROM	1024.
Periféricos de comunicación digital	1-UART,2-SPI,1-I2C.
Periféricos comparador/capturado/PWM.	1 Input capture, 1CCP, 6PWM.
Timers	2 x 8-bit, 1 x 16-bit.
Comparadores	1.
Rango de temperatura	-40 a 80°C.
Rango voltaje de operación	1.8 a 5.5.
Numero de pines	32.

Tabla 3- 1 Datos técnicos ATmega 328.

CAPÍTULO 4. INTERNET DE LAS COSAS.

4.1 Definición e historia.

es un concepto que se refiere a la interconexión digital de objetos cotidianos con Internet. Alternativamente, Internet de las cosas es la conexión de Internet con más "cosas u objetos" que personas. También se suele conocer como Internet de todas las cosas o Internet en las cosas.

El Internet de las cosas debería codificar de 50 a 100000 mil millones de objetos y seguir el movimiento de estos; puesto que se calcula que todo ser humano está rodeado de por lo menos de 1000 a 5000 objetos. Según la empresa Gartner, en 2020 habrá en el mundo aproximadamente 26 mil millones de dispositivos con un sistema de conexión a Internet de las cosas.

La empresa estadounidense Cisco, que está detrás de la iniciativa del Internet de las cosas, ha creado un "contador de conexiones" dinámico que le permite estimar el número de "cosas" conectadas desde julio de 2013 hasta el 2020. El concepto de que los dispositivos se conectan a la red a través de señales de radio de baja potencia es el campo de estudio más activo del Internet de las cosas. Este hecho se explica porque las señales de este tipo no necesitan ni Wi-Fi ni Bluetooth. Sin embargo, se están investigando distintas alternativas que necesitan menos energía y que resultan más económicas, bajo el nombre de "Chip Networks".



Figura 4- 1 Diagrama representativo IOT.

4.2 Aplicaciones de consumo.

Un porcentaje creciente de los dispositivos son creados para el consumo. Algunos ejemplos de aplicaciones de consumo incluyen automóviles conectados, entretenimiento, automatización del hogar, tecnología vestible, salud conectada y electrodomésticos como lavadoras, secadoras, aspiradoras robóticas, purificadores de aire, hornos, refrigeradores que utilizan Wi-Fi para monitoreo remoto.

CAPÍTULO 5. PROTOCOLO DE COMUNICACIÓN I2C.

5.1 Historia protocolo I2C.

I2C significa Circuito Interintegrado (Por sus siglas en Inglés Inter-Integrated Circuit) es un protocolo de comunicación serial desarrollado por Phillips Semiconductors allá por la década de los 80s. Básicamente se creó para poder comunicar varios chips al mismo tiempo dentro de los televisores.

El protocolo I2C toma e integra lo mejor de los protocolos SPI y UART. Con el protocolo I2C podemos tener a varios maestros controlando uno o múltiples esclavos. Esto puede ser de gran ayuda cuando se van a utilizar varios microcontroladores para almacenar un registro de datos hacia una sola memoria o cuando se va a mostrar información en una sola pantalla.

El protocolo utiliza solo dos vías o cables de comunicación.

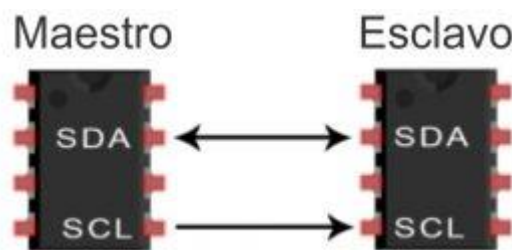


Figura 5- 1 vías o cables de comunicación protocolo I2C.

- SDA – Serial Data. Es la vía de comunicación entre el maestro y el esclavo para enviarse información.
- SCL – Serial Clock. Es la vía por donde viaja la señal de reloj.

5.2 I2C protocolo de comunicación serial.

el protocolo I2C envía información a través de una sola vía de comunicación. La información es enviada bit por bit de forma coordinada, por ende, transmite mediante información de forma serial.

A continuación, se presenta una tabla con especificaciones técnicas del protocolo.

Número de vías o cables	2
Velocidad máxima	Modo estándar (Sm) = 100Kbps
	Modo rápido (Fm) = 400Kbps.
	Modo High Speed (Fm+ `) = 3.4 Mbps
	Modo Ultra Fast (Hs-Mode) = 5Mbps
Síncrono o Asíncrono	Síncrono
Paralelo o Serial	Serial
Número máximo de maestros	Ilimitado
Número máximos de esclavos	1008

Tabla 5- 1 Especificaciones protocolo I2C.

5.3 ¿Cómo funciona el protocolo I2C?

Con el protocolo I2C la información viaja en mensajes. Los mensajes van divididos en tramas de datos. Cada mensaje lleva una trama con una dirección la cuál transporta la dirección binaria del esclavo al que va dirigido el mensaje, y una o más tramas que llevan la información del mensaje. También el mensaje contiene condiciones de inicio y paro, lectura y escritura de bits, y los bits ACK y NACK. Todo esto va entre cada sección de datos.

Mensaje								
Start	7 o 10 Bits	Bit para Leer/ Escribir	Bit para reconocer ACK/ NACK	8 Bits	Bit para reconocer ACK/ NACK	8 Bits	Bit para reconocer ACK/ NACK	Stop
Condición de inicio	Sección destinada para la dirección			Sección 1 para transportar información		Sección 2 para transportar información		Condición de paro

Tabla 5- 2 Trama de datos protocolo de comunicación I2C.

5.3.1 Descripción de trama de datos.

Condición de Inicio – Start: La vía SDA cambia de un nivel de voltaje Alto a un nivel de voltaje Bajo, antes de que el canal SCL cambie de Alto a nivel Bajo.

Condición de Paro – Stop: La vía SDA ahora cambia de un nivel de voltaje Bajo a Alto, después de que la vía SCL cambia de Bajo a Alto.

Trama de Dirección – Address Frame: Es una secuencia única que va de los 7 a los 10 bits. Esta sección (Frame) se envía a cada Esclavo, y va a identificar al Esclavo con el que el Maestro se quiere comunicar.

Bit para Lectura/Escritura A – Read/Write Bit A: Es un bit de información enviado a los Esclavos. Por medio de este bit el Maestro indica si le va enviar información al Esclavo (Nivel Bajo de voltaje = Escritura), o si el Maestro quiere solicitarle información al Esclavo (Nivel Alto de Voltaje = Lectura).

Bit ACK/NACK: Después de cada sección (Frame) de información enviada en un mensaje, podemos notar que lleva un bit acknowledge/no-acknowledge (reconocido/no-reconocido). Esto ayuda a identificar si la información fue enviada correctamente. En seguida de que se envía un Frame, si este fue recibido con éxito, se retorna un bit ACK al remitente. Si la información no fue recibida con éxito, se retorna un bit NACK.

5.4 Conceptos del I2C.

Dirección de envío – Address.

El protocolo I2C no cuenta con una línea de selección de esclavos (Como lo hace el protocolo SPI), así que se debe establecer una forma de notificación al Esclavo, para que éste se prepare para recibir información del Maestro. Mediante la sección de Dirección (Address) en el mensaje es como se notifica al Esclavo, es por eso que va en seguida del Bit de Inicio (Start).

Así que esta es la secuencia que se sigue para el direccionamiento de esclavos.

1.- El Maestro envía la dirección del Esclavo con el que quiere comunicarse, esta dirección se envía a todos los Esclavos que estén conectados 2.- Cada Esclavo recibe la dirección, y la compara con su propia dirección.3.- Si la dirección coincide con el Esclavo, en seguida el Esclavo envía un bit ACK con nivel de voltaje Bajo, de regreso al Maestro.4.- Si la dirección no coincide con el Esclavo, simplemente no se hace nada, y la vía SDA permanecerá en nivel de voltaje Alto.

Bit de Lectura/Escritura – Read/Write Bit.

Después de la dirección en el mensaje, se envía un bit, notificando al Esclavo si el Maestro quiere escribir información o leer información de él. Si el Maestro quiere enviarle información al Esclavo (Escribir), entonces manda un Bajo. Si el Maestro quiere solicitar información al Esclavo (Leer), el bit Read/Write será de nivel Alto.

Frame de información – Data Frame.

Después de que el Maestro recibe el bit ACK del Esclavo, la primera sección o Frame de información está lista para ser enviada al Esclavo.

El Frame para transportar información siempre es de 8 bits y se envía primero el bit más significativo. Después de enviar un Frame de información, inmediatamente se envía un bit ACK/NACK, para comprobar que el Frame de información efectivamente se ha llegado a su destino y se ha recibido satisfactoriamente. Independientemente de quién envía el Frame de información, si el Esclavo o el Maestro, siempre se debe enviar en seguida el bit ACK/NACK, antes de que se envíe el segundo Frame de información.

Una vez que se envían ambos Frames de información, el Maestro puede activar una condición de paro (Stop), y con esto se va a detener la transmisión. La condición de paro (Stop) es un cambio de voltaje de nivel Bajo a nivel Alto, en la vía SDA, y se activa cuando en la vía SCL se pasa de un nivel Bajo a un nivel Alto (La vía SCL, posteriormente, permanecerá en nivel Alto).

**CAPÍTULO 6. TEMPERATURA, HUMEDAD RELATIVA; MÉTODOS Y
PRINCIPIOS DE MEDICIÓN.**

6.1 TEMPERATURA.

6.1.1 Definición de variable.

La temperatura es una magnitud referida a las nociones comunes de calor medible mediante un termómetro. En física, se define como una magnitud escalar relacionada con la energía interna de un sistema termodinámico.

6.1.2 Métodos y principios de medición.

La medición de la temperatura puede realizarse mediante diferentes instrumentos basados en diferentes principios de operación:

- Termómetros de vidrio: el funcionamiento de estos termómetros está basado en la expansión de un líquido en función de la temperatura. Los más populares son los termómetros de mercurio, los cuales operan en un intervalo de temperatura entre -38°C y 540 °C. Para temperaturas más bajas se suelen usar termómetros de alcohol (-62°C), pentano (-200°C) y una mezcla de propano (-217°C). La precisión de estos instrumentos depende de su calidad y del rango de temperatura.

- Termómetros de presión: estos termómetros están basados en la expansión que sufre un fluido por efecto de la temperatura. Si el fluido está confinado (tubo de Bourdon o tubo capilar), el aumento de la temperatura generará un aumento de la presión.

- Pirómetros: Los pirómetros son sensores de temperatura que detectan la radiación electromagnética emitida por los cuerpos. El principio de funcionamiento de estos dispositivos se basa en que un cuerpo emite radiación electromagnética con una intensidad que depende de su temperatura. Los pirómetros se dividen en dos clases:
a) Pirómetros ópticos: para su implementación es necesario que el cuerpo emita en el espectro infrarrojo de longitud de onda. El rango de aplicación es entre 750°C y 2900°C.

- b) Pirómetros de radiación: son instrumentos que miden la tasa de emisión de energía por unidad de área en un intervalo de longitud de onda relativamente grande. Los pirómetros de radiación se diferencian por el tipo de detector, los cuales pueden ser térmicos y de fotones.

- Termopares: Un circuito cerrado, formado por dos hilos de metales diferentes, circula una corriente cuando existe una diferencia de temperatura entre la junta caliente y la referencia fría. Este fenómeno es denominado el efecto Seebeck. El

principio de operación de los termopares se fundamenta en la generación y medición de una fuerza electromotriz (FEM) cuando dos metales de diferente composición unidos por un extremo experimentan un gradiente de temperatura.

- Termistores: Un termistor es un resistor cuya resistencia eléctrica varía considerablemente con la temperatura. Los termistores están hechos de materiales semiconductores sinterizados, principalmente óxidos metálicos. Los termistores son de los sensores térmicos más precisos ($\pm 0.1^{\circ}\text{C}$; $\pm 0.2^{\circ}\text{C}$), sin embargo, el intervalo de temperatura de estos dispositivos es más limitado que el de los termopares ($0 - 100^{\circ}\text{C}$).

6.2 HUMEDAD RELATIVA.

6.2.1 Definición.

La humedad del aire se debe al vapor de agua que se encuentra presente en la atmósfera. La cantidad de vapor de agua que puede absorber el aire depende de su temperatura. El aire caliente admite más vapor de agua que el aire frío. El vapor de agua tiene una densidad menor que la del aire, por tanto, el aire húmedo (mezcla de aire y vapor de agua) es menos denso que el aire seco. Por otra parte, las sustancias al calentarse dilatan, lo que les confiere menor densidad

6.2.2 MÉTODOS Y PRINCIPIOS DE MEDICIÓN.

6.2.2.1 Higrómetro.

6.2.2.1.1 Mecánico.

Los higrómetros mecánicos usan la expansión y contracción de materiales orgánicos con los cambios de humedad. El elemento sensible puede ser cabello, un elemento textil o plástico. Los cambios de longitud se amplifican mediante la acción de palancas para mover la aguja en un dial, o un lápiz que registra un diagrama. Principalmente adecuado para condiciones ambientales normales. Incertidumbre de la medición es de alrededor de $\pm 5\%$ a $\pm 15\%$ de la humedad relativa

6.2.2.1.2 Bulbo húmedo y seco (sigrómetro).

6.2.2.2 Impedancia eléctrica.

6.2.2.2.1 Sensores capacitivos.

Responden más a la humedad relativa que al punto de rocío, con una buena linealidad a bajas humedades relativas. En general, los sensores capacitivos no resultan dañados por la condensación, aunque puede cambiar la calibración como resultado.

6.2.2.2.2 Sensores resistivos.

Responden más a la humedad relativa que al punto de rocío. La linealidad de los sensores resistivos es mejor a altas humedades. La mayoría de los sensores resistivos no pueden tolerar la condensación. Sin embargo, algunos están “protegidos contra la saturación”, mediante calentamiento automático para evitar la condensación.

CAPÍTULO 7. ELECCIÓN DE HORTALIZA A CULTIVAR.

7.1 ELECCIÓN DE HORTALIZA.

Dicha elección se realizó en base a un estudio comparado de cuatro de estas, siendo el Tomate, Lechuga, Apio y Brócoli. Se compararon sus requerimientos Edafoclimáticos (temperatura, Humedad, Luminosidad, suelo y fertilización) Pero solo teniendo en cuenta los dos parámetros fundamentales al aplicar el regadío por nebulización, los cuales son: Temperatura y Humedad relativa del ambiente. Ya que las principales características de nebulizar es reducir la temperatura ambiental y aumentar la humedad relativa del aire al interior del invernadero, entonces es necesario que las necesidades en el proceso de germinación, crecimiento y floración de la hortaliza sean acordes a lo entregado por el proceso.

A continuación, se presenta una tabla comparativa. Tabla 3-1. Donde se muestran los parámetros de Tº y Humedad necesarios para su desarrollo.

Hortaliza.	Temperaturas.	Humedad Relativa.
Tomate.	Día: 20ºC a 30ºC Noche: 1ºC a 17ºC	60-80%
Lechuga.	Germinación:18ºC a 20ºC Crecimiento: Día 14ºC a 18ºC. Noche: 5ºC a 8ºC Acogollado: Día: 12ºC. Noche: 3ºC a 5ºC.	60-80%
Apio.	Germinación: 17ºC a 20ºC. Crecimiento: 16ºC a 20ºC.	60-80%
Brócoli.	Crecimiento: 20ºC a 24ºC Floración. 10ºC a 15ºC.	60-75%

Tabla 7- 1 Condiciones climáticas óptimas para vegetales.

Como resultado de elección final se concluye la elección de la **“Lechuga”** como hortaliza a cultivar dentro del sistema invernadero.

Por los siguientes criterios:

- ✓ **Dentro del parámetro temperatura, no se encuentra un requerimiento de una elevada temperatura, por ende, el sistema de nebulización logra mantener los rangos requerido.**
- ✓ **Humedad relativa del aire elevada, por ende, posible control sobre esta.**

7.2. Requerimientos Edafoclimáticos.

7.2.1. Temperatura.

A continuación, se presenta una tabla con las temperaturas correspondientes al proceso de cultivo de la lechuga.

Tabla 4-1.

Proceso.	Tº
Germinación.	18-20ºC
Crecimiento.	14-18ºC Día / 5-8ºC Noche.
Acogollado	12ºC Día / 3-5ºC Noche.

Tabla 7- 2 Temperaturas cultivo lechuga

7.2.2 Humedad relativa.

El sistema radicular de la lechuga es muy reducido en comparación con la parte aérea por lo que es muy sensible a la falta de humedad. La Humedad relativa conveniente para la lechuga es de 60 a 80%.

7.2.3 Riego.

Los riegos se darán de manera frecuente y con poca cantidad de agua, procurando que el suelo este levemente seco en la superficie, para evitar podredumbres en el cuello y de la vegetación que toma contacto con el suelo.

CAPÍTULO 8. ELECCIÓN DEL MICROCONTROLADOR.

8.1 ENTRADAS Y SALIDAS ESTIMADAS.

La automatización del proyecto tiene como finalidad el control de variables y a si mismo parámetros, para obtener una producción ideal, dichas variables corresponden a Temperatura, Humedad relativa, Humedad de suelo y pH.

A raíz de esto se debe saber el número de entradas y salidas tanto análogas como discretas, correspondiente a sensores, etc. A continuación, se presenta un estimado de entradas y salidas posibles dentro del proyecto.

8.1.1 Tabla de entradas estimadas.

Se presenta el registro de las distintas entradas físicas que posee el sistema a través de la tabla 6-1. Junto a los detalles de estas.

Entradas.	Función.	Comentario.	Tipo de señal.
1	Sensor	Sensor temperatura	Análoga.
2	Sensor	Sensor humedad relativa.	Análoga.
3	sensor	Sensor humedad suelo.	Análoga.

Tabla 8- 1 número de entradas físicas, análogas y discretas estimadas.

8.1.2 Tabla de salidas estimadas.

Salidas.	Función.	Comentario.	Tipo de señal.
1	Extractor.	extractor interno salida a relé.	Discreta.
2	Bomba regadío.	Salida a relé	Discreta.
3	Calefactor	Salida a relé	Discreta.

Tabla 8- 2 Entradas estimadas

8.2 ELECCIÓN DEL MICROCONTROLADOR.

La elección del microcontrolador se realiza en base a criterios como de la cantidad de entradas y salidas del sistema, la robustez tanto física como tecnológica del proyecto y la economía monetaria que debe presentar.

Dado que no es un proyecto a “gran escala” la elección del controlador se enfocará dentro de un rango muy acotado en la gama de microcontroladores que existen en el mercado.

Los cuáles serán:

- Microcontroladores PIC.
- Microcontroladores o placas de desarrollo ARDUINO.
- Microcontroladores o placas de desarrollo RASPBERRY Pi.

8.2.1 MICROCONTROLADORES PIC.

8.2.1.1 Estructura central.

La arquitectura del PIC es sumamente minimalista. Está caracterizada por las siguientes prestaciones:

- Área de código y de datos separadas (Arquitectura_Harvard).
- Un reducido número de instrucciones de longitud fija.
- Implementa segmentación de tal modo que la mayoría de instrucciones duran 1 tiempo de instrucción (4 tiempos de reloj). Puede haber instrucciones de dos tiempos de instrucción (saltos, llamadas y retornos de subrutinas y otras) o inclusive con más tiempo de instrucción en PIC de gama alta. Esto implica que

el rendimiento real de instrucciones por segundo del procesador es de al menos 1/4 de la frecuencia del oscilador.

- Un solo acumulador (W), cuyo uso (como operador de origen) es implícito (no está especificado en la instrucción).
- Todas las posiciones de la RAM funcionan como registros de origen y/o de destino de operaciones matemáticas y otras funciones.
- Una pila de hardware para almacenar instrucciones de regreso de funciones.
- Una relativamente pequeña cantidad de espacio de datos direccionable (típicamente, 256 bytes), extensible a través de manipulación de bancos de simplemente, registros.
- memoria.
- El espacio de datos está relacionado con el CPU, puertos, y los registros de los periféricos.
- El contador de programa está también relacionado dentro del espacio de datos, y es posible escribir en él (permitiendo saltos indirectos).
- A diferencia de la mayoría de otros CPU, no hay distinción entre los espacios de memoria y los espacios de registros, ya que la RAM cumple ambas funciones, y esta es normalmente referida como "archivo de registros"

8.2.1.2 Programación.

Para transferir el código de un ordenador al PIC normalmente se usa un dispositivo llamado programador. La mayoría de PIC que Microchip distribuye hoy en día incorporan ICSP (*In Circuit Serial Programming*, programación serie incorporada) o LVP (*Low Voltage Programming*, programación a bajo voltaje), lo que permite programar el PIC directamente en el circuito destino. Para la ICSP se usan los pines RB6 y RB7 (En algunos modelos pueden usarse otros pines como el GP0 y GP1 o el RA0 y RA1) como reloj y datos y el MCLR para activar el modo programación aplicando un voltaje de 13 voltios. Existen muchos programadores de PIC, desde los más simples que dejan al software los detalles de comunicaciones, a los más complejos, que pueden verificar el dispositivo a diversas tensiones de alimentación e implementan en hardware casi todas las funcionalidades. Muchos de estos programadores complejos incluyen ellos mismos PIC preprogramados como interfaz para enviar las órdenes al PIC que se desea programar. Uno de los programadores más simples es el TE20, que utiliza la línea TX del puerto RS-232 como alimentación y las líneas DTR y CTS para mandar o recibir datos cuando el microcontrolador está en modo programación. El software de programación puede ser el ICprog.

8.2.1.3 Placa de desarrollo para PIC.

Generalmente la placa es independiente al chip microcontrolador, por lo cual se puede encontrar una gran variedad de tarjetas, las cuales también varían en características tales como E/S según el PIC, tipo de comunicación, y conexiones externas a sensores. etc.



Figura 8- 1 Placa de desarrollo PIC.

8.2.1.4 Costo.

A pesar de ser una tecnología relativamente “antigua” su precio dentro de la gama de las placas de desarrollo es intermedio, llegando a un promedio de 40 dólares, unos 23 mil pesos, todo esto dependiendo del tipo de PIC.

8.2.1.5 Compatibilidad con módulos externos (sensores, actuadores, relés etc.)

En este punto se busca que el microcontrolador tenga una compatibilidad optima con el proceso que se busca implementar, cuyo objetivo es el control mediante sensores y actuadores.

Dentro del mercado se encuentra una gran variedad de elementos de sensado referentes al proyecto, los cuales son 100% compatibles con el microcontrolador. Entregando una buena opción de uso de este.

8.2.2 ARDUINO.



Figura 8- 2 Logo Arduino.

Arduino es una plataforma de prototipos electrónica de código abierto (open-source) basad en hardware y software flexibles y fáciles de usar. Arduino puede “sentir” el entorno mediante la recepción de entradas desde una variedad de sensores y puede afectar a su alrededor mediante el control de luces, motores y otros artefactos. El microcontrolador de la placa se programa usando el *Arduino Programming Language* (basado en Wiring) y el *Arduino Development Environment* (basado en Processing). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador (por ejemplo, con *Flash*, *Processing*, *MaxMSP*, etc.). Arduino es posible conectarlo a internet mediante módulo de conexión Wi-Fi.

8.2.2.1 Código abierto y hardware extensible.

El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo.

8.2.2.2 Programación.

El entorno de programación de Arduino es fácil de usar para principiantes, pero su ciertamente flexible para que usuarios avanzados puedan aprovecharlo también. Está convenientemente basado en el entorno de programación Processing.

8.2.2.3 Código abierto y software extensible.

El software Arduino está publicado como herramientas de código abierto, disponible para extensión por programadores experimentados. El lenguaje puede ser expandido mediante librerías C++, y la gente que quiera entender los detalles técnicos pueden hacer el salto desde Arduino a la programación en lenguaje AVR C en el cual está basado. De forma similar.

8.2.2.4 Costo.

Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano.

El promedio de precio dentro del mercado es 15 mil pesos, dependiendo de la tipo, modelo y robustez del Arduino.

8.2.2.5 Multiplataforma.

El software de Arduino se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. La mayoría de los sistemas microcontroladores están limitados a Windows.

8.2.3 RASPBERRY PI.



Figura 8- 3 logo RaspBerry Pi

Más que una placa de desarrollo Raspberry es un computador de placa reducida de bajo costo, es decir, posee un sistema operativo predeterminado cargado en su CPU denominada Raspbian, aunque permite usar otros sistemas operativos, incluido una versión de Windows 10. En todas sus versiones incluye un procesador Broadcom, una memoria RAM, una GPU, puertos USB, HDMI, Ethernet (El primer modelo no lo tenía), 40 pines GPIO. Actualmente con conexión a internet inalámbrica.

8.2.3.1 Hardware.

La placa es relativamente pequeña, para tratarse de una "computadora", cuenta con múltiples conexiones desde salida de video HDMI, USB y las versiones más antiguas ETHERNET.

Una de las características que más resalta es que posee puerto GPIO (general purpose Input/Ouput), los cuales son configurables mediante programa, si estos son salidas o entradas.

8.2.3.2 Compatibilidad con elementos de sensado y actuadores.

Debido a sus pines, sus conexiones serial y USB la Raspberry Pi puede conectarse a sensores de:

- pH.
- Temperatura.
- Humedad.
- Conductividad eléctrica.
- Calidad del aire.
- Humedad del suelo.
- Estaciones meteorológicas.

En realidad, cualquier sensor puede ser conectado a la Raspberry Pi, incluso sensores análogos a través de tarjetas Arduino.

8.2.3.3 Costo.

Es un elemento confiable la cual es de bajo costo, si se habla de calidad versus precio, si se requiere un control para un proceso complejo, es una de las mejores opciones ya que su precio promedio es de 40 mil pesos aproximadamente.

8.2.3.4 Software y programación.

El software es open source, siendo su sistema operativo oficial una versión adaptada de Debian, denominada Raspbian, aunque permite usar otros sistemas operativos, incluido una versión de Windows 10.

Se programa en Python; es un lenguaje de programación muy potente y versátil para el registro y manejo de datos. Esto es una ventaja al momento de programar la Raspberry Pi, ya que se pueden conectar fácilmente las lecturas de los sensores con mecanismos de registro en la computadora y en servidores online.

8.3 ELECCIÓN FINAL DEL CONTROLADOR.

La elección se realiza bajo la creación de un sistema escalar de 1 a 7, donde 1 es deficiente y 7 es máximo. Donde se evaluarán ciertos criterios y/o requerimientos para el proyecto, los que consisten en:

- Robustez.
 - Física.
 - Tecnológica.
- Conocimiento programación.
- E/S necesarias.
- Conexión a internet.
- Costo.

Siempre teniendo en cuenta que se busca un proceso simple, **NO SOBREDIMENSIONADO**, por lo tanto, el controlador no debe exceder demasiado lo requerido, y que tampoco posea funciones innecesarias dentro del proyecto.

Es decir, se busca un promedio de evaluación entre 4 y 5.

A continuación, se presenta la tabulación de criterios y la debida evaluación de cada microcontrolador.

Controlador.	Robustez.		Conoci. Program.	E/S necesarias.	Conex. Internet.	Costo.	Promedio.
	Física	Tecnol.					
PIC.	4	4	2	5	6	4	4,1
Arduino.	4	4	6	5	6	3	4,6
RaspBerry	4	7	1	6	7	6	5,1

Tabla 8- 3 Elección microcontrolador.

“DADAS LAS CONDICIONES DE EVALUACIÓN PROPUESTAS Y EL RESULTADO OBTENIDO, SE PUEDE DEDUCIR QUE EL ELEMENTO DE CONTROL MEJOR EVALUADO, CORRESPONDE A **ARDUINO**. EL CUAL SERÁ PARTE FUNDAMENTAL DEL PROYECTO.”

8.4 ELECCIÓN MODELO ARDUINO.

para elegir de forma definitiva el modelo de Arduino a usar es necesario conocer todos los elementos que interactuarán dentro del proceso, convirtiéndose en entradas tanto digitales o analógicas y salidas digitales o analógicas.

Por ende, se debe tomar en cuenta las tablas de entradas y salidas dadas al comienzo del capítulo. De las cuales se obtienen los siguientes resultados:

- **Cantidad de entradas discretas: 0**
- **Cantidad de entradas analógicas: 4**
- **cantidad de salidas discretas: 3**
- **Cantidad de salidas analógicas: 0.**

Arduino presenta una enorme variedad de placas, las cuales tienen su mayor diferencia en su microcontrolador y la cantidad de entradas y salidas que poseen.

Dentro de estas diferencias, la más influyente en el proyecto, es la cantidad de E/S que posea la placa de desarrollo.

Gracias a esto, de manera fácil se puede elegir una de las placas más conocidas, ARDUINO UNO R3. La cual cumple con todas las necesidades requeridas.

8.5 ARDUINO UNO R3.

De manera resumida se presenta a continuación, la tabulación de las características técnicas más importante de esta placa de desarrollo

Microcontrolador ATmega 328.
Voltaje de entrada 7-12 volts.
Corriente 46 mA.
Potencia a 12 VDC, 552 mW.
14 pines digitales de I/O (6 salidas PWM).
6 entradas análogas.
32 k de memoria Flash.
Reloj de 16MHz de velocidad.

Tabla 8- 4 Ficha técnica Arduino UNO R3.

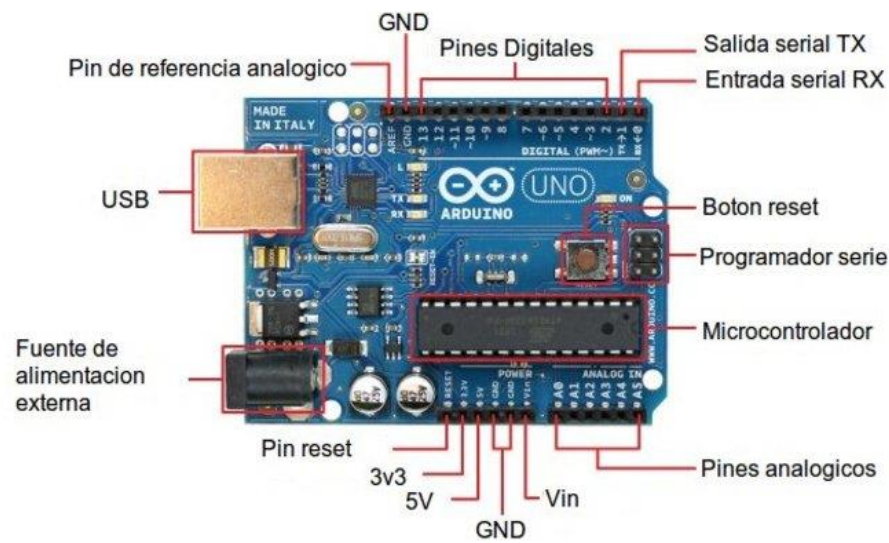


Figura 8- 4 Arduino UNO R3.

8.6 ARDUINO CONECTADO A INTERNET.

8.6.1 Modelo TCP/IP.

El modelo TCP/IP describe un conjunto de guías generales de diseño e implementación de protocolos de red específicos para permitir que un equipo pueda comunicarse en una red. TCP/IP provee conectividad de extremo a extremo especificando como los datos deberían ser formateados, direccionados, transmitidos, enrutados y recibidos por el destinatario.

La importancia del modelo TCP/IP es que es el modelo usado para acceder a Internet o a redes internas (Intranet) de ordenadores. Arduino va a permitir conectarse a Internet o a una red interna mediante TCP/IP y poder realizar múltiples operaciones o usarse como pasarela para conectar a Internet dispositivos que no tienen esa capacidad. La implementación de protocolos de TCP/IP en Arduino se hace mediante un shield o HW adicional que nos da la capa de acceso a red (ethernet o Wi-Fi), internet (IP) y transporte. La capa de aplicación deberemos implementarla dentro de Arduino ya sea directamente o mediante una librería.

En el caso del protocolo TCP/IP en el modelo OSI se simplifica:

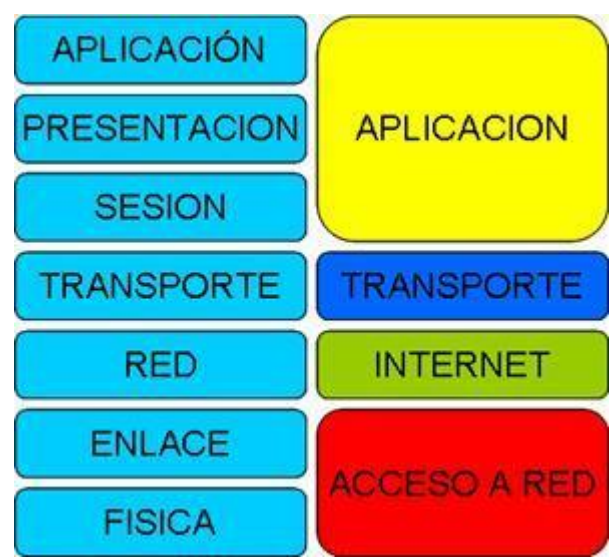


Figura 8- 5 TCP/IP a OSI.

8.6.2 HTTP.

Hypertext Transfer Protocol o HTTP (en español protocolo de transferencia de hipertexto) es el protocolo de comunicación que permite las transferencias de información en la WWW. Se trata de un protocolo de capa 7 de aplicación.

En Arduino con la librería ethernet solo trabaja con la capa de aplicación, todas las otras capas de TCP/IP ya están implementadas por Hardware, ya sea con la ethernet shield o el módulo Wifi. Aunque si se quiere realizar algunas funciones de capas inferiores, se puede hacer con los comandos adecuados comunicándonos con el chip ethernet o wifi vía SPI.

8.6.2.1 HTTP Request.

Un cliente HTTP debe formar una petición HTTP al servidor de una forma determinada para que sea entendida por el servidor. Cuando Arduino trabaja como cliente hay que programar esta petición correctamente.

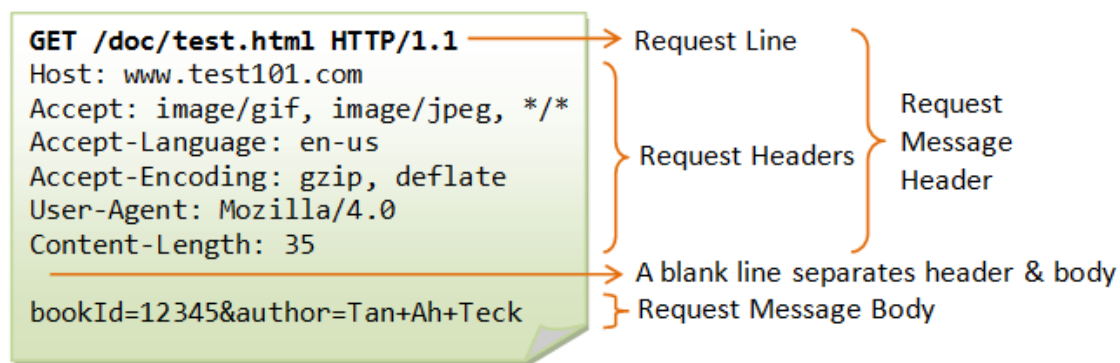


Figura 8- 6 HTTP Request.

8.6.2.2. HTTP Response.

Después de recibir e interpretar el servidor un HTTP request, el servidor debe responder con un mensaje de respuesta:

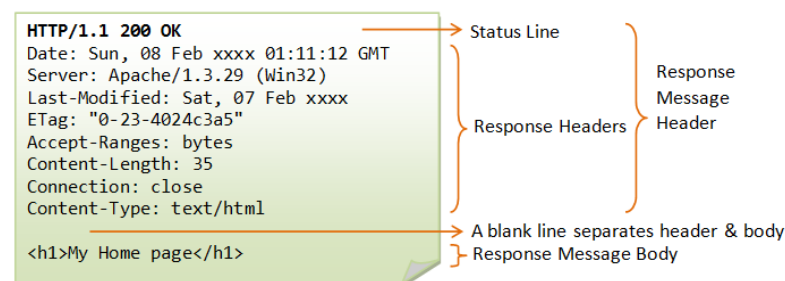


Figura 8- 7 HTTP Response.

8.6.3 Protocolo MQTT.

MQTT (MQ Telemetry Transport Message Queuing Telemetry Transport) es un protocolo de mensajería basado en publicación-suscripción estándar ISO (ICE / IEC PRF 20922) Funciona sobre el protocolo TCP / IP. Está diseñado para conexiones con ubicaciones remotas donde se requiere una "huella de código pequeña" o del ancho de banda de la red es limitado. El patrón de publicación publicación-suscripción requiere un intermediario de mensajes.

Es un protocolo usado para la comunicación machine-to-machine (M2M) en el "Internet of Things". Este protocolo está orientado a la comunicación de sensores, debido a que consume muy poco ancho de banda y puede ser utilizado en la mayoría de los dispositivos empujados con pocos recursos (CPU, RAM, ...). Un ejemplo de uso de este protocolo es la aplicación de Facebook Messenger tanto para android y Iphone. La arquitectura de MQTT sigue una topología de estrella, con un nodo central que hace de servidor o "broker" con una capacidad de hasta 10000 clientes. El broker es el encargado de gestionar la red y de transmitir los mensajes, para mantener activo el canal, los clientes mandan periódicamente un paquete (PINGREQ) y esperan la respuesta del broker (PINGRESP). La comunicación puede ser cifrada entre otras muchas opciones.

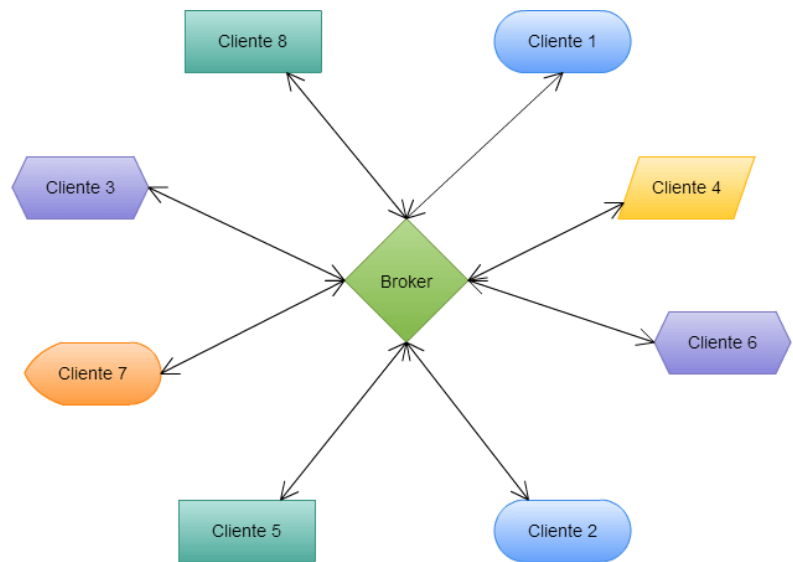


Figura 8- 8 Diagrama estructural Protocolo MQTT

La comunicación se basa en unos "topics" (temas), que el cliente que publica el mensaje crea y los nodos que deseen recibirlo deben subscribirse a él. La comunicación puede ser de uno a uno, o de uno a muchos. Un "topic" se representa mediante una cadena y tiene una estructura jerárquica. Cada jerarquía se separa con '/'. Y recibe datos como podemos ver a continuación:

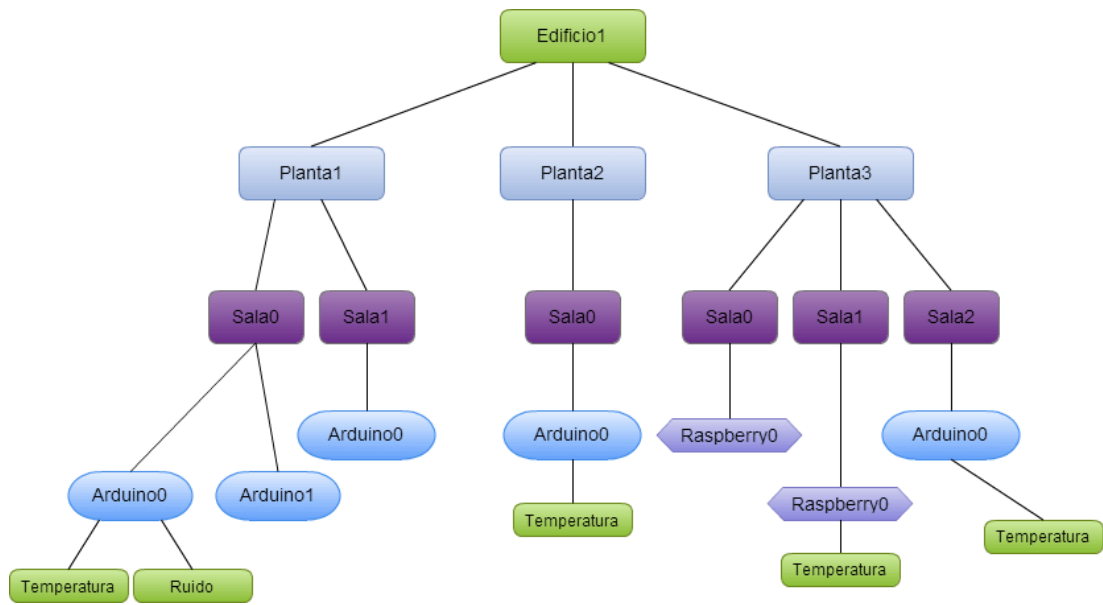


Figura 8- 9 Estructura jerárquica protocolo MQTT

8.7 Lenguaje y programación software.

La estructura del lenguaje de programación de Arduino es bastante simple y está compuesta de al menos dos partes. Ambas funciones son muy necesarias debido a que contienen declaraciones o instrucciones (también llamado estamentos).

void setup ()	// Esta función se invoca una sola
{	vez, cuando el programa
instrucciones;	empieza. Se utiliza para
	inicializar los modos de trabajo
	de los pins, o el puerto serie.
}	
	//Debe ser incluido en un programa,
	aunque no haya declaración que
	ejecutar.
 void loop ()	 // Después de llamar a setup (),
{	la función loop () hace
instrucciones;	precisamente lo que sugiere su
}	nombre, se ejecuta de forma
	cíclica, lo que posibilita que el
	programa este respondiendo
	continuamente ante los eventos que
	se produzcan en la tarjeta.

La estructura se puede apreciar “fácil” pero se deben tener en cuenta pequeños detalles para la correcta escritura de un programa, como lo es:

- **Entre llaves {}.** Estas definen el principio y el final de un bloque de instrucciones, suelen ir en los bloques de instrucciones de programación setup (), loop (), if, if – else, for, etc. Siempre una llave de apertura debe ir seguida de una llave de cierre, pues si no es así el programa dará errores.
- **Punto y coma.** Todas las instrucciones e incluyendo también las declaraciones de variables acaban con punto y coma, el uso de este signo es indispensable para separar instrucciones y si se olvidase esto se traducirá en un error de compilación.
- **El uso apropiado de las mayúsculas.** El lenguaje Arduino es Case – Sensitive, esto quiere decir que es totalmente diferente escribir una letra en mayúscula que en minúscula.
- **Comentarios.** En el entorno de la programación de Arduino existen dos formas de insertar comentarios, una de ellas es para crear tan solo una línea de comentarios y se comienza con // y se termina con la siguiente línea de código, mientras que la otra sirve para incluir un bloque de comentarios o multilínea de comentarios, este bloque comienza con /* y termina con */.
Los comentarios son áreas de texto ignorados completamente por el programa y su principal uso radica en ayudar a describir el programa para así facilitar la comprensión de este.

8.7.1 Variables.

Una variable es una manera de nombrar y almacenar un valor numérico para su uso posterior por el programa. Dicho valor numérico se puede obtener de diversas maneras: puede haber sido asignado literalmente (a esto se le llama “inicializar o asignación de una variable”), puede haber sido adquirido por un sensor, o puede haber sido el resultado de algún cálculo, etc. Provenza de donde provenga inicialmente, ese valor podrá ser siempre

cambiado en cualquier instante posterior de la ejecución del programa, solo si se sea y si el programa lo requiera.

Antes de poder usar alguna variable dentro del programa es necesario declararlas previamente y para lograr aquello es necesario definir qué tipo de variable será, a que se refiere con esto, quiere decir que la variable a escoger va a depender de que tipos de datos (números enteros, números decimales, cadena de caracteres, etc.) se quieran almacenar, dentro del conjunto de variables que podemos declarar tenemos las de tipo **int** (números enteros), **long** (largo), **float** (números decimales), **const**

(valor constante), **char** (carácter), **boolean** (booleano, puede adquirir dos valores solamente, '1' y '0') dentro de las más usadas en la programación.

El siguiente ejemplo se declarará la variable **VariableEjemplo** como una variable de tipo entero (int), y asignándole un valor inicial igual a cero. Esto se llama una asignación.

Por otro lado, una variable puede ser declarada en una serie de lugares del programa y en función del lugar en donde se lleve a cabo la definición, esto determinará en que partes del programa se podrá hacer uso de ella es decir que una variable puede ser declarada al inicio del programa antes de la parte de configuración setup () (**variable global**), a nivel local dentro de las funciones (**variable local**), y, a veces, dentro de un bloque, como para los bucles del tipo **if**, **for**, etc. En función del lugar de declaración de la variable determinará el ámbito de aplicación, o la capacidad de ciertas partes de un programa para hacer uso de ella.

- Una variable global es aquella que puede ser vista y utilizada por cualquier función y estamento de un programa. Esta variable se declara al comienzo del programa, antes de setup ().
- Una variable local es aquella que se define dentro de una función o como parte de un bucle. Sólo es visible y sólo puede utilizarse dentro de la función en la que se declaró.

8.7.2 Configuración de pines.

El poder llegar a configurar los pines es una parte importante de la programación de Arduino, esto se debe a que hay que definir si los pines serán entradas o salidas y esto

dependerá de que queramos recibir una señal del exterior o realizar un control con Arduino.

A continuación, se dará a conocer la sintaxis, el funcionamiento y utilidad de las instrucciones necesarias para poder configurar los pines, tanto digitales como analógicos:

- **pinMode (nº pin, modo):** Esta instrucción es utilizada en la parte de configuración setup () y sirve para configurar el modo de trabajo de un PIN pudiendo ser INPUT (entrada) u OUTPUT (salida).

```
pinMode (pin, INPUT);           // configura el pin como entrada
digitalWrite (pin, HIGH);        // activa las resistencias
                                internas
```

- **digitalRead (nº de pin):** Lee el valor de un pin (definido como digital previamente) dando un resultado HIGH (alto o '1') o LOW (bajo o '0').

- `digitalWrite (pin, estado)`: Envía al pin definido previamente como OUTPUT el valor HIGH o LOW (poniendo en 1 o 0 la salida).
- `analogRead (pin)`: Lee el valor de un determinado pin definido como entrada analógica con una resolución de 10 bits. El rango de valor que podemos leer oscila de 0 a 1023.

8.7.3 Comunicación puerto serial.

Prácticamente todas las placas Arduino disponen al menos de una unidad UART. Las placas Arduino UNO disponen de una unidad UART que operan a nivel TTL 0V / 5V, por lo que son directamente compatibles con la conexión USB. Por su parte, Arduino Mega y Arduino Due disponen de 4 unidades UART TTL 0V / 5V.

A continuación, se presentan las instrucciones relacionadas con la comunicación serial.

- **Serial.begin ()**: esta instrucción abre el canal serie para que pueda empezar la comunicación por él. Por tanto, su ejecución es imprescindible antes de realizar cualquier transmisión por dicho canal. Por eso normalmente se suele escribir dentro de la sección void setup ().
- `Serial.print ()`: envía a través del canal serie un dato (especificado como parámetro) desde el microcontrolador hacia el exterior.
- `Serial.println ()`: hace exactamente lo mismo que `Serial.print ()`, pero, se diferencia debido a que al final de ejecutar esta instrucción se efectúa un salto de línea.
- `Serial.available ()`: devuelve el número de bytes disponibles para ser leídos que provienen del exterior a través del canal serie (vía USB o vía pines TX / RX). Estos bytes ya han llegado al microcontrolador y permanecen almacenados temporalmente en una pequeña memoria de 64 bytes que tiene el chip TTL – UART –llamada buffer– hasta que sean procesados mediante la instrucción `Serial.read ()`. Si no hay bytes para leer, esta instrucción devolverá 0.
- `Serial.read ()`: devuelve el primer byte aún no leído de los que estén almacenados en el buffer de entrada del chip. Al hacerlo, lo elimina de ese buffer. Para leer el siguiente byte almacenado se ha de volver a ejecutar `Serial.read ()`. Y hacer así hasta que se hayan leído todos. Cuando no haya más bytes disponibles, `Serial.read ()` devolverá -1.

8.7.4 Bloques condicionales.

Para poder utilizar de forma correcta estos bloques es necesario conocer la forma de escritura de los operadores de comparación y de los operadores lógicos, su función es la de establecer la condicionante que hará que se ejecute el bloque o la función.

Los operadores de comparación son los siguientes.

1. Operadores de comparación.

- == Comparación de igualdad
- != Comparación de diferencia
- Comparación de mayor que
- >= Comparación de mayor o igual que
- < Comparación de menor que
- <= Comparación de menor o igual que

Los usados para unir dos o más comprobaciones dentro de una condición. Son los siguientes:

2. Operadores lógicos.

- && Operador AND
- || Operador OR
- ! Operador NOT

La función concreta de estos operadores es que por ellos ingresan dos señales (menos por el operador NOT) y en el caso del operador AND si ambas son TRUE (o '1') su salida será TRUE, pero si una de ellas es FALSE (o '0') su salida será FALSE. Caso contrario en el operador OR, si ambas son FALSE su salida será FALSE, pero basta con que tan solo una de ellas sea TRUE y su salida era TRUE. En el caso del operador NOT es solamente una entrada y si esta es de TRUE se salida será FALSE y viceversa.

8.7.5 Control de flujo.

El control de flujo permite variar, en un número ilimitado de ocasiones a lo largo de un programa, el orden "natural" de ejecución del código, omitiendo, ejecutando, o repitiendo sentencias o bloques de sentencias en función de una o varias condiciones. Las instrucciones para el control de flujo son comunes a muchos lenguajes de programación, y son *if*, *if...else*, *for*, *while*, y *do...while*.

8.7.5.1 if.

Realiza una comprobación de una condición. En el caso de que esta condición se cumpla, el código escrito entre llaves se ejecuta. Si, por el contrario, la condición entre paréntesis no se cumple, el código entre llaves es ignorado y el programa sigue ejecutándose normalmente

```
if (valorAComparar ?? valorDeReferencia) {  
    instruccionesAEjecutar();  
}
```

8.7.5.2 if...else.

Complementa en cierto modo al control de flujo que ofrece la sentencia *if*, permitiendo ejecutar un bloque de código en el caso de que se cumpla la condición entre paréntesis, y otro bloque de código si la condición no se cumple. Estos dos bloques de código son mutuamente excluyentes, por lo que nunca se podrán ejecutar los dos a la vez.

```
if (valorAComparar ?? valorDeReferencia) {  
    instruccionesSiCondicionSeCumple;  
} else {  
    instruccionesSiCondicionNoSeCumple;  
}
```

8.7.5.3 for.

Código encerrado entre llaves se repetirá siempre que se cumpla la condición. Con cada repetición del bloque de código, la condición vuelve a ser evaluada.

```
for (inicialización; condición; expresión) {  
    codigoAEjecutar();
```

8.7.5.4 while.

Ejecuta de forma continua un bloque de código mientras la condición de cabecera se cumpla. Al igual que pasaba con el bucle for, la condición se evalúa con cada repetición del bloque de código hasta que se deje de cumplir, continuando entonces con el código que haya después del bucle while.

```
while (valorAComparar ?? valorReferencia) {  
    codigoAEjecutar();  
}
```

8.7.5.5 do...while.

Es una modificación del bucle while en el que la condición se evalúa al final del bloque de código entre llaves. Por tanto, el bucle do...while no deja de ser un bucle while que se va a ejecutar al menos una vez.

```
do {  
    codigoaEjecutar();  
} while (valorAComparar ?? valorReferencia);
```

**CAPÍTULO 9. DESARROLLO DEL PROYECTO, DEFINICIÓN DE
SENSORES, ACTUADORES Y CONDICIONES DE TRABAJO.**

Toda elección de sensores y actuadores se hará en base a
PROTOTIPO, se considerará una superficie de invernadero de 6m²
aproximadamente.

9.1 CONTROL DE TEMPERATURA.

La realización del control de temperatura del proyecto se realiza en base a la toma de datos de un sensor, el cual permite ejercer un control directo mediante la programación del Arduino; en dos actuadores, en este caso corresponde a un ventilador de aire, el cual permite el enfriamiento del proceso y un calefactor el cual permite el aumento de temperatura.

A continuación, se presenta el diagrama de bloques correspondiente al proceso que se implementará.

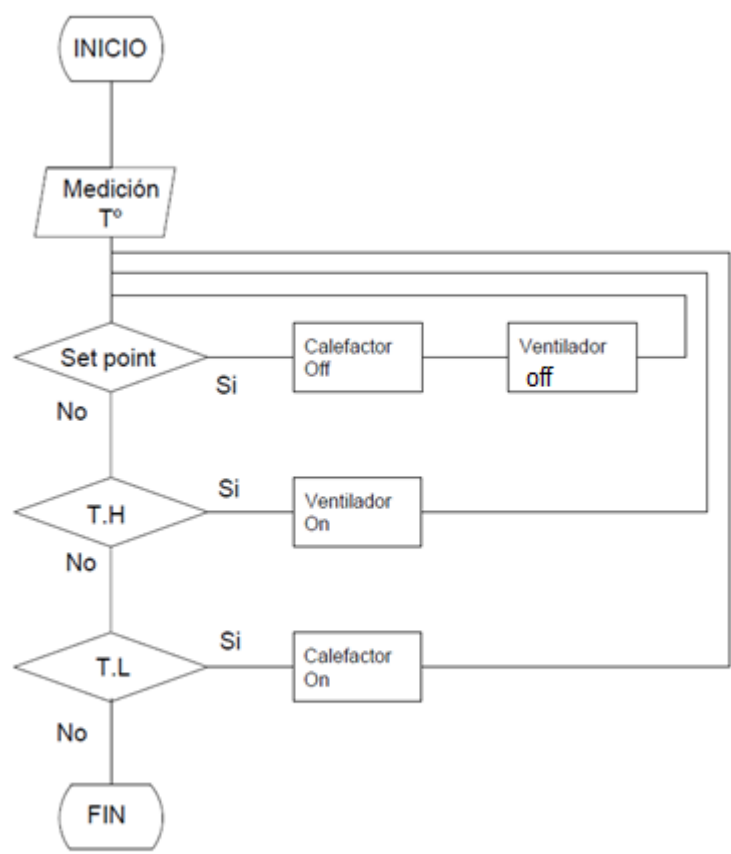


Figura 9- 1 Diagrama de bloques control de T°.

9.1.1 Sensor.

El sensado se realizará mediante un sensor diseñado para Arduino; El **DHT22**. Este sensor a parte de su tamaño reducido, posee la ventaja que ya viene calibrado, es decir, no existe la necesidad de componentes adicionales por lo que basta con conectarlo para comenzar a tomar mediciones de humedad relativa y temperatura. Por lo que asegura una alta calidad y fiabilidad a lo largo del tiempo, ya que contiene un microcontrolador de 8 bits integrado.

Para medir la humedad, cuenta con dos electrodos con un sustrato en medio de ellos que retiene la humedad del ambiente. Entonces, cuando la humedad varía, la conductividad del sustrato y resistencia entre los electrodos también varía.

Para medir la temperatura, estos sensores cuentan con un termistor NTC, que consiste en una resistencia variable la cual cambia su valor conforme varía la temperatura del ambiente. Dicha resistencia disminuye a medida que la temperatura aumenta. Su respuesta es rápida en las medidas,

Puede medir la Humedad entre el rango 0% hasta 100% aproximadamente y la **Temperatura entre el rango -40°C y 80°C.**



Figura 9- 2 Sensor DHT22.

A continuación, se presenta las principales características que posee el sensor DHT22.

Especificaciones.	
Voltaje de alimentación.	3.3-5.5 V.
Corriente durante medición.	2.5 mA.
Consumo durante medición.	12.5 mW
Corriente en modo espera	40-50 µA.
Humedad.	0-100 RH.
Rango de Temperatura	-40°C a 80°C.
Precisión.	±2%RH.

Tabla 9- 1 Especificaciones técnicas sensor DHT22.

DHT22 se caracteriza por su reducido tamaño, el cual es en el orden de centímetros. A continuación, se presentan las dimensiones exactas de este.

Todas las medidas se encuentran en milímetros [mm].

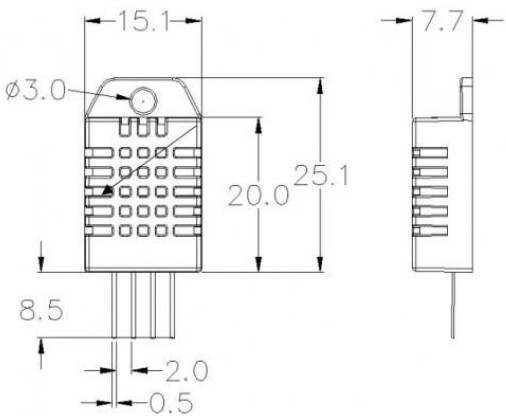


Figura 9- 3 Dimensiones DHT22.

Dicho sensor posee 4 pines, aunque la conexión solo se realiza mediante 3 de estos, los que se presentan a continuación, con su debida descripción.

PIN	Nombre	Descripción
1	Vcc	Fuente de alimentación (3.3-5.5V)
2	SDA	Datos puerto serie
3	NULL	No se conecta
4	GND	Tierra

Tabla 9- 2 PinOut sensor DHT22.



Figura 9- 4 PIN OUT DHT22

9.1.1.1 Conexión con Arduino.

A continuación, se muestra la conexión del Arduino con el sensor y su simplicidad, solo presenta una condicionante; El Pin 2 del sensor DHT22 se conecta primero a una resistencia en el orden de los K Ω y luego al Vcc del Arduino.

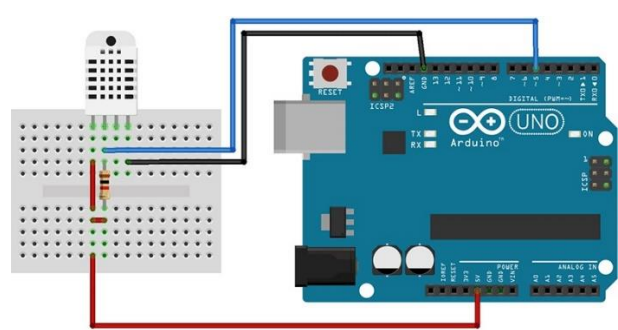


Tabla 9- 3 Conexión DHT22 con Arduino.

9.1.1.2 Comunicación con Arduino.

El protocolo de comunicación es a través de un único hilo, correspondiendo al protocolo 1-wire, por lo tanto, hace que la integración de este sensor con el proyecto sea rápida y sencilla.

La longitud de comunicación y la capacidad de transmisión de señal es hasta 20 metros de distancia.

9.2 CONTROL DE HUMEDAD.

El objetivo de esta parte del proceso es mantener una Humedad Relativa cercana a un *Set Point*, que corresponde al porcentaje de humedad ideal para el cultivo. Al igual que el control de Temperatura, el control de Humedad se realizará bajo las mediciones del sensor DHT22, el cual posee esta función. Este control también directo es a partir de programación, la cual toma los datos de humedad relativa y los procesa para tomar una acción, mediante actuadores. Estos actuadores serán al igual que en el control de temperatura, ventilador, calefactor y agregando en este caso también el sistema de nebulización.

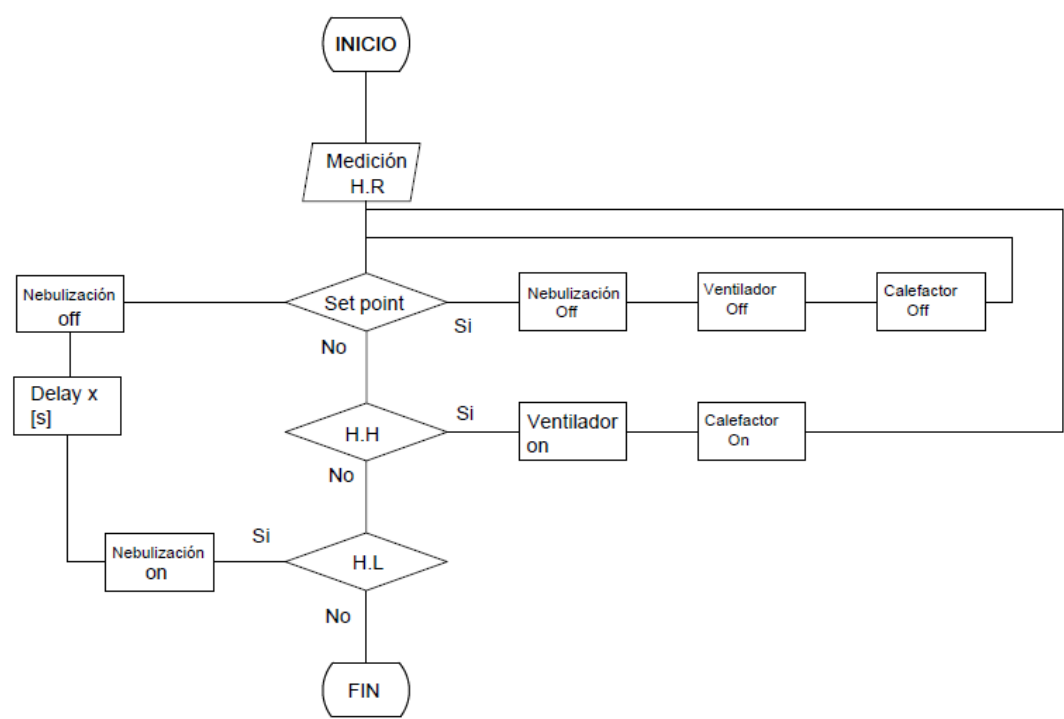


Figura 9- 5 Diagrama de bloques control de Humedad.

9.2.1 Condiciones de trabajo.

Humedad deficiente: En esta condición de trabajo anormal, el proceso debe activar la nebulización dentro del invernadero, aumentando así la humedad relativa del aire. Permitiendo obtener un valor cercano al *Set Point*. (es posible dar un margen de error permisible, ya que es una variable difícil de controlar).

Humedad excesiva: En esta condición de trabajo anormal, el proceso debe activar el calefactor y ventilador para así reducir la humedad y hacer circular de forma forzada el aire al interior del invernadero.

9.3 Actuadores.

Destacar que los actuadores correspondientes al proceso de control de Temperatura y humedad, **NO PUEDEN CONECTARSE DIRECTAMENTE A LA PLACA ARDUINO** ya que son de alimentación monofásica 220v, entonces es imperativo conectarlo mediante un *relé*, el cual conmutará de acuerdo a la señal de un *PIN* de salida discreta del Arduino, permitiendo la energización o des energización de los actuadores.

9.3.1 Extractor de aire.

Para obtener una mayor eficiencia es necesario considerar una cierta “regla” que se considera estándar en cuanto a la extracción y cambio de aire en un invernadero. La cual dice que para la correcta extracción y cambio de aire dentro de un invernadero el extractor debe tener un caudal mínimo de 10 veces mayor al volumen del recinto.

Entonces para realizar la correcta elección se debe calcular el volumen total del invernadero para luego obtener el caudal del extractor que se necesita.

El recinto está compuesto por dos prismas, los cuales son un prisma rectangular y otro triangular, para obtener el volumen total es necesario calcular ambos por separado, ya que se dichos cálculos son de forma distinta uno del otro.

I. Cálculo volumen prisma Rectangular.

$$\text{Volumen} = \text{Ancho} * \text{Largo} * \text{Alto.}$$

$$\begin{aligned} \text{Volumen} &= 2 \text{ m} * 3 \text{ m} * 2 \text{ m} \\ V &= 12 \text{ m}^3. \end{aligned}$$

II. Cálculo volumen prisma Triangular.

$$\text{Volumen} = A_{\text{base}} * h$$

$$A_{\text{base}} = \frac{1}{2} * \text{Base} * \text{altura} \rightarrow A_{\text{base}} = \frac{1}{2} * (200 \text{ cm}) * (20 \text{ cm}) \rightarrow A_{\text{base}} = 0,2 \text{ m}^2$$

$$\text{Volumen} = 0,2 \text{ m}^2 * 3 \text{ m} = 0,6 \text{ m}^3.$$

III. Volumen total del recinto y volumen necesario de extractor.

$$V_{\text{Total}} = 0,6 \text{ mts}^3 + 12 \text{ mts}^3 = 12,6 \text{ m}^3.$$

$$V_{\text{Extractor}} = 12,6 \text{ mts}^3 * 10 = 126 \text{ m}^3/\text{h}.$$

Dentro del mercado no existen valores tan exactos a los requeridos por lo tanto se aproxima el valor 130 m³/hora.



Figura 9- 6 Extractor axial.

9.3.2 Calefactor

Para calcular las necesidades de calefacción, se deben contemplar los siguientes factores:

- 1) Salto térmico entre la Tº exterior y la deseada en el interior.
- 2) Superficie de construcción.
- 3) Material con el que está construido.

Para obtener la pérdida total de calor del invernadero, se aplica la siguiente fórmula:

$$Ct = K * S * (Ti - Te)$$

Donde:

- Ct= Kilocalorías necesarias.
- K= Coeficiente de transmisión de calor de la cubierta.
- S= Superficie en metros cuadrados de paredes y techo.
- Ti= Tº mínima al interior deseada.
- Te= Temperatura media/mínima exterior.

Ya que dentro del estudio del proyecto no se considera la construcción del invernadero, La ecuación permitirá calcular las calorías necesarias según el material que se implemente.

A modo de ejemplo y porque es un material muy usado en la construcción de invernaderos de usará el polietileno, el cual posee un coeficiente "K" igual a 5.

Entonces:

- Ct= ?
- K=5.
- S= 26,52 m²
- Ti= 3ºC
- Te= -2 ºC

$$Ct = 5 * 26,52 * (3 - (-2)) = 132.6 \text{ Kcal/h.}$$

Entonces según los cálculos la cantidad de Kcal/h mínima necesaria para mantener una Tº óptima dentro del invernadero (en el caso que este cubierto por polietileno), es de **133 Kcal/h.**

Existen tabulaciones y conversiones directas entre KW/h y Kcal/h. esto permitirá determinar la potencia necesaria para cumplir con los requerimientos del sistema.

La conversión dice que 1 Kcal/h equivale a 1,163 W/h.

Entonces:

$$133 \text{ kcal/h} * 1,163 \text{ w/h} = 153,52 \text{ W/h}$$

A continuación, se calcula el valor de corriente nominal (I_n) de trabajo de acuerdo a las condiciones de trabajo anteriormente descritas.

$$I = \frac{P}{V} \rightarrow I = \frac{153,52 \text{ W}}{220 \text{ V}} = 690 \text{ mA.}$$

En resume, el calefactor que cumple con los requerimientos mínimos debe ser alimentado por una fuente de 220 VAC, suministrar un mínimo de 133 Kcal/h y consumir una potencia de 153 W.

Aunque en lo real es difícil encontrar esos requerimientos, porque generalmente los calefactores de alimentación de 220 V son de 1500 watts aproximadamente, ya que son resistivos consumen una gran potencia eléctrica. Por ello el control deberá ser ajustado al tiempo de funcionamiento de este. Esto debe ser netamente programación.

9.4 Módulo relé 4 canales optoacoplado.

La razón por la cual es necesario usar relés con optoacopladores es que permite la aislación del circuito de control proveniente desde el Arduino y el circuito de fuerza proveniente de la fuente de poder designada, ya sea de 220 VAC o 12 VDC.

Esto permitirá obtener una mayor protección de los elementos sensibles a subidas de voltajes, tales como: Placa de desarrollo Arduino, sensores etc.

Y así también de posibles ruidos eléctricos provenientes de la alimentación del sistema de fuerza y al realizar la conmutación.

Los módulos necesariamente deben ser de una alimentación 12 V en bobina y soportar tensiones mínimas de 220 VAC.

Estas características se encuentran presentes en la gran mayoría de los módulos diseñados para placas de desarrollo, por ende, dentro del mercado es fácil encontrar. Estos pueden encontrarse de forma individual o en módulos más grandes los cuales poseen 2, 4 u 8 relés con sus respectivos optoacopladores. Ya que dentro del proceso existen 3 elementos de potencia, los cuales son: extractor, calefactor y la bomba nebulizadora; son necesarios 3 relés.

La mejor opción es utilizar un módulo de 4 relés, quedando uno disponible para alguna futura aplicación.

Básicamente estos módulos son genéricos y no existe diferencia en entre un modelo de otro.

Existen módulos de relés para montaje en riel din, pero difícilmente se encuentran en 12 V y con optoacopladores.



Figura 9- 7 Módulo relés 4 canales optoacoplados.

A continuación, se presentan las principales características de dicho módulo.

Característica.	
Canal de salida	4
Tensión de carga.	125 VAC/250 VAC. 28 VDC/30 VDC.
Corriente operación carga.	10 A.
Tensión de bobina.	12 V.
Diodo de protección.	En cada bobina.
Consumo de corriente por bobina	20 mA.

Tabla 9- 4 Características módulo relé 4 canales.

Conexión con Arduino.

La conexión debe hacerse de forma directa a la bobina que posee el relé, eso se consigue conectado algún pin de salida digital y el GND de Arduino hacia los contactos del relé designados para el control de energización y desenergización de la bobina de 12 V.

Pin Out módulo relé.

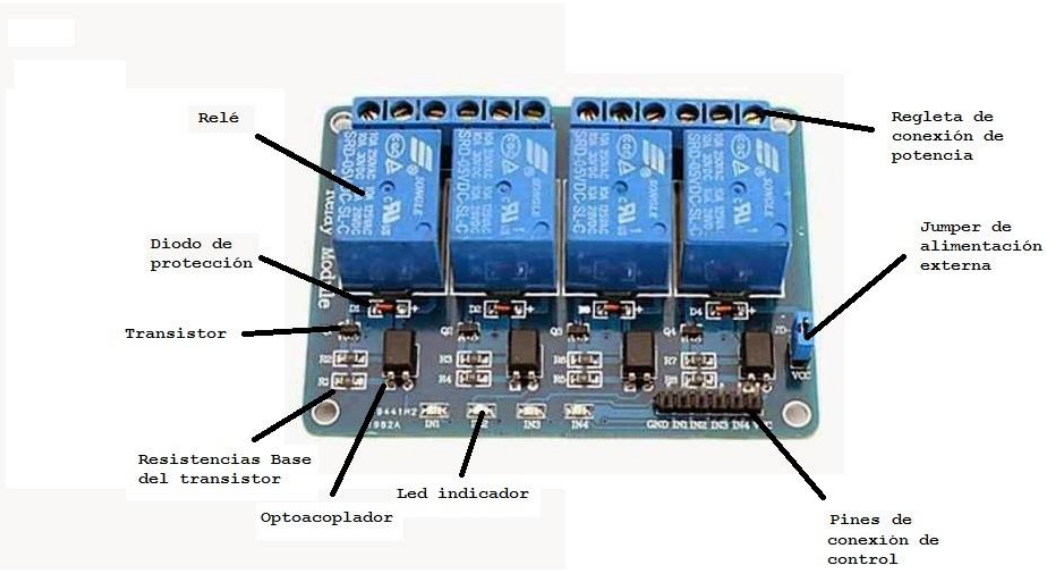


Figura 9- 8 Pin-Out Módulo relé 4 canales

9.5 CONTROL RIEGO POR NEBULIZACIÓN.

El riego mediante nebulización se efectuará de forma periódica, para mantener una humedad de suelo óptima. Para lograr esto es necesario conocer la humedad a la cual se encuentra dicho suelo, esto mediante un sensor.

El sensor a usar es el sensor de humedad de suelo **FC-28**, entregando este una medida real al software.

Lo siguiente es que el software aplique una acción directa mediante un actuador, en este caso una bomba de pulverización, con sus respectivos elementos, como boquillas, tuberías flexibles etc.

A continuación, se presenta el diagrama de bloques correspondiente al proceso de control a efectuar.

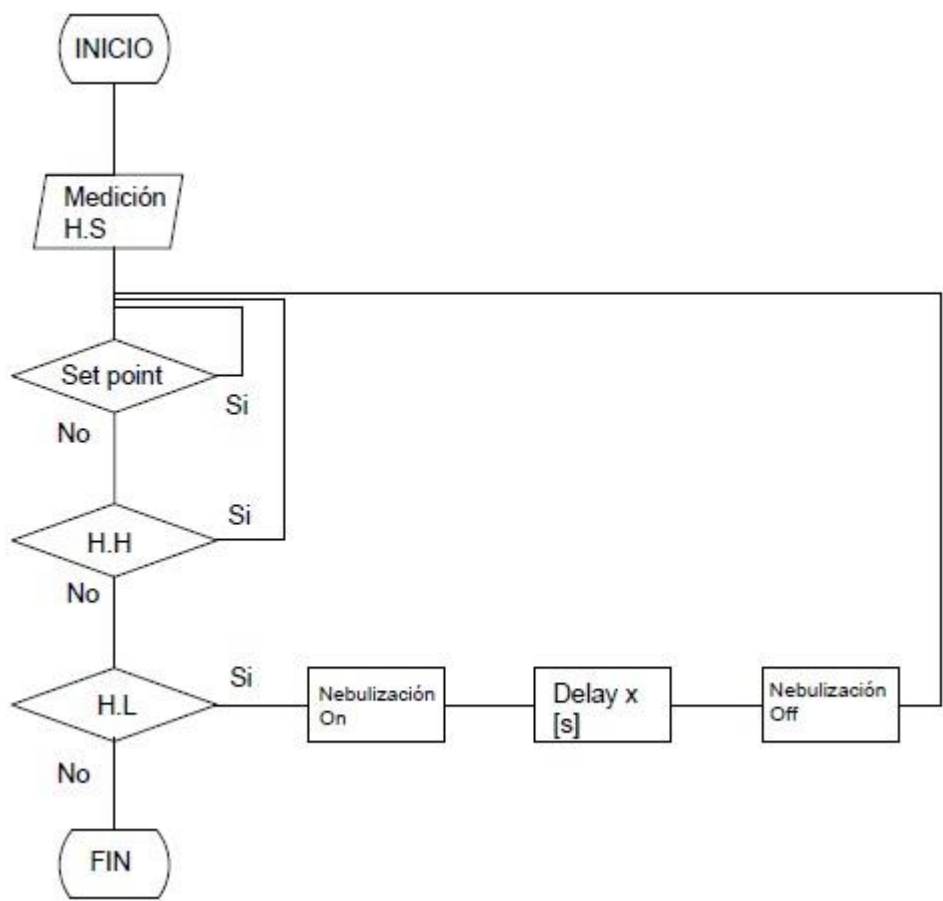


Figura 9- 9 Diagrama de bloques Control riego por nebulización.

9.5.1 Sensor FC-28.

El módulo FC-28, un sensor de humedad de suelo resulta ser otro módulo que utiliza la conductividad entre dos terminales para determinar ciertos parámetros relacionados a agua, líquidos y humedad. El principio de funcionamiento de este dispositivo consiste en dos placas separadas entre sí por una distancia determinada. Ambas placas están recubiertas de una capa de material conductor. 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 LM393 que será el encargado de transformar la conductividad registrada a un valor analógico que podrá ser leído por Arduino.

Posee dos tipos de salidas, una analógica y una digital.

- La salida digital entregará un pulso bajo cuando haya conductividad suficiente entre cada una de las puntas. El umbral de disparo se puede establecer ajustando el potenciómetro del circuito de control.
- La salida analógica el nivel de voltaje dependerá directamente de cuanta humedad haya en el suelo. Es decir, dependiendo de cuanta conductividad (producto del agua en el suelo) haya entre las puntas del módulo, así variará el valor entregado.

Los valores entregados varían entre **0 y 1023**, donde 0 corresponde a un suelo muy húmedo y 1023 suelo demasiado seco.

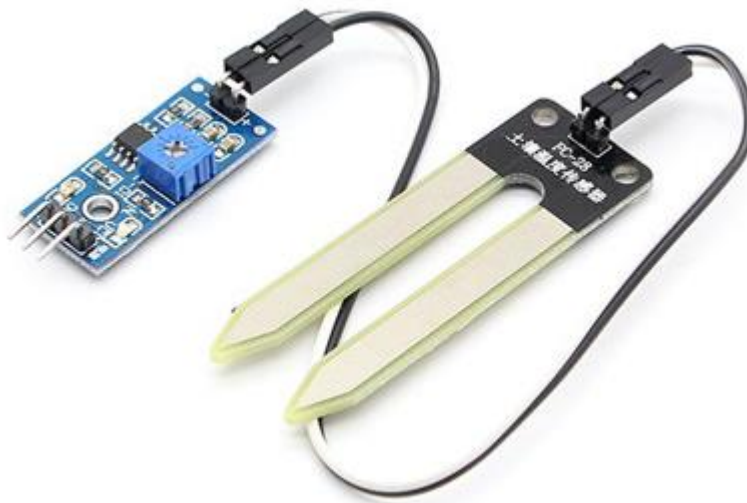


Figura 9- 10 Sensor de Humedad FC-28.

A continuación, se presenta una tabla de características eléctricas del sensor.

Característica.	
Tensión alimentación.	3.3 VDC – 5 VDC.
Corriente trabajo	20 mA.
Potencia trabajo.	100 mW.

Tabla 9- 5 Características técnicas FC-28

A continuación, se presenta el *Pin Out* del sensor y su esquema de montaje y conexionado correspondiente.

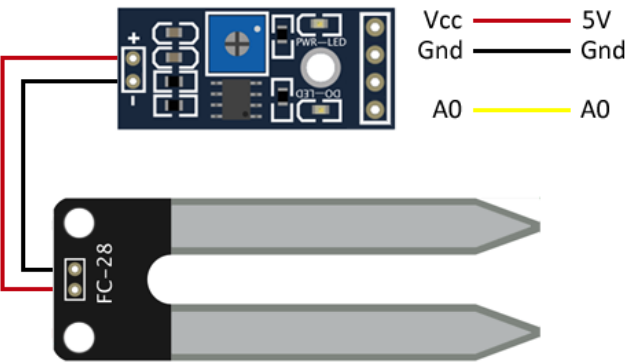


Figura 9- 11 Pin Out FC-28

9.5.1.1 Conexión con Arduino.

Como ya se ha visto en el *Pin Out*, la conexión es bastante sencilla, siendo de la siguiente forma:

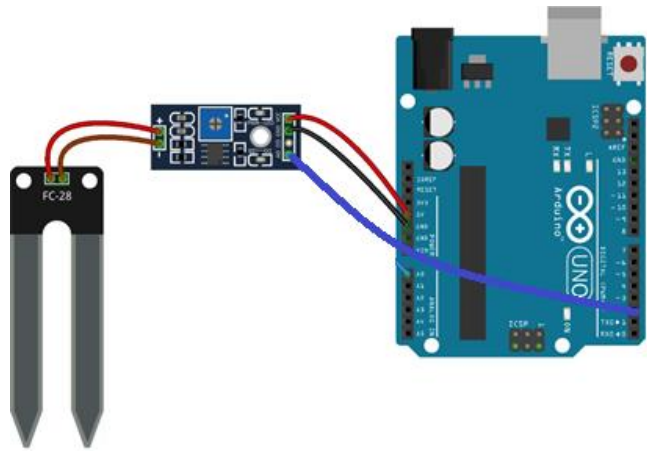


Figura 9- 12 Conexión FC-28 con Arduino

La longitud de comunicación y la capacidad de transmisión de señal es hasta 20 metros de distancia.

9.5.2 SISTEMA DE BOMBEO.

El sistema de bombeo cuenta con diversos elementos, tales como:

- Bomba
- Mangueras.
- Boquillas.

Los que en conjunto permiten realizar un regadío de manera correcta. Cada uno se eligió en base al área donde se deberá suministrar el riego (6m²), es decir el sistema elegido no es sobredimensionado a las necesidades del proyecto.

Destacar que al igual que los actuadores del control de Temperatura y Humedad, la bomba no se puede conectar directamente al Arduino, es decir, se necesita obligatoriamente un relé para ejercer la acción de activación o desactivación del proceso de riego.

9.5.2.1 Bomba.

Ya que el área de riego no es grande, no es necesario instalar una bomba muy robusta, por ende, dentro de la gran variedad se eligió una pequeña bomba de refuerzo de alta presión de diafragma, marca Propumps. La cual se integra satisfactoriamente al sistema.

A continuación, se muestra de forma detallada las características que posee.



Figura 9- 13 Bomba nebulizadora.

9.5.2.1.1 Características técnicas.

Descripción.	
Marca.	Pro Pumps.
Modelo.	HG-MNP-50
Material.	Plástico.
V. alimentación	12 VDC.
Corriente trabajo.	5 A.
Potencia.	60 W
Caudal.	5 L/Min.
Presión de trabajo.	160 PSI.
Tamaño	16.5*10*6 cm
IP	53
Conexiones mangueras.	Hilo macho 1/2 "

Tabla 9- 6 Datos técnicos bomba nebulizadora.

Ya que la alimentación es de 12 VDC se debe considerar una fuente de tensión de 12 VDC. Para así poder alimentar la bomba.

9.5.3 Sistema de Nebulización.

El sistema de nebulización consta con dos elementos principales, que permiten el transporte y la expulsión del agua proveniente de la bomba.
Estos son: Mangueras y boquillas de alta presión.

9.5.3.1 Tuberías.

El medio de transmisión se elige de acuerdo a las necesidades de presión, la presión nominal de trabajo del sistema de riego, por ende, es necesario tener un sistema de mangueras que entregue la seguridad de no ruptura.

El largo a ocupar se relaciona directamente con el área de riego, en este caso 6 m².
A continuación, se presenta la tubería a usar y sus características.



Figura 9- 14 Tuberías nebulización

9.5.3.2 Boquillas de asiento.



Figura 9- 15 Boquillas de asiento nebulización.

9.5.3.3 Boquillas de nebulización.



Figura 9- 16 Boquilla de nebulización

9.5.4 VISUALIZACIÓN DE VARIABLES.

Para poder conocer los valores de las variables mediadas de forma directa, es necesario establecer una interfaz entre el usuario y el controlador, por ende, y ya que las variables y valores son reducidos, no es necesario contar con un elemento sofisticado. En este caso una **pantalla LCD** creada para Arduino.

9.5.4.1 Pantalla LCD 16x2.

Es una pantalla alfanumérica que consta de 2 filas y 16 columnas, con un módulo integrado de conversión de comunicación a protocolo I2C, el cual permite la interacción del controlador con la pantalla mediante dos hilos de conexión. También es necesario instalar una librería para la comunicación del LCD, están se encuentran de forma gratuita en internet.



Figura 9- 17 Pantalla LCD 16x2 con módulo I2C.

A continuación, se presenta la tabulación de las características de la pantalla LCD.

Características.
Backlighth led azul
Ángulo de visión amplio y alto contraste.
Controlador integrado estándar HD44780.
Puede mostrar 2 líneas x 16 caracteres.
Alimentación 5 [V].
Consumo de corriente con luz de fondo total: 25 mA máx.
Dimensión del módulo: 80mm x 35mm x 11mm.
Tamaño de la zona de display: 64.5mm x 16mm

Tabla 9- 7 Datos técnicos pantalla LCD 16x2.

9.5.4.2 Pin Out adaptador I2C.

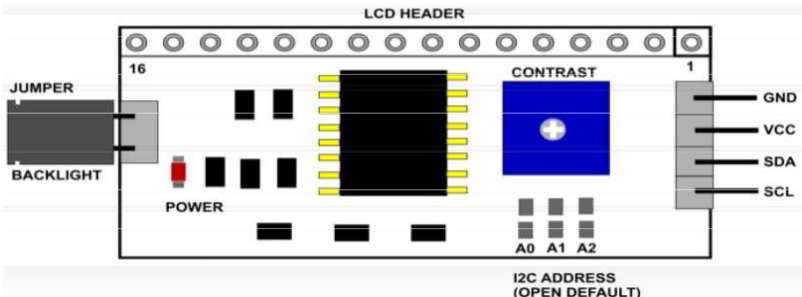


Figura 9- 18Pin-Out adaptador I2C para pantalla LCD 16x2.

GND	Tierra
VCC	Alimentación 5 [V].
SDA	A4
SCL	A5

Tabla 9- 8 Pin Out Adaptador I2C.

9.5.4.3 Conexión con Arduino.

Se presenta la conexión que se debe efectuar para establecer comunicación entre el módulo de conversión a I2C de la pantalla LCD con la placa de desarrollo Arduino.

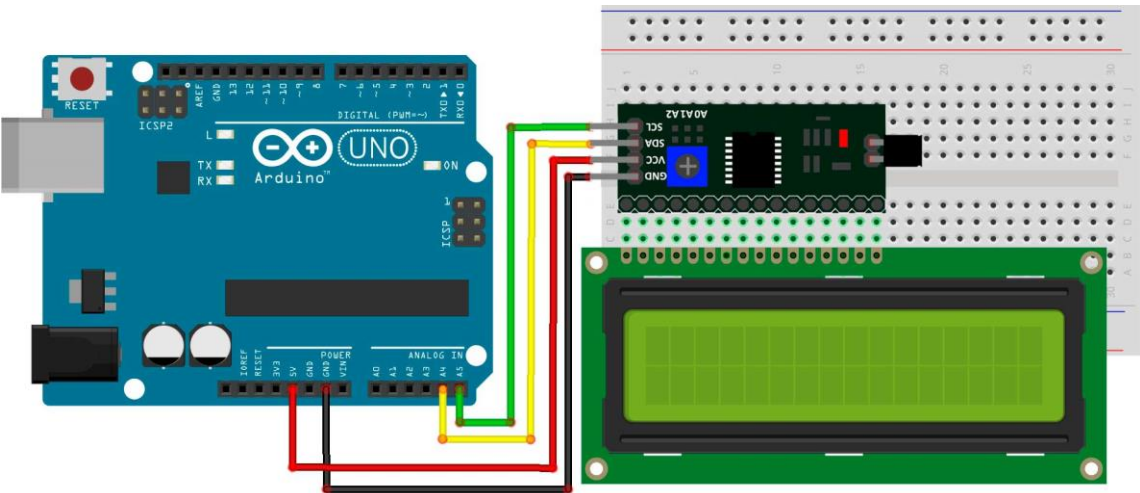


Figura 9- 19 Conexión pantalla LCD con Arduino.

CAPÍTULO 10. INTEGRACIÓN INTERNET DE LAS COSAS

10.1 ESTABLECER COMUNICACIÓN DE ARDUINO A INTERNET.

Ya que la placa Arduino UNO no posee de manera integrada acceso a una red de internet, es necesario realizarlo mediante módulos.

Uno de los módulos adaptadores WI-FI más conocidos es el **ESP8266 ESP-01**, el cual permite una conexión a la red de forma rápida y fácil.

El objetivo de realizar la comunicación es el monitoreo a través de la nube, esto quiere decir, poder recibir los datos y estados de los procesos que contiene el sistema, ya sea la temperatura existente, la humedad tanto de suelo como relativa y obtener información sobre el tiempo de riego ejecutado. También en casos de problemas de algún proceso, este genere alarmas las cuales se transmitirán a la nube

10.1.2 ESP8266 ESP-01.

Este pequeño módulo permite a los microcontroladores conectarse a una red Wi-Fi y realizar conexiones TCP/IP sencillas utilizando comandos de tipo **AT**.

Es compatible con el estándar IEEE802.11b/g/n. Su CPU de 32 bits de baja potencia puede funcionar como procesador de aplicaciones.

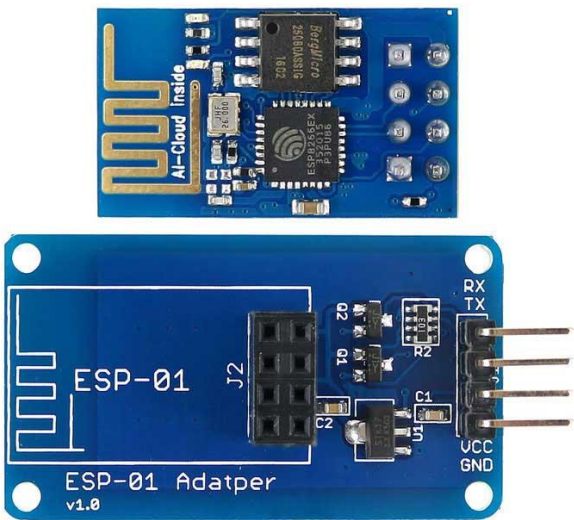


Figura 10- 1 Módulo WIFI ESP-01.

A continuación, se presenta la tabulación de sus especificaciones físicas y eléctricas más importantes:

PARÁMETRO.	
Dimensiones.	25*14*1 [mm].
Peso.	7 [g]
Potencia de salida.	802.11b mode+19.5 [dB].
Wifi	2.4 GHz, seguridad WPA/WPA2
Encendido y transferencia de paquetes.	<2 [ms]
Consumo de energía en espera.	>1.0 [mW]
Consumo de energía en operación.	264 mW
Corriente en operación.	80 mA
Rango Tº operación.	-40 a 125 [°C.]
Voltaje de alimentación.	3.3 [V] máximo.
Comunicación serial.	TX/RX.

Tabla 10- 1 Especificaciones Módulo WIFI ESP-01.

10.1.2.1 Pin Out.

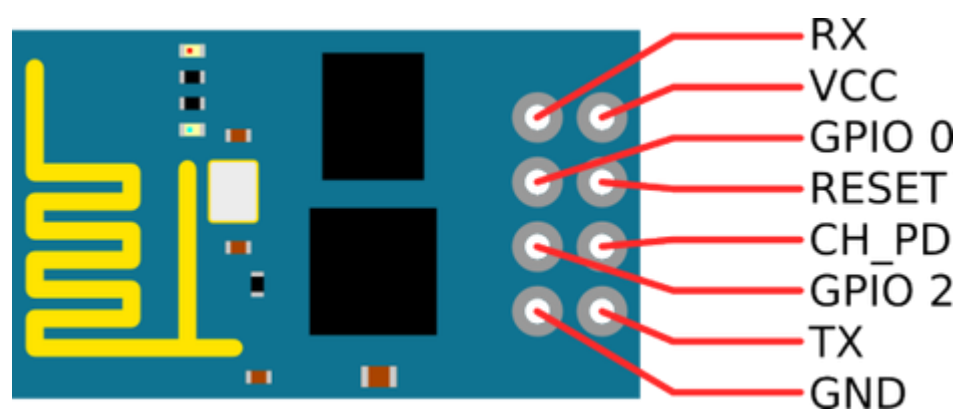


Figura 10- 2 Pin Out ESP-01.

Los Pines necesitados son:

- TX.
- RX.
- GND.
- VCC.
- CH_PD.

10.1.2.2 Conexión con Arduino.

Al hacer la conexión para la programación del módulo, es necesario quitar el microcontrolador que posee Arduino para así solo programar el microcontrolador del módulo, y que se necesita conectar el módulo de forma directa al puerto USB.

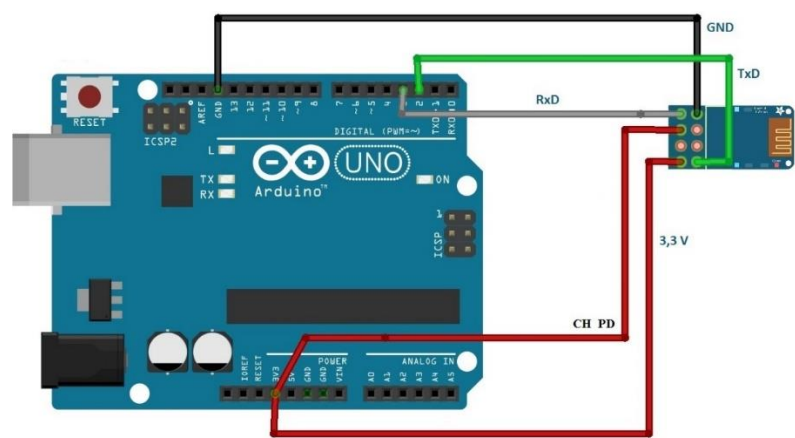


Figura 10- 3 Conexión módulo Wi-fi ESP8266 ESP-01.

Existe un pequeño problema con la conexión la cual es el consumo de corriente de encendido del módulo es mayor a la proporcionada por el Arduino, por ende, el módulo no iniciara y es capaz de quemar el Arduino. La solución es integrar una fuente de alimentación externa al Arduino o al módulo.

Esta alimentación típicamente debe ser una fuente de poder diseñada para Arduino la cual posee una tensión de 3.3 y 5 [V]. Pero la cual solo está diseñada para su instalación en protoboard.

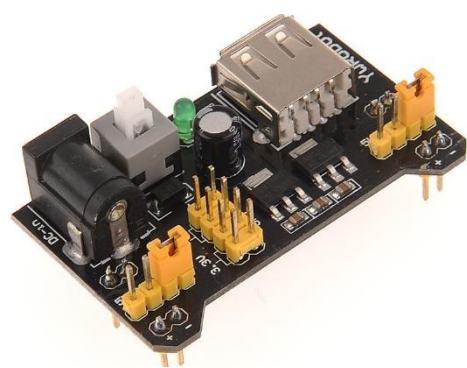


Figura 10- 4 Fuente de poder para protoboard 3.3 [V]-5 [V].

Dentro del mercado existen los denominados conversores o reguladores de voltaje, los cuales posibilitan tener una tensión de salida variable y estable. Y son más versátiles en su instalación ya que poseen separadores como elemento de montaje, y, por ende, aislar los contactos de la placa impresa. Siendo está la mejor opción.

Los requerimientos para que sea compatible con el sistema son: Permitir un voltaje de salida estable de 3.3 VDC, corriente de salida $< 80 \text{ mA}$ y una potencia $< 264 \text{ mW}$.

10.1.2.3 Convertidor de voltaje.

El convertido elegido es permite mantener un ajuste rápido y seguro mediante un voltímetro y mostrar la tensión de salida mediante un display. Las características técnicas se muestran a continuación.

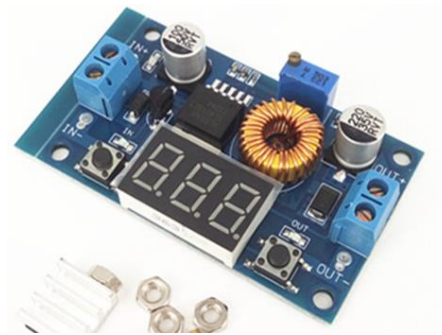


Figura 10- 5 Convertidor DC/DC.

10.1.2.3.1 Especificaciones técnicas conversor DC/DC.

Parámetro	
Tensión entrada.	4 - 38VDC.
Tensión salida.	1.35–36 VDC (continuamente ajustable)
Corriente de salida max.	5 A.
Potencia de salida.	Hasta 75 W.
Error voltímetro.	±0.05 V.
Eficiencia.	96%
Regulación de cargas.	$S(I) \leq 0.8\%$
Regulación de voltajes.	$S(u) \leq 0.8\%$
Dimensiones.	6.6*3.9*1.8 cm
Protección.	Sobrecalentamiento y Corto Circuito.

Tabla 10- 2 Datos técnicos conversor DC/DC

10.1.2.4 Programación Comandos AT.

Es un lenguaje desarrollado por la compañía Hayes Communications que prácticamente se convirtió en estándar abierto de comandos para configurar y parametrizar módems. Los caracteres “AT”, que preceden a todos los comandos, significan “Atención”, e hicieron que se conociera también a este conjunto de comandos como **comandos AT**.

A continuación, se presenta la lista completa de los comandos admisibles por el módulo WI-FI para realizar comunicación y transferencia de datos con software de Arduino.

AT+RST - Reinicia el módulo		
AT+RST	OK	Resetea el módulo

AT+CWMODE - Modo Wifi		
¿AT+CWMODE=?	+CWMODE:(1-3) OK	Lista los modos validos
¿AT+CWMODE?	+CWMODE: modo OK	Pregunta en qué modo AP está actualmente el módulo

AT+CWMODE=modo	OK	Establece el módulo en el modo dado 1 = Modo estación (cliente) 2 = Modo AP (huésped) 3 = Modo AP + Estación (modo dual)
AT+CWLAP - Lista APs disponibles		
AT+CWLAP	AT+CWLAP:ecn,ssid, rssi,mac OK	Lista los Access Points disponibles para conectarse. ecn: codificación, puede ser: 0 = Abierto 1 = WEP 2 = WPA PSK 3 = WPA2 PSK 4 = WPA WPA2 PSK ssid: String que contiene el SSID del AP rssi: Fuerza de la señal mac: String que contiene la dirección MAC
AT+CWLAP=ssid,mac,ch	+CWLAP:ecn,ssid, rssi,mac OK	Busca Access Points disponibles para conectarse con las condiciones especificadas

AT+CWJAP - Unirse a un Access Point		
AT+CWJAP?	+ CWJAP:ssid OK	Imprime el SSID al que el módulo está conectado
AT+CWJAP=ssid,pwd	OK	El módulo se conecta a la red con el nombre ssid indicado y la contraseña pwd suministrada

AT+CWQAP - Desconectarse de una Access Point		
AT+CWQAP	OK	Se desconecta de la red que está actualmente conectado

AT+CWSAP - Configurar el softAP del módulo		
AT+CWSAP?	+CWSAP:ssid,pwd,ch,ecn OK	Pregunta la configuración actual del softAP
AT+CWSAP=ssid,pwd,ch,ecn	OK	Configura el softAP con ssid: String con el nombre de la red pwd: Contraseña, no mayor a 64 caracteres ch: Canal inalámbrico ecn: Tipo de codificación 1 = Abierto 2 = WPA_PSK 3 = WPA2_PSK 4 = WPA_WPA2_PSK

STATUS - Información acerca de la conexión

STATUS	status STATUS:id,type,addr tetype	status: 2 = Se obtuvo IP 3 = Conectado 4 = Desconectado ID de la conexión en caso de multiconexión (1-4) type: Tipo de conexión, "TCP" o "UDP" addr: Dirección IP de la conexión port: Numero del puerto tetype: 0 = El módulo corre como cliente 1 = El módulo corre como servidor
---------------	---	---

AT+CIPMUX - Habilitar o deshabilitar multiples conexiones		
AT+CIPMUX=mode	OK	mode: 0 = Conexión unica 1 = Múltiples conexiones, hasta 4
AT+CIPMUX?	+CIPMUX:mode OK	Imprime el mode, el modo de conexión actual

AT+CIPSTART - Establece una conexión TCP o registra un puerto UDP e inicia la conexión		
AT+CIPSTART=type,addr,port	OK	Empieza una conexión como cliente (en modo conexión única) type: puede ser "TCP" o "UDP" addr: String que contenga la dirección IP remota port: String que contenga el puerto remoto
AT+CIPSTART=id,type,addr,port	OK	Empieza una conexión como cliente (En modo conexión múltiple) id: ID de la conexión (1-4)
AT+CIPSTART=?	[+CIPSTART:(id)("type"),("ip address"),(port)] OK	Lista los posibles comandos

AT + CIPCLOSE - Cierra la conexión TCP o UDP		
AT+CIPCLOSE=?	OK	
AT+CIPCLOSE=id	OK	Cierra la conexión TCP o UDP con el ID "id" en modo conexión múltiple
AT+CIPCLOSE	OK	Cierra la conexión TCP o UDP para modo de conexión única

AT+CIPSEND - Enviar datos		
AT+CIPSEND=?	OK	
AT+CIPSEND=length	SEND OK	Establece la longitud de datos a enviarse (máximo 2048). Para un envío normal (modo conexión única)
AT+CIPSEND=id,length	SEND OK	Establece la longitud de datos a enviarse en la conexión número "id". Para un envío normal (modo conexión múltiple)
AT+CIPSEND		Envía datos sin adornos cada 20ms. El módulo retorna ">" después ejecutar el comando, si se recibe el comando "+++" se regresa al modo comando.

AT+CIFSR - Obtener la dirección IP local		
AT+CIFSR=?	OK	
AT+CIFSR	+CIFSR:ip OK	Retorna la dirección IP local del módulo como cliente.
AT+CIFSERVER - Configurar como servidor		
AT+CIPSERVER=mode[,port]	OK	Configura el módulo como servidor donde el modo: 0 = Borrar servidor 1 = Crear servido puerto: número del puerto, por defecto es el 333

AT+CIOBAUD Cambiar la velocidad de transmisión serial		
AT+CIOBAUD=?	+CIOBAUD:(9600-921600) OK	Nos informa que las velocidades de transmisión permitidas están en este rango
AT+CiOBAUD?	+CIOBAUD:baudrate OK	Nos indica que el módulo está actualmente configurado a 'baudrate'
AT+CIOBAUD=baudrate	OK	Configura la velocidad de transmisión a 'baudrate'

Tabla 10- 3 Comandos AT.

CAPÍTULO 11. SISTEMA DE ALIMENTACIÓN Y TABLERO DE CONTROL.

11.1 ALIMENTACIÓN SISTEMA DE CONTROL.

La elección de la fuente de alimentación se debe considerar fundamental ya que es el elemento más importante dentro del sistema, esta debe suministrar una tensión continua de 12 V, y que el dimensionamiento de esta debe ser acorde a lo consumido (watts) del sistema general, incluyendo los sensores conectados al microcontrolador principal. También debe basarse la elección, en la corriente necesaria para el conjunto de elementos, es decir, la sumatoria de todas las corrientes máximas.

El tamaño, considerado como requisito en todos los elementos del sistema, debe ser acorde, esto básicamente se refleja en el tipo de fuente que se considerará.

Otro aspecto relevante y que se debe tomar en cuenta a la hora de elegir, es el factor de rizado, el cuál es un indicador de la efectividad de filtro que posee la fuente. Y así obtener una señal de tensión continua más pura, atenuando o eliminando componentes de alterna, Los cuales pueden interferir en las señales de control o directamente en el microcontrolador, entregando señales de medición y lecturas erróneas de acuerdo a la realidad del proceso, por ende, crear una desestabilización del proceso general.

Cuanto menor sea el factor de rizado, mejor será el filtro. El factor de rizado puede reducirse incrementando el valor del condensador del filtro.

11.1.1 Potencia de suministro necesaria.

Para dimensionar la potencia necesaria para la fuente es necesario saber las potencias máximas de todas las cargas, esto incluye el microcontrolador, los sensores, relés y la bomba nebulizadora. A continuación, se presenta la tabulación de datos correspondientes a los consumos de cada elemento.

Elemento	Potencia	Nº elementos	Potencia total.
Arduino	552 mW	1	552 mW
Sensor Humedad DHT22	12,5 mW	2	25 mW
Sensor Tº DHT22	12,5 mW	2	25 mW
Sensor Humedad suelo FC-28	100 mW	2	200 mW
Bomba nebulizadora	60 W	1	60 W
Pantalla y módulo I2C	125 mW	1	125 mW
Módulo Wi-Fi	264 mW	1	264 mW
Relé	100 mW	4	400 mW
		Total	61,591 W

Tabla 11- 1 Tabulación potencias.

Dada la tabulación anterior se obtiene la totalidad de potencia consumida por el sistema, se puede decir que la potencia de suministro de la fuente debe ser como mínimo de 62 Watts aproximadamente.

11.1.2 Corriente de suministro necesaria.

Uno de los parámetros que siempre entrega la fuente es la corriente que esta es capaz de suministrar, por ende, es imperativo tener claro cuál es el consumo de corriente que tendrá el sistema. Para ello solo se necesita realizar la sumatoria de las corrientes de consumo de cada elemento a energizar por la fuente.

A continuación, se presenta la tabulación y sumatoria de corrientes de cada elemento, los cuales son entregados por su respectivo datasheet.

Elemento	Corriente	Nº elementos	Corriente total.
Arduino	46 mA	1	46 mA
Sensor Humedad DHT22	2,5 mA	2	5 mA
Sensor Tº DHT22	2,5 mA	2	5 mA
Sensor Humedad suelo FC-28	20 mA	2	40 mA
Bomba nebulizadora	5 A	1	5 A
Pantalla y módulo I2C	25 mA	1	25 mA
Módulo Wi-Fi	80 mA	1	80 mA
Relé	20 mA	4	80 mA
		Total	6,281 A

Tabla 11- 2 Tabla consumo de corrientes.

Dada la tabulación anterior se obtiene la totalidad de corriente consumida por el sistema, se puede decir que el suministro mínimo de corriente de la fuente debe ser de 6.3 Amperes.

11.2 Tipo de fuente.

Dentro de la categorización de las fuentes de alimentación podemos encontrar las denominadas lineales (fuente PC, fuente de poder variable), estas son usadas generalmente para proyectos en laboratorio o pruebas de equipamientos. tienen un diseño relativamente simple, que puede llegar a ser más complejo cuanto mayor es la corriente que deben suministrar, sin embargo su regulación de tensión es poco eficiente.

Las fuentes conmutadas o "Switching"., de la misma potencia que una lineal, será más pequeña y normalmente más eficiente, pero será más compleja y por tanto más susceptible a averías. En tableros generalmente se usa este tipo, encontrando una gran variedad en el mercado para este objetivo, ya sea para montaje en riel din o anclaje a placa. Siendo el montaje riel din más caro.

La elección no resulta difícil, pero para tener claro los requerimientos más importantes del proyecto, se comparará Eficiencia, Tamaño y Precio de dichas fuentes de alimentación.

11.2.1 Fuentes Lineales.



Figura 11- 1 Fuentes lineales.

- Eficiencia.

En aplicaciones normales las fuentes lineales tienen una eficiencia entre en 30 al 60%. Esto significa que por cada Watt los costos irán aumentando.

- Tamaño.

El solo hecho de que posee un transformador interno, la hace significativamente más robusta en comparación a una fuente conmutada.

- Precio.

Basándose en los requerimientos de tensión, corriente y potencia dentro del mercado se encuentran precios desde los 100 mil pesos aproximadamente, dependiendo de la calidad.

11.2.2 Fuentes conmutadas.



Figura 11- 2 Fuentes conmutadas.

- Eficiencia.

Su eficiencia estándar está entre el 68 y el 90%. Esto hace reducir significativamente el costo de los dispositivos de potencia.

- Tamaño.

Ya que mayormente se encuentra construida por elementos electrónicos y la falta de transformador, la hacen significativamente más pequeña de una fuente lineal de las mismas características.

- Precio.

Basándose en los requerimientos de tensión, corriente y potencia dentro del mercado se encuentran precios desde.

Fuente montaje riel din: 50 mil pesos aproximadamente

Fuente de gabinete: 8 mil pesos aproximadamente.

11.3 ELECCIÓN FINAL FUENTE ALIMENTACIÓN.

Fácilmente se puede concluir que la mejor opción para el proyecto es una fuente conmutada, ya que su eficiencia es notoriamente mejor a la de una fuente lineal, su tamaño reducido, precio inferior. Y destacar que es la más comúnmente usada en gabinetes, por ello las características necesarias serán.

Fuente de alimentación conmutada o Switching.
Corriente de salida mínimo 6,3 A. De preferencia <10 A para obtener un grado de ampliación importante.
Potencia de suministro mínimo 63 Watts, preferente <100 W para obtener un grado de ampliación importante
Factor de rizado pequeño

Para tener una buena elección final es necesario identificar las características individuales de cada tipo de fuente. Dentro del mercado existe una infinidad de productos y precios, pero cabe destacar que están relacionados por precio calidad, y la calidad se ve reflejada muchas veces en la señal de salida, pudiendo tener una fuente de bajo precio un factor de rizado alto, pudiendo esto afectar directamente en las señales de control del proceso.

11.3.1 Fuente conmutada montaje riel din.

Esta fuente es un poco más industrial, por ende, está diseñada para alimentar elementos más específicos, en áreas donde se necesite una mayor protección, sin duda la calidad de la fuente es mejor, por ende, su precio es bastante elevado.

Su montaje fácil y rápido permite montarlo en un riel din sin problemas.

11.4 Fuente conmutada montaje en placa.

Ya que el proyecto está basado en prototipo, no es necesario utilizar una fuente de gama muy alta, y ya que la fuente riel din es de ese tipo. es conveniente utilizar una fuente conmutada de gabinete, la cual se adapta completamente al proyecto.

Esta fuente es más común y se utiliza principalmente en proyectos de gabinetes más pequeños, proyectos de laboratorio, etc.

Su calidad es acorde a su precio. Siendo de gama media en su tipo. Su montaje es mediante pernos de anclaje.

Una de las marcas más conocida de venta de equipamiento de automatización industrial es "Men well", los cuales poseen fuentes de poder de diferentes gamas, donde se encuentran según montaje, calidad, parámetros de voltaje, potencia y corriente.

En la gama media se puede encontrar las fuentes de gabinete, y en dicha categoría y según los requisitos necesarios anteriormente descritos; V salida:12 VDC, $C_{suministrada}$: <10 A y $P_{suministrada}$: <100 w. Se encuentra una fuente de poder NES-150 series.

A continuación, se muestra y tabula los datos técnicos de la fuente de poder Mean well NES-150 series.



Figura 11- 3 Fuente conmutada montaje en placa.

11.4.1 Datos técnicos.

Parámetro.	
Tensión entrada.	180-264 VAC.
Tensión salida.	12 VDC.
Corriente suministro.	12,5 A.
Potencia suministro.	150 W.
Eficiencia.	83%.
Voltaje rizado.	120 mV.
Tº operación.	-20 a 60 °C
Humedad operación.	20-90%

Tabla 11- 3 Datos técnicos fuente conmutada montaje en placa

El factor de rizado de la fuente se debe calcular, para así obtener una noción exacta de cuanto es el riesgo de que la señal contenga excesiva componente de alterna, y pueda afectar en las lecturas.

A continuación, se realiza el correspondiente cálculo.

$$r = 100 * \left(\frac{V_r}{V_{cd}}\right) ; \text{ Donde : } r = \text{Factor de rizado.}$$

Vr= Voltaje de rizado eficaz.

Vcd= Voltaje total de salida.

11.4.2 Cálculo factor de rizado.

$V_r = 120 \text{ mV} / 1000 = 0.12 \text{ V.}$

$V_{cd} = (V_{\text{salida}} + V_r) = (12 \text{ V} + 0.12 \text{ V}) = 12.12 \text{ V.}$

$$r = 100 * \left(\frac{0.12}{12.12}\right) = 0.99 \%$$

El factor de rizado equivale a solo el 0,99 % de la tensión de salida de la fuente, por ende, es un componente de corriente alterna muy pequeña que altera la señal, no teniendo problemas de ruido eléctrico. Por ellos obteniendo una alimentación óptima

11.5 TABLERO CONTROL.

Los componentes básicos de un gabinete son regletas, riel din, prensas estopas y necesariamente su placa de montaje.

Es necesario una placa de montaje, la cual puede ser de material blando o metálico, para así realizar un montaje seguro sobre este, o en riel DIN, ya que algunos componentes como por ejemplo Arduino, poseen contactos descubiertos. Es necesario que la placa sea aislada.

Pero para mayor seguridad en cuanto a asegurar un buen contacto y un fácil montaje al Riel DIN, se opta por usar un módulo adaptador de bloque de terminales de tornillo para Arduino UNO R3. La cual es una PCB, diseñada para montarla al riel y permitir conectar a través conectores con tornillo que permiten una gama más amplia de cableado, y no limitando solo a conectores "Dupont". También permite una fácil extracción de la placa de desarrollo, en caso de alguna falla o daño, este puede ser fácilmente reemplazado.

Módulo adaptador riel Din para Arduino UNO R3.

Difícil de encontrar, se puede generalmente en páginas de ventas extranjeras.

A continuación, se entregan las especificaciones técnicas del elemento.

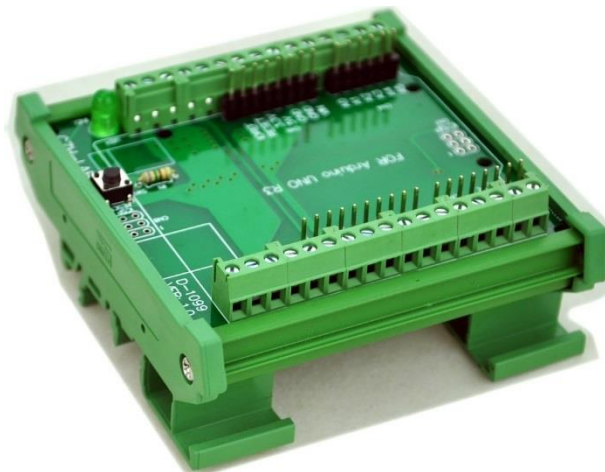


Figura 11- 4 Módulo adaptador Arduino UNO R3 montaje Riel DIN.

Datos Técnicos.

Material construido.	Fibra de vidrio FR-4, capas cobre doble.
Bloque terminales.	Rango de cables 26-16 AWG.
Dimensiones PCB.	80mm x 72.5mm (W x L).
Dimensiones módulo.	83mm x 87mm x 45mm (a x L x H).

Tabla 11- 4 Datos técnicos módulo Arduino riel DIN.



Figura 11- 5 Módulo montado sobre Riel DIN.

Hay elementos para los cuales no existen los módulos de montaje a riel din, como lo son: Fuente de poder 12V, módulo relés, pantalla 16x2 y conversor DC/DC. Por ende, se debe realizar el montaje sobre separadores de placas PCB, los cuales permiten mantener una aislación entre masa y las placas PCB, tales como: Sensor FC-28, Módulo Wi-Fi y pantalla LCD. A continuación, se presenta dicho elemento.



Figura 11- 6 Separador roscado hexagonal M3.

11.6 GABINETE.

Se considera la elección del tablero bajo la norma NCH 4 2003, específicamente los puntos:

6.2.1.8.- El tamaño de caja, gabinete o armario se seleccionará considerando que:

- El cableado de interconexión entre sus dispositivos deberá hacerse a través de bandejas no conductoras que permitan el paso cómodo y seguro de los conductores.
- Deberá quedar un espacio suficiente entre las paredes de las cajas, gabinetes o armarios y las protecciones o dispositivos de comando y/o maniobra de modo tal de permitir un fácil mantenimiento del tablero.
- Se deberá considerar un volumen libre de 25% de espacio libre para proveer ampliaciones de capacidad del tablero.

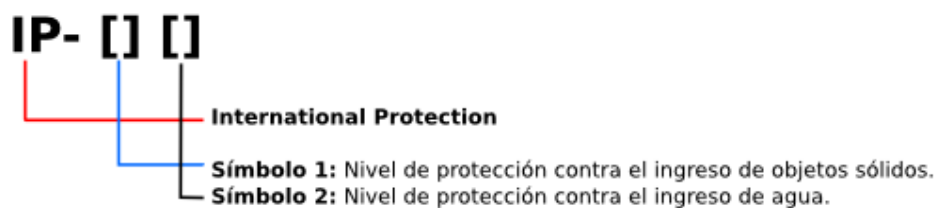
6.2.1.9.- Las cajas, gabinetes o armarios en que se monten los tableros podrán ser contruidos con placas de acero o materiales no conductores.

11.6.1 Protección IP.

También se considera un grado de protección al ambiente. Para ello se hace referencia a la norma internacional IEC 60529, la cual indica el índice de protección IP, utilizado generalmente en los datos técnicos de equipamientos, en uso general industrial.

Este estándar ha sido desarrollado para calificar de una manera alfa-numérica a equipamientos en función del nivel de protección que sus materiales contenedores le proporcionan contra la entrada de materiales extraños. Mediante la asignación de diferentes códigos numéricos, el grado de protección del equipamiento puede ser identificado de manera rápida y con facilidad.

11.6.1.1 Nomenclatura IEC 60529.



- La norma IEC 60529 establece para el primer dígito de que el equipo a ser certificado debe cumplir con alguna de las siguientes condiciones

Nivel	Tamaño del objeto entrante	Efectivo contra
0	—	Sin protección
1	<50 mm	El elemento que debe utilizarse para la prueba (esfera de 50 mm de diámetro) no debe llegar a entrar por completo.
2	<12.5 mm	El elemento que debe utilizarse para la prueba (esfera de 12,5 mm de diámetro) no debe llegar a entrar por completo.
3	<2.5 mm	El elemento que debe utilizarse para la prueba (esfera de 2,5 mm de diámetro) no debe entrar en lo más mínimo.
4	<1 mm	El elemento que debe utilizarse para la prueba (esfera de 1 mm de diámetro) no debe entrar en lo más mínimo.
5	Protección contra polvo	La entrada de polvo no puede evitarse, pero el mismo no debe entrar en una cantidad tal que interfiera con el correcto funcionamiento del equipamiento.
6	Protección fuerte contra polvo	El polvo no debe entrar bajo ninguna circunstancia

Tabla 11- 5 Niveles protección contra sólidos.

- La norma IEC 60529 establece para el segundo dígito que el equipo a ser certificado debe cumplir con alguna de las siguientes condiciones.

Nivel	Protección frente a	Método de prueba	Resultados
0	Sin protección.	Ninguno	El agua entrará en el equipamiento en poco tiempo.
1	Goteo de agua	Se coloca el equipamiento en su lugar de trabajo habitual.	No debe entrar el agua cuando se la deja caer, desde 200 mm de altura respecto del equipo, durante 10 minutos (a razón de 3-5 mm ³ por minuto)
2	Goteo de agua	Se coloca el equipamiento en su lugar de trabajo habitual.	No debe entrar el agua cuando se la deja caer, durante 10 minutos (a razón de 3-5 mm ³ por minuto). Dicha prueba se realizará cuatro veces a razón de una por cada giro de 15° tanto en sentido vertical como horizontal, partiendo cada vez de la posición normal de trabajo.
3	Agua nebulizada. (spray)	Se coloca el equipamiento en su lugar de trabajo habitual.	No debe entrar el agua nebulizada en un ángulo de hasta 60° a derecha e izquierda de la vertical a un promedio de 11 litros por minuto y a una presión de 80-100 kN/m ² durante un tiempo que no sea menor a 5 minutos.
4	Chorros de agua	Se coloca el equipamiento en su lugar de trabajo habitual.	No debe entrar el agua arrojada desde cualquier ángulo a un promedio de 10 litros por minuto y a una presión de 80-100 kN/m ² durante un tiempo que no sea menor a 5 minutos.
5	Chorros de agua.	Se coloca el equipamiento en su lugar de trabajo habitual.	No debe entrar el agua arrojada a chorro (desde cualquier ángulo) por medio de una boquilla de 6,3 mm de diámetro, a un promedio de 12,5 litros por minuto y a una presión de 30 kN/m ² durante un tiempo que no sea menor a 3 minutos y a una distancia no menor de 3 metros.
6	Chorros muy potentes de agua.	Se coloca el equipamiento en su lugar de trabajo habitual.	No debe entrar el agua arrojada a chorros (desde cualquier ángulo) por medio de una boquilla de 12,5 mm de diámetro, a un promedio de 100 litros por minuto y a una presión de 100 kN/m ² durante no menos de 3 minutos y a una

			distancia que no sea menor de 3 metros.
7	Inmersión completa en agua.	El objeto debe soportar sin filtración alguna la inmersión completa a 1 metro durante 30 minutos.	No debe entrar agua.
8	Inmersión completa y continua en agua.	El equipamiento eléctrico / electrónico debe soportar (sin filtración alguna) la inmersión completa y continua a la profundidad y durante el tiempo que especifique el fabricante del producto con el acuerdo del cliente, pero siempre que resulten condiciones más severas que las especificadas para el valor 7.	No debe entrar agua.
9K	Potentes chorros de agua a alta temperatura	Protegido en contra de chorros de corto alcance a alta presión y de alta temperatura.	Duración del Test: Volumen de agua: 14-16 litros por minuto Presión: [8000-10000 kPa / 80-100 Bar] distancia de 0.1-0.15 m Temperatura del agua: 80 C

Tabla 11- 6 Niveles protección contra líquidos.

Dado que los recintos invernaderos se encuentran a intemperie, se considera el gabinete del sistema también ubicado a intemperie, entonces es necesario designar un tipo de tablero con un índice de protección necesario para protegerlo de las condiciones climáticas.

Para ello el requerimiento mínimo de índice de protección, equivale a IP 65, ya que según la nomenclatura el número 6 es el grado máximo de protección contra el polvo y 5

11.6.2 Tamaño.

El tamaño es seleccionado bajo la norma chilena 4/2003 punto **6.2.1.8** antes descrito.

La elección es de forma aproximada al conocer el tamaño real de los elementos a instalar dentro del tablero. Para ello se creó un esquema aproximado a mano alzada donde se determina que el tamaño requerido es aproximadamente de 500 x 450 mm, la profundidad del tablero en este proyecto no es considerada ya que ningún elemento es demasiado robusto.

Se selecciona un tablero que cumple con lo requerido el cual posee las siguientes características:

Ficha técnica
Libre de halógenos.
Paneles desmontables
Grado de protección IP66.
Sujetador y tornillo ¼ de vuelta.
Adecuado para instalaciones de -25º a +60º C
Capacidad de disipación de calor 72 W
Elevada robustez.
Con placa de montaje.

Tabla 11- 7 Ficha técnica tablero control.

11.6.3 Caja de paso.

Para ahorro de cables y menos perforaciones en el gabinete, es conveniente usar un tablero de paso el cual permitirá la distribución de 5 volts correspondiente a la alimentación de los sensores, y también a la adquisición de los cables de señales provenientes de los sensores, DHT22 y FC-28.

Dicho tablero consta con 4 bornes de 2 niveles correspondiente a la alimentación 5 Volts y 3 bornes 2 niveles para las 6 señales de los sensores. También debe instalarse los acondicionadores de señal de los sensores FC-28.

11.7 PROTECCIONES.

Los sistemas de Protección se utilizan en los sistemas eléctricos de potencia para evitar la destrucción de equipos o instalaciones por causa de una falla que podría iniciarse de manera simple y después extenderse sin control en forma encadenada. Los sistemas de protección deben aislar la parte donde se ha producido la falla buscando perturbar lo menos posible la red, limitar el daño al equipo fallado, minimizar la posibilidad de un incendio, minimizar el peligro para las personas, minimizar el riesgo de daños de equipos eléctricos adyacentes.

Dentro del proyecto se usarán dos tipos de protecciones; Disyuntores y fusibles. Los disyuntores generalmente usados para equipos de fuerza y los fusibles para equipos electrónicos.

A continuación, se presenta los cálculos de protecciones correspondiente al sistema total y sub sistemas presentes

Según la norma Nch 4/2003 específicamente el punto:

12.2.6.- En grupos de motores en que existan motores de régimen permanente, periódico, intermitente y/o de breve duración, la sección de los conductores que alimentan al grupo deberá permitir una capacidad de transporte para una corriente que se determina como sigue:

- La suma de las corrientes de plena carga de los motores de régimen permanente, más
- La suma de las corrientes de plena carga de los motores de régimen no permanente, afectada por el factor que corresponda, determinado de acuerdo a la tabla Nº 12.28, más •
- 0,25 veces la corriente de plena carga del motor de mayor potencia afectada por el factor correspondiente de acuerdo a la tabla Nº 12.28 si el motor no es de régimen permanente

Entonces ya que el sistema consta de diferentes tipos de motores se debe considerar una corriente de corto circuito del 25 % mayor a la corriente máxima del equipo y/o circuito a proteger.

A partir de esto se tabula y se calcula las protecciones para los siguientes circuitos del sistema.

Circuito	Corriente máxima	Corriente protección	Tipo protección	Recomendada
General	6 A	7,85 A	Interruptor	16 A
Calefactor	6.8 A	8.5 A	Interruptor	10 A
Extractor	55 mA	68,8 mA	fusible	80 mA
Arduino	46 mA	58 mA	fusible	60mA
Bomba	5 A	6 A	Interruptor	6 A
Módulo relés	80 mA	100 mA	fusible	100 mA
Fuente 12 V.	5 A	6,4 A	fusible	7 A

Tabla 11- 8 Valor protecciones.

Los valores de corriente resultantes deben ser ajustados a los existentes en el mercado, ya que será difícil encontrar valores tan específicos.

También destacar la elección de bornes portafusibles para el menor uso de espacio dentro del tablero de control.

11.7.1 Curva de disparo.

Las curvas de disparos para las protecciones para interruptores automáticos según la norma UNE-EN 60898 define la corriente a la que debe suceder la desconexión del sistema, estas se clasifican como:

- **Curva B:** Protección de generadores, de personas y grandes longitudes de cables.
Sobrecarga: térmico estándar.
Cortocircuito: I_m entre 3 y 5 I_n o 3,2 y 4,8.
- **Curva C:** Protección de cables alimentando receptores básicos.
Sobrecarga: Térmico estándar.
Cortocircuito: I_m entre 5 y 10 I_n o 7 y 10.
- **Curva D:** Protección de cables alimentando receptores con fuertes peak arranque
Sobrecarga: Térmico estándar.
Cortocircuito: I_m entre 10 y 14 I_n .
- **Curva MA:** Protección arranque de motores.
Sobrecarga: NO hay protección.
Cortocircuito: I_m fijado a 12 I_n .
- **Curva Z:** Protección de circuitos electrónicos.
Sobrecarga: Térmico estándar.
Cortocircuito: I_m entre 2,4 y 3,6 I_n .

Donde:

I_m = Intensidad de disparo magnético.

I_n = Intensidad de regulación de disparo magnético.

Ya que los elementos son elementos básicos, la corriente y protección general calculada se aproxima a 10 A, la curva apropiada es CURVA C. La cual debe ser elegida en todas las protecciones.

11.8 CANALIZACIÓN.

La canalización tanto interna como externa al tablero se realizará mediante elementos que permitan la protección y orden de los conductores tanto de alimentación como de control.

Estos elementos se clasifican como: Canalización Interna y Canalización Externa.

11.8.1 Canalización interna.

Se realiza mediante canaletas ranuradas las cuales son diseñadas para el interior de tableros eléctricos. La cual permite un fácil montaje y tendido de cables, fácil reconocimiento de cables y orden estético al tablero.

El tamaño es variado, adecuándose al volumen de conductores a instalar. Y ya que el volumen es menor, no es necesario una canaleta muy grande. Por ello se consideran las medidas 25x20 mm x 2m.



Figura 11- 7 Canaleta ranurada.

11.8.2 Canalización externa.

Dado que las condiciones climáticas y ambientales influyen directamente en los conductores que deben salir del tablero, es necesario un tipo de canalización que mantenga un buen aislamiento y protección al medio ambiente.

Por ello es que la canalización exterior se realizará mediante Conduit flexible hermético, compuesto por cinta de acero galvanizado y con revestimiento externo de polivinilo negro, con protección ip 66, el cual permite la instalación en intemperie en

lugares húmedos y polvorientos. También resistente a la abrasión y esfuerzos mecánicos.

Dado que la canalización debe permitir transportar los cables desde dentro del gabinete, se debe instalar prensas estopas las cuales permiten conservar la protección IP de este, las cuales deben ser consideradas con un diámetro acorde al conduit.



Figura 11- 8 Conduit hermético 1”.

11.9 CONDUCTORES.

11.9.1 Dimensionamiento cables alimentación.

Un mal dimensionamiento de cable puede producir un funcionamiento anormal del sistema eléctrico, generando problemas tales como:

- Variaciones de voltaje.
- Pérdida de energía.
- Sobrecalentamiento de líneas.

Los principales factores que se deben considerar para realizar un correcto dimensionamiento son:

- Que la sección del conductor pueda transportar la corriente necesaria.
- Que la caída de tensión esté dentro de los parámetros de funcionamiento del equipo eléctrico (norma eléctrica).

Los alimentadores deben ser calculados bajo la norma chilena, la cual indica la caída de tensión permisible, en el punto:

7.1.1.3.- La sección de los conductores de los alimentadores o sub-alimentadores será tal que la caída de tensión provocada por la corriente máxima que circula por ellos determinada, no exceda del 3% de la tensión nominal de la alimentación, siempre que la caída de tensión total en el punto más desfavorable de la instalación no exceda del 5% de dicha tensión.

Cálculos.

Lo necesario para comenzar es saber el valor del permisible de caída de tensión, el cual es el 3% de la tensión de alimentación nominal, en este caso 220 VAC. Entonces:

$$\Delta V = 220 \text{ VAC} * 0,03 = 6,6 \text{ VAC.}$$

Donde “ ΔV ” es la caída de tensión permisible.

Siendo está de 6,6 volts, es decir la mínima tensión en la carga debe ser de 214 VAC. Para mantener el circuito bajo norma.

A continuación, se realiza el cálculo de sección basado en caída de tensión

$$S = \frac{2 \cdot \rho \cdot l \cdot P}{\Delta V \cdot V}$$

Donde:

S= Sección en milímetros cuadrados.

ρ = Resistividad material Cu.

l= Largo en metros dos líneas.

P= Potencia eléctrica.

ΔV = Caída de tensión permisible.

V= Tensión alimentación

Entonces:

$$S = \frac{2 \cdot 0,0171 \Omega \text{mm}^2/\text{m} \cdot 12 \text{m} \cdot 1662 \text{w}}{6,6 \text{ V} \cdot 220 \text{ V}} = 0,046 \text{ mm}^2$$

Es decir, la sección mínima necesaria para la alimentación es de 0,046 mm².

Importante.

Ya que no es posible encontrar una norma que se adecue perfectamente a los requerimientos de proyectos de control, se considera la norma chilena 4/2003 la cual indica instalaciones en baja tensión, la cual considera una sección mínima de instalación en el punto mostrado a continuación.

8.2.19.11.- La sección mínima de estos conductores será de 1,5 mm². Se autorizará el uso de secciones menores en caso de usar cables multiconductores.

Entonces considerando esta norma, se define una sección de 1,5 mm² para la alimentación en 220 V del sistema y cables de fuerza, estos son para alimentación de equipos, en este caso; Calefactor, extractor y bomba nebulizadora.



Figura 11- 9 Cordón alimentación.

11.9.2 Cable de control.

Al igual que los alimentadores, el cable de control a usar se puede definir mediante el cálculo de sección, pero ya que la norma chilena 4/2003 no especifica claramente "Cable de control" ni detalles de estas ni condiciones de uso ya que la sección mínima mencionada refiere a circuitos de fuerza, alimentación e iluminación.

Generalmente la sección del cable de control es menor a 1,5 mm² por ellos la nomenclatura del calibre se menciona en AWG, que significa "Calibre de alambre estadounidense" (en inglés american wire gauge).

Una medida común es 18 AWG, la cual es un valor intermedio para cables de control. A continuación, se presenta una tabla de conversión de dicho calibre

AWG	Diámetro	Área	R eléctric.en Cu	I admisible en Cu a 40º C	Equivalencia aprox. Estándar métrico.
	(mm)	(mm ²)	(Ω/Km)	(A)	
18	0.0403	0.823	20.948	5	>24/0.2

Tabla 11- 9 Conversión calibre AWG.



Figura 11- 10 Cable control.

11.9.3 Cable instrumentación.

Ya que los sensores que envían la información de las variables hacia el microcontrolador usan un hilo para hacerlo es necesario aislar la señal de datos.

Para esto es necesario un conductor especializado para señales, el cual es cable de instrumentación.

Dado que el número de señales son 6: dos señales de humedad relativa, dos señales de temperatura y dos de humedad de suelo; Es necesario y es óptimo usar un cable de instrumentación multiconductor (Cordón), el cual permitirá llevar las señales a través de solo una canalización y obtener una señal limpia sin ruidos, gracias a la pantalla que posee, la que puede ser pantalla colectiva o individual.

También se debe alimentar los sensores con una tensión de 5 volts provenientes desde la placa Arduino Uno R3, los que se conectarán a bornes dentro de la caja de paso ubicada en el invernadero, para esto se usará cable de instrumentación 1x 18 AWG (1 par, calibre 18 AWG), esto para evitar ruidos en la señal de alimentación.



Figura 11- 11 Cordón multipar apantallado.

CONCLUSIONES.

Como resultado del estudio de tecnologías y elementos que en conjunto permiten desarrollar el proyecto, se puede concluir que los objetivos específicos propuestos, fueron cumplidos en su totalidad.

Ya que se logró determinar un proyecto teórico en base a control cerrado que permiten la automatización de regadío, temperatura y humedad del proceso en su totalidad dentro de un recinto invernadero a escala.

La descripción de los elementos, tanto, controlador, como sensores y actuadores, también el costo monetario necesario para su implementación, añadiendo a esto, elementos de conexión, canalización y protección del sistema, todo con su debida justificación técnica.

A modo de ayuda se describió las variables de software las cuales permitirán la futura programación.

También se determinó como y con qué elementos se puede implementar el concepto "IOT" (internet de las cosas) a dicho proyecto, cuyo objetivo es un monitoreo remoto Destacar que esto es algo más complejo y que solo es permisible mediante conexión internet, siendo esta una limitación.

Finalmente se dejan planos de conexión y montaje, los cuales permitirán el desarrollo físico del proyecto.

BIBLIOGRAFÍA.

<https://aprendiendoarduino.wordpress.com/2016/12/21/arduino-conectado-a-internet/>

<http://rduinostar.com/documentacion/programacion-arduino/control-de-flujo-ii/>

<https://www.luisllamas.es/arduino-puerto-serie/>

<https://github.com/esp8266/Arduino/issues/2839>

<https://www.coldfire-electronica.com/esp/item/169/32/sensor-dht22-humedad-y-temperatura>

<http://www.madnesselectronics.com/tutorial-sensor-de-humedad-de-suelo/>

<https://www.solerpalau.com/es-es/blog/climatizacion-de-un-invernadero/>

<http://www.solerpalau.cl/es-cl/hojas-tecnicas-la-climatizacion-de-invernaderos/>

http://hydroenv.com.mx/catalogo/index.php?main_page=page&id=138

<http://www.zonaindustrial.cl/product/rs-75>

<https://repositorio.usm.cl/>

https://rua.ua.es/dspace/bitstream/10045/18432/1/Tema%201_Introduccion.pdf

<http://isa.uniovi.es/docencia/IngdeAutom/transparencias/Sistemas%20Automatizados.pdf>

