

**UNIVERSIDAD RICARDO PALMA  
FACULTAD DE INGENIERÍA  
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRONICA**



**DISEÑO DEL FITOTRON APLICANDO  
LOGICA DIFUSA**

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

**PRESENTADA POR**

**Bach. REYNA SALAS, RICHARD RONALD  
Bach. PUENTE CAMASCA, RAUL IGNACIO**

**Asesor: Ing. RIVAS LEÓN JAVIER HIPÓLITO**

**LIMA-PERÚ**

**2021**

## **DEDICATORIA**

A nuestros padres y familiares por mostrarnos el camino hacia la superación, y a mis compañeros por permitirnos aprender el valor de la amistad.

## **AGRADECIMIENTOS**

Un agradecimiento especial a todas las personas que valoran la electrónica como mis profesores que han inculcado año tras año un sentimiento especial por esta rama de la ingeniería.

A la familia, a nuestros padres y por supuesto a nuestro asesor Ing. Javier Rivas, que nos brindó su apoyo en la realización de esta tesis.

## ÍNDICE GENERAL

RESUMEN .....	X
ABSTRACT.....	XI
INTRODUCCIÓN.....	1
CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA.....	2
1.1. Presentación de la problemática.....	2
1.2. Formulación del problema.....	3
1.2.1. Problema general.....	3
1.2.2. Problemas específicos.....	3
1.3. Objetivos .....	3
1.3.1. Objetivo general.....	3
1.3.2. Objetivos específicos.....	3
1.4. Alcance y limitaciones de la investigación.....	3
1.5. Justificación de la investigación .....	4
1.5.1. Justificación de control de calidad .....	4
1.5.2. Justificación comercial.....	4
1.5.3. Justificación tecnológica .....	4
1.6. Estructura de la tesis .....	5
CAPÍTULO II: MARCO TEÓRICO .....	7
2.1. Antecedentes de la investigación.....	7
2.1.1. Antecedentes nacionales.....	7
2.1.2. Antecedentes internacionales.....	7
2.2. Selección de variables.....	7
2.3. Bases teóricas.....	8
2.3.1. Concepto de germinación.....	8
2.3.2. Condiciones de temperatura.....	9
2.3.3. Condiciones de humedad .....	11
2.3.4. Condiciones de luminosidad .....	11
2.3.5. Concepto de cámara de cultivo “fitotrón”.....	13
2.3.6. Concepto de controladores arduino.....	15

2.4. Características de los sensores y actuadores elegidos.....	18
2.4.1. Sensor para el control de temperatura.....	18
2.4.2. Sensor para el control Humedad.....	19
2.4.3. Sensor para el control luminosidad.....	21
2.4.4. Actuador para incrementar la temperatura.....	23
2.4.5. Actuador para bajar la temperatura.....	24
2.4.6. Actuador de Luminosidad.....	27
2.5. Características de control clásico y avanzado.....	28
2.5.1. Control Clásico ON/OFF.....	29
2.5.2. Control Clásico PID.....	29
2.5.3. Control avanzado lógica difusa.....	32
2.6. Características del software.....	38
2.6.1. Software Eclipse.....	38
2.6.2. Matlab.....	38
CAPÍTULO III: DISEÑO Y SIMULACIÓN.....	40
3.1. Requerimientos del diseño.....	40
3.2. Software a emplear.....	40
3.3. Control difuso de la temperatura.....	40
3.3.1. Declaración de las variables lingüísticas.....	40
3.3.2. Declaración de las reglas de la lógica difusa.....	42
3.3.3. Salidas y simulación.....	44
CAPITULO IV: IMPLEMENTACIÓN Y RESULTADOS.....	47
4.1. Estructura del fitotrón.....	47
4.1.1. Materiales a usar.....	47
4.1.2. Circuito de calefacción.....	50
4.1.3. Circuito de refrigeración.....	51
4.2. Implementación del sistema de control de la temperatura.....	53
4.3. Implementación del sistema de monitoreo de humedad.....	55
4.4. Implementación del sistema de monitoreo de luminosidad.....	56
4.5. Costos de la implementación.....	61
CONCLUSIONES.....	63
OBSERVACIONES.....	64
RECOMENDACIONES.....	65

BIBLIOGRAFÍA Y REFERENCIAS .....	66
ANEXOS .....	70

## LISTA DE TABLAS

Tabla 1: Variable independiente .....	8
Tabla 2: Variable dependiente .....	8
Tabla 3: Temperatura y días en que tarda en germinar la semilla .....	10
Tabla 4: Comparación y elección del controlador .....	17
Tabla 5: Cuadro comparativo de sensores MCP9701 y LM35.....	19
Tabla 6: Cuadro comparativo de sensores HIH4000 y DHT11 .....	21
Tabla 7: Cuadro comparativo de sensores Temt6000 y Tls2561.....	23
Tabla 8: Declaración de reglas.....	42
Tabla 9: Normas difusas para la temperatura .....	44
Tabla 10: Materiales empleados para la construcción del fitotrón .....	48
Tabla 11: Lista de materiales empleados y precios .....	61
Tabla 12: Costo del diseño del fitotrón.....	62
Tabla 13: Costo energético del fitotrón .....	62

## LISTA DE FIGURAS

Figura 1: Estructura de la tesis.....	5
Figura 2: Imagen de una germinación .....	9
Figura 3: Frecuencia de riego .....	11
Figura 4: Efecto de la luz en el desarrollo de una planta .....	13
Figura 5: Fitotrón australiano (1961).....	14
Figura 6: Esquema descriptivo de entradas y salidas de un microcontrolador .....	16
Figura 7: Arduino Mega 2560 .....	16
Figura 8: Sensor MCP9701.....	18
Figura 9: Sensor de humedad usado en el proyecto.....	20
Figura 10: Respuesta lineal del sensor HIH 4000.....	20
Figura 11: Sensor de luminosidad temt 6000 .....	22
Figura 12: Sensibilidad del sensor de luminosidad .....	22
Figura 13: Características del alambre de nicrom .....	24
Figura 14: Funcionamiento de una celda de peltier .....	25
Figura 15: Códigos de la celda de Peltier .....	27
Figura 16: Consumo de energía para variedades de emisores de luz .....	28
Figura 17: Comportamiento de un control ON/OFF .....	29
Figura 18: Características de la señal PID .....	30
Figura 19: Esquema de un controlador PID.....	31
Figura 20: Aplicaciones de la lógica difusa.....	32
Figura 21: Operación de complemento.....	33
Figura 22: Operación de unión de dos conjuntos. ....	33
Figura 23: Operación de intersección de dos conjuntos. ....	34
Figura 24: Configuración básica de un sistema tipo Mamdani .....	35
Figura 25: Función de pertenencia.....	35
Figura 26: Forma de las reglas.....	36
Figura 27: Sistema de inferencia tipo Mamdani .....	37
Figura 28: Imagen del centroide .....	38
Figura 29: Entradas y salidas de lógica difusa.....	40
Figura 30: Valores membresía de temperatura .....	41
Figura 31: Valores membresía de setpoint .....	42



Figura 32: Declaración de las reglas en Matlab.....	43
Figura 33: Representación de las reglas mediante su función de pertenencia.....	45
Figura 34: Representación tridimensional del sistema .....	46
Figura 35: Esquema del fitotrón .....	47
Figura 36: Estructura del fitotrón.....	49
Figura 37: Diagrama de flujo de la implementación del fitotrón de la estructura .....	50
Figura 38: Calefactor puesto en el fitotrón .....	51
Figura 39: Conexión del disipador del Peltier .....	52
Figura 40: a) disipador del Peltier b) cooler expansor del frio en el fitotrón .....	53
Figura 41: Sensor de temperatura instalado.....	54
Figura 42: Declaración y promedio de la variable de temperatura en el programa.....	54
Figura 43: Visualización de la temperatura en la pantalla LCD.....	55
Figura 44: Sensor de humedad instalado .....	55
Figura 45: Visualización del sensor de humedad en la Pantalla LCD.....	55
Figura 46: Declaración de humedad relativa .....	56
Figura 47: Sensor de luminosidad instalado .....	56
Figura 48: Panel de los leds .....	57
Figura 49: Visualización del sensor temt 6000 en la pantalla LCD .....	57
Figura 50: Función del programa para encender y apagar la luz.....	57
Figura 51: Variación de pwm para variar la intensidad de luz .....	58
Figura 52: Esquema del circuito implementado en el Fitotrón.....	59
Figura 53: Diagrama de flujo del control de temperatura.....	60

## RESUMEN

La presente tesis busca mejorar varios temas con respecto a la agricultura, llevando la tecnología a los lugares donde el clima no es muy favorable, haciendo que la producción de plantas para el consumo humano sea constante.

Para comenzar este proyecto, se estudió los factores principales que afectan el crecimiento de una planta, se definieron las variables que inciden en la germinación de una semilla de planta comestible tales, como: la temperatura, la humedad y la luminosidad, de todos estos factores cual íbamos a controlar, y se determinó manejarlo bajo un sistema de control difuso para la automatización del fitotrón. La simulación del sistema de control difuso se desarrolló utilizando el Matlab, y la programación del controlador para el Arduino se realizó con el software eclipse.

Se utilizó un calefactor para incrementar la temperatura basado en una resistencia de 64 ohm a 3.5 Amperes cuando está en su máxima potencia y un sistema de refrigeración implementado con celdas de Peltier modelo TEC1-12706 que acompañados por un sistema de disipación hacen que se obtengan valores óptimos al bajar la temperatura, así como también el mejorar la vida útil de las celdas.

La energía que consume todo el fitotrón es de aproximadamente un total 1216 watios y el costo total de la implementación es de s/.2246.00 soles.

**Palabras Claves:** Arduino, Control lógica difusa, Invernadero.

## ABSTRACT

The thesis seeks to improve several topics regarding agriculture, bringing technology to places where the climate is not very favorable to make the production of plants for human consumption constant.

To begin the project we studied the main factors that affect the growth of a plant, defined the variables that affect the germination of an edible plant seed, such as temperature, humidity and luminosity, of all these factors that we were going to control, and it was determined to handle it under a diffuse control system for the automation of the phytotron, the simulation of the diffuse control system was developed using the Matlab, and the controller programming for the Arduino was developed done with the eclipse software.

We use a heater to increase the temperature based on a resistance from 64 ohm to 3.5 Ampere and a cooling system implemented with Peltier cells model TEC1-12706 that accompanied by a dissipation system make us have optimal values just as we also improve the lifespan of cells.

The energy consumed by the entire phytotron is approximately. total 1216 watts. and the total cost of implementation is s/.2246.

**Key Words:** Arduino, Fuzzy logic Control, Greenhouse.

## INTRODUCCIÓN

En la actualidad nos encontramos con un avance sustancial en el uso de la lógica difusa en el ámbito de la industria, ya que su uso es muy difundido en cualquier aparato electrónico disponiendo de funciones complejas e inteligentes que generan autonomía en los sistemas. Basado en este principio y la elección de un controlador que se adecue muy bien a este tipo de control automatizado, se eligió el Arduino, ya que al ser una plataforma libre favorece la implementación del fitotrón de una manera sencilla y eficaz.

Este proyecto de diseño del fitotrón, el cual se basa en la germinación de semillas, tendrá que simplificar su diseño y reducir el costo de los equipos, debido a que en el mercado presentan un elevado precio añadido a la importación de estos, haciendo desfavorable la compra de alguno de estos aparatos por parte del consumidor. Ante esto se hace la pregunta. ¿Será posible reducir el costo y simplificar el diseño de estos equipos para el mercado actual?

La consecuencia de esta implementación dejará un abanico de posibilidades a desarrollar en un futuro, y planteando las bases para el mejoramiento del crecimiento sostenible del fitotrón, ya sea en su estructura como a nivel de software.

# CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

## 1.1. Presentación de la problemática

En las épocas actuales, la biotecnología comprende técnicas basadas en el análisis del estudio de la biología celular y/o molecular, que pueden ser usadas en una gran variedad industrial que utilice microorganismos o células vegetales y animales.

Este tipo de tecnología va a permitir un gran avance en el ámbito de la agricultura principalmente en países que basan su economía en dicha área; el Perú es uno de ellos, donde mejoraría su exportación sin desabastecer el mercado local.

También mencionaremos que tiene un impacto para otras industrias originadas del carbono, como por ejemplo la reutilización de los desechos, energía gas, etc.

Las biotecnologías agrícolas favorecen en la adaptación al cambio climático sosteniendo los recursos naturales. Cosa que por el momento los pequeños agricultores no se ven favorecidos en ese aspecto.

La FAO (La Organización de las Naciones Unidas para la Alimentación y la Agricultura) menciona lo siguiente. “Con el fin de producir alimentos de forma sostenible para 2000 millones de personas más para 2050, el enfoque actual no será suficiente. Esto es especialmente cierto en vista del cambio climático y otros factores que amenazan a recursos naturales como la biodiversidad, la tierra y el agua, que son esenciales para la producción de alimentos y la agricultura, incluyendo la silvicultura y la pesca. Para hacer frente a estos retos, la ingeniería, la ciencia y la aplicación de las biotecnologías, así como las tecnologías convencionales jugarán un papel clave”.

Uno de los principales objetivos de este trabajo fue dar una solución práctica a los problemas que se presenta en el Laboratorio de Biotecnología e Ingeniería Genética de la facultad de Biología en la Universidad Ricardo Palma, donde no se cuenta con un ambiente totalmente controlado y automatizado, además de la falta de control en el consumo de energía eléctrica y el mal sensado de las principales variables. Además de bajar el costo de estos equipos automatizados.

## 1.2. Formulación del problema

La pregunta es: ¿Será posible diseñar e implementar un sistema fitotrón para la germinación de semillas a bajo costo?.

### 1.2.1. Problema General

¿Cómo producir la germinación de semillas utilizando tecnologías modernas?

### 1.2.2. Problemas Específicos

- ¿Cómo manipular las variables del fitotrón para una operación inteligente?
- ¿Qué componentes usar para el diseño del equipo y a su vez que no eleve su precio?
- ¿Cómo generar un entorno virtual de operación inteligente del fitotrón?

## 1.3. Objetivos

A continuación, mencionamos los objetivos del proyecto.

### 1.3.1. Objetivo general

Diseñar e implementar el Fitotrón automatizado para la germinación de semillas a bajo costo.

### 1.3.2. Objetivos específicos

- Diseñar el control de las principales variables del fitotrón utilizando lógica difusa a través del Arduino.
- Elegir los componentes ya sea sensores y actuadores adecuados para este proyecto
- Utilizar Matlab para la simulación de la lógica difusa.

## 1.4. Alcance y limitaciones de la investigación

Este proyecto se centra en la creación, simulación e implementación del fitotrón, utilizando el Arduino como su controlador.

- **Creación** del programa que maneja el Arduino para controlar las variables en el equipo.
- **La simulación** de la lógica difusa bajo el entorno del Matlab,
- **La implementación** del fitotrón usando los sensores y actuadores adecuados

A continuación, mencionamos las limitaciones del proyecto:

- Para no encarecer el costo del proyecto se trabajó solo con 6 de celdas de Peltier para bajar la temperatura del módulo, si se desea bajar aún más la temperatura,

añadiremos más celdas de Peltier acompañado de un sistema de disipación de calor adecuado para cada una de ellas.

- El espacio utilizado para la germinación de semillas no es grande debido a que será utilizado para el laboratorio de biología a modo de pruebas experimentales.

## **1.5. Justificación de la investigación**

A continuación, presentamos las justificaciones de este proyecto.

### **1.5.1. Justificación de control de calidad**

El ingreso de semillas transgénicas ha hecho que la calidad de nuestros productos vaya en declive con lo que respeta en años anteriores; es por eso, que este trabajo se basa en procurar el buen desarrollo de las semillas y su pronta germinación libre de impurezas, ya que al estar en un ambiente cerrado estará más controlado de ciertas plagas que afectarían en si su producción.

### **1.5.2. Justificación comercial**

La implementación de este proyecto es viable debido a que utilizaremos componentes electrónicos comerciales encontrados en el mercado actual. Siendo totalmente realizable reducir el costo del producto debido a esto.

### **1.5.3. Justificación tecnológica**

Para automatizar el control del fitotrón se basa en el uso de la lógica difusa y el Arduino como su principal controlador, siendo ambos muy usados en la actualidad.

## 1.6. Estructura de la tesis

Esta tesis se ha diseñado con una introducción, 4 capítulos, además de los costos, conclusiones, observaciones, referencias bibliográficas y anexos Véase figura 1.

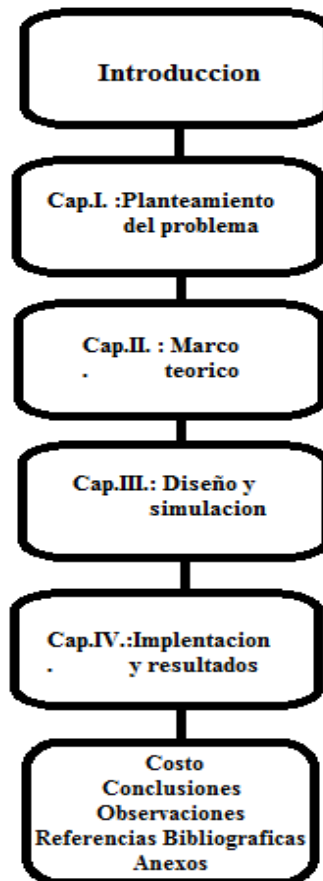


Figura 1: Estructura de la tesis

Fuente: Elaboración propia

### **Capítulo I. Planteamiento del problema**

En este capítulo se abarca la presentación de la problemática, detallando los problemas, las limitaciones y la estructura de la tesis.

### **Capítulo II. Marco teórico**

En este capítulo se mencionan los antecedentes que se basó la tesis para el presente trabajo, las variables y las bases teóricas en las que se apoya el proyecto.

### **Capítulo III. Diseño y simulación**

En este capítulo se hace mención a los programas usados, así como también las distintas simulaciones en Matlab.



#### **Capítulo IV. Implementación y resultados**

En este capítulo se describe el proceso de implementación y los resultados obtenidos, del proyecto.

#### **Costos, Conclusiones, Observaciones, Recomendaciones, Referencias Bibliográficas y Anexos.**

En este apartado se presenta el costo, las conclusiones, observaciones y recomendaciones del proyecto. Y finalmente, las referencias bibliográficas y los anexos que sirven de apoyo a la tesis.

## **CAPÍTULO II: MARCO TEÓRICO**

### **2.1. Antecedentes de la Investigación**

A continuación, mencionamos antecedentes de la investigación.

#### **2.1.1. Antecedentes Nacionales**

Gonzales Huisa, A.R. (2015) esta tesis se basó en el tema de la captura de imágenes con diversas longitudes de onda en un tiempo no menor a los 200ms que sirven para el estudio de los cultivos.

Huamán Lizana, K.F (2018) en la tesis se menciona la utilización de PLCs para poder controlar el riego en la facultad de su universidad.

#### **2.1.2. Antecedentes Internacionales**

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

Mazquiarán Andrade, C. (2017) de la Universidad Politécnica de Madrid, con su trabajo de tesis “Control Climático de un invernadero mediante lógica difusa”, el cual concluyo que este tipo de control difuso para manejar la temperatura es el más adecuado comparado con el uso de un PID.

Duarte Galván C. (2011) de la Universidad Autónoma de Querétaro con su trabajo “Sistema de control climático de bajo costo para invernadero basado en FPGA “, quien empleo un método inalámbrico para comunicarse con los sensores utilizando un módulo xbee de la compañía digi internacional.

Barrera Peña, N.F. y Maldonado Peña, J.C. (2008) de la Universidad de San Buenaventura en Bogotá Colombia con su tesis “Diseño, control e implementación de factores climáticos y ambientales en una cámara para el cultivo hidropónico de lechuga” en el cual utiliza un microcontrolador MC68HC908AP16 para manejar las variables de temperatura en su diseño.

### **2.2. Selección de variables**

A continuación, se detallan las distintas variables, descritas en la tabla 1 y 2.

Tabla 1: Variable independiente

Variable	Dimensión	indicadores
Lógica difusa	Arduino	Definición de las reglas de correspondencia

Fuente: Elaboración propia

Tabla 2: Variable dependiente

Variable	Dimensión	indicadores
Implementación del fitotrón	Monitoreo	Humedad
	Control	Intensidad lumínica
		Temperatura

Fuente: Elaboración propia

## 2.3. Bases teóricas

A continuación, se menciona características de la cámara de cultivo y germinación.

### 2.3.1. Concepto de Germinación

La Germinación es el proceso por el cual un embrión emerge de una semilla al encontrar condiciones internas (maduración, viabilidad) y externas (temperatura, luz, humedad, oxígeno) favorables. Cuando no se encuentra cualquiera de estas condiciones la semilla no termina de germinar y a este periodo se le conoce como periodo de latencia, llegando inclusive a permanecer años antes de brotar, posiblemente perdiendo su capacidad de germinación, Véase figura 2.



Figura 2: Imagen de una germinación

Fuente: <https://www.portalfruticola.com/noticias/2017/11/02/temperaturas-tiempos-la-germinacion-las-semillas/>

Sobre la germinación, Portalfruticola.com®, s.f. comenta:

Es el proceso mediante el cual una semilla se desarrolla hasta convertirse en una nueva planta. Este proceso se lleva a cabo cuando el embrión se hincha y la cubierta de la semilla se rompe. Para lograr esto, toda nueva planta requiere de elementos básicos para su desarrollo: luz, agua, oxígeno y sales minerales.

### **2.3.2. Condiciones de Temperatura**

La temperatura es un factor termodinámico que estimula el crecimiento de las plantas al generar transferencia de energía.

La temperatura es uno de los principales factores de la distribución y productividad de las plantas, para poder germinar las semillas es necesario tener una temperatura adecuada que varía según la planta a desarrollar, en ese sentido lo mostramos en la tabla 3, donde detallamos la temperatura promedio que se requiere para la germinación de las semillas.

Tabla 3: Temperatura y días en que tarda en germinar la semilla

<b>PLANTA</b>	<b>Días que tarda en germinar (aprox)</b>	<b>Temperatura ideal °C</b>
<b>Acelga</b>	9	18-22
<b>Ajo</b>	10	25-30
<b>Alcachofa</b>	25	16-27
<b>Apio</b>	20-25	18-25
<b>Brócoli</b>	6	18-25
<b>Berenjena</b>	15	20-25
<b>Col</b>	6	8-35.
<b>Coliflor</b>	6	10-30.
<b>Espinaca</b>	7	10-25.
<b>Haba</b>	15	4-30.
<b>Tomate</b>	5-8.	15-30
<b>Zanahoria</b>	10-18.	4-30.
<b>Maíz</b>	7	20
<b>Papa</b>	30	15
<b>Sandia</b>	5-8.	30
<b>Lechuga</b>	7	6-25.
<b>Melón</b>	5-8.	15-35
<b>Pepino</b>	5-8.	16-40
<b>Perejil</b>	25-30	10-22.

Fuente: <https://rockymountainseeds.org/images/pdfs/KnottsHandbook2012.pdf>

El factor de la temperatura es el más importante para la germinación, debido a que influyen en las enzimas regulando la velocidad de las reacciones bioquímicas que realizan las semillas una vez rehidratada. Estas enzimas poseen actividades que las realizan dentro de unos valores de temperatura entre mínimos y máximos permitidos, por eso su importancia en este proceso. Por este motivo las semillas solo germinaran cuando se encuentren dentro de estos valores óptimos, si no las hallara no se producirá la germinación a pesar que las otras condiciones sean favorables.

### 2.3.3. Condiciones de humedad

El líquido vital en el planeta tierra es el agua por lo tanto las plantas lo requieren para su desarrollo se dice que tan solo el 3% de este líquido que absorben es utilizado para los procesos de metabolismo o fotosíntesis. El resto lo usan para transportar los nutrientes del suelo hacia las hojas donde se evaporarán por transpiración.

Entonces la relevancia de este componente está más que claro, si hay poca humedad la planta no recibirá todos los nutrientes necesarios que requiere y no se podrá desarrollar normalidad, a su vez si encontrara un exceso este sería perjudicial ya que podría ahogarse de manera que arrastraría a las demás plantaciones produciendo plagas y alejando los nutrientes que necesita.

A continuación, mostramos la figura 3, donde detallamos la frecuencia de riego en algunos vegetales. Cabe mencionar que en la etapa de verano es donde más se requiere de la humedad

Alto riego	Ajo	Cebolla	Lenteja
Medio riego	Guisante	Tomate	Pepino
Bajo riego	Acelga	Alcachofa	Calabacín

Figura 3: Frecuencia de riego

Fuente: [http://oa.upm.es/47865/1/TFG\\_CESAR\\_MAZQUIARAN\\_ANDRADE.pdf](http://oa.upm.es/47865/1/TFG_CESAR_MAZQUIARAN_ANDRADE.pdf)

### 2.3.4. Condiciones de luminosidad

Otro factor importante externo para el crecimiento de una planta viene a ser la luz, la longitud de onda del espectro electromagnético donde el ser humano puede visualizar los colores esta dado en un rango de 380 nm y 750 nm.

Para el desarrollo de las plantas se ha estudiado que las luces con determinadas longitudes de onda favorecen en su crecimiento con mucha mayor intensidad que otras, se ha llegado a la conclusión según los estudios que son la luz azul y la luz roja los que más influencia tienen para su desarrollo. Un punto a mencionar es que debido a la poca absorción de la luz verde sobre las hojas de las plantas hacen que se reflejen en ellas observando que sea este el color característico en la vegetación (Véase Figura 4).

También mencionaremos que para el caso de la germinación en algunas semillas este factor no es tan trascendental como en otras.

Los niveles de luz afectan directamente a la transpiración, la absorción de agua, el florecimiento, la germinación, el crecimiento intermodal, etc. dentro de la planta.

Mencionamos un párrafo de Pro-mix®, s.f.

La luz es importante, ya que las plantas usan parte de la energía radiante que emite el sol para la fotosíntesis. Generalmente, la luz se confunde con energía. La calidad de la luz es la longitud de onda (en nanómetros, nm) dentro del espectro electromagnético mencionaremos que la luz azul es de 450 nm y la luz roja es de 660 nm. Cada longitud de onda tiene un paquete diferente de radiación o energía (en  $\text{J mol}^{-1}$ ) llamado fotones. La energía de un fotón es inversamente proporcional a su longitud de onda; mientras más corta la longitud de onda, mayor es la energía y mientras más larga la longitud de onda, menor es la energía, expresada en la ecuación de Planck.

(Véase la ecuación de abajo); es decir que mientras la energía de la luz azul de onda corta es de  $2,66 \times 10^5 \text{ J}\cdot\text{mol}^{-1}$ , la luz roja de onda más larga es de  $1,81 \times 10^5 \text{ J}\cdot\text{mol}^{-1}$ , para el cálculo de la energía utilizaremos la ecuación de Planck:

$$E = h \cdot c / \lambda$$

E = energía (Joule)

h = constante de Planck ( $6.63 \cdot 10^{-34} \text{ Joule}\cdot\text{s}$ )

c = velocidad de la Luz ( $3 \cdot 10^8 \text{ m/s}$ )

$\lambda$  = Longitud de onda (nm)

Energía irradiada por la luz azul

$$E = (6.63 \cdot 10^{-34}) \cdot (3 \cdot 10^8) / 450 \cdot 10^{-9} = 0.0442 \cdot 10^{-17} \text{ J}$$

Lo multiplicamos por el número de Avogadro y da como resultado  $2,66 \times 10^5 \text{ J}\cdot\text{mol}^{-1}$

Energía irradiada por la luz roja

$$E = (6.63 \cdot 10^{-34}) \cdot (3 \cdot 10^8) \cdot /660 \cdot 10^{-9} = 0.03 \cdot 10^{-17} \text{ J}$$

Lo multiplicamos por el número de Avogadro y da como resultado  $1,81 \times 10^5 \text{ J} \cdot \text{mol}^{-1}$

Para que la energía expresada en Joules, nos dé un resultado final con respecto a moles lo multiplicamos por la constante de Avogadro “Na” (que viene hacer la cantidad de fotones, átomos, moléculas, etc., encontrada en un mol de cualquier sustancia).

$$Na = 6.03 \cdot 10^{23} / \text{mol}.$$

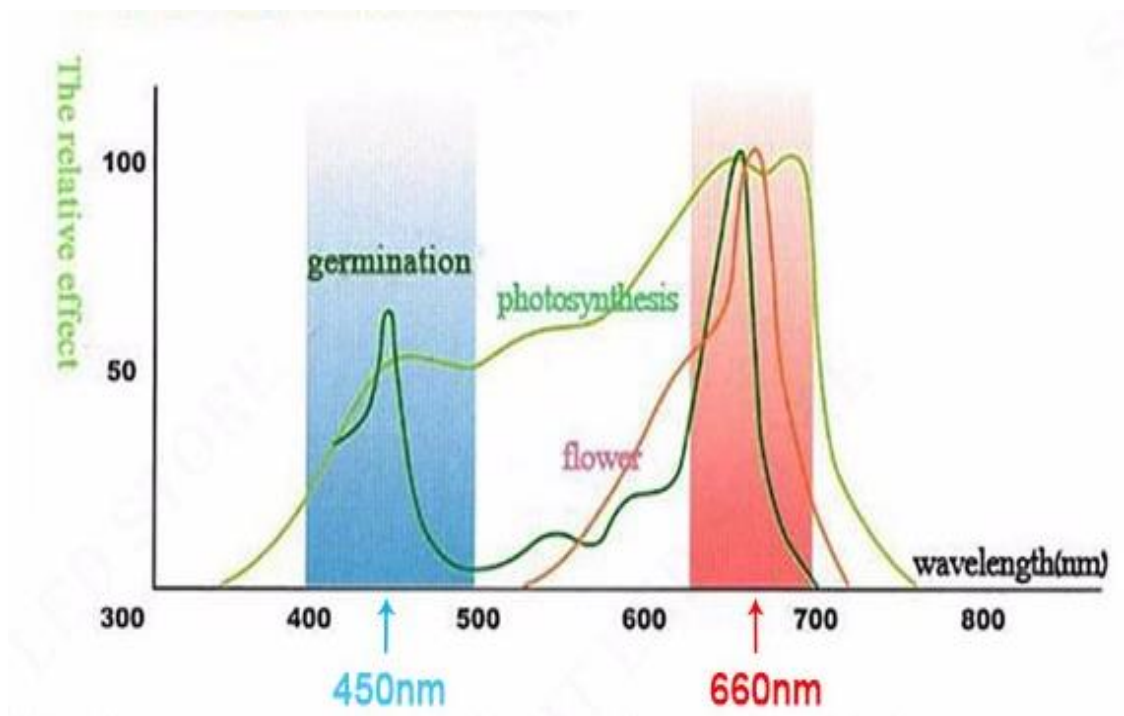


Figura 4: Efecto de la luz en el desarrollo de una planta

Fuente: <https://es.dhgate.com/product/8pcs-100w-led-grow-lights-full-spectrum-150leds/438946050.html>

### 2.3.5. Concepto de cámara de cultivo “fitotrón”

Un Fitotrón es una cámara de cultivo térmicamente aislado y automatizado, que simula condiciones climáticas favorables para el crecimiento de las plantas, deben poseer dimensiones con espacio suficiente para poder albergar dentro de ellas plantas, semillas, etc debido a que sus parámetros son monitoreados y controlados



constantemente por un controlador, crecen en un ambiente libre de contaminación externa reduciendo ampliamente de manera favorable las plagas o agentes patógenos que existen en espacios abiertos, cosa que en el campo sería muy difícil de conseguir, de esta manera garantizamos el desarrollo lo más natural posible y la producción constante de las plantas.

El primer Fitotrón moderno se desarrolló en el año 1949 bajo el mando del doctor Neerlandes “Frits Warmolt Went” en el instituto de tecnología de california “*Caltech*”. Se construyó un gran complejo de habitaciones con un ambiente controlado que inicialmente se conoció con el nombre de “*laboratorio de investigación de plantas earhart*”, que brindan condiciones experimentales definidas para que los investigadores analicen los efectos del clima en las plantas para su germinación y desarrollo.



Figura 5: Fitotrón australiano (1961)

Fuente: <http://www.worldoftrons.com/blog/2017/2/25/phytotron>

Mencionamos un párrafo de worldoftrons.com@,2017

Los fitotrones unificaron y extendieron los esfuerzos parciales de años anteriores para reclamar el control total de todo el entorno. Tanto en cuartos de almacenamiento como en gabinetes más pequeños, los fitotrones produjeron y reprodujeron climas complejos de muchas variables. En los primeros fitotrones, cada cuarto individual se mantuvo a una temperatura única constante. El fitotrón australiano (figura 5) , por ejemplo, tenía habitaciones que mantenían 9 ° C, 12 ° C, 16 ° C, 20 ° C, 23 ° C, 26 ° C, 30

° C, 34 ° C. Debido a que algunos de los primeros experimentos con ambiente controlado mostraron que las plantas reaccionaban de manera diferente a las temperaturas diurnas y nocturnas, los primeros experimentos observaron el efecto (s) de variar la temperatura diurna en comparación con la temperatura nocturna, terminado el curso del día rutinariamente vieron a los investigadores mover sus plantas de altas a más bajas temperaturas, o cualquier otra variable o constante. Esto hizo que la variable "temperatura" se pueda controlar experimentalmente. Incluso un enfoque rápido que probara cada variable ambiental sucesiva y cada variedad de planta serviría para identificar condiciones ambientales específicas para maximizar el crecimiento. Con la expectativa de que seguramente se obtendría más conocimiento de una mayor tecnología, la próxima generación de fitotrones se expandió en alcance tecnológico, en sus rangos de variables ambientales y también en el grado de control sobre cada variable.

### **2.3.6. Concepto de controladores Arduino**

El Arduino es una herramienta muy práctica para poder procesar la información que entregan los sensores, al ser de lenguaje libre y su programación no es tan compleja además que no requiere de un circuito programador adicional.

Posee una placa que tiene un microcontrolador, Véase Figura 6, (que para nuestro proyecto es el atmega 2560) y varios puertos de entrada/salida digitales y analógicos, existen complementos conocidos como (shields), estos complementos brindan una mayor autonomía a programador y usuario de la placa y hacen de Arduino un entorno muy amigable y simple de trabajar.

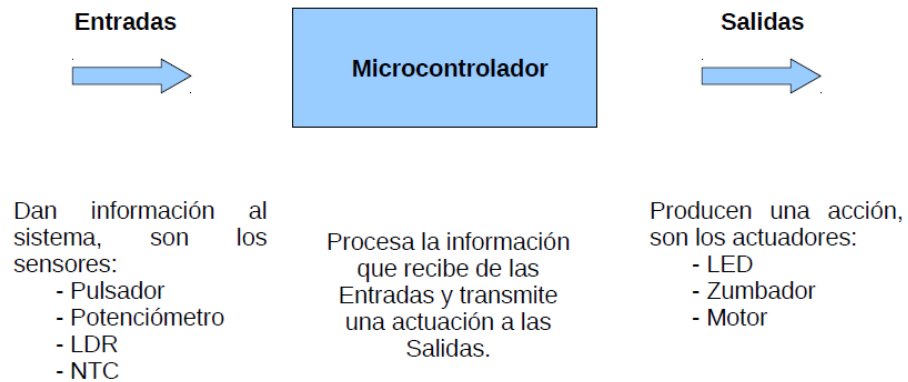


Figura 6: Esquema descriptivo de entradas y salidas de un microcontrolador

Fuente: <https://educarparaelcambio.com/arduino/que-es-arduino-y-para-que-sirve/>

Mencionamos un párrafo de arduino.cl®, 2019

El Arduino Mega 2560, véase Figura 7, es una placa de desarrollo basada en el microcontrolador ATmega2560. Tiene 54 entradas/salidas digitales (de las cuales 15 pueden ser usadas como salidas PWM), 16 entradas analógicas, 4 UARTs, un cristal de 16Mhz, conexión USB, jack para alimentación DC, conector ICSP, y un botón de reseteo. La placa Mega 2560 es compatible con la mayoría de shields compatibles para Arduino UNO.



Figura 7: Arduino Mega 2560

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

En la Tabla 4, mostramos un cuadro comparativo

Tabla 4: Comparación y elección del controlador

CUADRO COMPARATIVO DE CONTROLADORES				
	ARDUINO MEGA 2560	ARDUINO DUE	ARDUINO NANO	Selección
Características principales	Microcontrolador: ATmega2560 Led: Pin 13 Conexión USB y por jack Cristal de 16 MHz	Microcontrolador: AT91sam3x8e  Cristal de 84 MHz	Microcontrolador: ATmega328 Arquitectura: AVR Led: Pin 13 Cristal de 16 MHz	<p><b>ARDUINO MEGA 2560</b></p> <p><i>1. Cuenta con la cantidad de pines de entrada y salida necesarios para el fitotrón</i></p> <p><i>2. El voltaje de operación se adecua a casi todos los módulos del mercado y en mayor medida al proyecto,</i></p> <p><i>3. La capacidad de memoria está dentro de lo necesario, como la temperatura tiene una respuesta lenta, no hay necesidad de contar con un microcontrolador de mayor frecuencia de trabajo,</i></p> <p><i>4. El costo del Arduino mega es mucho más económico con respecto al Arduino due.</i></p> <p><i>5. Posee memoria eeprom que es de utilidad a la hora almacenar mayor cantidad de datos.</i></p>
Pines	Digitales Entrada/Salida: 54 PWM Entrada/Salida: 15 Análogos: 16 Corriente en los pines: 40 mA Corriente en el pin de 3.3 V: 50mA	Digitales Entrada/Salida: 54 PWM Entrada/Salida: 12 Análogos: 12 Corriente en los pines: 130 mA Corriente en el pin de 3.3 V: 800mA	Digitales Entrada/Salida: 22 PWM Entrada/ Salida: 6 Corriente en los pines: 40 mA	
Alimentación	Voltaje de operación: 5V Voltaje de entrada recomendado: 7-12 V  Voltaje de entrada límite: 6-20 V	Voltaje de operación: 3.3V Voltaje de entrada recomendado: 7-12 V  Voltaje de entrada límite: 6-20 V	Voltaje de operación: 5V Voltaje de entrada recomendado: 7-12 V	
Memoria	Memoria flash: 256 KB SRAM: 8 KB EEPROM: 4 KB	Memoria flash: 512 KB SRAM: 96KB EEPROM: No posee	Memoria flash: 32 KB SRAM: 2KB EEPROM: 1 KB	
Tamaño y peso	Longitud: 101.52 mm Ancho: 53.3 mm Peso: 37 g	Longitud: 101.52 mm Ancho: 53.3 mm Peso: 36 g	Longitud: 45 mm Ancho: 18mm Peso: 7g	

Fuente: <https://pdf1.alldatasheet.com/datasheet-pdf/view/897466/ATMEL/MEGA2560.html>

Fuente: <https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf>

Fuente: <https://store.arduino.cc/usa/due>

## 2.4. Características de los sensores y actuadores elegidos

Los sensores forman parte importante de este proyecto debido a que su precisión de medición nos dará un indicativo de cómo va el ambiente dentro de la cámara y de esta manera poder corregir valores y tomar acciones que harán que la planta logre germinar de manera óptima. Es la encargada de convertir la acción física (medición de temperatura, humedad y luminosidad) en un valor eléctrico que sea entendido y procesado por el controlador.

A continuación, se detalla los sensores y actuadores elegidos.

### 2.4.1. Sensor para el control de temperatura

Para este proyecto se ha escogido el sensor de temperatura MCP9701 de (Microchip technology ®, 2019), véase figura 8, convierte la temperatura en un voltaje análogo el cual nos ofrece una resolución de  $19,53 \text{ mV} / ^\circ \text{C}$ , y es de bajo costo, factores suficientes para haber elegido este sensor. A continuación, presentamos una tabla donde hacemos la comparación con otro sensor de similares características el LM35. Véase tabla 5.



Figura 8: Sensor MCP9701

Fuente: <https://www.microchip.com/wwwproducts/en/MCP9701>

Tabla 5: Cuadro comparativo de sensores MCP9701 y LM35

CUADRO COMPARATIVO DE SENSORES DE TEMPERATURA			
	Opción 1	Opción 2	Selección
Temperatura	MCP9701	LM35	MCP9701
	Salida lineal con voltaje análogo, proporcional a la temperatura	Entrada de temperatura y Salida con voltaje análogo,	<i>El sensor resistivo MCP9701 es de una respuesta lineal se conecta directamente al Arduino no necesita de un circuito o componente adicional para mantener su estabilidad al ser más inmune al ruido y presenta una mayor ganancia y mejor ahorro de corriente con respecto a la otra opción ya que solo necesitamos sensar de cero a 50 ° C es suficiente para el proyecto.</i>
	Tiempo de respuesta sin flujo de aire <1min	Tiempo de respuesta sin flujo de aire <3min	
	Sensibilidad a ruidos externos: INMUME	Sensibilidad a ruidos externos: SENSIBLE	
	Tecnología: termistor	tecnología: Diodo + transistor	
	Ganancia: 19.53mv/°c	Ganancia: 10mv/°c	
	Exactitud: ±1° C	Exactitud: ±1° C	
corriente: 6 uA	corriente: 60 uA		

Fuente: <http://pdf1.alldatasheet.com/datasheet-pdf/view/109864/MICROCHIP/MCP9701.html>

Fuente: <http://www.ti.com/lit/ds/symlink/lm35.pdf>

Este sensor de temperatura mcp9701 a diferencia del lm35, no necesita un voltaje negativo para poder medir señales por debajo de los cero grados centígrados, y su baja impedancia en la salida, hacen que su elección sea el mejor al ser leído fácilmente por la entrada analógica del Arduino.

#### 2.4.2. Sensor para el control Humedad

Para poder medir la humedad dentro de la cámara de cultivo y que sea reflejada en la pantalla LCD de 16x4, hemos elegido el sensor de humedad HIH 4000 de la marca Honeywell. véase Figura 9.

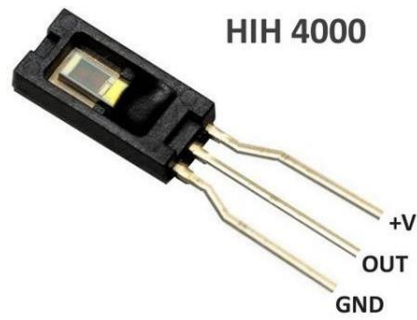


Figura 9: Sensor de humedad usado en el proyecto

Fuente: <https://www.indiamart.com/proddetail/honeywell-humidity-sensor-hih-4000-series-7801617430.html>

Este sensor es de bajo consumo llega a  $200\mu\text{A}$  y ofrece una respuesta lineal en la salida como se observa en la gráfica de la figura 10, que fueron tomados con voltaje de entrada de 5v a temperatura ambiente de  $25^\circ\text{C}$ .

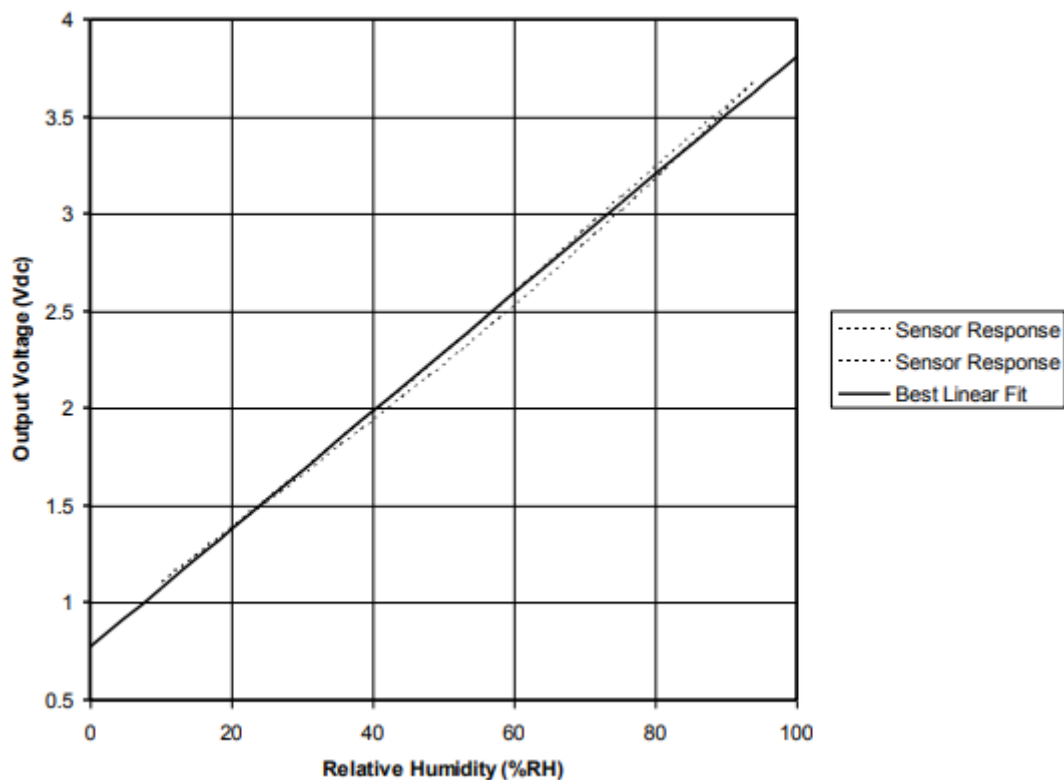


Figura 10: Respuesta lineal del sensor HIH 4000

Fuente: <https://sensing.honeywell.com/honeywell-sensing-hih4000-series-product-sheet-009017-5-en.pdf>

A continuación, mostramos (Véase tabla 6) una comparación con el sensor de Humedad DHT11.

Tabla 6: Cuadro comparativo de sensores HIH4000 y DHT11

CUADRO COMPARATIVO DE SENSORES DE HUMEDAD			
	Opción 1	Opción 2	Selección
Humedad relativa	HIH 4000	DHT11	HIH4000
	Salida con voltaje análogo, proporcional a la humedad	Salida digital de 8 bits, en un pin	<i>El sensor capacitivo HIH4000 de respuesta lineal presenta una mejor precisión además de una respuesta mucha más rápida que el otro sensor, necesario para el proyecto.</i>
	Rango de medición: 0-100% (HR)	Rango de medición: 20-90% (HR)	
	Precisión: $\pm 3.5\%$ RH.	Precisión: $\pm 5\%$ RH	
	Tiempo de respuesta: 5 segundos	Tiempo de respuesta: 10-15 segundos	
	corriente: 200 uA	corriente: 0.5 mA	

Fuente: <https://sensing.honeywell.com/honeywell-sensing-hih4000-series-product-sheet-009017-5-en.pdf>

Fuente: <https://datasheet4u.com/datasheet-parts/DHT11-datasheet.php?id=785590>

### 2.4.3. Sensor para el control luminosidad

Para la etapa del proyecto de iluminación de la cámara del fitotrón hemos optado por colocar un sensor de lumínico TEMT 6000, véase figura 11, que viene hacer un fototransistor de tipo NPN, que ofrece características adecuadas para el proyecto.



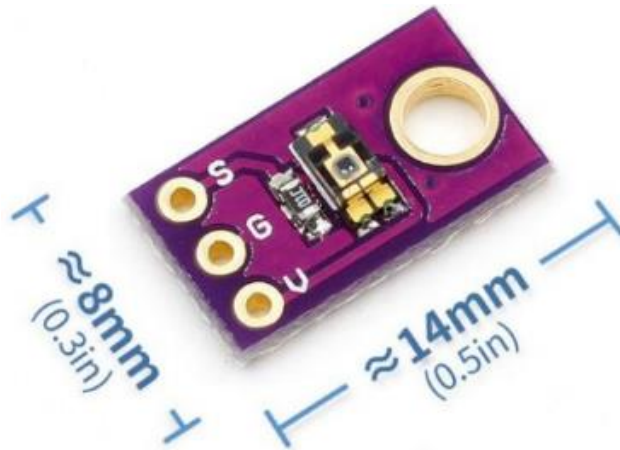


Figura 11: Sensor de luminosidad temt 6000

Fuente: <https://www.vs-elec.fr/es/lumiere/2171-sensor-de-luz-ambiental-de-brillo-temt-6000-3665662014382.html>

Una de las características importantes que posee son sus dimensiones que son 8mm x 14mm fácil de instalar dentro de la cabina, tiene una salida análoga que está conectada a la entrada del Arduino. Tiene un rango de lectura óptimo de 440 nm a 800 nm del espectro magnético, siendo 570nm el valor de mayor sensibilidad (véase figura 12), de acuerdo a esta información, el sensor no detecta longitud de onda infrarroja ni ultravioleta.

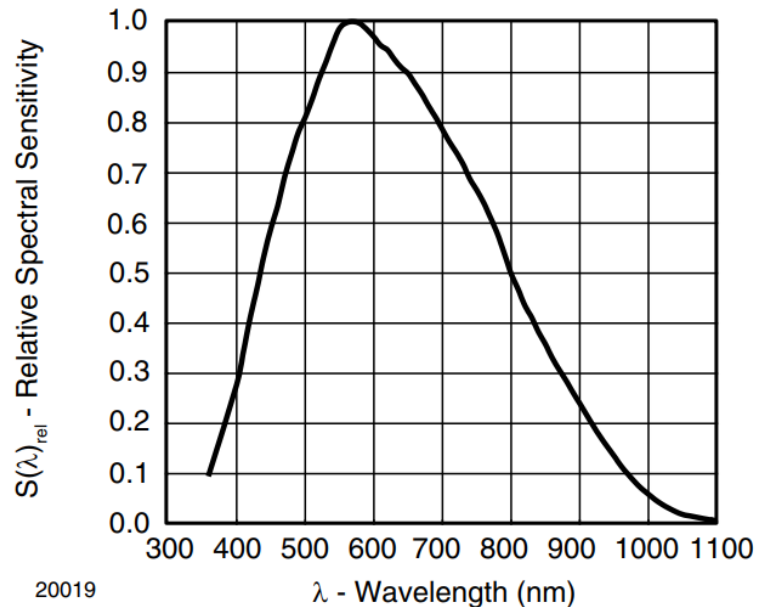


Figura 12: Sensibilidad del sensor de luminosidad

Fuente: <https://www.vishay.com/docs/81579/temt6000.pdf>

A continuación, mostramos un cuadro comparativo, Véase tabla 7

Tabla 7: Cuadro comparativo de sensores Temt6000 y Tls2561

CUADRO COMPARATIVO DE SENSORES DE LUMINOSIDAD			
	Opción 1	Opción 2	Selección
Luminosidad	TEMT6000	TLS2561	TEMT6000
	Su corriente del colector es proporcional a la entrada Luz, analógico	Salida digital es proporcional a la entrada de luz, comunicación I2C	<i>El fototransistor NPN TEMT6000 posee mayor inmunidad al ruido con un arreglo de condensadores se soluciona, además de tener mayor rapidez de respuesta y considerando que solo nos centraremos en el rango del espectro de frecuencias visible al ojo humano para medir y cabe mencionar el voltaje de entrada compatible para el proyecto.</i>
	Sensibilidad a ruidos externos: al ser señal analógica con un arreglo de condensadores se soluciona	Sensibilidad a ruidos externos: Si hay mucho ruido no hay comunicación	
	Voltaje alimentación: 5V	Voltaje de alimentación: 3V	
	Tiempo de respuesta: < 15u segundos	Tiempo de respuesta: < 420 ms	
	Corriente colector: 20 mA	corriente: 0.6mA	

Fuente: <https://www.sparkfun.com/datasheets/Sensors/Imaging/TEMT6000.pdf>

Fuente: <https://cdn-shop.adafruit.com/datasheets/TSL2561.pdf>

#### 2.4.4. Actuador para incrementar la temperatura

Los actuadores son dispositivos finales que generan una acción física en un sistema automatizado ya sea eléctrica, neumática, hidráulica, etc

Se propone una resistencia de alambre de nicrom de 64 ohm para elevar la temperatura dentro de la cabina, está compuesto de una aleación de níquel (80%) y cromo (20%) resistente a la oxidación y además a la corrosión, logrando elevar la temperatura al paso de la corriente eléctrica en pocos segundos. Fue patentado por Albert Marsh en el año de 1960. Sus propiedades de transferencia de calor son muy difundidas en la actualidad además de poseer un punto de fusión de 1400

grados centígrados. A continuación, mostramos algunas de sus características, véase figura 13.

<b>alambre de nicrom</b>				
Diámetro en m/m	Sección m/m <sup>2</sup>	Ohmios por Metro $\Omega * m = \rho$	Gramos por metro	Intensidad máxima En Amperios a: 800 <sup>o</sup> c
3	7	0,166	36	65,4
2,5	4,88	0,20	29	56,2
2	3,14	0,33	26,7	45,3
1,6	2,10	0,53	16	32,4
1,3	1,31	0,83	10	23,4
1,15	1,04	1,10	8,35	19,8
1	0,78	1,35	6,6	16,8
0,9	0,64	1,67	5,2	14,2
0,8	0,50	2,12	4,23	12
0,72	0,408	2,67	3,34	10,20
0,64	0,32	3,38	2,64	8,65
0,57	0,25	4,24	2,10	7,33
0,51	0,204	5,36	1,67	6,20
0,45	0,159	6,76	1,33	5,27
0,40	0,125	8,57	1,02	4,47
0,36	0,102	10,74	0,84	3,80
0,32	0,08	13,64	0,65	3,23
0,28	0,06	16,95	0,52	2,71
0,25	0,049	21,65	0,41	2,30
0,22	0,038	27,33	0,326	1,95
0,20	0,031	33,82	0,263	1,66
0,16	0,02	54,52	0,164	1,18
0,12	0,011	86,59	0,100	0,85
0,1	0,007	135,26	0,067	0,613
0,08	0,005	240,53	0,037	0,439

Figura 13: Características del alambre de nicrom

Fuente: <https://www.yoreparo.com/electricidad/electricidad-industrial/preguntas/1567229/calculo-de-alambre-de-nicrom>

#### 2.4.5. Actuador para bajar la temperatura.

Este factor es sumamente importante en la implementación del fitotrón, hemos elegido la celda de Peltier debido a que su implementación es muy sencilla y se adapta muy bien en el proyecto.

La celda se basa en el efecto Peltier que fue descubierto por científico el francés Jean Peltier en el año 1834, en el cual consiste en generar una diferencia de temperatura al pasar corriente eléctrica en una juntura de dos materiales de diferentes aleaciones, con el pasar del tiempo y la aparición de los semiconductores tipo P y N, fue mejorando en su rendimiento, La celda se considera un dispositivo termoelectrico capaz de generar frio a través del paso de la corriente eléctrica, durante el funcionamiento una cara se calienta mientras

que la otra se enfría de una manera sumamente rápida, si se invierte la polaridad de la fuente de alimentación que da energía a la placa se invertirá también la temperatura en ambas caras de la celda, su uso más frecuente es la de enfriar debido a que existe elementos más eficientes de generar calor. “Durante su aplicación debemos dar salida al lado caliente de la celda por ese motivo se coloca un disipador con un ventilador en ese lado de la cara “. Así mismo también se recomienda poner un disipador en el lado frío.

Temperatura máxima alcanzada  $T_{max} = 83\text{ }^{\circ}\text{C}$   
 Temperatura mínima alcanzada  $T_{min} = -55\text{ }^{\circ}\text{C}$

La vida media de las celdas es de 200,000 horas.

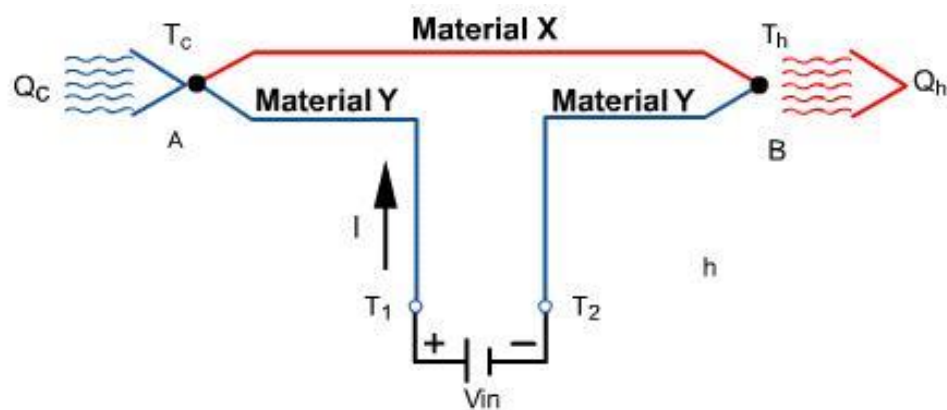


Figura 14: Funcionamiento de una celda de peltier

Fuente: <https://www.monografias.com/trabajos16/modulos-peltier/modulos-peltier.shtml>

Donde:

T<sub>c</sub> = Temperatura fría

T<sub>h</sub> = Temperatura Caliente

A, B = puntos de Junturas.

Q<sub>c</sub> = Transferencia de temperatura fría

Q<sub>h</sub> = Transferencia de temperatura caliente

V<sub>in</sub> = Voltaje de entrada

$$Q = K \cdot i \cdot t$$

Calor transferido (Q) es proporcional al coeficiente (k) característico de la pareja de semiconductores, y de la intensidad (I) y el tiempo (t) de funcionamiento

En la figura 14, se observa que si aplicamos voltaje v<sub>in</sub> en los terminales T<sub>1</sub> y T<sub>2</sub>, circulará una corriente I en el circuito, esto provocará que en la junta A absorba el

calor por ende se enfriara , por otro lado , en la juntura B generara calor como consecuencia se calentara.

Las celdas de Peltier consta de:

- Zona semiconductor Canal Tipo P Y N alternándose.
- Placas conductoras de cobre que unen las Zonas semiconductoras P y N de manera alterna
- Placas cerámicas que cubren el sistema en ambos lados, una cara será la caliente y la otra la fría.

Las ventajas de usar celdas de Peltier:

- Fácil funcionamiento
- Bajo costo
- No es contaminante, al no usar gases para enfriar
- No se requiere de un mantenimiento
- Funcionan en una variedad de entornos

y sus desventajas son:

- Un elevado consumo en corriente.
- La cara caliente necesita de una buena disipación
- Baja eficiencia.

En la figura 15, mostramos las variedades de celdas de Peltier que se encuentran en la actualidad.

TEC1 series——TEC1-127 (40\*40) (50\*50) (62\*62)

Type	Couples	I <sub>max</sub> (A)	U <sub>max</sub> (V)	Q <sub>cmax(w)</sub> ΔT=0	ΔT <sub>max</sub> ℃ Q <sub>c</sub> =0	Dimension (mm)			R Ω		
						Th=27℃					
						L	M	H			
TEC1-12703	127	3	15.4	26.7	68	40	40	4.92	3.42		
TEC1-12704		4		35.6	68	40	40	4.54	3.02		
TEC1-12705		5		44.5	68	40	40	4.20	2.40		
TEC1-12706		6		53.3	68	40	40	3.82	1.98		
TEC1-12707		7		62.2	68	40	40	3.80	1.70		
TEC1-12708		8		71.1	68	40	40	3.80	1.50		
TEC1-12709		9		80.1	68	40	40	3.37	1.36		
TEC1-12710		10		88.9	67	40	40	3.30	1.08		
TEC1-12708		127		8	15.4	71.1	68	50	50	5.11	1.50
TEC1-12710				10		88.9	68	50	50	4.50	1.12
TEC1-12712	12		106.7	68		50	50	4.20	0.90		
TEC1-12714	14		124.4	68		50	50	4.28	0.81		
TEC1-12715	15		133.3	68		50	50	3.61	0.75		
TEC1-12710	127	10	15.4	88.9	68	62	62	6.10	1.05		
TEC1-12730		30		266.7	68	62	62	4.80	0.27		

TEC 1-127 06

致冷器件的外形圖示

TE----Thermo Electrical

C---- Area of section of N/P > 1mm<sup>2</sup>

1----Stage 1

127----Couples(127couples)

06----Current(6A)

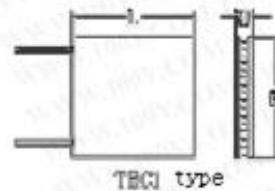


Figura 15: Códigos de la celda de Peltier

Fuente: <http://controlcbautista.blogspot.com/>

#### 2.4.6. Actuador de Luminosidad.

Consideramos para iluminar la cabina del fitotrón el uso de las luces led por ser un emisor de luz que no eleva la temperatura debido que transforma la mayor parte de la energía en luz y no en calor, además que nos permitirá abarcar todo el espectro de onda necesarias para el crecimiento y desarrollo de las plantas ,esta tecnología se inició en la década de los sesenta como componentes de aparatos electrónicos que emitían luz infrarroja de baja intensidad ,pero su difusión actual en el sector doméstico se realizó en los últimos años. A continuación, se muestra una comparación con el consumo de energía vs la cantidad de luz que emiten.

A continuación, en la figura 16, mencionamos algunas características de los diferentes tipos de emisor de luz

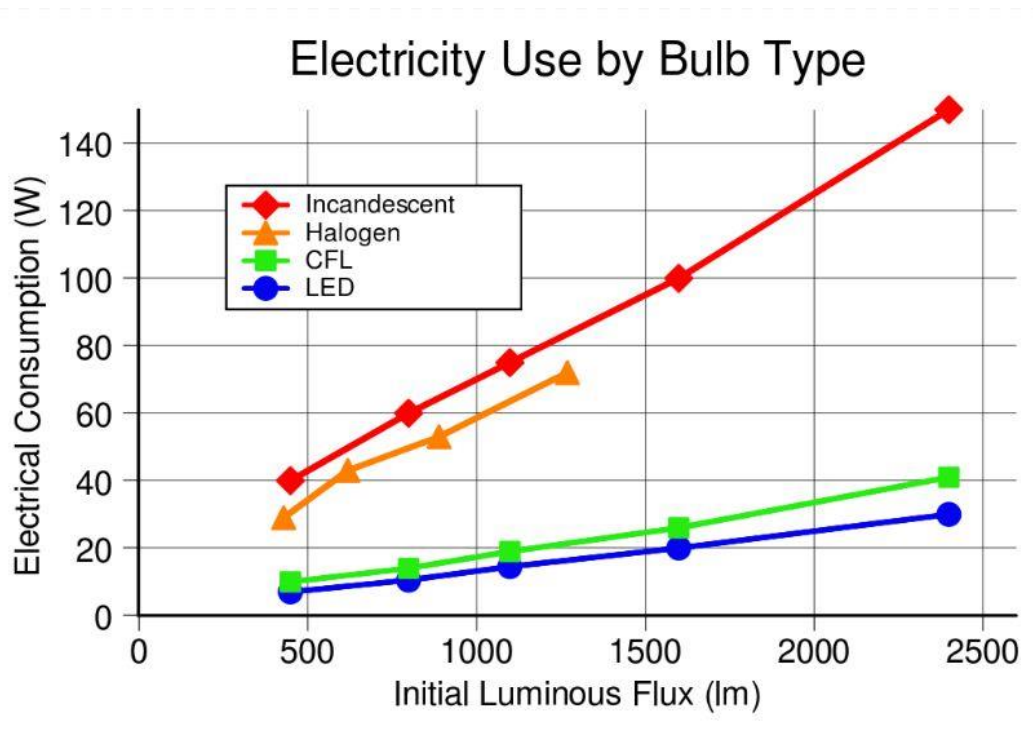


Figura 16: Consumo de energía para variedades de emisores de luz

Fuente: [https://es.wikipedia.org/wiki/L%C3%A1mpara\\_led#/media/Archivo:Electricity\\_use\\_by\\_lightbulb\\_type.svg](https://es.wikipedia.org/wiki/L%C3%A1mpara_led#/media/Archivo:Electricity_use_by_lightbulb_type.svg)

## 2.5. Características de control clásico y avanzado.

Los controles clásicos se caracterizan por manejar un sistema lineal y si no lo son, se hacen estrategias de linealización, además de trabajar con señales que sean continuas en el tiempo. Mientras que el control avanzado gracias a la llegada de los sistemas digitales y la aparición del procesador, así como también de los semiconductores, permitieron manejar sistemas no lineales y una amplia variedad de variables, este último tipo de control se basa a partir de varias ramas de estudio siendo la más importantes, el control robusto, control adaptivo y control inteligente (lógica difusa, redes neuronales y algoritmos genéticos).

El control clásico y avanzado no están separados uno del otro ya que gracias a los antecedentes del control clásico el control avanzado ha podido evolucionar hasta lo que conocemos hoy en día, más aún que en numerosos procesos se usan ambos trabajando a la par, donde uno va cubriendo las deficiencias del otro hasta convertirlo en un sistema completo y autosuficiente con amplias ventajas.



### 2.5.1. Control Clásico ON/OFF.

Este es un tipo de control que se basa en valor extremos, ya sean máximos o mínimos, al usar un algoritmo simple evaluando si el valor sobrepasa o no el setpoint indicado.

A manera de ejemplo: Cuando tenemos un aire acondicionado, El termostato activa el aire frio, si (ON) la temperatura es mayor a la de referencia o setpoint y lo desactivan (OFF) cuando la temperatura es menor (o igual) al setpoint.

Este tipo de control presenta desgaste en los componentes mecánicos cuando el cambio de estado es muy frecuente. Véase figura 17.

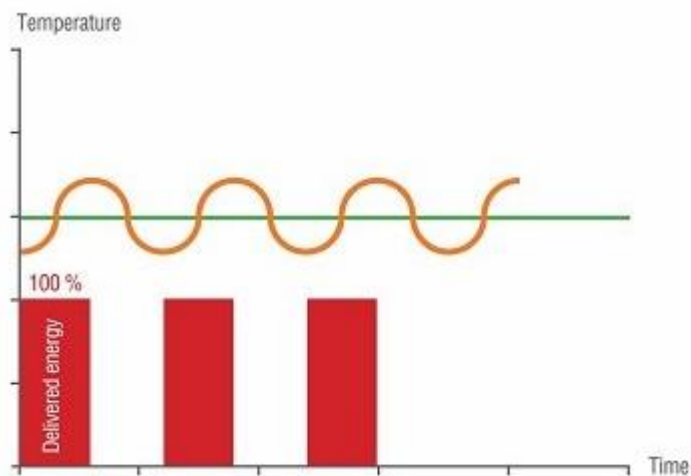


Figura 17: Comportamiento de un control ON/OFF

Fuente:<https://instrumentacionycontrol.net/control-on-off-o-todo-nada/>

### 2.5.2. Control Clásico PID

Este tipo de controlador está basado en una combinación de hardware y software, que funciona mediante el monitoreo constante de la señal de error (es la diferencia de las señales que se desean y los valores reales). Este proceso está dado mediante una realimentación, Donde la señal de salida se compara y se obtiene una diferencia que es enviada al controlador obteniéndose una señal de control con la finalidad de llevar la señal de salida al valor deseado, Véase figura 18.



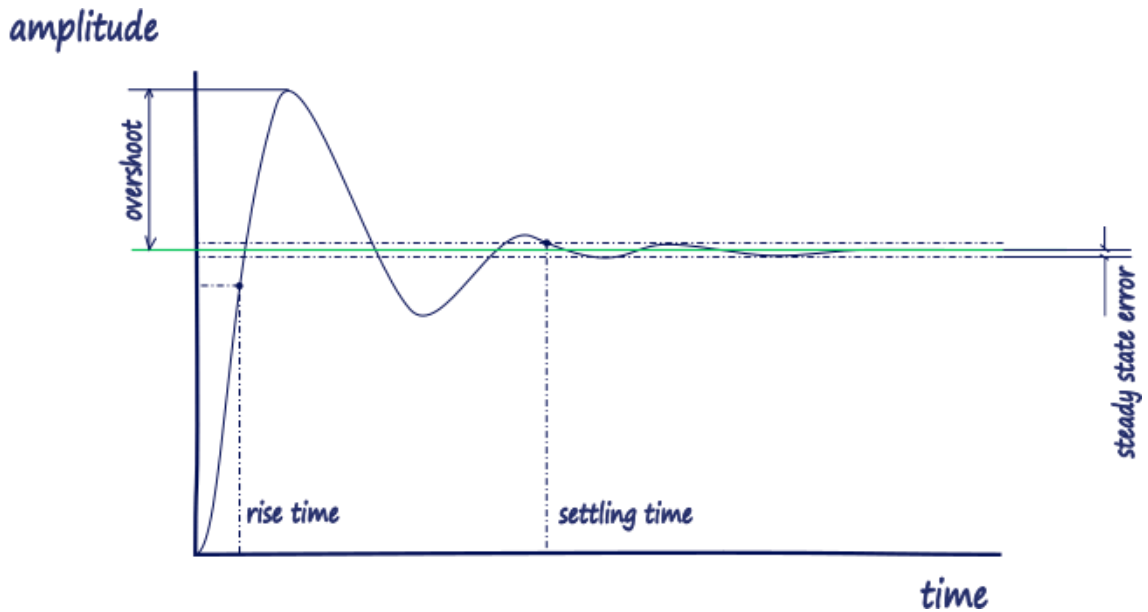


Figura 18: Características de la señal PID

Fuente: <http://nbozov.com/article/post/32/PID-Controller-basics>

**Rise time:** el tiempo que tarda la producción de la planta y en elevarse por encima del 90% del nivel deseado por primera vez.

**Overshoot:** Es el nivel máximo más alto que el estado estacionario, normalizado contra el estado estacionario.

**Settling time:** el tiempo que tarda el sistema en converger a su estado estable.

**Steady-state Error:** la diferencia entre la salida de estado estacionario y la salida deseada. Este sistema de control está compuesto de tres acciones Proporcional, Integral y Derivativo

- a) **Acción Proporcional:** Se entrega una constante como función de transferencia para el controlador P, denominada Ganancia Proporcional ( $K_p$ ). La acción proporcional proporciona una respuesta instantánea al error de control. Esto es útil para mejorar la respuesta de un sistema estable pero no puede controlar un sistema inestable por sí mismo. Además, la ganancia es la misma para todas las frecuencias que salen del sistema con un error de estado estable distinto de cero
- b) **Acción integral:** La acción de integral impulsa al error de estado estable hacia 0 pero ralentiza la respuesta.

c) Acción derivativa: Proporciona una respuesta rápida a diferencia de la acción integral, pero no puede aceptar errores constantes, Los derivados tienen una fase de +90 grados que conducen a una respuesta anticipatoria o predictiva. Sin embargo, el control derivado producirá grandes señales de control en respuesta a errores de alta frecuencia tales como cambios conjuntos de puntos (comando paso) y el ruido de medición (la respuesta derivativa es altamente sensible al ruido).  
 A continuación, se muestra una gráfica, véase figura 19, representando el funcionamiento del controlador P.I.D.

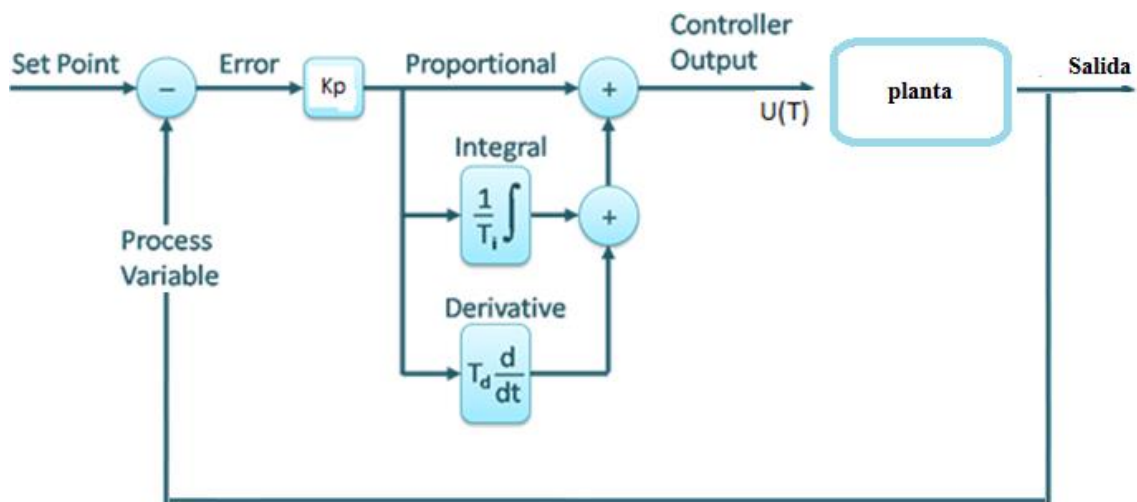


Figura 19: Esquema de un controlador PID

Fuente: <https://www.elprocus.com/the-working-of-a-pid-controller/>

Donde :

$$U(t) = \underbrace{K_p * e(t)}_P + \underbrace{\frac{K_p}{t_i} * \int_0^t e(t) dt}_I + \underbrace{K_p * T_d * \frac{de(t)}{dt}}_D$$

$K_p$ : ganancia proporcional

$e(t)$ : error

$T_i$ : tiempo requerido para que la acción integral iguale a la acción proporcional

$T_d$ : tiempo requerido para que la acción derivativa iguale a la acción proporcional

### 2.5.3. Control avanzado lógica difusa

La lógica difusa emergió como una alternativa para trabajar con información imprecisa o incompleta, así como para tomar decisiones sobre procesos con incertidumbres no aleatorias, de manera similar a como lo hace el lenguaje natural. La noción de conjunto difuso provee un conveniente punto de vista para construir un sistema de inferencia, pero en una forma más general que con conjuntos clásicos. En la lógica difusa, la idea principal es trabajar con variables lingüísticas más fáciles de entender en lugar de usar variables numéricas.

Un sistema difuso es básicamente es un conjunto de reglas, las reglas difusas son procesadas por un sistema de inferencia que las evalúa para llegar a una conclusión, puede ser visto como una interpolación entre diferentes subsistemas. Las reglas difusas determinan qué acción hacer mediante variables lingüísticas, las cuales se definen mediante funciones de pertenencia, Véase en la figura 20 las aplicaciones.

Áreas de aplicación de la teoría difusa			
	Problemas de interfaces Hombre/máquina	Problemas no lineales variantes en el tiempo	Clasificación de problemas
Problemas de los métodos convencionales	<ul style="list-style-type: none"> <li>• Dificultad para expresar numéricamente los objetivos del control</li> <li>• Evaluación del control por interpretación humana</li> </ul>	<ul style="list-style-type: none"> <li>• La dinámica de la planta varía con el tiempo.</li> <li>• Plantas no lineales</li> </ul> <p>Sobreflujo oscilación</p>	<ul style="list-style-type: none"> <li>• La acción a tomar no es clara</li> <li>• No es posible describir todas las trayectorias de solución</li> </ul> <p>Limitaciones hardware/velocidad</p>
Aplicaciones	<ul style="list-style-type: none"> <li>• Control de suspensión</li> <li>• Transmisiones automáticas</li> <li>• Metro de Sendai</li> </ul>	<ul style="list-style-type: none"> <li>• Control de temperatura</li> <li>• Control de posición de las cabezas de un disco duro</li> <li>• Pilotos automáticos</li> </ul>	<ul style="list-style-type: none"> <li>• Auto ZOOM</li> <li>• Reconocimiento de patrones escritos a mano</li> <li>• Transmisiones automáticas</li> </ul>

Figura 20: Aplicaciones de la lógica difusa

Fuente: <https://www.monografias.com/trabajos106/fundamentos-logica-difusa-control-difuso/fundamentos-logica-difusa-control-difuso2.shtml>

En la actualidad la aplicación de la lógica difusa es diversa, por ejemplo, se está usando en el sector financiero para elaborar el análisis de crédito, el análisis de inversión (costo del capital, análisis de equilibrio, etc.), gestión en la cadena de suministros, son solo algunos de los trabajos representativos del uso de las nuevas tecnologías. La lógica difusa permite incluir en la mayoría de modelos financieros la incertidumbre de manera diferente a como lo hace actualmente la teoría de probabilidades.

**a) Operaciones con conjuntos difusos**

Las operaciones básicas en los conjuntos difusos son tres AND, OR Y NOT. y como ejemplo definimos los conjuntos de pertenencias  $\mu_A(x)$  y  $\mu_B(x)$  entonces:

- El conjunto complementario not (Véase figura 21),  $\bar{A}$  de un conjunto, cuya función característica está definida por:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

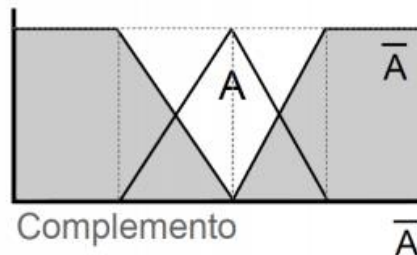


Figura 21: Operación de complemento

Fuente: [http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG\\_2017\\_Diciembre\\_SanahujaSamuel.pdf?sequence=1](http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG_2017_Diciembre_SanahujaSamuel.pdf?sequence=1)

- La unión de dos conjuntos OR (Véase figura 22), A y B, es un conjunto difuso  $A \cup B$ , cuya función de pertenencia es:

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

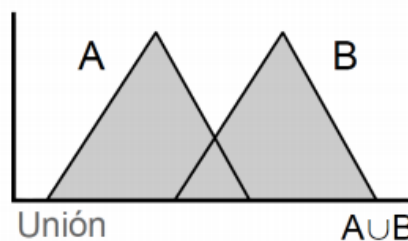


Figura 22: Operación de unión de dos conjuntos.

Fuente:[http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG\\_2017\\_DiciembreSanahuja\\_Sa\\_muel.pdf?sequence=1](http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG_2017_DiciembreSanahuja_Sa_muel.pdf?sequence=1)

- La intersección de dos conjuntos difusos AND (Véase figura 23), A y B, es un conjunto difuso,  $A \cap B$ , cuya función característica es:

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

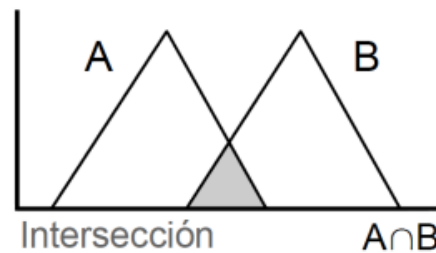


Figura 23: Operación de intersección de dos conjuntos.

Fuente:[http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG\\_2017\\_DiciembreSanahuja\\_Sa\\_muel.pdf?sequence=1](http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG_2017_DiciembreSanahuja_Sa_muel.pdf?sequence=1)

Existe una gran variedad de tipos de métodos difusos, dos grandes grupos son los que en general se emplearán, estos son el método de Mamdani y el método difuso de Takagi-Sugeno (TS, para abreviar). Para nuestro proyecto el que mejor se ajusta es el método Mamdani debido a que es intuitivo y está bien adaptado a incorporar conocimiento y experiencia.

## b) Funciones de membresía

Estas funciones representan el grado de pertenencia de un elemento dado a un subconjunto definido por una etiqueta. Hay una gran variedad de formas para estas funciones, las más comunes son del tipo trapezoidal, triangular, singleton, S.

## Método difuso Mamdani

Es un método muy difundido en la actualidad a continuación mostramos una gráfica (figura 24) de cómo está compuesto.

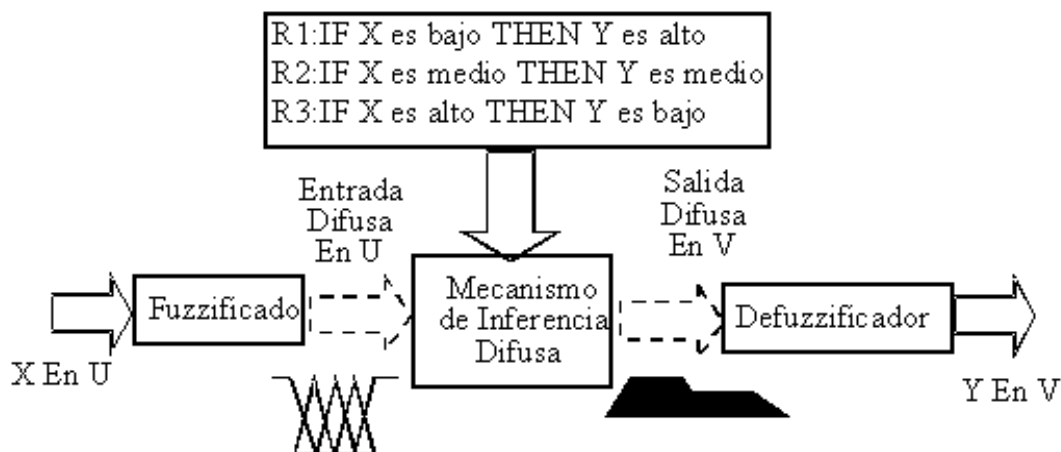


Figura 24: Configuración básica de un sistema tipo Mamdani

Fuente: [http://members.tripod.com/jesus\\_alfonso\\_lopez/FuzzyIntro2.html](http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html)

Este sistema Mamdani consta de las siguientes partes:

### a) Fusificación.

Es un proceso por el cual permite asociar a un valor numérico un conjunto difuso. Donde se asignan grados de pertenencia según un término lingüístico a cada una de las variables de entrada con relación a los conjuntos difusos, dicho proceso responde a un conjunto de normas preestablecidas, conceptualizadas a partir del conocimiento que brinda el razonamiento humano a través de un sistema implementado vía software. Véase figura 25.

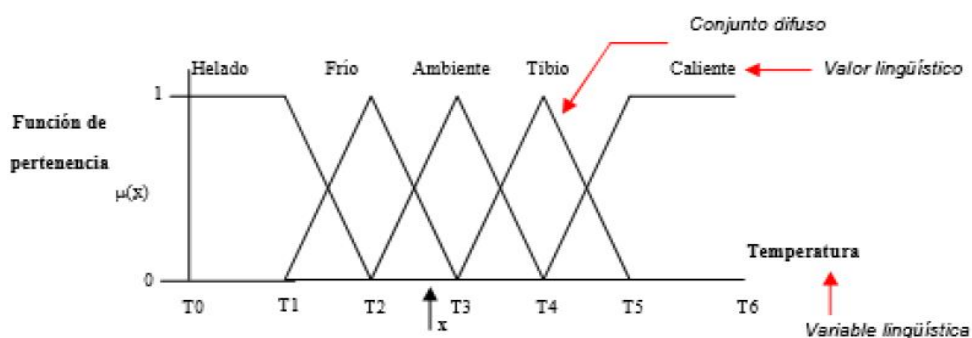


Figura 25: Función de pertenencia

Fuente: [http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG\\_2017\\_DiciembreSanahuja\\_Samuel.pdf?sequence=1](http://repositori.uji.es/xmlui/bitstream/handle/10234/173788/TFG_2017_DiciembreSanahuja_Samuel.pdf?sequence=1)

## b) Bases de reglas.

Estas reglas son usadas por los controladores, combinando entradas llamadas antecedentes como también salidas llamado consecuentes o consecuencia. Son afirmaciones del tipo “*if-else*” donde los conjuntos difusos se asocian con operadores lógicos AND, OR, etc. Son reglas lingüísticas de control que originan la forma de comportamiento del sistema.

Para expresar el conocimiento de manera más completa se determinan y detallan más de una regla conocido como base de reglas, es aquí donde es necesario la experiencia y conocimiento del operario para tomar la correspondiente estrategia de control. Las bases de las reglas generalmente se representan por tablas, donde el aumento de las variables lingüísticas hará que la tabla crezca, junto a una regla puede estar asociado un valor entre cero y uno que pesa a tal regla, esto puede ser importante cuando una regla tiene menor fuerza que otras de la base de reglas. Véase figura 26.

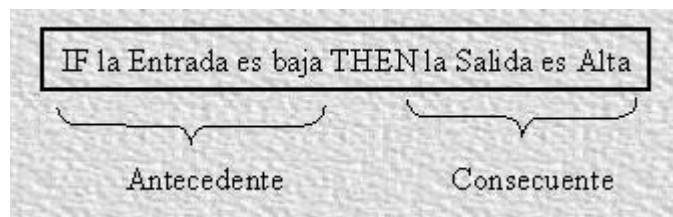


Figura 26: Forma de las reglas

Fuente: [http://members.tripod.com/jesus\\_alfonso\\_lopez/FuzzyIntro2.html](http://members.tripod.com/jesus_alfonso_lopez/FuzzyIntro2.html)

## d) Inferencia difusa.

La inferencia difusa cumple la función de calcular el resultado de las variables de entrada al interpretar las reglas del tipo “*if-then*” con la finalidad de la obtener conclusiones en términos de variables lingüísticas de salida. Relaciona los conjuntos difusos de entrada y salida para representar las diferentes reglas que definirán el sistema. Véase figura 27.

Las reglas difusas nos representan el conocimiento además de la estrategia de un sistema de control, pero si le asignamos información específica cada variable de entrada en el antecedente, la inferencia difusa se hace necesaria para el resultado de las variables de salida en términos difusos.

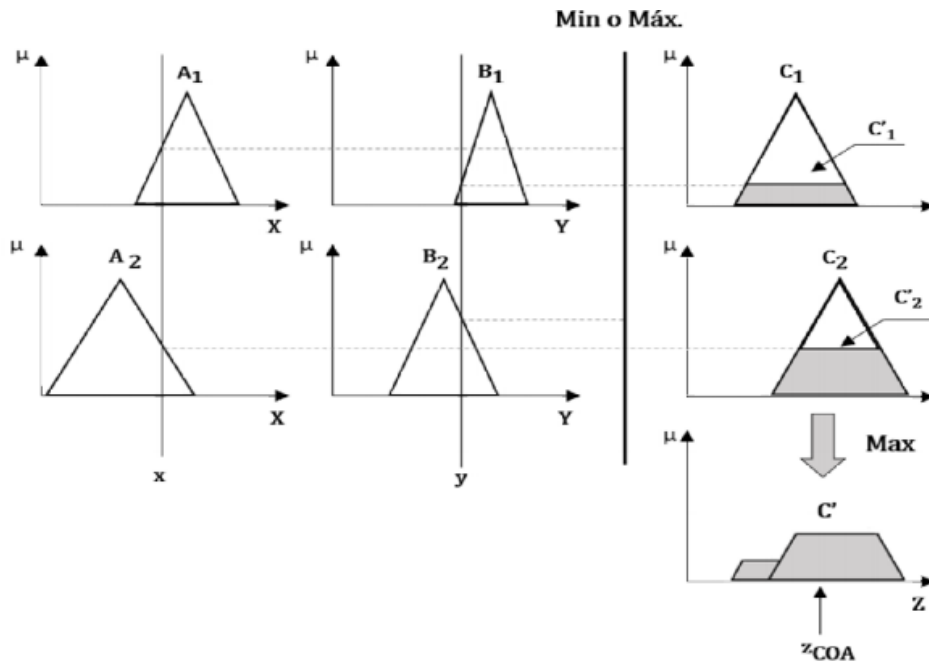


Figura 27: Sistema de inferencia tipo Mamdani

Fuente: <http://www.issr-journals.org/links/papers.php?journal=ijisr&application=pdf&article=IJISR-15-321-02>

### e) Defusificación.

Es el proceso inverso a la Fusificación, nos brinda la posibilidad de adecuar los valores difusos generados en la inferencia en valores que se usarán en el proceso de control. Esto quiere decir que lo que queremos obtener es valor numérico y no un valor difuso, uno de los métodos más usados es del centroide (calcula el centro del área definida por el conjunto difuso). Como se muestra en la figura 28.

$$C = \frac{\int_S xu(x) dx}{\int_S u(x) dx}$$



## Desdifusificación

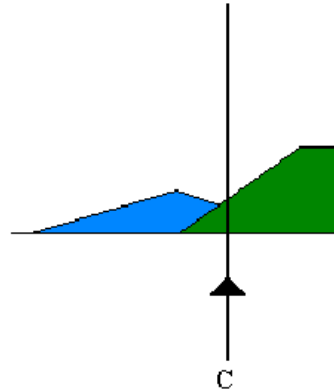


Figura 28: Imagen del centroide

Fuente: Elaboración propia

C: es el centroide, donde  $\mu(x)$  es la función de pertenencia del conjunto de salida, donde la variable es  $x$  y  $S$  viene a ser el dominio o rango de integración.

## 2.6. Características del software.

A continuación, mencionamos los softwares utilizados.

### 2.6.1. Software Eclipse.

Viene a ser una plataforma de software compuesto por varias herramientas de programación con código abierto de multiplataforma para desarrollar, opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (IDE),

Eclipse es también una comunidad de usuarios básicamente programadores.

Mencionamos un párrafo de Wikipedia®, 2019.

Eclipse originalmente fue desarrollado por IBM . En la actualidad posee un nombre, Eclipse es una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

### 2.6.2. Matlab.

Es un programa para realizar cálculos numéricos con vectores y matrices. Como caso particular puede también trabajar con números escalares tanto reales como complejos, con cadenas de caracteres y con otras estructuras de información más complejas. Una de las capacidades más atractivas es la de

realizar una amplia variedad de gráficos en dos y tres dimensiones. Matlab tiene también un lenguaje de programación propio.

Se menciona un párrafo de [juancarlosusomatlab2015.weebly.com](http://juancarlosusomatlab2015.weebly.com)®, 2019

“Es un programa interactivo orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos”.

A continuación, mencionamos sus usos:

- Cálculos numéricos
- Desarrollo de algoritmos
- Modelado, simulación y pruebas de prototipos
- Análisis de datos, exploración y visualización
- Graficación de datos con fines científicos o de ingeniería
- Desarrollo de aplicaciones que requieran de una interfaz gráfica de usuario (GUI).

## CAPÍTULO III: DISEÑO Y SIMULACIÓN

### 3.1. Requerimientos del diseño

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

- Ser de bajo costo.
- Ser de tamaño compacto.

### 3.2. Software a emplear.

A continuación, mencionamos los programas utilizados para el proyecto.

- Software eclipse.
- Matlab.

### 3.3. Control difuso de la temperatura

A continuación, mencionamos las variables usadas para la lógica de la temperatura

#### 3.3.1. Declaración de las variables lingüísticas

Para comenzar ,declaramos nuestras variables lingüísticas, figura 29.

Variables de entrada

- Temperatura
- Setpoint

Variables de salida

- Output1

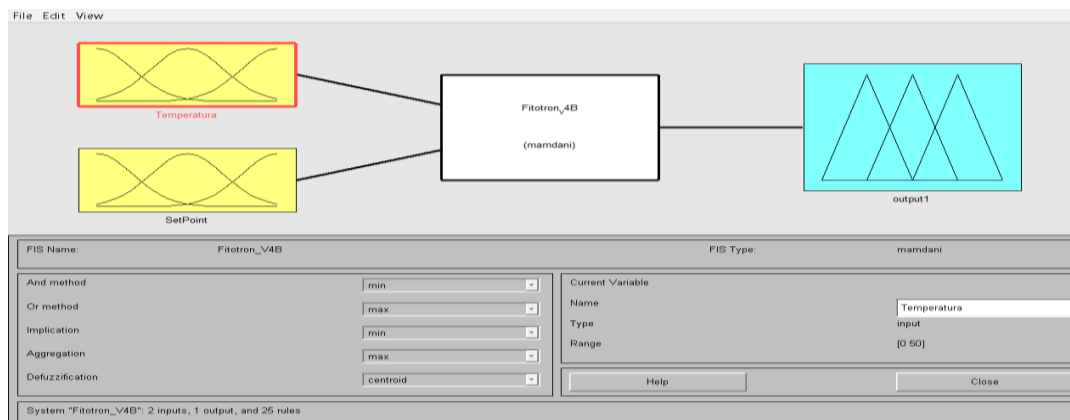


Figura 29: Entradas y salidas de lógica difusa

Fuente: Elaboración con Matlab

Una vez definidos nuestras variables, generamos nuestros valores lingüísticos, así como también se muestra gráficamente, Véase figura 30 y 31.

- Muy frio
- Frio
- Normal
- Caliente
- Muy caliente.

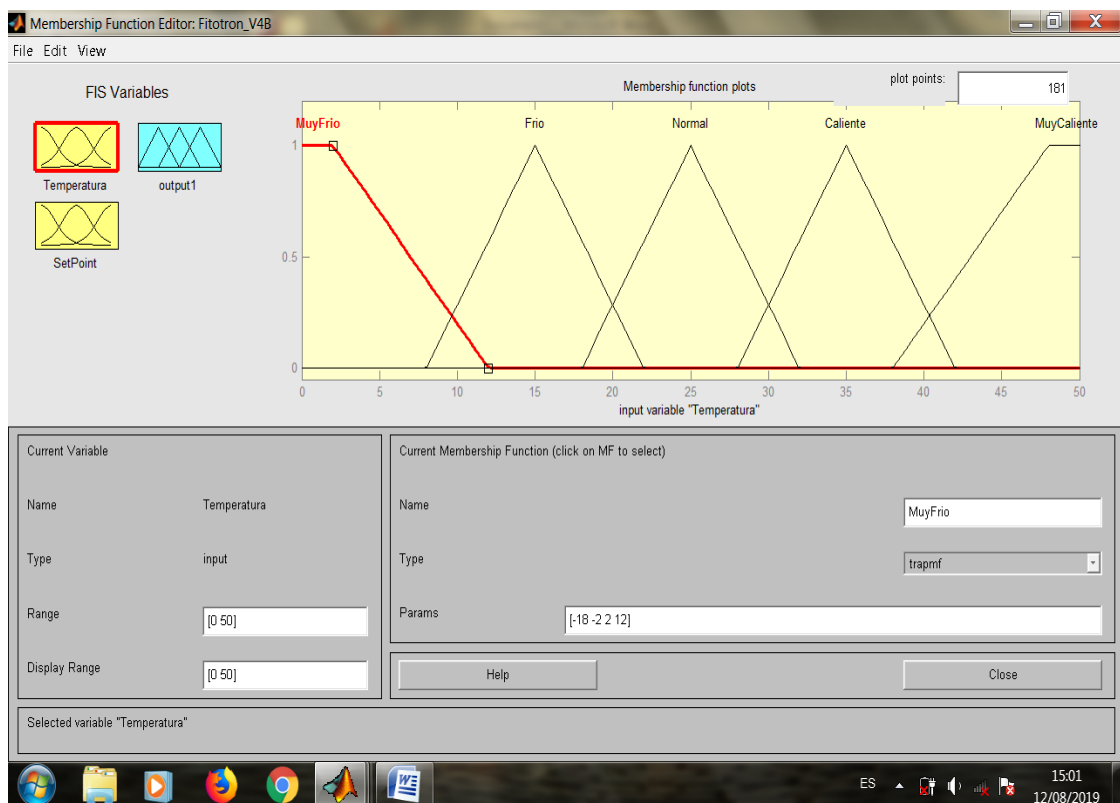


Figura 30: Valores membresía de temperatura

Fuente: Elaboración con Matlab

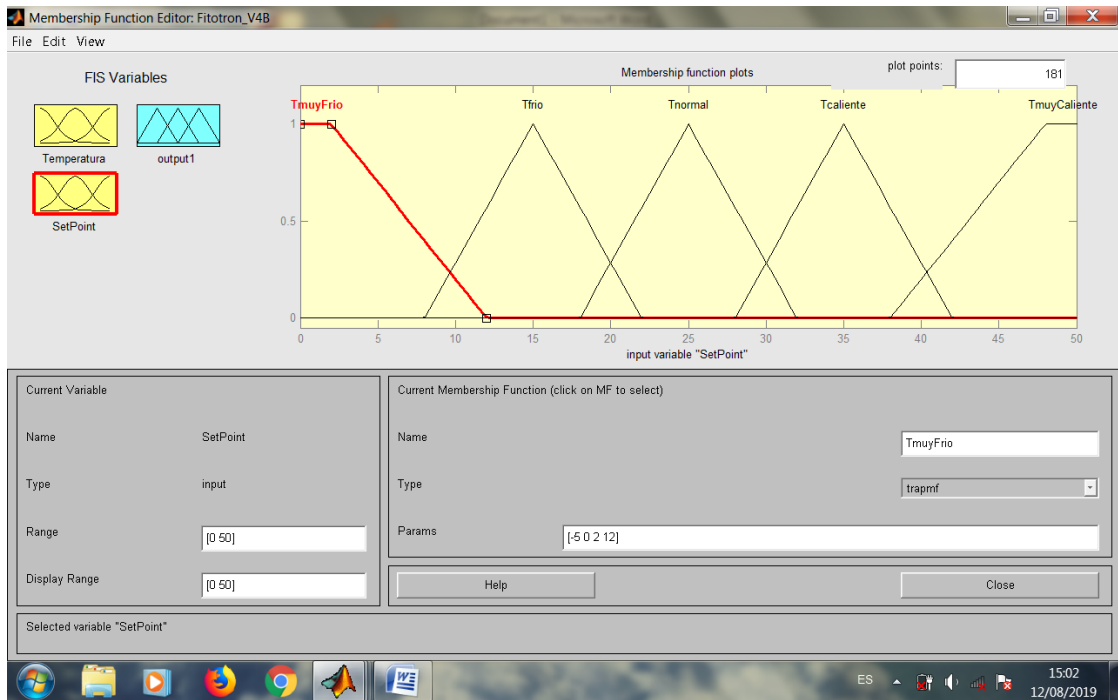


Figura 31: Valores membresía de setpoint

Fuente: Elaboración Matlab

### 3.3.2. Declaración de las reglas de la lógica difusa

A continuación, declaramos las reglas implementadas véase tabla 8 y figura 33, que se usaran para controlar la temperatura. Cabe destacar que mientras más reglas usemos mucha mejor respuesta de salida se obtendrá, pero también elevaremos el consumo de procesamiento de nuestro controlador.

Tabla 8: Declaración de reglas

Reglas	Temperatura	Set Temperatura	Salida-PWM
1	MuyFrio	TMuyFrio	Estable
2	Frio	TMuyFrio	AccFrioN
3	Normal	TMuyFrio	AccFrioNB
4	Caliente	TMuyFrio	AccFrioM
5	MuyCaliente	TMuyFrio	AccFrioA
6	MuyFrio	TFrio	Calef-N
7	Frio	TFrio	Estable
8	Normal	TFrio	AccFrioN
9	Caliente	TFrio	AccFrioNB
10	MuyCaliente	TFrio	AccFrioM
11	MuyFrio	TNormal	Calef-NB
12	Frio	TNormal	Calef-N
13	Normal	TNormal	Estable

Reglas	Temperatura	Set Temperatura	Salida-PWM
14	Caliente	TNormal	AccFrioN
15	MuyCaliente	TNormal	AccFrioNB
16	MuyFrio	TCaliente	Calef-M
17	Frio	TCaliente	Calef-NB
18	Normal	TCaliente	Calef-N
19	Caliente	TCaliente	Estable
20	MuyCaliente	TCaliente	AccFrioN
21	MuyFrio	TMuyCaliente	Calef-A
22	Frio	TMuyCaliente	Calef-M
23	Normal	TMuyCaliente	Calef-NB
24	Caliente	TMuyCaliente	Calef-N
25	MuyCaliente	TMuyCaliente	Estable

Fuente: Elaboración Propia

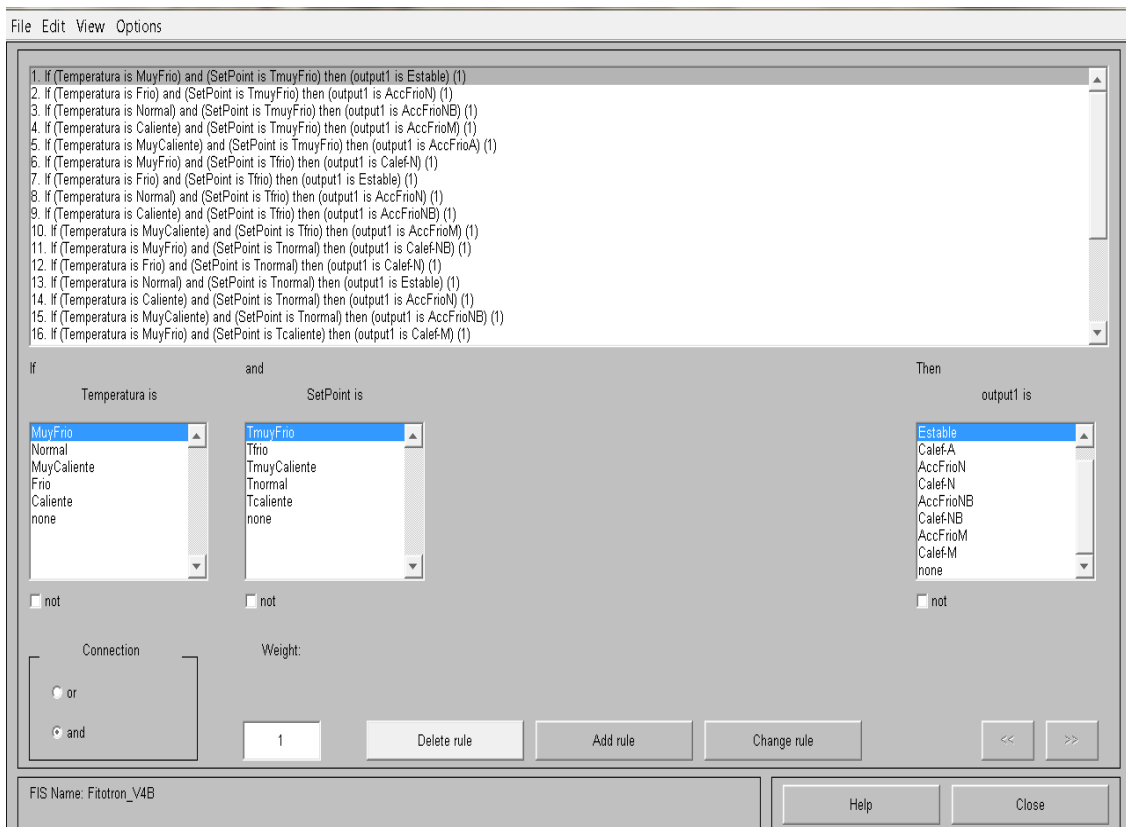


Figura 32: Declaración de las reglas en Matlab

Fuente: Elaboración con Matlab

Las reglas enumeradas previamente se pueden expresar en forma de tabla, véase en la tabla 9, para que sea mucho más fácil su entendimiento.

Tabla 9: Normas difusas para la temperatura

	<b>Setpoint</b>				
<b>Output</b>	<b>TMuyFrio</b>	<b>TFrio</b>	<b>TNormal</b>	<b>TCaliente</b>	<b>TMuyCaliente</b>
<b>MuyFrio</b>	Estable	Calef-N	Calef-NB	Calef-M	Calef-A
<b>Frio</b>	AccFrioN	Estable	Calef-N	Calef-NB	Calef-M
<b>Normal</b>	AccFrioNB	AccFrioN	Estable	Calef-N	Calef-NB
<b>Caliente</b>	AccFrioM	AccFrioNB	AccFrioN	Estable	Calef-N
<b>MuyCaliente</b>	AccFrioA	AccFrioM	AccFrioNB	AccFrioN	Estable

Fuente: Elaboración Propia

Donde:

Acc (Aire acondicionado): Se activa la celda de Peltier y los extractores, para generar el aire frio dentro de la cámara de germinación, también es activado las 2 bombas de agua del sistema de disipador del Peltier.

Calef (Calefactor): Se enciende la resistencia de nicrom y los extractores, para elevar la temperatura dentro de la cabina extrayendo el aire frio y haciendo ingresar el aire caliente

-A(Alto): el sistema reconoce que la temperatura deseada está lejana por lo tanto se activan los actuadores a su máxima potencia.

-M(medio): Cuando se requiere aumentar o disminuir la temperatura de una manera regular.

-N(Normal): Cuando el sistema requiere ralentizar para el aumento o disminución de temperatura dentro de la cámara de germinación.

-NB (Normal bajo): Cuando el sistema no tiene gran necesidad para cambiar la temperatura se activan los actuadores con baja potencia.

### 3.3.3. Salidas y simulación

Podemos ver gráficamente las reglas, expresadas con las funciones de pertenencias y del conjunto a defusificar, véase en la Figura 33.

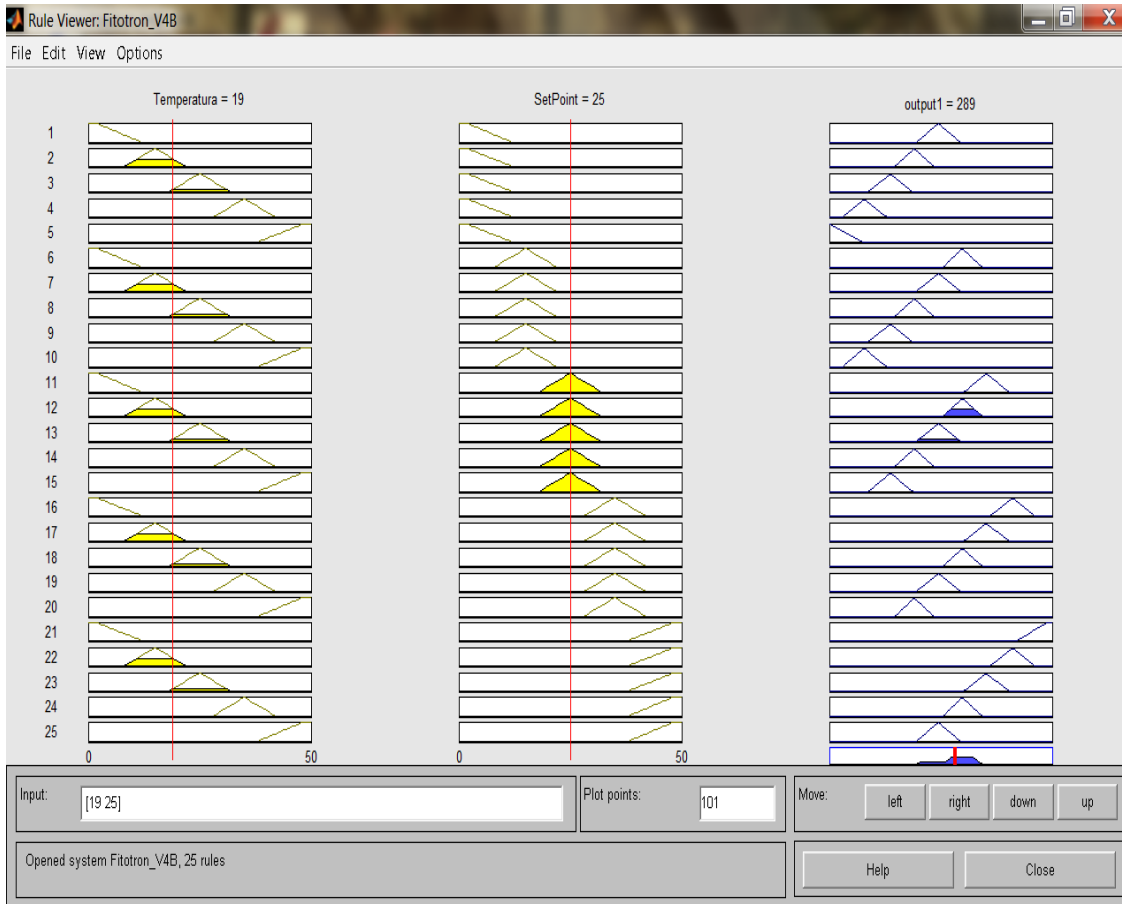


Figura 33: Representación de las reglas mediante su función de pertenencia.

Fuente: Elaboración con Matlab

Una vez creado el controlador se puede observar, véase figura 35, una representación 3D del sistema en el toolbox fuzzy del Matlab, la superficie de control que es el resultado de aplicar la base de reglas y demás parámetros del controlador difuso, esta superficie de control es no-lineal.



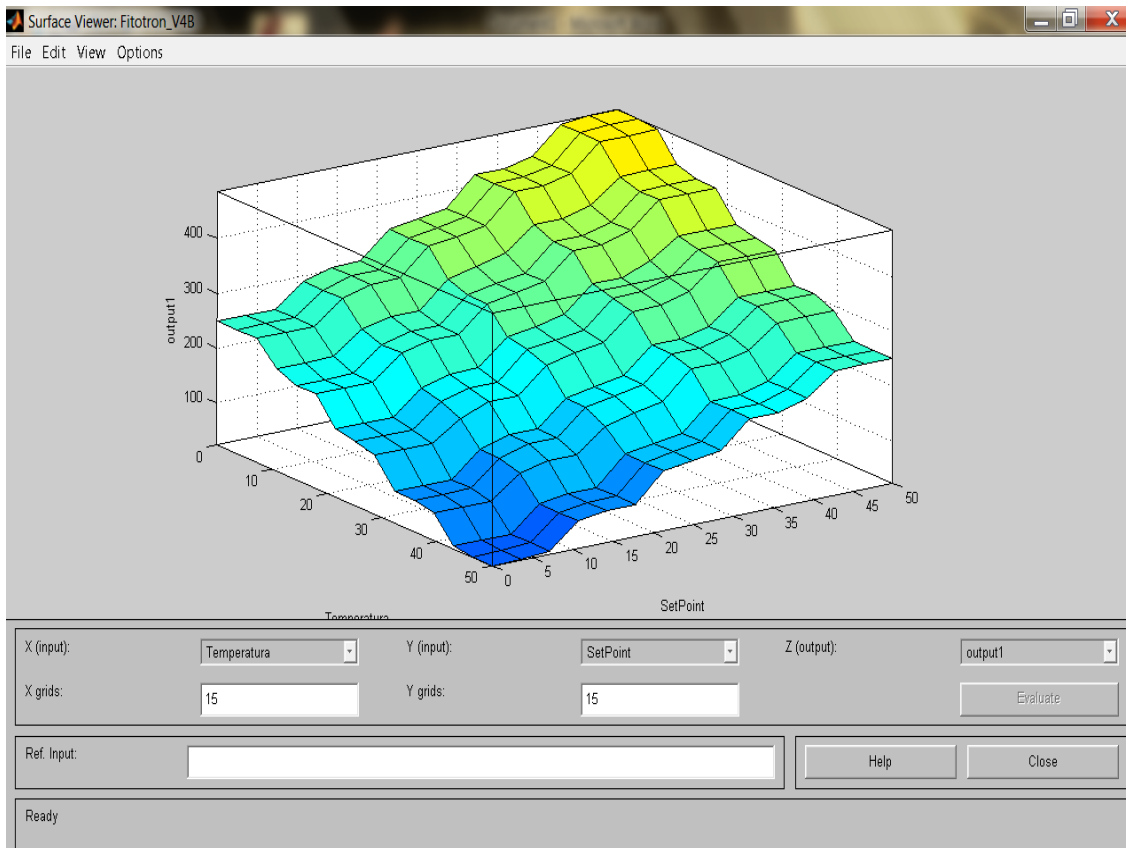


Figura 34: Representación tridimensional del sistema

Fuente: Elaboración con Matlab.

## CAPITULO IV: IMPLEMENTACIÓN Y RESULTADOS

### 4.1. Estructura del fitotrón

El fitotrón se ha diseñado de la siguiente manera, véase figura 36.



Figura 35: Esquema del fitotrón

Fuente: Elaboración propia

#### 4.1.1. Materiales a usar

Los materiales usados para el armazón del fitotrón son básicamente láminas de aluminio (pintadas con spray de color negro), recubiertas con 2 láminas de vidrio en la parte central para garantizar que no haya transferencia de temperatura del exterior al interior o viceversa separado por una distancia de 3mm entre ellas, para colocar las macetas donde van a germinar las semillas se

colocaron tres rejillas pintadas de color blanco en su interior. De los materiales utilizados se mencionan en la tabla 7, se utilizó tornillos y pegamento para unir la estructura, finalmente se colocaron 4 ruedas “garruchas”, en su base para fácil traslado. En la figura 36 se observa esta estructura, y en la figura 37 un diagrama de flujo de su construcción.

Tabla 10: Materiales empleados para la construcción del fitotrón

Ítem	Unidades
Láminas de aluminio 1/2"	15
tubos cuadrados aluminio 3/2"	12
plancha de aluminio 1/4 espesor"	2
rejilla	3
Pegamento	2
Tornillos (caja)	1
ruedas	4
Spray color negro	5

Fuente: Elaboración propia



Figura 36: Estructura del fitotrón

Fuente: Elaboración propia

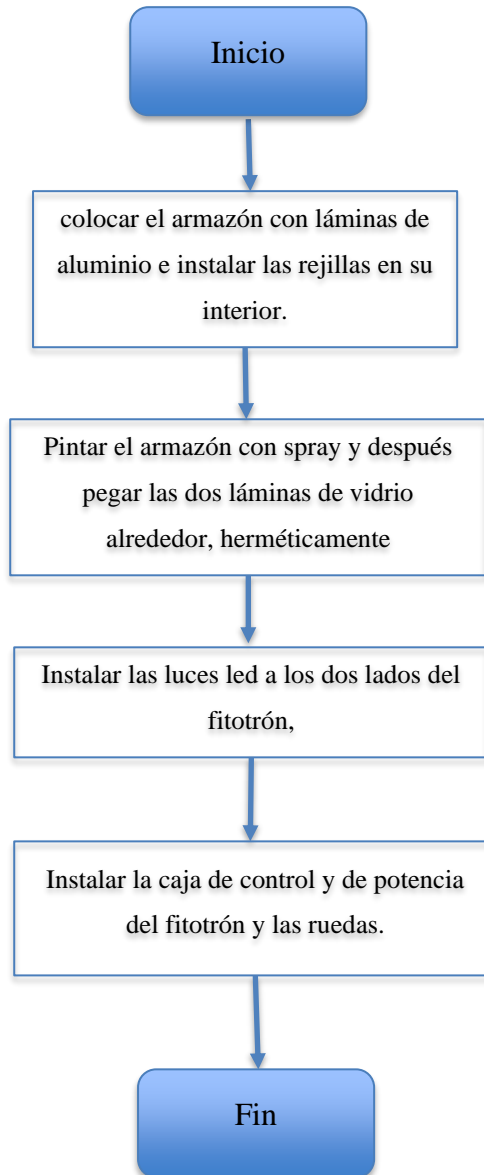


Figura 37: Diagrama de flujo de la implementación del fitotrón de la estructura

Fuente: Elaboración propia

#### 4.1.2. Circuito de calefacción

Para el circuito de calefacción se ha utilizado una resistencia de 64 ohm que se va a calentar cuando se active un voltaje de 220v AC sobre sus contactos, están recubierto con un tubo PVC, (Véase figura 38) y posee extractores de aire tanto en la entrada como en la salida, haciendo que haya un fluido de aire contribuyendo al cambio de temperatura al interior de la cabina. La temperatura máxima lo

fijamos con el dimmer que ha sido seteado manualmente, aproximadamente giramos el potenciómetro ajustándose a 3k ohm.

a) Sensores

- Sensores MCP 9701

b) Actuadores

- 2 extractores
- 1 Resistencia de 64 ohm
- 1 dimmer

c) Elementos para visualización

- Pantalla LCD 16x4

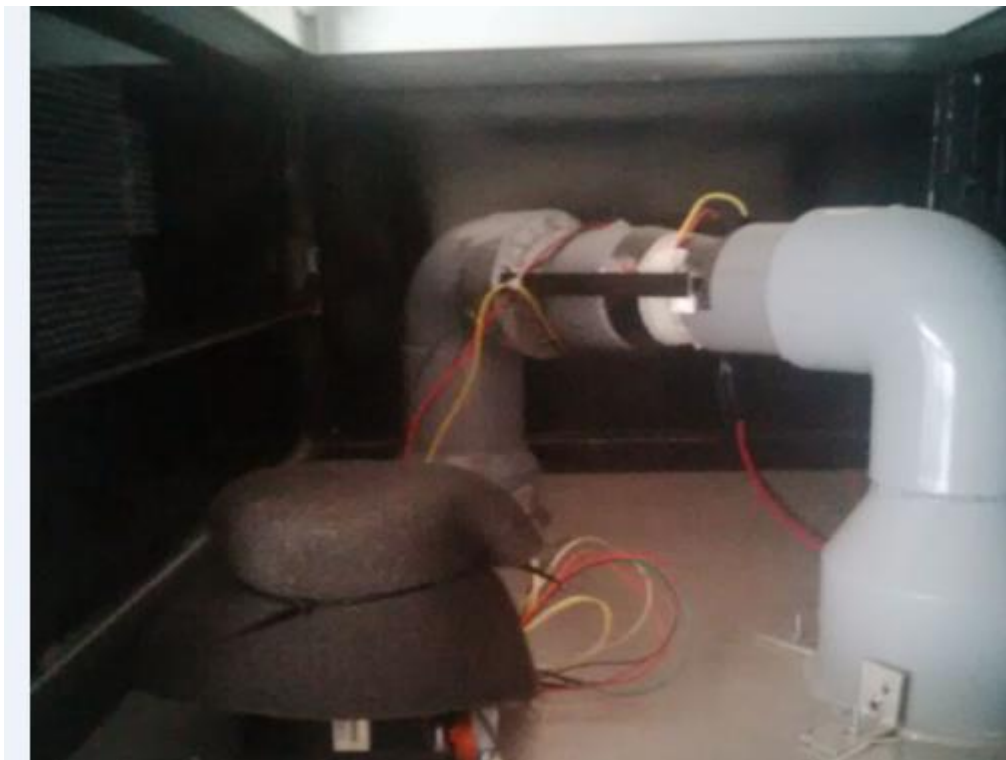


Figura 38: Calefactor puesto en el fitotrón

Fuente: Elaboración propia

### 4.1.3. Circuito de refrigeración.

Para disminuir la temperatura al interior de la cabina utilizamos 6 celdas de Peltier que están puestas 3 de ellas (correspondiente al lado caliente del Peltier) sobre un disipador de aluminio (Figura 39) y las otras 3 en el otro disipador.

#### a) Sensores

- Sensores MCP 9701

#### b) Actuadores

- Celda de Peltier/disipador de calor
- 1 ventilador de aire
- 2 bombas de agua /radiador
- 1 relay

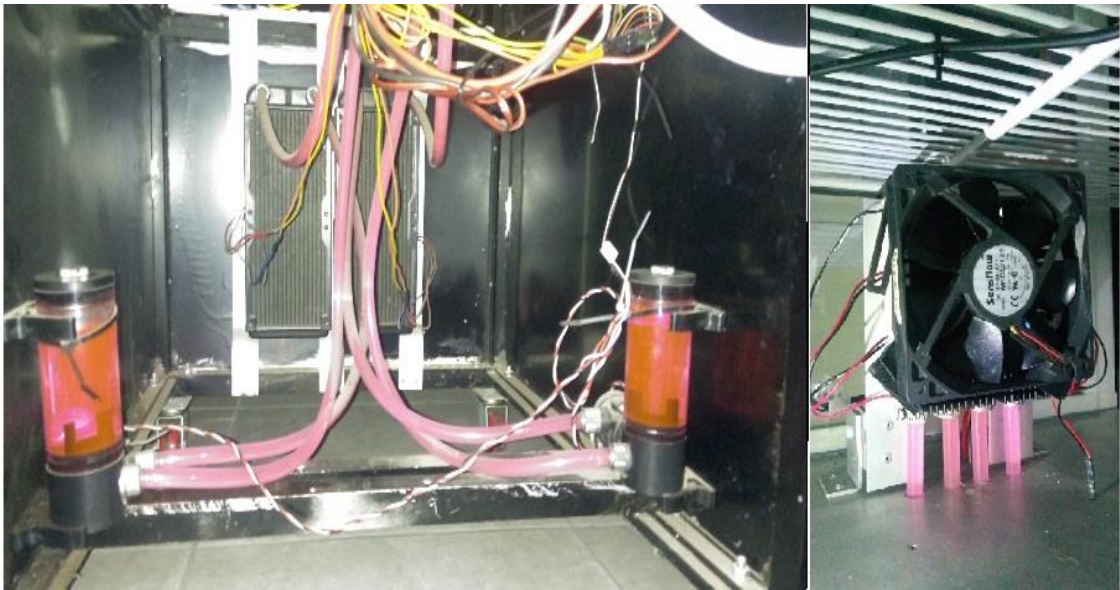
#### c) Elementos para visualización

- Pantalla LCD 16x4



Figura 39: Conexión del disipador del Peltier

Fuente: Elaboración propia



(a)

(b)

Figura 40: a) disipador del Peltier b) cooler expansor del frio en el fitotrón

Fuente: Elaboración propia

En la parte fría de las 6 celdas de Peltier hemos puesto un extractor de aire, Figura 40 (b) que extraerá el aire frío y lo expenderá en toda la cabina.

#### 4.2. Implementación del sistema de control de la temperatura

Se instaló en el interior de la cámara del fitotrón tres sensores de temperatura del modelo mcp9701, están ubicados en el punto medio de las tres rejillas (ver Figura 41), de esta manera cubrimos toda el área interna., haciendo que la lectura de temperatura sea la más correcta posible.



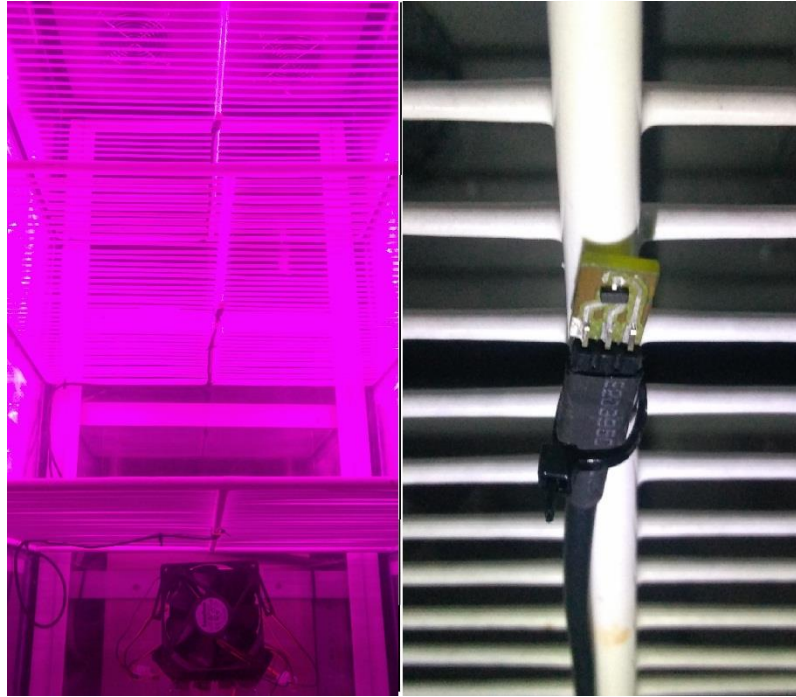


Figura 41: Sensor de temperatura instalado

Fuente: Elaboración propia

Se ha realizado un promedio de los tres sensores de temperaturas, como se observa en la Figura 42

```
346     sensorValue1 = analogRead(A0); // sensor de temp superior
347     sensorValue2 = analogRead(A1); // sensor de temp medio
348     sensorValue3 = analogRead(A2); // sensor de temp inferior
349     sensorValue4 = analogRead(A4); // sensor de humedad
350     sensorValue5 = analogRead(A3); // sensor de Intensidad luminica
351
352     ValorPromTc = (sensorValue1+sensorValue2+sensorValue3)/3.0; //
353     TcRes=((ValorPromTc*(5000/1024.0))-500)/10.0; // Convercion de d
```

Figura 42: Declaración y promedio de la variable de temperatura en el programa

Fuente: Elaboración con software eclipse

Y el resultado se visualiza en la pantalla LCD instalado en la parte externa del fitotrón, donde Tc (temperatura compensada) que es la temperatura actual y Ts (temperatura seteada) que es la temperatura que se desea. Figura 43.



Figura 43: Visualización de la temperatura en la pantalla LCD

Fuente: Elaboración propia

### 4.3. Implementación del sistema de monitoreo de humedad

Para poder monitorear la humedad del fitotrón se colocó dentro del módulo un sensor de humedad HIH 4000 (Figura 44) ubicado en la parte superior, que es de baja potencia este tomará valores y los enviará al Arduino donde se verá visualizado en una pantalla LCD que está ubicado en la parte delantera del fitotrón.

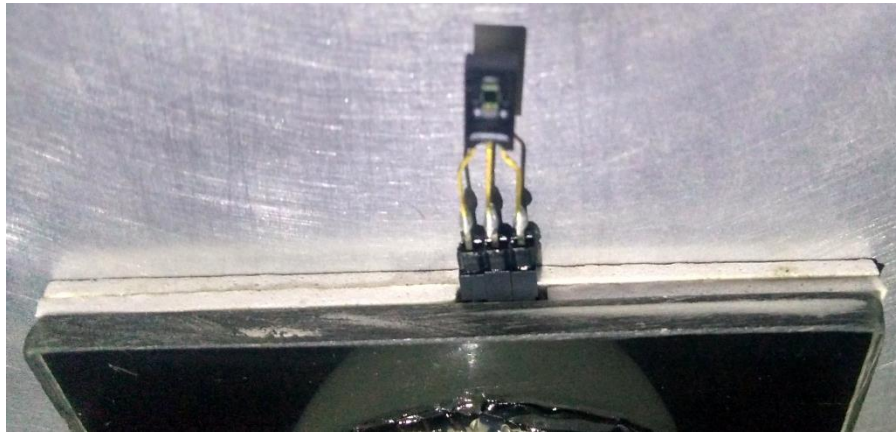


Figura 44: Sensor de humedad instalado

Fuente: Elaboración propia

En Figura 45 se visualiza en la pantalla LCD el sensado de la humedad dentro de la cabina donde

RH%: humedad relativa

RH\_C%: humedad relativa compensada



Figura 45: Visualización del sensor de humedad en la Pantalla LCD

Fuente: Elaboración propia

A continuación, en la Figura 46 vemos la declaración de la función de humedad realizado con el programa Eclipse

```
204 TcRes=((valorFijado*(5000/1024.0))-500)/10.0; // Conversion de digital a grados centigrados
265 RH=(((sensorValue4*156.76)/1024.0)-24.61); // Conversion de digital a Humedad relativa
266 RHcomp=(1000.0*RH)/(1054.6-2.16*TcRes); // Humedad relativa compensado con temperatura
267
```

Figura 46: Declaración de humedad relativa

Fuente: Elaboración con software eclipse

#### 4.4. Implementación del sistema de monitoreo de luminosidad

El sensor de luminosidad TEMT 6000 escogido para este proyecto ha sido puesto en la parte superior, dentro de la cámara del fitotrón situado en el punto medio del mismo. Está cubierto con un foco tipo empavonado, para garantizar que la luz que llegue al sensor sea de manera uniforme, de esta forma tendremos un mejor sentido de luminosidad. Véase figura 47.



Figura 47: Sensor de luminosidad instalado

Fuente: Elaboración propia

Además, se colocó dos paneles de luces en ambos lados del fitotrón que iluminarán la cámara con focos led, contando principalmente con un espectro de colores rojo y azul, funcionan con un Voltaje de 12 VDC. Podemos apreciar estas luces en la figura 48.

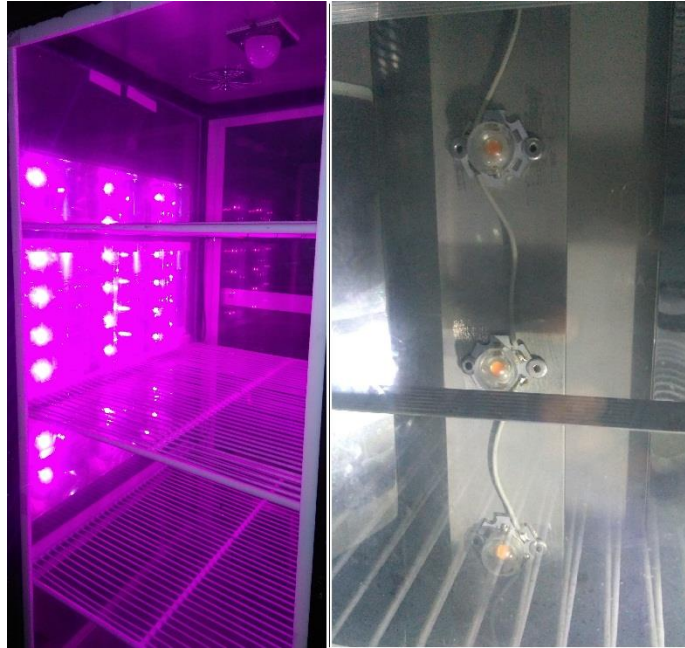


Figura 48: Panel de los leds

Fuente: Elaboración propia

En la siguiente imagen se visualiza la salida del sensor de luz en la pantalla LCD en esta oportunidad se muestra una medición de intensidad de luz de 602.5 lux, figura 49.

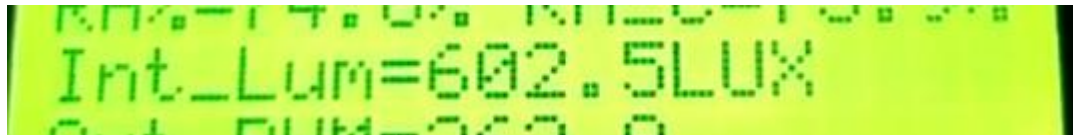


Figura 49: Visualización del sensor temt 6000 en la pantalla LCD

Fuente: Elaboración propia

Mediante programación se realizará el encendido y el apagado de las luces, Figura 50, y para variar la intensidad se ingresa valores al PWM. Figura 51

```

316     pinMode(R_Luces,OUTPUT);
317
318     digitalWrite(R_Luces,HIGH);           // SE ACTIVA
319     digitalWrite(D_DeltaT,LOW);         // SE ACTIVA O DE

```

Figura 50: Función del programa para encender y apagar la luz

Fuente: Elaboración con software eclipse

```
320     digitalWrite(K_Calificador,LOW); // SE ACTI
321
322     analogWrite(pwm_Luces,255);
323     analogWrite(num_Fxt_Sun 0);
```

Figura 51: Variación de pwm para variar la intensidad de luz

Fuente: Elaboración con software eclipse

En la figura 52 se observa el diagrama circuital del fitotrón y en la figura 53 se muestra el diagrama de flujo del mismo.

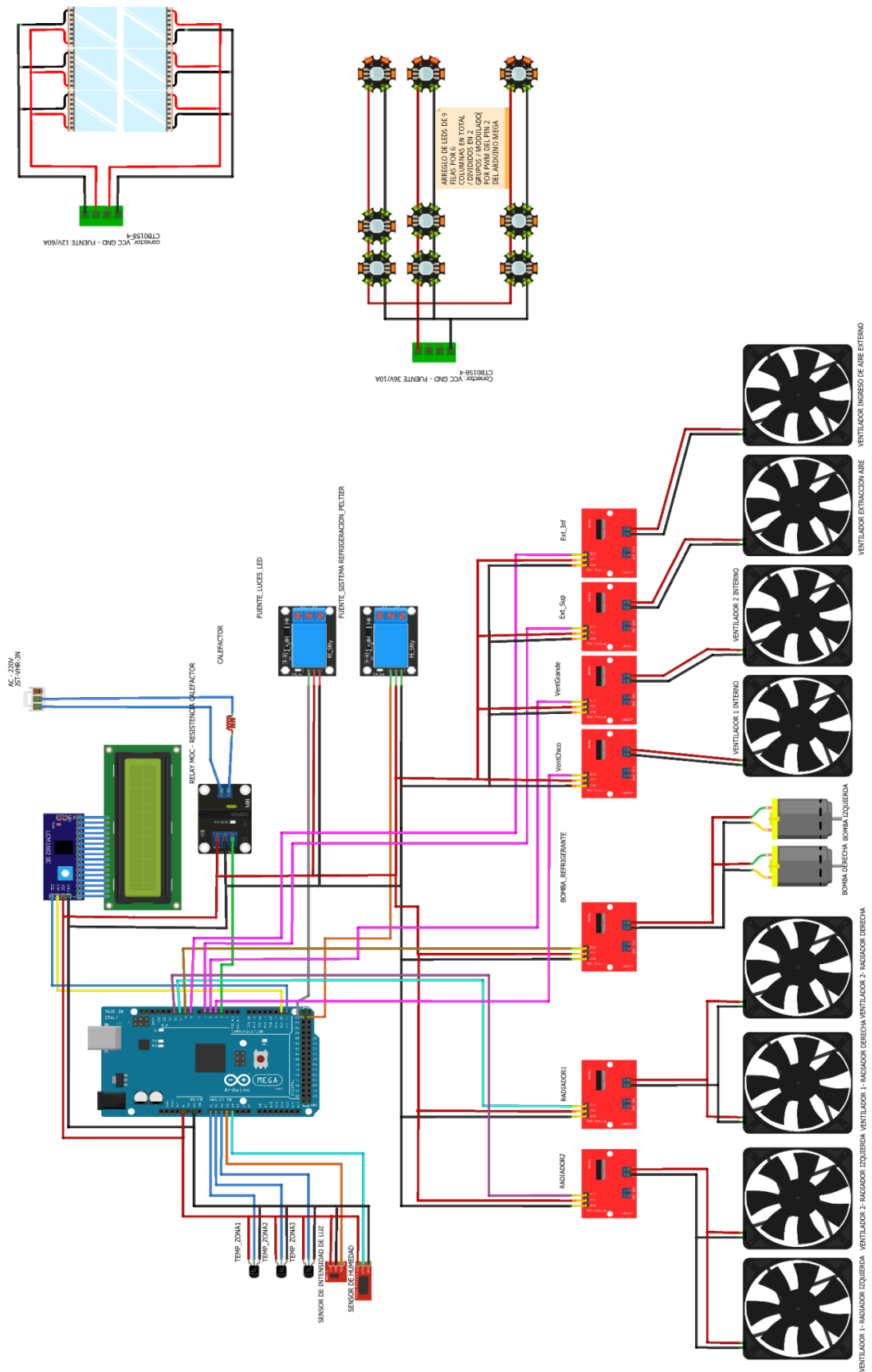


Figura 52: Esquema del circuito implementado en el Fitotrón  
 Fuente: Elaboración usando fritzing

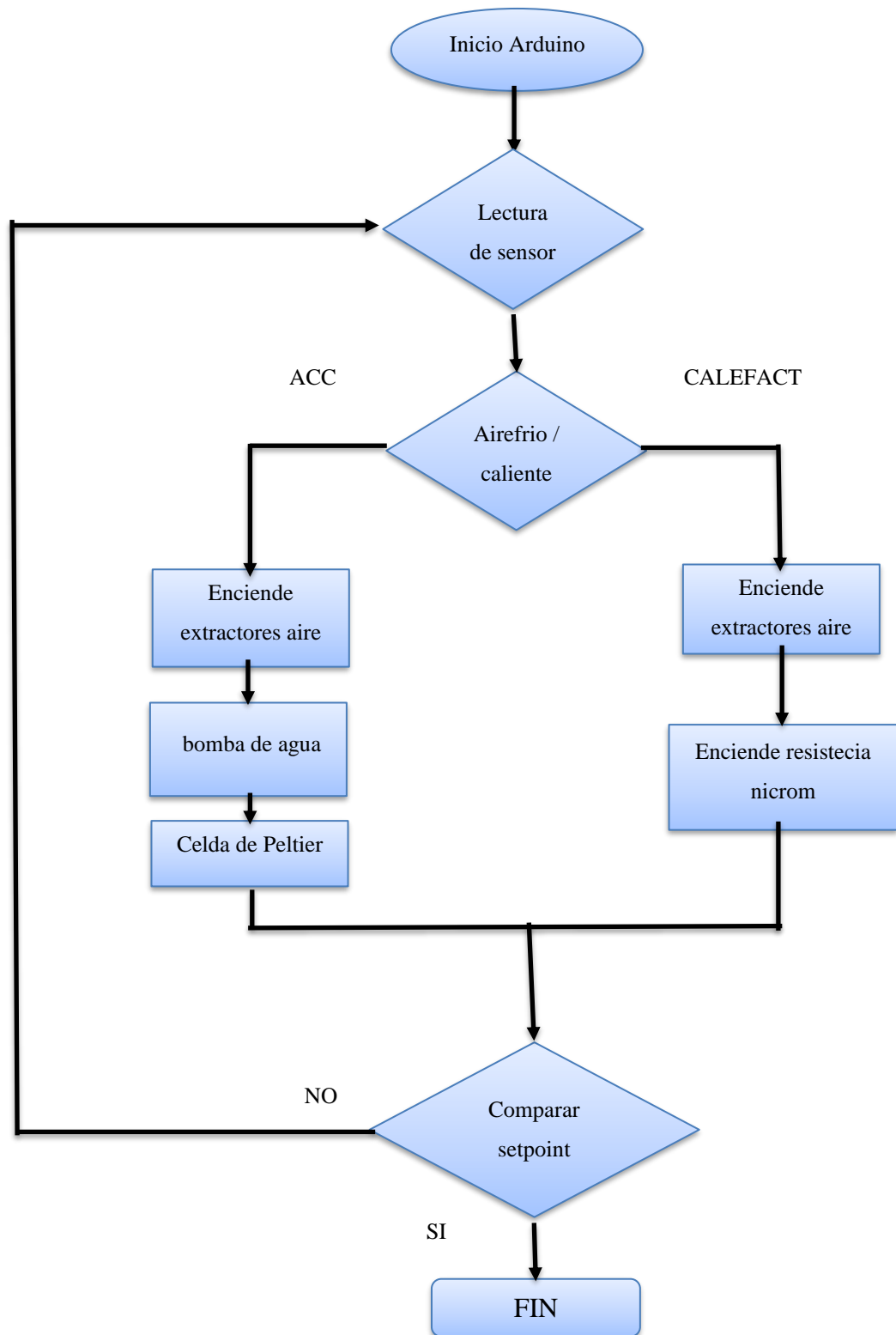


Figura 53: Diagrama de flujo del control de temperatura

Fuente: Elaboración propia

## 4.5. Costos de la implementación.

### 1. Costo equipamiento

Tabla 11: Lista de materiales empleados y precios

Item	Unidades	Precio unitario	Subtotal (S/)
Arduino mega	1	70.00	70.00
abrazaderas	8	0.50	4.00
armazón	1	450.00	450
bomba de agua	2	121.00	242.00
base luces led	2	30.00	60.00
cable 14 AWG	6	2.50	15.00
control de iluminación	1	80.00	80.00
caja de control del fitotrón	1	40.00	40.00
canaletas, silicona	1	40.00	40.00
disipador de aluminio	2	36.00	72.00
fuelle 12v 10 A	1	35.00	35.00
fuelle 24v 10 A	1	45.00	45.00
fuelle 12 v 60A	1	176.00	176.00
garruchas	4	5.00	20.00
llave térmica	1	40.00	40.00
luces led	1	60.00	60.00
Peltier	6	15.00	90.00
radiador	2	101.00	202.00
resistencia de calefacción	1	30.00	30.00
rejillas de división	3	30.00	90.00
regleta de baquelita	4	5.00	20.00
sensor de temperatura	3	10.00	30.00
sensor de humedad	1	30.00	30.00
sensor de luminosidad	1	30.00	30.00
tarjeta LCD	1	30.00	30.00
tarjeta I2C	1	10.00	10.00
tarjeta de potencia DC	2	15.00	30.00
tarjeta de control AC	1	15.00	15.00
tarjeta de dimmer	1	15.00	15.00
tarjeta relay	1	15.00	15.00
ventiladores	4	25.00	100.00
ventiladores del calefactor	2	10.00	20.00
vidrio	4	10.00	40.00
Subtotal			
Total			2246.00

Fuente: Elaboración propia



## 1. Costo diseño

Tabla 12: Costo del diseño del fitotrón

Ítem	Unidades	Precio unitario	Subtotal (S/)
Diseño del fitotrón	72h	12.50	900.00
Total			900.00

Fuente: Elaboración propia

## 2. Costo energético

Tabla 13: Costo energético del fitotrón

Consumo energético del fitotrón en un día		
N°	Ítem	Potencia (Watts)
1	Peltier (6 celdas)	432
2	calefactor	756
3	Bomba agua (2 motores)	12
4	Luces (2 paneles)	151
5	Ventiladores	28
6	Arduino Mega 2560	5
<b>Total aprox</b>		1384

Fuente: Elaboración propia

## CONCLUSIONES

Este proyecto se basó en la necesidad de crear una forma segura de germinar semillas a bajo costo controlando la temperatura y monitoreando la humedad relativa y la luminosidad, de lo cual podemos mencionar lo siguiente:

1. Se logró implementar las variables de control y monitoreo (temperatura, luminosidad y humedad) necesarias para el funcionamiento del fitotrón, fueron ingresadas al controlador Arduino usando la lógica difusa utilizando el método Mamdani.
2. Se consiguió los sensores y actuadores en el mercado local, de manera que se ahorró el costo de importación a su vez nos proporcionó una fácil instalación al no ocupar mucho espacio dentro del equipo. La adquisición de datos de los sensores fueron los necesarios para el fitotrón.
3. Se logró simular la salida del controlador Difuso en función de las variables de entrada como se mostró en la imagen 3D.
4. La electrónica del fitotrón resulto no ser muy compleja en comparación con otros equipos de similares características.
5. El programa Eclipse que se adecuó muy bien a las necesidades que se requerían, y redujo la línea de comandos en comparación que si se hubiera utilizado otro programa.
6. La celda de Peltier requiere de un buen disipador en la parte caliente, para que la disminución de temperatura sea la eficiente.

## **OBSERVACIONES**

1. Considerar que el módulo es solo para la germinación de semillas luego de eso se sembrará la planta en otro lugar más apropiado.
2. Mencionar la importancia de la protección de los cables que estén instalados de forma segura, y aislados. Además, para que no se presente interferencia con las señales de control.
3. Cabe señalar que la corriente que ingresa a la resistencia del calefactor se estableció de forma manual con un dimmer para que al encenderse no se dispare la temperatura en la cabina.
4. La implementación del fitotrón se realizó lo más hermético posible para que no exista fugas de temperatura en la cabina.

## RECOMENDACIONES

1. Usar otra celda de Peltier si se requiere bajar aún más la temperatura, tener en cuenta que esto también haría que se eleve el consumo de energía.
2. Usar un módulo inalámbrico para evitar el cableado en varias partes del circuito.
3. Implementar una línea de comandos adicional para ingresar los datos directamente desde la pantalla Led
4. Emplear un pequeño humidificador que se active cuando el sensor detecte valores de humedad por debajo de los requeridos

## BIBLIOGRAFÍA Y REFERENCIAS

- Barrera Peña, N. F., & Maldonado Peña, J. C. (2008). Diseño, control e implementación de factores climáticos y ambientales en una cámara para el cultivo hidropónico de lechuga. (Tesis de pregrado). Universidad de San Buenaventura, Colombia.
- Bautista Angeles, C. (2012). *Códigos de la celda de Peltier*. [Figura]. Obtenido de <http://controlcbautista.blogspot.com/>
- Boteron, F. (s.f.). Control clásico y moderno. Universidad de misiones, Argentina. Obtenido de <https://slideplayer.es/slide/8887567/>
- Bozov, N. (2019). Características de la señal PID. [Figura]. Obtenido de <http://nbozov.com/article/post/32/PID-Controller-basics>
- California institute of technology®. (1982). *Interview with James F. Bonner Eeuu*: <http://oralhistories.library.caltech.edu/>. Obtenido de [http://oralhistories.library.caltech.edu/15/1/OH\\_Bonner\\_J.pdf](http://oralhistories.library.caltech.edu/15/1/OH_Bonner_J.pdf)
- DHgate.com®. (s.f.). *Efecto de la luz en el desarrollo de una planta*. [Figura]. Obtenido de <https://es.dhgate.com/product/8pcs-100w-led-grow-lights-full-spectrum-150leds/438946050.html>
- Diaz Arca, J. (2018). Ejercicios aplicando lógica difusa en riegos financieros. Obtenido de <https://medium.com/@javierdiazarca/1%C3%B3gica-difusa-ejercicios-propuestos-b99603ef1bc0>
- Duarte Galván, C. (2011). Sistema de control climático de bajo costo para invernadero basado en FPGA. (Tesis de pregrado). Universidad Autónoma de Querétaro, México.
- educarparaelcambio.com®. (s.f.). *Esquema descriptivo de entradas y salidas del microcontrolador*. [Figura]. Obtenido de <https://educarparaelcambio.com/arduino/que-es-arduino-y-para-que-sirve/>
- Elprocus®. (2019). *Esquema de un controlador PID*. [Figura]. Obtenido de <https://www.elprocus.com/the-working-of-a-pid-controller/>
- Gonzales Huisa, A. R. (2015). Diseño de una cámara multiespectral de 4 bandas para agricultura de precisión (Tesis de pregrado). Pontificia Universidad Católica del Perú, Lima.

- Grupo Nova Àgora. (s.f.). *Feria virtual medición y control*. Obtenido de <http://www.interempresas.net/Medicion/FeriaVirtual/Producto-Camaras-climaticas-para-crecimiento-de-plantas-y-germinacion-Fitotron-SGC-120-174381.html>
- Herrador Moreno, M. (s.f.). *Estado del arte de las distintas tecnologías de refrigeración solar*. Obtenido de [http://bibing.us.es/proyectos/abreproy/70166/fichero/Archivo\\_Cap%C3%ADulo+3.pdf](http://bibing.us.es/proyectos/abreproy/70166/fichero/Archivo_Cap%C3%ADulo+3.pdf)
- Honeywell International®. (2010). *Respuesta lineal del sensor HIH 4000*. [Figura]. Obtenido de <https://sensing.honeywell.com/honeywell-sensing-hih4000-series-product-sheet-009017-5-en.pdf>
- Huamán Lizana, K. F. (2018). Diseño de un sistema automático de riego tecnificado para las áreas verdes de la escuela de ingeniería electrónica y facultad de ciencias físicas y matemáticas (Tesis de pregrado). Universidad Nacional Pedro Ruiz Gallo, Lambayeque.
- International Society for Horticultural Science®. (2019). *Phytotronics and horticulture*. Eeuu: <https://www.ishs.org/>. Obtenido de [https://www.actahort.org/books/39/39\\_6.htm](https://www.actahort.org/books/39/39_6.htm)
- Ircservices.com®. (s.f.). *Propiedades del alambre de nicrom*. Obtenido de [http://www.ircservices.com/propiedades-del-alambre-del-nicrom\\_XOEXwwnN/](http://www.ircservices.com/propiedades-del-alambre-del-nicrom_XOEXwwnN/)
- Livrosdeamor®. (s.f.). Control clásico vs Control moderno. Obtenido de <https://livrosdeamor.com.br/documents/control-clasico-vs-control-moderno-1-5bc9620419876>
- Maynard, D., & Hochmuth, G. (2012). Temperatura y días en que tarda en germinar la semilla [tabla]. Handbook for vegetables grower. Obtenido de <https://rockymountainseeds.org/images/pdfs/KnottsHandbook2012.pdf>
- Mazquiarán Andrade, C. (2017). Control Climático de un invernadero mediante lógica difusa. (Tesis de pregrado). Universidad Politécnica de Madrid. España.
- Microchip technology ®. (2019). Sensor Mcp9701. [Figura]. Obtenido de <https://www.microchip.com/wwwproducts/en/MCP9701>
- Monografias.com®. (s.f.). Funcionamiento de una celda de Peltier. [Figura]. Obtenido de <https://www.monografias.com/trabajos16/modulos-peltier/modulos-peltier.shtml>

- Moreno–Ortiz, D. G., & García–Romero, D. J. (2017). Prototipo generador de agua con celdas de Peltier (Tesis de pregrado). Universidad distrital Francisco José de Caldas, Colombia.
- Picturingmeteorology.com®. (s.f.). *Caltechs fabulous weather factory:1949-1971*. Obtenido de <https://picturingmeteorology.com/home/2017/4/26/caltechs-fabulous-weather-factory-1949-71>
- Pontigo Vásquez, L. A. (2011). Rediseño y construcción de un termo-estimulador. (Tesis de pregrado). Universidad Austral, Chile.
- Portafruticola.com®. (2017). *Temperaturas y tiempos de la germinación de las semillas. Chile: <https://www.portafruticola.com/>*. Obtenido de <https://www.portafruticola.com/noticias/2017/11/02/temperaturas-tiempos-la-germinacion-las-semillas/>
- Premiertech®. (s.f.). *Como medir la calidad y cantidad de luz Francia: <https://www.pthorticulture.com/fr/>*. Obtenido de <https://www.pthorticulture.com/es/centro-de-formacion/como-medir-la-calidad-y-la-cantidad-de-luz/>
- Rubio Ramírez, C., Martheyn Lizarazo, G., & Vera Duarte, E. (2017). *Termoelectricidad: Uso de las celdas de Peltier en el campo de la refrigeración y sus principales aplicaciones*. Obtenido de <https://revistas.uniminuto.edu/index.php/Inventum/article/download/1502/1452>
- Texas Instruments ®. (2017). *Sensor de temperatura LM35. [tabla]*. Obtenido de <http://www.ti.com/lit/ds/symlink/lm35.pdf>
- Universidad Nacional de mar del Plata. (s.f.). Historia del control PID. Obtenido de <http://www3.fi.mdp.edu.ar/control4c7/APUNTES/Clase%207%20-%20PID.pdf>
- Universidad politécnica de Valencia. (s.f.). *Germinación de las semillas. España: <http://www.euita.upv.es/>*. Obtenido de [http://www.euita.upv.es/varios/biologia/temas/tema\\_17.htm](http://www.euita.upv.es/varios/biologia/temas/tema_17.htm)
- usroasterie.com®. (2019). *Usos de un alambre de nicrom*. Obtenido de <https://www.usroasterie.com/usos-de-un-alambre-de-nicrom.html>
- Villajulca, J. C. (2019). Control ON/OFF. [Figura]. Obtenido de <https://instrumentacionycontrol.net/control-on-off-o-todo-nada/>
- Vishay intertechnology, inc. (2017). Sensibilidad del sensor de luminosidad. [Figura]. Obtenido de <https://www.vishay.com/docs/81579/temt6000.pdf>

vs-elec.fr®. (2019). *Sensor de luminosidad tmt6000*. [Figura]. Obtenido de <https://www.vs-elec.fr/es/lumiere/2171-sensor-de-luz-ambiental-de-brillo-temt-6000-3665662014382.html>

Weiss Technik India Private. (s.f.). *Environmental Simulations*. Obtenido de <https://www.indiamart.com/weissttechnik/environmental-simulations.html>

Wikipedia®. (2019). Consumo de energía para las variedades de emisores de luz. [Figura]. Obtenido de [https://es.wikipedia.org/wiki/L%C3%A1mpara\\_led#/media](https://es.wikipedia.org/wiki/L%C3%A1mpara_led#/media)

Wikipedia®. (2019). *Phytotron*. Obtenido de <https://en.m.wikipedia.org/wiki/Phytotron>

Worldoftrons®. (2017). *Fitotrón australiano*[figura]. *Blog Phytotron*. Obtenido de <http://www.worldoftrons.com/blog/2017/2/25/phytotron>

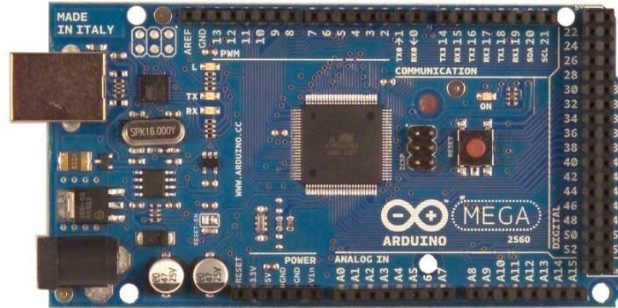
Yoreparo.com®. (s.f.). Características del alambre de nicrom. [Figura]. Obtenido de <https://www.yoreparo.com/electricidad/electricidad-industrial/preguntas/1567229/calculo-de-alambre-de-nicrom>



## ANEXOS

Anexo 1: DATASHEET DE CONTROLADOR .....	71
Anexo 2: DATOS TÉCNICOS DEL PELTIER.....	76
Anexo 3: CÓDIGO FINAL .....	77

## Anexo 1: DATASHEET DE CONTROLADOR



### Overview

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

The Mega 2560 is an update to the Arduino Mega, which it replaces.

### Summary

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

## Schematic & Reference Design

EAGLE files: arduino-mega2560-reference-design.zip

Schematic: arduino-mega2560-schematic.pdf

## Power

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

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

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

The power pins are as follows:

- **vin.** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5v.** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3v3.** A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **gnd.** Ground pins.

## Memory

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

## Input and Output

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

- **Serial:** 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX). Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts:** 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.
- **Pwm:** 0 to 13. Provide 8-bit PWM output with the `analogWrite()` function.
- **Spi:** 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS). These pins support SPI communication using the SPI library. The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.
- **Led: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **I<sup>2</sup>c: 20 (sda) and 21 (scl).** Support I<sup>2</sup>C (TWI) communication using the Wire library (documentation on the Wiring website). Note that these pins are not in the same location as the I<sup>2</sup>C pins on the Duemilanove or Diecimila.

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

There are a couple of other pins on the board:

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

### **Programming**

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

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

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

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

### **Automatic (Software) Reset**

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

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

data.

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

### **USB Overcurrent Protection**

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

### **Physical Characteristics and Shield Compatibility**

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

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

## **Anexo 2: DATOS TÉCNICOS DEL PELTIER**

### Características de la Celda Peltier Tec1-12706:

- Fabricada en material cerámico
- Cables de alimentación de 30cm
- Temperatura lado caliente: 50-57°C
- Diferencial de temperatura: 66-75°C
- Corriente máxima: 6.4A
- Voltaje nominal: 12V
- Voltaje máximo: 16.4V
- Potencia nominal: 72W
- Resistencia de la celda 1.98-2.30 Ohms

Fuente: <https://www.geekfactory.mx/tienda/otros-accesorios/celda-peltier-tec1-12706/>

### Anexo 3: CÓDIGO FINAL

```
//Programación realizada en el programa eclipse
#include "Arduino.h"
// LIBRERIA PARA CONTROL LCD POR I2C
#include <wire.h>
#include<LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x3F,20,4);

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

//Instanciando un objeto de biblioteca
Fuzzy* fuzzy = new Fuzzy();
//Funciones de Pertenencia de Entrada - TEMPERATURA
FuzzySet* MF = new FuzzySet(0, 0, 5, 12);
FuzzySet* F = new FuzzySet(8, 15, 15, 22);
FuzzySet* N = new FuzzySet(18, 25, 25, 32);
FuzzySet* C = new FuzzySet(28, 35, 35, 42);
FuzzySet* MC = new FuzzySet(38, 45, 50, 50);
//Funciones de Pertenencia de Entrada - SetPoint Temperatura
FuzzySet* STMF = new FuzzySet(0, 0, 5, 12);
FuzzySet* STF = new FuzzySet(8, 15, 15, 22);
FuzzySet* STN = new FuzzySet(18, 25, 25, 32);
FuzzySet* STC = new FuzzySet(28, 35, 35, 42);
FuzzySet* STMC = new FuzzySet(38, 45, 50, 50);
//Funciones de Pertenencia de Salida - Output(PWM)
FuzzySet* ACCF4 = new FuzzySet(0, 0, 0, 75);
FuzzySet* ACCF3 = new FuzzySet(30, 80, 80, 130);
FuzzySet* ACCF2 = new FuzzySet(90, 140, 140, 190);
FuzzySet* ACCF1 = new FuzzySet(150, 195, 195, 240);
```



```

FuzzySet* EST = new FuzzySet(200, 250, 250, 300);
FuzzySet* CF1 = new FuzzySet(260, 305, 305, 350);
FuzzySet* CF2 = new FuzzySet(310, 360, 360, 410);
FuzzySet* CF3 = new FuzzySet(370, 420, 420, 470);
FuzzySet* CF4 = new FuzzySet(430, 500, 511, 511);

//.

int sensorValue1 = 0;    // valor de los 3 sensores de temperatura
int sensorValue2 = 0;
int sensorValue3 = 0;
int sensorValue4 = 0;    // sensor de humedad relativa
int sensorValue5 = 0;

float ValorPromTc = 0;    // promedio de temperaturas
float RH = 0;            // Humedad Relativa 1
float RHcomp = 0;       // RH% compensado con la temperatura
float TcRes = 0;        // temperatura final
float IntLumLUX = 0;    // Intensidad lumínica en LUX

//-          -PINES DE SALIDAS DIGITALES-
//Salidas Digitales
const uint8_t R_Luces=22;
const uint8_t R_Peltier=23;
const uint8_t R_Calefactor=3;
//Salidas moduladas - PWM
const uint8_t pwm_Luces=2;
const uint8_t pwm_Ext_Sup=6;
const uint8_t pwm_Ext_Inf=8;
const uint8_t pwm_VentChico_cab=5;
const uint8_t pwm_VentGrande_cab=4;
const uint8_t pwm_BombasAgua=9;
const uint8_t pwm_Rad_I=10;
const uint8_t pwm_Rad_II=11;

int ON_peltier=0;
int OFF_peltier=0;

void setup()
{

```

```

//Conjuntos Difusos de Entrada
FuzzyInput* Temperatura = new FuzzyInput(1);

Temperatura->addFuzzySet(MF);
Temperatura->addFuzzySet(F);
Temperatura->addFuzzySet(N);
Temperatura->addFuzzySet(C);
Temperatura->addFuzzySet(MC);

fuzzy->addFuzzyInput(Temperatura);

FuzzyInput* SetTemperatura = new FuzzyInput(2);

SetTemperatura->addFuzzySet(STMf);
SetTemperatura->addFuzzySet(STF);
SetTemperatura->addFuzzySet(STN);
SetTemperatura->addFuzzySet(STC);
SetTemperatura->addFuzzySet(STMc);

fuzzy->addFuzzyInput(SetTemperatura);

//Conjunto Difuso de Salida
FuzzyOutput* PWM1 = new FuzzyOutput(1);

PWM1->addFuzzySet(ACCF4);
PWM1->addFuzzySet(ACCF3);
PWM1->addFuzzySet(ACCF2);
PWM1->addFuzzySet(ACCF1);
PWM1->addFuzzySet(EST);
PWM1->addFuzzySet(CF1);
PWM1->addFuzzySet(CF2);
PWM1->addFuzzySet(CF3);
PWM1->addFuzzySet(CF4);

fuzzy->addFuzzyOutput(PWM1);

//DISEÑANDO LAS REGLAS DIFUSAS

//++++ANTECEDENTES++++
FuzzyRuleAntecedent* ifMFandSTMf = new FuzzyRuleAntecedent();

```

```

ifMFandSTMF->joinWithAND(MF, STMF);
FuzzyRuleAntecedent* ifFandSTMF = new FuzzyRuleAntecedent();
ifFandSTMF->joinWithAND(F, STMF);
FuzzyRuleAntecedent* ifNandSTMF = new FuzzyRuleAntecedent();
ifNandSTMF->joinWithAND(N, STMF);
FuzzyRuleAntecedent* ifCandSTMF = new FuzzyRuleAntecedent();
ifCandSTMF->joinWithAND(C, STMF);
FuzzyRuleAntecedent* ifMCandSTMF = new FuzzyRuleAntecedent();
ifMCandSTMF->joinWithAND(MC, STMF);
FuzzyRuleAntecedent* ifMFandSTF = new FuzzyRuleAntecedent();
ifMFandSTF->joinWithAND(MF, STF);
FuzzyRuleAntecedent* ifFandSTF = new FuzzyRuleAntecedent();
ifFandSTF->joinWithAND(F, STF);
FuzzyRuleAntecedent* ifNandSTF = new FuzzyRuleAntecedent();
ifNandSTF->joinWithAND(N, STF);
FuzzyRuleAntecedent* ifCandSTF = new FuzzyRuleAntecedent();
ifCandSTF->joinWithAND(C, STF);
FuzzyRuleAntecedent* ifMCandSTF = new FuzzyRuleAntecedent();
ifMCandSTF->joinWithAND(MC, STF);
FuzzyRuleAntecedent* ifMFandSTN = new FuzzyRuleAntecedent();
ifMFandSTN->joinWithAND(MF, STN);
FuzzyRuleAntecedent* ifFandSTN = new FuzzyRuleAntecedent();
ifFandSTN->joinWithAND(F, STN);
FuzzyRuleAntecedent* ifNandSTN = new FuzzyRuleAntecedent();
ifNandSTN->joinWithAND(N, STN);
FuzzyRuleAntecedent* ifCandSTN = new FuzzyRuleAntecedent();
ifCandSTN->joinWithAND(C, STN);
FuzzyRuleAntecedent* ifMCandSTN = new FuzzyRuleAntecedent();
ifMCandSTN->joinWithAND(MC, STN);
FuzzyRuleAntecedent* ifMFandSTC = new FuzzyRuleAntecedent();
ifMFandSTC->joinWithAND(MF, STC);
FuzzyRuleAntecedent* ifFandSTC = new FuzzyRuleAntecedent();
ifFandSTC->joinWithAND(F, STC);
FuzzyRuleAntecedent* ifNandSTC = new FuzzyRuleAntecedent();
ifNandSTC->joinWithAND(N, STC);
FuzzyRuleAntecedent* ifCandSTC = new FuzzyRuleAntecedent();
ifCandSTC->joinWithAND(C, STC);
FuzzyRuleAntecedent* ifMCandSTC = new FuzzyRuleAntecedent();
ifMCandSTC->joinWithAND(MC, STC);
FuzzyRuleAntecedent* ifMFandSTMC = new FuzzyRuleAntecedent();

```

```

    ifMFandSTMC->joinWithAND(MF, STMC);
    FuzzyRuleAntecedent* ifFandSTMC = new FuzzyRuleAntecedent();
    ifFandSTMC->joinWithAND(F, STMC);
    FuzzyRuleAntecedent* ifNandSTMC = new FuzzyRuleAntecedent();
    ifNandSTMC->joinWithAND(N, STMC);
    FuzzyRuleAntecedent* ifCandSTMC = new FuzzyRuleAntecedent();
    ifCandSTMC->joinWithAND(C, STMC);
    FuzzyRuleAntecedent* ifMCandSTMC = new FuzzyRuleAntecedent();
    ifMCandSTMC->joinWithAND(MC, STMC);

//****CONSECUENTES****
    FuzzyRuleConsequent* thenEST = new FuzzyRuleConsequent();
    thenEST->addOutput(EST);
    FuzzyRuleConsequent* thenACCF1 = new FuzzyRuleConsequent();
    thenACCF1->addOutput(ACCF1);
    FuzzyRuleConsequent* thenACCF2 = new FuzzyRuleConsequent();
    thenACCF2->addOutput(ACCF2);
    FuzzyRuleConsequent* thenACCF3 = new FuzzyRuleConsequent();
    thenACCF3->addOutput(ACCF3);
    FuzzyRuleConsequent* thenACCF4 = new FuzzyRuleConsequent();
    thenACCF4->addOutput(ACCF4);
    FuzzyRuleConsequent* thenCF1 = new FuzzyRuleConsequent();
    thenCF1->addOutput(CF1);
    FuzzyRuleConsequent* thenCF2 = new FuzzyRuleConsequent();
    thenCF2->addOutput(CF2);
    FuzzyRuleConsequent* thenCF3 = new FuzzyRuleConsequent();
    thenCF3->addOutput(CF3);
    FuzzyRuleConsequent* thenCF4 = new FuzzyRuleConsequent();
    thenCF4->addOutput(CF4);

//+*+*REGLAS*+*+
    //Regla 1
    FuzzyRule* LDregla1 = new FuzzyRule(1, ifMFandSTMF, thenEST);
    fuzzy->addFuzzyRule(LDregla1);
    //Regla 2
    FuzzyRule* LDregla2 = new FuzzyRule(2, ifFandSTMF, thenACCF1);
    fuzzy->addFuzzyRule(LDregla2);
    //Regla 3
    FuzzyRule* LDregla3 = new FuzzyRule(3, ifNandSTMF, thenACCF2);
    fuzzy->addFuzzyRule(LDregla3);

```

```

//Regla 4
FuzzyRule* LDregla4 = new FuzzyRule(4, ifCandSTMF, thenACCF3);
fuzzy->addFuzzyRule(LDregla4);
//Regla 5
FuzzyRule* LDregla5 = new FuzzyRule(5, ifMCandSTMF, thenACCF4);
fuzzy->addFuzzyRule(LDregla5);
//Regla 6
FuzzyRule* LDregla6 = new FuzzyRule(6, ifMFandSTF, thenCF1);
fuzzy->addFuzzyRule(LDregla6);
//Regla 7
FuzzyRule* LDregla7 = new FuzzyRule(7, ifFandSTF, thenEST);
fuzzy->addFuzzyRule(LDregla7);
//Regla 8
FuzzyRule* LDregla8 = new FuzzyRule(8, ifNandSTF, thenACCF1);
fuzzy->addFuzzyRule(LDregla8);
//Regla 9
FuzzyRule* LDregla9 = new FuzzyRule(9, ifCandSTF, thenACCF2);
fuzzy->addFuzzyRule(LDregla9);
//Regla 10
FuzzyRule* LDregla10 = new FuzzyRule(10, ifMCandSTF, thenACCF3);
fuzzy->addFuzzyRule(LDregla10);
//Regla 11
FuzzyRule* LDregla11 = new FuzzyRule(11, ifMFandSTN, thenCF2);
fuzzy->addFuzzyRule(LDregla11);
//Regla 12
FuzzyRule* LDregla12 = new FuzzyRule(12, ifFandSTN, thenCF1);
fuzzy->addFuzzyRule(LDregla12);
//Regla 13
FuzzyRule* LDregla13 = new FuzzyRule(13, ifNandSTN, thenEST);
fuzzy->addFuzzyRule(LDregla13);
//Regla 14
FuzzyRule* LDregla14 = new FuzzyRule(14, ifCandSTN, thenACCF1);
fuzzy->addFuzzyRule(LDregla14);
//Regla 15
FuzzyRule* LDregla15 = new FuzzyRule(15, ifMCandSTN, thenACCF2);
fuzzy->addFuzzyRule(LDregla15);
//Regla 16
FuzzyRule* LDregla16 = new FuzzyRule(16, ifMFandSTC, thenCF3);
fuzzy->addFuzzyRule(LDregla16);
//Regla 17

```

```

FuzzyRule* LDregla17 = new FuzzyRule(17, ifFandSTC, thenCF2);
fuzzy->addFuzzyRule(LDregla17);
//Regla 18
FuzzyRule* LDregla18 = new FuzzyRule(18, ifNandSTC, thenCF1);
fuzzy->addFuzzyRule(LDregla18);
//Regla 19
FuzzyRule* LDregla19 = new FuzzyRule(19, ifCandSTC, thenEST);
fuzzy->addFuzzyRule(LDregla19);
//Regla 20
FuzzyRule* LDregla20 = new FuzzyRule(20, ifMCandSTC, thenACCF1);
fuzzy->addFuzzyRule(LDregla20);
//Regla 21
FuzzyRule* LDregla21 = new FuzzyRule(21, ifMFandSTMC, thenCF4);
fuzzy->addFuzzyRule(LDregla21);
//Regla 22
FuzzyRule* LDregla22 = new FuzzyRule(22, ifFandSTMC, thenCF3);
fuzzy->addFuzzyRule(LDregla22);
//Regla 23
FuzzyRule* LDregla23 = new FuzzyRule(23, ifNandSTMC, thenCF2);
fuzzy->addFuzzyRule(LDregla23);
//Regla 24
FuzzyRule* LDregla24 = new FuzzyRule(24, ifCandSTMC, thenCF1);
fuzzy->addFuzzyRule(LDregla24);
//Regla 25
FuzzyRule* LDregla25 = new FuzzyRule(25, ifMCandSTMC, thenEST);
fuzzy->addFuzzyRule(LDregla25);

//Serial.begin(9600);

lcd.init();
lcd.backlight();
lcd.print("---- FITOTRON ----");
lcd.setCursor(0,1);
lcd.print("Esc. Ing. Electronica");
lcd.setCursor(0,2);
lcd.print(" Richard Reyna Salas");
lcd.setCursor(0,3);
lcd.print(" Raul Puente Kamasca");
delay(5000);
lcd.clear();

```

```

/*

// - PINES DE SALIDAS DIGITALES
// Salidas Digitales
const uint8_t R_Luces=22;
const uint8_t R_Peltier=23;
const uint8_t R_Calefactor=3;
// Salidas moduladas - PWM
const uint8_t pwm_Luces=2;
const uint8_t pwm_Ext_Sup=6;
const uint8_t pwm_Ext_Inf=8;
const uint8_t pwm_VentChico_cab=5;
const uint8_t pwm_VentGrande_cab=4;
const uint8_t pwm_BombasAgua=9;
const uint8_t pwm_Rad_I=10;
const uint8_t pwm_Rad_II=11;

*/

// - - - INICIALIZACION DE SALIDAS - - - -
pinMode(R_Luces,OUTPUT);
pinMode(R_Peltier,OUTPUT);
pinMode(R_Calefactor,OUTPUT);

pinMode(pwm_Luces,OUTPUT);
pinMode(pwm_Ext_Sup,OUTPUT);
pinMode(pwm_Ext_Inf,OUTPUT);
pinMode(pwm_VentChico_cab,OUTPUT);
pinMode(pwm_VentGrande_cab,OUTPUT);
pinMode(pwm_BombasAgua,OUTPUT);
pinMode(pwm_Rad_I,OUTPUT);
pinMode(pwm_Rad_II,OUTPUT);

digitalWrite(R_Luces,HIGH); // SE ACTIVA O DESACTIVA LA
FUENTE DE ALIMENTACION DE LAS LUCES USANDO EL RELAY
digitalWrite(R_Peltier,LOW); // SE ACTIVA O DESACTIVA LA FUENTE DE
ALIMENTACION DEL PELTIER USANDO EL RELAY
digitalWrite(R_Calefactor,LOW); // SE ACTIVA O DESACTIVA EL CALEFACTOR
USANDO EL RELAY

analogWrite(pwm_Luces,255);

```

```

    analogWrite(pwm_Ext_Sup,0);
    analogWrite(pwm_Ext_Inf,0);
    analogWrite(pwm_VentChico_cab,128);
    analogWrite(pwm_VentGrande_cab,0);
    analogWrite(pwm_BombasAgua,0);
    analogWrite(pwm_Rad_I,0);
    analogWrite(pwm_Rad_II,0);

}

void loop()
{
    //

    int stpCab=22; // TEMPERATURA DE ESTABILIZACION (SET POINT)

    sensorValue1 = analogRead(A0); // sensor de temp superior
    sensorValue2 = analogRead(A1); // sensor de temp medio
    sensorValue3 = analogRead(A2); // sensor de temperatura inferior
    sensorValue4 = analogRead(A4); // sensor de humedad
    sensorValue5 = analogRead(A3); // sensor de Intensidad lumínica en lux

    ValorPromTc = (sensorValue1+sensorValue2+sensorValue3)/3.0; //
    Promedio de valores digitales de sensores de temperatura
    TcRes=((ValorPromTc*(5000/1024.0))-500)/10.0; // Conversión de digital
a grados centigrados
    RH=((sensorValue4*156.76)/1024.0)-24.61; // Conversión de digital a
Humedad relativa
    RHcomp=(1000.0*RH)/(1054.6-2.16*TcRes); // Humedad relativa compensado
con temperatura
    IntLumLUX = sensorValue5*0.9765625; //

    //

    //int tempCab=; // VALOR ACTUAL MEDIDO EN LA CABINA

    fuzzy->setInput(1, TcRes); //Ingreso de valor de Temperatura de Cabina
    fuzzy->setInput(2, stpCab); //Ingreso de Valor de SetPoint de Temperatura
    fuzzy->fuzzify();

```



```

/*

Serial.println("Temp_Cabina = "+String(TcRes,DEC)+"°C");
Serial.println("SetPoint Temp_Cabina = "+String(stpCab,DEC)+"°C");

// Serial.print("T = " );
//Serial.print(TcRes);
Serial.print("C / RH=");
Serial.print(RH);
Serial.print("% / RH-Compensado=");
Serial.print(RHcomp);
Serial.println("% ");
Serial.print("Xd_T=");
Serial.print(ValorPromTc);
Serial.print(" / Xd_RH=");
Serial.println(sensorValue4);
Serial.print("    LUX =");
Serial.println(IntLumLUX);
*/

float outPWM = fuzzy->defuzzify(1);

// Serial.print("Salida de control Defuzificada: ");
// Serial.println(outPWM);

// ENVIO DE DATOS AL LCD POR COMUNICACION I2C
  lcd.setCursor(0,0);
  lcd.print("Tc=");
  lcd.print(TcRes,1);
  lcd.print("°C");
  lcd.setCursor(10,0);
  lcd.print("Ts=");
  lcd.print(stpCab,1);
  lcd.print("°C");
  //Segunda fila
  lcd.setCursor(0,1);
  lcd.print("RH%=");
  lcd.print(RH,1);
  lcd.print("%");

```

```

lcd.setCursor(10,1);
lcd.print("RH_C=");
lcd.print(RHcomp,1);
lcd.print("%");
//Tercera fila
lcd.setCursor(0,2);
lcd.print("Int_Lum=");
lcd.print(IntLumLUX,1);
lcd.print("LUX");

lcd.setCursor(0,3);
lcd.print("Out_PWM=");
lcd.print(outPWM,1);

// ACTIVACION DEL CALEFACTOR
if(outPWM > 260)
{
    digitalWrite(R_Calefactor,HIGH);
    analogWrite(pwm_VentChico_cab, map(outPWM, 260, 511, 128, 255));
    analogWrite(pwm_VentGrande_cab, map(outPWM, 260, 511, 128, 255));
}else
{
    digitalWrite(R_Calefactor,LOW);
    analogWrite(pwm_VentChico_cab, 128);
    analogWrite(pwm_VentGrande_cab, 0);
}

// ACTIVACION DEL AIRE ACONDICIONADO - PELTIER
if(outPWM < 260)
{

    analogWrite(pwm_Rad_I,255);
    analogWrite(pwm_Rad_II,255);
    analogWrite(pwm_BombasAgua, 255);
    if(ON_peltier==0 and OFF_peltier==0)
    {
        delay(30000);
        OFF_peltier=1;
    }
}

```

```
        digitalWrite(R_Peltier,HIGH);
        ON_peltier=1;

    }else
    {
        digitalWrite(R_Peltier,LOW);
        if(ON_peltier==1 and OFF_peltier==1)
        {
            delay(30000);
            delay(30000);
            ON_peltier=0;
        }
        analogWrite(pwm_Rad_I,0);
        analogWrite(pwm_Rad_II,0);
        analogWrite(pwm_BombasAgua,0);
        OFF_peltier=0;
    }

}
```