



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO**

ESCOM

Trabajo Terminal

***“Sistema para el Monitoreo y
Control de una Hortaliza – DG2”***

2014-B027

Presenta

Escamilla Martínez David

Director

M. en C. Cervantes de Anda Ismael



Diciembre 2015



**ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA**

**DEPARTAMENTO DE FORMACIÓN INTEGRAL E
INSTITUCIONAL**

COMISIÓN ACADÉMICA DE TRABAJO TERMINAL



México, D.F. a 16 de Diciembre de 2015.

**DR. FLAVIO ARTURO SÁNCHEZ GARFIAS
PRESIDENTE DE LA COMISIÓN ACADÉMICA
DE TRABAJO TERMINAL
P R E S E N T E**

Por medio del presente, se informa que el alumno que integra el **TRABAJO TERMINAL:**
2014-B027, titulado "Sistema para el Monitoreo y Control de una Hortaliza – DG2" concluyo satisfactoriamente su trabajo.

Los discos (DVDs) fueron revisados ampliamente por su servidor y corregidos, cubriendo el alcance y el objetivo planteados en el protocolo original y de acuerdo a los requisitos establecidos por la Comisión que Usted preside.

ATENTAMENTE


**NOMBRE Y FIRMA DEL DIRECTOR
DEL TRABAJO TERMINAL**

Advertencia

“Este documento contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional, a partir de datos y documentos con derecho de propiedad y por lo tanto, su uso quedará restringido a las aplicaciones que explícitamente se convengan.”

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n Teléfono: 57296000, extensión 52000.

Agradecimientos

Quiero agradecer a mi madre y hermanas por todo el apoyo que siempre me brindaron, por siempre alentarme a seguir adelante y continuar con la carrera, por creer y por hacerme creer que podía llegar hasta este punto; nadie me dijo que mi vida estudiantil duraría tanto, pero ellas sin embargo, han estado desde que comenzo.

A mi director el Profesor Ismael Cervantes, por todo el apoyo que ofreció durante la realización de este Trabajo Terminal, al profesor Alejandro Soto, por ser un director más que un sinodal.

A todos mis compañeros, por estar conmigo desde el principio de mi estancia en ESCOM, a Ivonne por siempre ofrecer una sonrisa durante cualquier situación, a Daniel, porque sin él nada de esto hubiera sido posible, a Tavo porque se va a sentir mal si no lo agrego en los agradecimientos a todos ustedes, gracias amigos y a Lour porque aunque quizás ella no estuvo desde el principio de este ciclo, llego al cierre, justo para iniciar uno nuevo.



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



10 de Diciembre del 2015

No de Registro del Trabajo Terminal 2014-B027

Reporte Técnico

“Sistema para el Monitoreo y Control de una Hortaliza–DG2”

Presenta

Escamilla Martínez David

Director

Cervantes de Anda Ismael

Resumen

El sistema desarrollado permite el control y la automatización de las variables físicas necesarias para el desarrollo de una hortaliza casera aplicando conceptos de domótica como el ahorro de energía y que así dicha hortaliza sea capaz de proveer productos para consumo. Con una interfaz de usuario para supervisar los sensores y el control de los actuadores.

Índice

Capítulo I	13
1.1 Introducción.....	13
1.2 Problemática.....	14
1.3 Justificación.....	14
1.4 Estado del Arte.....	14
Capítulo II	16
2.1 Metodología.....	16
2.2 Modelo.....	16
2.3 Partes del Sistema.....	17
2.4 Requerimientos del Sistema.....	18
3.1 Variables a considerar en el control climático de una hortaliza.....	20
3.1.1 Temperatura.....	20
3.1.2 Humedad Relativa.....	20
3.1.3 Luminosidad.....	21
3.1.4 CO2.....	21
3.1.5 Tabla de características de diferentes Hortalizas.....	22
3.2 Sensores de Humedad.....	23
3.2.1 Sensor HIH-4000.....	23
3.2.2 Sonda YL-69 y Módulo YL-38.....	23
3.2.3 Sensor DHT22.....	23
3.3 Sensores de Temperatura.....	24
3.3.1 Sensor DS600/ (U) (U+) (U/T&R) (U+/T&R).....	24
3.3.2 Sensor STH7x (71, 75).....	24
3.3.3 Sensor LM35HA.....	24
3.3.3 La elección de los sensores.....	25
3.4 Actuadores.....	25
3.4.1 Relevador.....	26
3.4.2 Electro Válvulas.....	26
3.4.3 Ventiladores.....	26
3.5 Lenguajes de programación para el módulo de control y el servidor.....	27
3.5.1 Lenguaje Processing/Wiring.....	27
3.5.2 Lenguaje ensamblador.....	28

3.5.3	Lenguaje Javascript/Node.js.	28
3.5.4	La elección del lenguaje para el modulo.....	29
3.6	Sistema Gestor de Base de Datos.	29
3.6.1	MySQL.....	30
3.6.2	MongoDB.....	30
3.6.3	La elección del Sistema de Base de Datos.	31
3.7	Tarjeta de Desarrollo.	31
3.7.1	Arduino Mega 2560.....	31
3.7.2	Raspberry Pi B+.	32
3.7.3	Tessel 2.....	33
3.7.4	La elección de la tarjeta de desarrollo.....	34
3.8	Costos del sistema.	34
3.8.1	Coste de la implementación del prototipo.....	34
3.8.2	Costo del desarrollo del servidor e interfaz de usuario.	36
Capítulo IV	38
4.1	Diagrama a bloques general del sistema.	38
4.2	Diagrama a bloques del Servidor/Módulo de control.	39
4.3	Diagramas de Casos de Uso.	41
4.3.1	Diagramas de casos de uso para el actor Usuario.	41
4.3.2	Diagrama de casos de uso para el usuario Módulo de control.	42
4.3.3	Especificación de Casos de Uso.	43
4.4	Diagramas de Secuencia.....	48
4.4.1	Diagrama de Secuencia Visualizar Valor Muestreado.	48
4.4.2	Diagrama de Secuencia Modificar Estado de Actuador.	49
4.4.3	Diagrama de Secuencia Configurar Sensor.....	49
4.4.4	Diagrama de Secuencia Módulo de control.	50
4.5	Diagramas de Estados.....	51
4.5.1	Visualizar Variable de Temperatura.	51
4.5.2	Modificar Actuador de Temperatura.....	51
4.5.3	Obtener Estado de Actuador.	52
4.5.4	Visualizar Variable de Humedad.	52
4.5.5	Modificar Estado de Actuador de Humedad.	53
4.5.6	Modificar Sensor.	54
4.5.7	Seleccionar Hortaliza.....	54

4.5.8	Evaluar Variable de Temperatura.	55
4.5.9	Almacenar Datos Muestreados.	55
4.5.10	Evaluar Variable de Humedad.	56
4.6	Esquema de la base de datos.	57
4.7	Diseño preliminar de la interfaz.	58
4.7.1	Diseño de la pantalla inicial.	58
4.7.2	Diseño de la pantalla de Temperatura de Hortaliza.	58
4.7.3	Diseño de la pantalla de Humedad de la Hortaliza.	59
4.7.4	Diseño de la pantalla Editar/Añadir Hortaliza.	60
4.8	Pruebas del sistema.	61
4.8.1	Pruebas Sensor - Módulo de control.	61
4.8.2	Módulo de control – Actuadores.	66
Capítulo V	67
5.1	Implementación de la Base de Datos.	67
5.1.1	Modelos de Colecciones para la Base de Datos.	70
5.1.2	Colección Hortalizas.	70
5.1.3	Colección Actuator.	71
5.1.4	Colección Sensor.	71
5.1.5	Colección Value.	72
5.1.6	Colección Activation.	72
5.2	Implementación del módulo de control.	73
5.2.1	Prototipo 1	73
5.2.2	Prototipo 2.	75
5.3	Implementación de la Interfaz de Usuario.	77
5.3.1	Prototipo 1.	77
5.3.2	Prototipo 2.	80
6.1	Trabajo a futuro.	81
6.2	Conclusiones.	81
Anexo	82
Bibliografía	84

Índice de tablas.

Tabla 1: Características de sistemas similares.	15
Tabla 2: Requerimientos Básicos.	18
Tabla 3: Requerimientos Funcionales.	18
Tabla 4: Requerimientos No Funcionales.	18
Tabla 5: Características de las Hortalizas.	22
Tabla 6: Comparación entre los Sensores de Humedad.	25
Tabla 7: Comparación de los Sensores de Temperatura.	25
Tabla 8: Comparativa entre los Lenguajes de Programación.	29
Tabla 9: Comparación entre los Gestores de Base de Datos.	31
Tabla 10: Comparativa de tarjetas Arduino de gama media.	32
Tabla 11: Comparativa entre Tarjetas de desarrollo.	34
Tabla 12: Costes de diseño del prototipo.	35
Tabla 13: Costos del material para el Módulo de Control.	35
Tabla 14: Costo de la Implementación del Prototipo.	36
Tabla 15: Estimación de Líneas de Código.	36
Tabla 16: Criterios de salarios.	37
Tabla 17: Coste total del sistema.	37
Tabla 18: Caso de uso de ejemplo: descripción.	41
Tabla 19: Caso de uso CU-01. Visualizar variable Temperatura.	43
Tabla 20: Caso de uso CU-02. Modificar el estado del actuador de Temperatura.	43
Tabla 21: Caso de uso CU-03. Obtener el estado del Actuador.	44
Tabla 22: Caso de uso CU-04. Visualizar variable Humedad.	44
Tabla 23: Caso de uso CU-05. Modificar el estado del actuador de Humedad.	44
Tabla 24: Caso de uso CU-06. Configurar Sensor.	45
Tabla 25: Caso de uso CU-07. Seleccionar Hortaliza.	45
Tabla 26: Caso de uso CU-08. Obtener lectura de Temperatura.	45
Tabla 27: Caso de uso CU-09. Evaluar variable de Temperatura.	46
Tabla 28: Caso de uso CU-10. Almacenar datos muestreados.	46
Tabla 29: Caso de Uso CU-11. Obtener lectura de Humedad.	46
Tabla 30: Caso de uso CU-12. Evaluar variable de Humedad.	47
Tabla 31: Pruebas de muestreo del sensor LM35.	62
Tabla 32: Serie de pruebas sensor de humedad.	64
Tabla 33: Pruebas de tiempo de los Relés.	66

Índice de ilustraciones.

Figura 1: El modelo en espiral.....	17
Figura 2: Diagrama General del Sistema.....	19
Figura 3: Diagrama a bloques del sistema.....	38
Figura 4: Diagrama a bloques del Servidor.....	39
Figura 5: Diagrama de Casos de uso para el Usuario.....	41
Figura 6: Diagrama de Casos de uso para el Módulo de Control.....	42
Figura 7: Diagrama de Secuencia Visualizar Valor Muestreado.....	48
Figura 8: Diagrama de Secuencia Modificar Estado de Actuador.....	49
Figura 9: Diagrama de Secuencia Configurar Sensor.....	49
Figura 10: Diagrama de Secuencia para el Módulo de control.....	50
Figura 11: Diagrama de Estados Visualizar Variable de Temperatura.....	51
Figura 12: Diagrama de Estados Modificar Actuador de Temperatura.....	52
Figura 13: Diagrama de Estados Obtener Estado de Actuador.....	52
Figura 14: Diagrama de Estados Visualizar Variable de Humedad.....	53
Figura 15: Diagrama de Estados Modificar Actuador de Humedad.....	53
Figura 16: Diagrama de Estados Modificar Sensor.....	54
Figura 17: Diagrama de Estados Seleccionar Hortaliza.....	54
Figura 18: Diagrama de Estados Evaluar Variable de Temperatura.....	55
Figura 19: Diagrama de Estados Almacenar Datos Muestreados.....	56
Figura 20: Diagrama de Estados Evaluar Variable de Humedad.....	56
Figura 21: Esquema de la base de datos.....	57
Figura 22: Pantalla Inicial.....	58
Figura 23: Pantalla Temperatura de Hortaliza.....	59
Figura 24: Pantalla Humedad de la Hortaliza.....	59
Figura 25: Pantalla Editar/Añadir Hortaliza.....	60
Figura 26: Grafica de pruebas del sensor de temperatura.....	62
Figura 27: Valores muestreados a temperatura ambiente.....	63
Figura 28: Valores muestreados a fuego directo.....	63
Figura 29: Grafica de pruebas del sensor de humedad.....	65
Figura 30: Valores muestreados de Humedad Relativa.....	65
Figura 31: Estructura de un objeto JSON.....	67
Figura 32: Tipos de valores admitidos.....	68
Figura 33: Definición de un esquema en Mongoose.....	68
Figura 34: Colecciones para la Base de Datos.....	70
Figura 35: Esquema Hortaliza.....	70
Figura 36: Esquema Actuador.....	71
Figura 37: Esquema Sensor.....	71
Figura 38: Esquema Value.....	72
Figura 39: Esquema Activation.....	72
Figura 40: Prototipo 1 Simulado.....	73
Figura 41: Registro de Temperatura.....	73
Figura 42: Simulación de Actuador.....	74
Figura 43: Prototipo 1 - Simulación de Actuador.....	74

Figura 44: Prototipo 2	75
Figura 45: Amplificador Operacional - No Inversor.	75
Figura 46: Circuito de acondicionamiento del sensor LM35.	76
Figura 47: Prototipo 2 - Activación de Actuador.....	76
Figura 48: Prototipo 2 - Actuador Activado.	76
Figura 49: Prototipo 1 - Pantalla principal.....	77
Figura 50: Prototipo 1 - Detalles de hortaliza.	78
Figura 51: Prototipo 1 - Grafica en tiempo real.....	79
Figura 52: Prototipo 2 - Pantalla Principal	80
Figura 53: Prototipo 2 - Detalles de Hortaliza.	80

Objetivo

Objetivo general:

Realizar un prototipo de sistema autónomo para el control y monitoreo de una hortaliza casera mediante el uso de un servidor web escalable capaz de controlar actuadores y sensores, para así proporcionar condiciones óptimas de crecimiento y que esta pueda proveer productos para consumo.

Objetivos específicos:

- Realizar muestreo de variables físicas mediante sensores para su posterior procesamiento, realizar una evaluación y mediante esta, emitir una respuesta hacia los actuadores.
- Optimización de recursos como tiempo, agua y energía eléctrica mediante el control automatizado de los factores ambientales necesarios para el desarrollo de la hortaliza.
- Realizar el almacenamiento de datos muestreados para una posterior consulta sobre las actividades realizadas por el sistema.
- Implementar una interfaz de usuario final la cual permita el monitoreo y configuración de los diferentes actuadores y sensores conectados al sistema.

Capítulo I

1.1 Introducción.

Los huertos caseros tienen una importancia propiamente, porque gracias a esta práctica encontramos plantas de todo tipo, ya sean medicinales, comestibles, ornamentales, condimentales, frutales, etc.

La mayoría de los ejemplos de huertos caseros de finales del siglo XIX y la primera mitad del siglo XX que se reportan están relacionados con períodos de escasez de alimentos, durante las dos Guerras Mundiales, el programa *Liberty and Victory Gardens* estimuló la construcción de huertos familiares y comunitarios. Se consideró que 20 millones de norteamericanos respondieron a la llamada y como resultado, en 1943 el 40 por ciento de las frutas y vegetales frescos consumidos en Estados Unidos provenían de huertos familiares, escolares o comunitarios [1].

Para muchas familias hoy en día es importante tener huertos completos en sus patios o traspatios, de esta manera se puede ahorrar en la compra de productos comestibles en el mercado. En cambio muchas de las familias vivimos diariamente utilizando alimentos provenientes de otros lugares, lo que más adoptamos es tener jardines hermosos, y no plantas para nuestro propio beneficio.

Grandes empresas en la rama de la tecnología como Panasonic y Sharp están incursionando en este medio, implementando “huertos inteligentes” en las inmediaciones de sus instalaciones o implementado cuartos especializados llamados “Huertos 2.0”.

Dichos recintos están dotados con una gama específica de lámparas LED y tienen instalado un sistema de purificación de aire de alta tecnología el cual está destinado a la eliminación de gérmenes y bacterias; además el centro cuenta con un complejo sistema de monitorización donde se controla las variables de cara a recrear las condiciones exactas para el crecimiento de los cultivos. Estas instalaciones han tenido un coste cercano a los 100,000 dólares [2].

1.2 Problemática.

Está previsto que la población mundial alcanzará los 9.2 mil millones de personas para el 2050 [3]. En el año 2012, La Organización Mundial de Alimentos y Agricultura, FAO por sus siglas en inglés (*The Food and Agricultural Organization*) estimó que una de cada ocho personas dormirían con hambre, esta situación sólo se agravara las condiciones actuales.

Factores como el aumento que ha tenido el costo de producción de los alimentos, los cambios climáticos y las técnicas poco sustentables de producción hacen a las empresas plantear soluciones ante este problema, pero estas soluciones solo se encuentran encaminadas hacia la industrialización y la producción en masa con precios muy elevados.

1.3 Justificación.

El prototipo funcional en este Trabajo Terminal propone llevar las acciones que están tomando las grandes empresas con respecto a los “Huertos 2.0” a un nivel doméstico para poder llevar el control del entorno de una hortaliza casera de manera automática y que esta pueda ser una fuente sustentable de alimento de la siguiente manera:

- Elaborando un sistema que mediante el uso de sensores pueda monitorear variables del entorno de la hortaliza para poder proveer de manera autónoma condiciones óptimas de crecimiento mediante actuadores, consiguiendo una mejor gestión de recursos como el tiempo, el suministro de agua, entre otros; frente a un cuidado y control del entorno de manera manual, el cual requiere una mayor atención por parte del usuario, más tiempo para el monitoreo de las variables físicas de manera personalizada, las cuales pueden sufrir cambios repentinos que no pueden ser controlados y por lo tanto pueden ser pasados por alto afectando directamente al cuidado de las hortalizas.

1.4 Estado del Arte.

Existen empresas encargadas al control y automatización de huertos y/o invernaderos las cuales controlan variables como la humedad, la radiación solar, la temperatura, el PH, la cantidad de agua suministrada, etc., algunos ejemplos en México son la empresa HYDROCULTURA o la empresa NOVEDADES AGRICOLAS S.A. DE C.V., ambas encargadas de acondicionar grandes complejos para el control y monitoreo de variables ambientales, pero al enfocarse en grandes locaciones sus precios son muy elevados.

A principios del año 2014 la empresa francesa PARROT lanzó al mercado el “Flower Power”, el cual es un dispositivo que puede monitorear, la luminosidad, humedad, temperatura y cantidad de fertilizante que hay en una hortaliza o planta ornamental, sin embargo no dispone de ningún módulo de control para poder manipular dichas variables volviéndolo únicamente un dispositivo sensor.

Empresa/Dispositivo	Características	Precio en el Mercado
HYDROCULTURA	<ul style="list-style-type: none"> • Ofrece el monitoreo de temperatura, PH, humedad, radiación solar. • Cursos de inducción a la jardinería. • Control de riego y fertilizante. 	Según sean las características y el tamaño del recinto oscila entre \$5,000.00 hasta los \$15,000.00 Pesos M.N.
NOVEDADES AGRICOLAS S.A. de C.V.	<ul style="list-style-type: none"> • Permite el monitoreo de temperatura, humedad, luminosidad y control de plagas mediante tecnología inalámbrica. • Control de riego y fertilizante. 	Desde \$7,000.00 Pesos M.N.
PARROT- Flower Power [4]	<ul style="list-style-type: none"> • Permite el monitoreo de temperatura, humedad, luminosidad y cantidad de fertilizante. • Aplicación para dispositivos móviles con sistema operativo iOS y Android. 	\$1,050.00 Pesos M.N.
Solución Propuesta	<ul style="list-style-type: none"> • Permite el censado de variables como temperatura, humedad y luminosidad. • Control sobre actuadores como el riego. • Interfaz de usuario final. 	\$ 94,226.00 Pesos M.N

Tabla 1: Características de sistemas similares.

Capítulo II

2.1 Metodología.

Para el desarrollo del sistema como metodología se usará Prototipado Evolutivo.

Es frecuente que los requerimientos de un sistema cambien conforme avanza su desarrollo, lo que hace que trazar una trayectoria lineal hacia el producto final sea poco realista. Por ello cuando se comprende bien el producto básico se lanza una versión limitada que ayude a resaltar detalles y/o extensiones que aún están por definirse o no estaban del todo contempladas. En estas situaciones es necesaria una metodología que se adapte a un producto que evoluciona con el tiempo.

Con respecto al Trabajo Terminal, este se compone tanto de partes de hardware y de software lo cual por lo cual al elegir una metodología iterativa se permitirá el desarrollo de prototipos cada vez más complejos y detallados del sistema que con el tiempo y la refinación formarán parte final del sistema.

2.2 Modelo.

El modelo a utilizar en el sistema será El Modelo Espiral [5], es un modelo evolutivo que se acopla con la naturaleza iterativa de hacer prototipos, posee los aspectos controlados y sistémicos del modelo Cascada y además otorga el potencial de un desarrollo rápido de versiones cada vez más complejas.

Características principales:

- Como el sistema evoluciona a medida que el proceso avanza se pueden definir detalles con cada evolución.
- Se puede comprender y reaccionar de manera más estratégica a los riesgos en cada evolución.
- Mantiene el enfoque de escalón sistemático sugerido por el ciclo de vida clásico pero le otorga un enfoque iterativo haciendo el proceso de vida más realista.

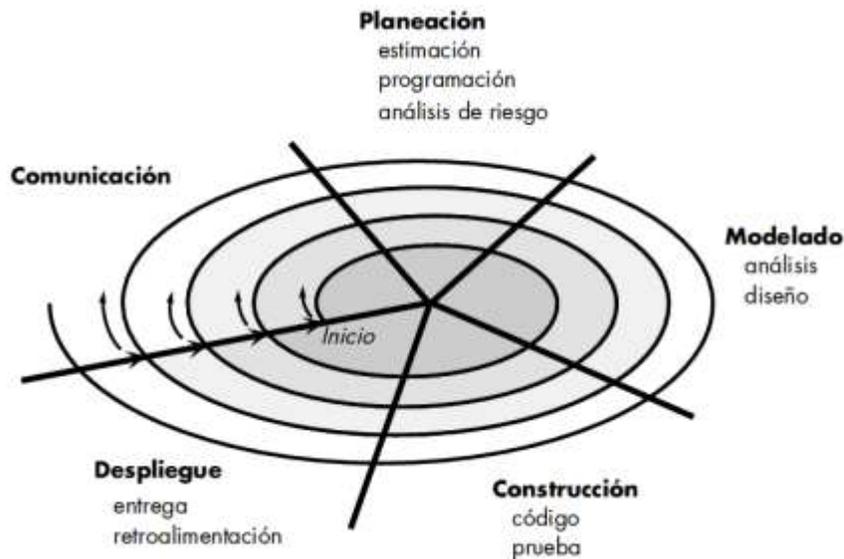


Figura 1: El modelo en espiral.

2.3 Partes del Sistema.

El sistema está planeado para constar de las siguientes partes:

Servidor y Módulo de control:

Esta parte será la encargada del control de los actuadores en base a los datos que serán recopilados por los sensores. Además de proveer la funcionalidad de un servidor web asíncrono realizando las conexiones con la aplicación del lado del cliente que será la interfaz de usuario.

Módulo de Sensado.

Esta será la parte que contendrá los sensores los cuales recopilarán datos y los enviarán al Módulo de control.

- Sensor de temperatura.
- Sensor de humedad.
- Sensor de luminosidad

Actuador.

Esta parte será la encargada de controlar los sistemas como el de riego.

Interfaz.

Por medio de ella el usuario visualizará los datos recopilados y la frecuencia de riego, además de poder controlar la frecuencia en la que se realizan las diferentes actividades del sistema.

2.4 Requerimientos del Sistema.

Requerimientos Básicos.

Los requerimientos básicos del sistema se describen en la siguiente tabla:

Requerimiento	Descripción
Comunicación	Comunicación entre los sensores, actuadores y el módulo de control al servidor.
	Comunicación entre el Servidor y la Interfaz

Tabla 2: Requerimientos Básicos.

Requerimientos Funcionales.

Requerimiento	Descripción
RF 1	Comunicación de los sensores con el módulo central.
RF 2	Muestreo de variables físicas.
RF 3	Almacenamiento de datos.
RF 4	Evaluación de los datos muestreados.
RF 5	Control sobre los actuadores.
RF 6	Comunicación entre el servidor y la interfaz de usuario.

Tabla 3: Requerimientos Funcionales.

Requerimientos No Funcionales.

Requerimiento	Descripción
RNF 1	Interfaz Grafica
RNF 2	Portabilidad.
RNF 3	Control manual de los actuadores.
RNF 4	Despliegue de datos almacenados.

Tabla 4: Requerimientos No Funcionales.

Capítulo III

Un Sistema Autónomo es un sistema capaz de operar de forma automática por periodos prolongados sin la intervención humana, evitando, reaccionando o adaptándose a situaciones imprevistas y cambiantes del entorno.

Para llevar a cabo esto y ser considerado un sistema autónomo el sistema hará uso de un conjunto de sensores que serán los encargados de muestrear valores de las variables físicas del entorno, dichos valores serán procesados y el sistema actuará de manera acorde mediante un conjunto de actuadores que manipularán dichas variables, todo esto sin la intervención del usuario. Una tarjeta de desarrollo será la encargada de realizar dichas funciones mediante un servidor web capaz de realizar dicho monitoreo y control y que además se encargara de ofrecer conexiones con la interfaz de usuario.

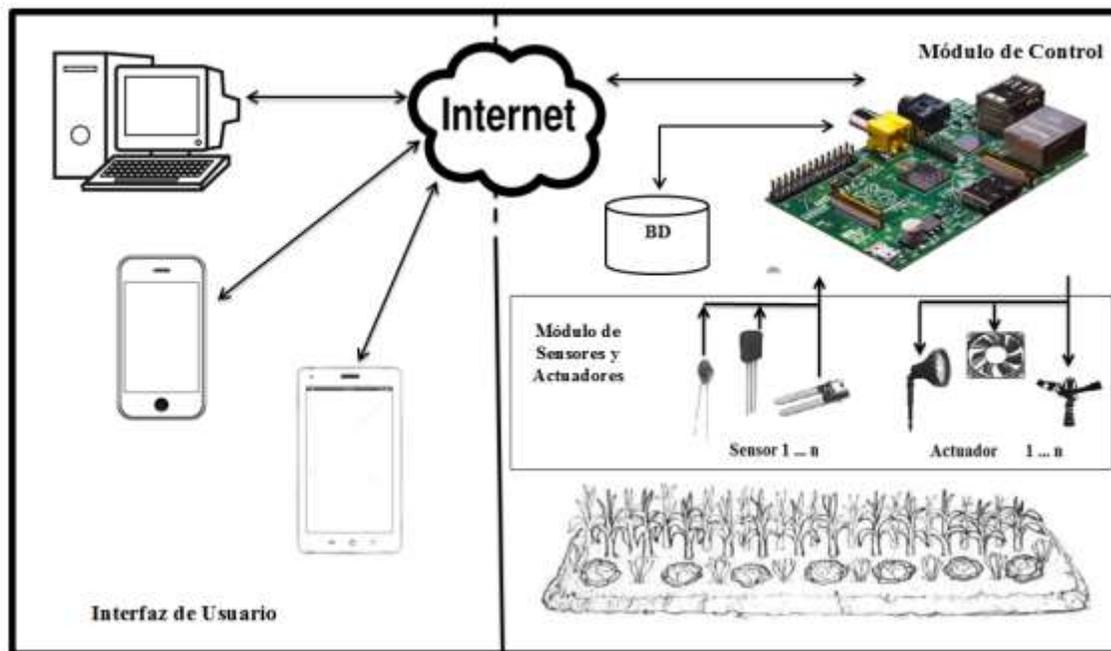


Figura 2: Diagrama General del Sistema.

Un sensor es un dispositivo capaz de transformar magnitudes físicas, llamadas variables de instrumentación, en magnitudes eléctricas. Un sensor está siempre en contacto con la variable a medir o controlar.

La señal que nos entrega un sensor no solo sirve para medir la variable, sino también para tener una relación directa con los cambios de la variable censada dentro de un rango y para fines de control sobre dicha variable en un proceso.

3.1 Variables a considerar en el control climático de una hortaliza.

El desarrollo de un cultivo, está condicionado por cuatro factores ambientales o climáticos: temperatura, humedad, luz y CO₂. Para que las plantas puedan realizar sus funciones es necesaria la conjunción de estos factores dentro de unos límites máximos y mínimos, fuera de estos límites las plantas cesan su metabolismo y pueden llegar a morir. [6]

3.1.1 Temperatura.

Este parámetro es uno de los más importantes ya que influye directamente en el crecimiento y desarrollo de las plantas. Normalmente la temperatura óptima para las plantas se encuentra entre los 10 y 30° C.

Para el manejo de la temperatura es importante conocer las necesidades y limitaciones de la especie cultivada. Así mismo se deben aclarar los siguientes conceptos de temperaturas, que indican los valores objetivos a tener en cuenta para un buen crecimiento.

- Temperatura mínima y máxima letal: Por debajo y sobre estas temperaturas se producen daños en la planta.
- Temperatura máximas y mínimas biológicas: Indican los valores, por encima o por debajo respectivamente del cual, no es posible que la planta alcance una determinada fase vegetativa, como floración, fructificación, etc. [7]
- Temperatura óptima: Indican los valores aconsejados para un correcto desarrollo de la planta.

3.1.2 Humedad Relativa.

La humedad es la masa de agua en unidad de volumen. La humedad relativa es la cantidad de agua contenida en el aire o en la tierra. Al igual que la temperatura cada especie tiene una humedad idónea para poder vegetar en perfectas condiciones, esta humedad ronda entre el 50 y el 80%.

Cuando la humedad es excesiva las plantas reducen la transpiración y disminuyen su crecimiento, si por el contrario, es muy baja las plantas transpiran en exceso y se deshidratan.

3.1.3 Luminosidad.

Aunque los cultivos dejan de depender de ciertos factores externos aun es necesaria la presencia del sol para que estos tengan un crecimiento saludable. Una buena luminosidad promueve la fotosíntesis.

3.1.4 CO₂.

La concentración normal de CO₂ en la atmosfera es del 0.03%, Este índice aumenta a límites de 0.1-0.2%, cuando los demás factores de la producción vegetal son óptimos, de esta manera se aprovechan al máximo la actividad fotosintética de las plantas. Es por eso que los niveles óptimos dependen de la especie, la radiación solar, humedad y temperatura.

Las dos variables mínimas para el adecuado desarrollo de una hortaliza son la temperatura y la humedad según el manual de Agricultura Urbana “Huerto en casa” [8] ya que la concentración de CO₂ rara vez alcanza valores demasiado elevados y la luminosidad no es requerida a menos de que se trate de una instalación con cuidados más exhaustivos; en la **Tabla 5** se muestra diferentes hortalizas y sus valores mínimos y máximos tanto de temperatura cómo de humedad, además de otros factores.

3.1.5 Tabla de características de diferentes Hortalizas.

Hortaliza	Pimiento	Lechuga	Tomate	Papa	Espinacas	Pepino	Berenjena	Zanahoria
Temporada de cultivo.	Primavera	Todo el año	Finales de la primavera.	Primavera	Todo el año	Primavera	Invierno	Primavera, otoño e invierno.
Profundidad de las raíces.		10 cm.	40 cm.	7-8 cm.	2 cm.	5 cm.	25-30 cm	25 cm.
<i>Humedad.</i>	50-70%	60-80%	65-70%	85-90%	90%	70-80%	90%	80%
Riego.	Abundante, 4 o 5 veces por semana.	2 veces por semana, poco abundante.	Abundante, 3 veces por semana.	Muy abundante 6 o 7 veces por semana.	Poco abundante, 4 o 5 veces por semana.	Abundante, 4 veces por semana.	Abundante, 4 o 5 veces por semana.	Abundante.
Como Cultivar.	15 cm entre cada planta y 60 cm entre líneas.	30 cm entre cada planta y 20 cm entre líneas.	60 cm entre cada planta y 90 cm entre líneas.	30 cm entre cada planta y 75 cm entre líneas.	10 cm entre cada planta y 30 cm entre líneas.	12 cm entre cada planta y 15 cm entre líneas.	60 cm entre cada planta y 80 cm entre líneas.	8 cm entre cada planta y 30 entre líneas.
<i>Temperatura Mínima Letal.</i>	-1 °C	0-1 °C	0-2 °C	0-1 °C	-4 °C	-1 °C	0 °C	1 °C
<i>Temperatura mínima biológica.</i>	10-12 °C	13-15 °C	10-12 °C	10-12 °C	5-12 °C	10-12 °C	10-12 °C	10-12 °C
<i>Temperatura óptima.</i>	16-18 °C	18-21 °C	13-16 °C	15-18 °C	15-19 °C	18-18 °C	17-22 °C	15-20 °C
<i>Temperatura máxima biológica.</i>	23-27 °C	20-25 °C	21-27 °C	20-27 °C	20-26 °C	20-25 °C	22-27 °C	20-26 °C
<i>Temperatura máxima letal.</i>	33-38 °C	31-35 °C	33-35 °C	31-35 °C	30-35 °C	31-35 °C	43-53 °C	31-35 °C

Tabla 5: Características de las Hortalizas.

3.2 Sensores de Humedad.

Los sensores de humedad más usuales son:

- **Mecánicos:** Este tipo de sensores aprovechan los cambios de dimensiones que sufren ciertos tipos de materiales en presencia de la humedad.
- **Basados en sales higroscópicas:** Deducen el valor de la humedad en el ambiente a partir de una molécula cristalina que tiene mucha afinidad con la absorción de agua.
- **Por conductividad:** La presencia de agua en el ambiente permite que a través de unas rejillas de oro circule una corriente. Ya que el agua es buena conductora de corriente; según la medida de corriente se deduce el valor de la humedad.
- **Capacitivos:** Están basados en el cambio de la capacidad que sufre un condensador en presencia de humedad.
- **Infrarrojos:** Este tipo dispone de 2 fuentes infrarrojas que lo que hacen es absorber parte de la radiación que contiene el agua.
- **Resistivos:** Aplican un principio de conductividad de la tierra. Es decir, cuanta más cantidad de agua hay en la muestra, más alta es la conductividad de la tierra.

3.2.1 Sensor HHH-4000.

El HHH-4000 es un sensor de humedad relativa de salida analógica cuyo rango de medición va desde 0 a 100% con una precisión de $\pm 2\%$; su voltaje de operación es entre 4 y 5.8 volts.

3.2.2 Sonda YL-69 y Módulo YL-38.

La sonda YL-69 es un sensor de salida analógica y digital que tiene la capacidad de medir la humedad del suelo por medio de la resistencia que genera la humedad de la tierra al aplicar voltaje en sus terminales; por otro lado el módulo YL-38 es el circuito de control encargado de alimentar a la sonda YL-69, su voltaje de operación es entre 3.3 y 5 volts.

3.2.3 Sensor DHT22.

El DHT22 es un sensor de humedad relativa de salida digital con un rango de 0 a 100% con una precisión de máximo $\pm 5\%$; su voltaje de operación es de entre 3.3 y 6 volts, tiene un periodo de 2 segundos entre las lecturas.

3.3 Sensores de Temperatura.

Los tipos de sensores más comunes son:

- Termopares: Se crea cuando dos metales diferentes se juntan y el punto de contacto produce un pequeño voltaje como una función de la temperatura. Este voltaje termoeléctrico se usa para calcular la temperatura; en pequeños cambios este voltaje es aproximadamente lineal.
- Detectores de temperatura resistivos (RTD): Este tipo de sensores están basados en la variación de la resistencia de un conductor con la temperatura, al calentarse la resistencia aumenta, al enfriarse, esta disminuye. Pasar corriente a través de un RTD genera un voltaje; al medir este voltaje se puede determinar la resistencia y por lo tanto la temperatura.
- Termistor: Al igual que con los RTDs, pasar una corriente a través de un termistor genera voltaje en el termistor y de esta manera se puede determinar su temperatura. A diferencia de los RTDs, los termistores tienen más alta resistencia (2,000 a 10,000 Ω) y una sensibilidad mucho más alta ($\sim 200 \Omega/^\circ\text{C}$), permitiéndoles alcanzar más alta sensibilidad en un rango de temperatura limitado (hasta 300°C).

3.3.1 Sensor DS600/ (U) (U+) (U/T&R) (U+/T&R).

El DS600 es un sensor de temperatura de salida analógica con una precisión de ± 0.5 grados Centígrados; su voltaje de operación es entre 2.7 y 5.5 volts y su rango de medición de temperatura es de -20°C a 100°C .

Para el resto de sus versiones (U) (U+) (U/T&R) (U+/T&R) el rango de temperatura es de -40°C a 125°C .

3.3.2 Sensor STH7x (71, 75).

El STH7x es un sensor de temperatura provee una salida digital con una precisión de ± 0.3 grados Centígrados, su voltaje de operación es entre 2.4 y 5.5 volts; su rango de medición es entre -40°C a 123°C .

3.3.3 Sensor LM35HA.

El LM35HA es un sensor que permite medir la temperatura en un rango entre -55°C a 150°C , su factor de escala es de 10 mili-Volts por cada 1°C con una precisión de ± 0.2 grados Centígrados lo cual permite medir en un rango muy pequeño, su voltaje de operación es entre 4 y 20 volts.

3.3.3 La elección de los sensores.

En la Tabla 6 y 7 se puede apreciar una comparativa entre los sensores de humedad y temperatura respectivamente.

SENSOR	HIH-4000	YL-38 & YL-69	DHT22
Rango de medida (%HR)	0 a 100	0 a 100	0 a 100
Precisión (%HR)	±0.5	±0.2	±0.2
Rango de Operación(V)	4 a 5.8	2.4 a 5.5	4 a 20
Precio	~39 Pesos M.N.	~40 Pesos M.N.	~110 Pesos M.N.

Tabla 6: Comparación entre los Sensores de Humedad.

SENSOR	DS600	STH7x	LM35HA
Rango de medida (°C)	-20 a 100	-40 a 123	-55 a 150
Precisión (°C)	±0.5	±0.3	±0.2
Factor de Escala (mV/°C)	6.45	--	10
Rango de Operación(V)	2.7 a 5.5	3.3 a 5	3.3 a 6
Precio	~45 Pesos M.N.	~39 Pesos M.N.	~25 Pesos M.N.

Tabla 7: Comparación de los Sensores de Temperatura.

Considerando el rango de medición, la precisión, el rango de voltaje de operación y el costo, de los tres modelos comerciales de sensores de temperatura y 3 de humedad se eligieron la Sonda YL-69 y Módulo YL-38 para la medición de la humedad y el sensor LM35HA por parte de la temperatura.

Los sensores de luminosidad y de CO2 no fueron contemplados para la realización del prototipo porque aunque es posible monitorear dichas variables, el control sobre ellas resultaría complicado y costoso para este prototipo.

3.4 Actuadores.

Se denominan actuadores a aquellos elementos que pueden provocar un efecto sobre un proceso automatizado. Los actuadores son dispositivos capaces de generar una fuerza a partir de líquidos, de energía eléctrica y gaseosa. El actuador recibe la orden de un regulador o controlador y da una salida necesaria para activar a un elemento final de control como por ejemplo las válvulas.

3.4.1 Relevador.

Un relevador, también conocido como relé, es un dispositivo que controla el estado de un interruptor mediante una entrada eléctrica. En su interior, posee una bobina que al energizarse induce una fuerza magnética que cambia el estado del interruptor. [9]

Se empleó este dispositivo debido a que es un dispositivo de carácter universal con respecto a la parte de potenciación, útil para energizar cualquier dispositivo (actuador) ya sea este de corriente alterna o corriente directa.

El dispositivo utilizado para este proyecto es una tarjeta con dos relés, la cual puede ser utilizada directamente con tarjetas de desarrollo. Cada relé es activado con una señal de 5 volts, es capaz de manejar una corriente alterna que va desde los 125 hasta los 250 volts y un corriente directa desde los 5 hasta los 28 volts; cuenta con un opto acoplador [10] que aísla la señal del circuito de salida (actuador) de la del circuito de entrada (módulo de control), dicho aislamiento es útil ya que los actuadores pueden utilizar mucho más voltaje que nuestro módulo de control.

3.4.2 Electro Válvulas.

Las electroválvulas son dispositivos que responden a pulsos eléctricos. Esto es gracias a la corriente que circula a través de una bobina es posible abrir o cerrar la válvula controlando, de esta forma, el flujo de fluidos. Al circular corriente por la bobina genera un campo magnético que atrae el núcleo móvil y al finalizar el efecto del campo magnético, el núcleo vuelve a su posición, en la mayoría de los casos, por efecto de un resorte, algo similar a lo que ocurre con un relé.

Este tipo de dispositivos son más fáciles de controlar mediante un programa de software, lo que lo vuelve ideal para la automatización del riego en el prototipo. El modelo contemplado para el prototipo opera con un voltaje de 9 a 12V, posee un caudal desde 0.05 hasta 1m³/h y puede soportar temperaturas de hasta 43°C.

3.4.3 Ventiladores.

Un ventilador es una máquina de fluido que es capaz de producir una corriente de aire. Uno de los modelos más comunes actualmente es el eléctrico y consiste en un rodete con aspas que giran gracias a un motor eléctrico produciendo una diferencia de presión que es utilizada para desplazar aire o gas de un lugar a otro.

Este dispositivo al igual que la electroválvula es fácil de controlar mediante programas de software, la función de este dispositivo dentro del prototipo sería para reducir la

temperatura cuando el riego es limitado. El ventilador contemplado opera a 12 volts, a 2000 R.P.M (revoluciones por minuto), con un flujo de aire de 69 CFM (pies cúbicos por minuto).

3.5 Lenguajes de programación para el módulo de control y el servidor.

Dentro de los lenguajes que podemos utilizar para programar el microcontrolador tenemos tres posibilidades.

- Lenguaje Processing/Wiring.
- Lenguaje ensamblador.
- Lenguaje JavaScript/Node.js.

3.5.1 Lenguaje Processing/Wiring.

El lenguaje Wiring es un conjunto de funciones, variables y constantes que encapsulan el funcionamiento del hardware facilitando el uso del mismo, adicionalmente se pueden utilizar las características del lenguaje C++ dentro del ambiente de desarrollo lo que permite crear prototipos de funciones, punteros, clases y objetos e incluso utilizar lenguaje de máquina y otras características propias del compilador para micro controladores de AVR [11].

Ventajas:

- Está basado en el Lenguaje C++ por lo que hereda y permite el uso de sus características.
- Incluye soporte para múltiples arquitecturas de hardware.
- Permite crear sketches sin utilizar herramientas adicionales.
- El ambiente de desarrollo permite crear proyectos complejos rápidamente.
- Ofrece gran flexibilidad al momento de crear proyectos.

Desventajas:

- Dado que la programación es de alto nivel, el uso de las librerías provoca retrasos en la ejecución del sketch sobre todo a la hora de hacer una adquisición de datos.
- Únicamente se encarga del control del micro controlador haciendo necesario otro lenguaje para el servidor

3.5.2 Lenguaje ensamblador.

Es un lenguaje de programación de bajo nivel para los computadores, microprocesadores, microcontroladores, y otros circuitos integrados programables. Implementa una representación simbólica de los códigos de máquina binarios y otras constantes necesarias para programar una arquitectura dada de CPU y constituye la representación más directa del código máquina específico para cada arquitectura legible por un programador.

Ventajas:

- Por ser lenguaje más cercano y trabajar directamente con el microprocesador, es procesado más rápido.
- No ocupa mucho espacio en memoria ya que no tiene que cargar librerías y demás como son los lenguajes de alto nivel.
- Se pueden hacer tareas específicas que en un lenguaje de alto nivel no se pueden llevar a cabo porque tienen ciertas limitantes que no se lo permite.

Desventajas:

- Al ser un lenguaje de bajo nivel requiere más instrucciones para realizar el mismo proceso, en comparación con un lenguaje de alto nivel.
- Cada microcontrolador usa un diferente set de mnemónicos y de banco de registros.
- No es una selección apropiada ya que al ser muy específico con el microcontrolador en turno posee poca portabilidad al migrar los programas a otros microcontroladores.

3.5.3 Lenguaje Javascript/Node.js.

Node.js es una plataforma basada en eventos, streams y está construido encima del motor de JavaScript V8, el cual es el mismo con el que funciona el motor JavaScript de Google Chrome [12], Se programa del lado del servidor, aunque no se limita a ello, además es muy útil en la creación de programas en red altamente escalables.

Ventajas:

- Node.js tiene compatibilidad con el protocolo Firmata [13] lo cual nos brinda la compatibilidad con una amplia gama de tarjetas de desarrollo como Arduino y Raspberry Pi. La librería Firmata proporciona los protocolos de comunicación serie que permitirán comunicar al servidor con el microcontrolador.
- Nos permite encapsular el funcionamiento del microcontrolador del lado del servidor brindando una comunicación directa.
- Debido a su enfoque orientado a eventos se vuelve un modelo liviano y eficiente de cara a aplicaciones real-time.

Desventajas:

- Al ser una plataforma en crecimiento pueden producirse incompatibilidad entre las actualizaciones de la API.
- Si no se está familiarizado con la programación asíncrona el código obtenido puede ser difícil de entender y mantener.

3.5.4 La elección del lenguaje para el modulo.

Con base a la investigación realizada, el lenguaje de programación idóneo para el desarrollo del proyecto es JavaScript/Node.js debido a que es ejecutado del lado del servidor, nos permite crear una aplicación altamente escalable y al ser orientado a eventos lo vuelve una muy buena opción para trabajar con sensores y actuadores.

La compatibilidad que posee con el protocolo Firmata nos permite elegir una amplia gama de tarjetas de desarrollo y encapsular el funcionamiento de los sensores y actuadores dentro del servidor sin la necesidad de un script exclusivo para esta función.

LENGUAJE	Ensamblador	Processing	Node.js
Aspectos positivos.	El procesamiento es más rápido.	Basado en Lenguaje C.	Alta escalabilidad.
+	Instrucciones más específicas.	No es necesario herramientas adicionales.	Enfoque orientado a eventos.
+	Ocupa poco espacio en memoria.	Permite crear proyectos complejos rápidamente.	Comunicación directa con un servidor.
Aspectos Negativos.	Únicamente se hace cargo del micro controlador.	Únicamente se hace cargo del micro controlador.	Múltiples actualizaciones.
-	Mnemónicos especiales para cada microcontrolador.	Ocurren retrasos durante la adquisición de datos.	

Tabla 8: Comparativa entre los Lenguajes de Programación.

3.6 Sistema Gestor de Base de Datos.

El Sistema Gestor de Base de Datos será una parte vital del sistema, para la cual se han evaluado dos alternativas.

- MySQL

- MongoDB

3.6.1 MySQL.

MySQL es un sistema de administración de base de datos, el cual nos permite velocidad y flexibilidad. Las tablas están conectadas por relaciones definidas que hacen posible combinar datos de diferentes tablas según sea el requisito. Es un software de fuente abierta, lo que quiere decir que cualquier persona puede descargarlo modificarlo y/o usarlo según sus necesidades.

Ventajas:

- Es software Open Source.
- Facilidad de instalación y configuración.
- Baja probabilidad de corromper los datos, incluso si los errores no se producen en el propio gestor sino en el sistema en el que está.

Desventajas:

- Un gran número de utilidades de MySQL no están documentadas.

3.6.2 MongoDB.

MongoDB es un sistema de base de datos NoSQL multiplataforma orientado a documentos, de esquema libre; esto quiere decir que cada entrada o registro puede tener un esquema de datos diferente, con atributos que no tienen por qué repetirse de un registro a otro.

El Shell y la manipulación de los datos son manejados con código muy similar al de JavaScript.

Puede ser obtenido de manera gratuita bajo licencia pública general.

Ventajas:

- La velocidad de consulta es más rápida frente a una base de datos relacional.
- Al ser libre de Schemas los datos no se dividen en diferentes tablas.
- Permite una alta frecuencia de lecturas y escrituras.

Desventajas:

- Tiene problemas de escalabilidad cuando el tamaño de las colecciones supera los 100 GB.
- Suele ser un poco más difícil de implementar que una base de datos relacional.

3.6.3 La elección del Sistema de Base de Datos.

El Sistema Gestor De Base De Datos a utilizar será MongoDB, ya que al ser una base de datos no relacional ofrece una mayor velocidad de consulta que MySQL, es un enfoque libre de Schemas lo que permite registros con diferentes atributos, sin olvidar que es software libre y su licencia no tiene costo.

SGBD	MySQL	MongoDB
Tipo de enfoque.	Sistema basado en tablas.	Sistema basado en colecciones de datos.
Relaciones.	—	Permite documentos embebidos.
	—	Mayor velocidad de lectura y escritura.
Esquema Dinámico.		X
Manejo de tipos de datos.	X	X
Intuitivo para Programadores.		X

Tabla 9: Comparación entre los Gestores de Base de Datos.

3.7 Tarjeta de Desarrollo.

En la actualidad han aparecido en el mercado muchas tarjetas de desarrollo las cuales están enfocadas en su mayoría hacia los desarrolladores de software en lugar de los de hardware, como la línea Arduino o la línea Raspberry Pi cada una con diferentes prestaciones y se espera que aparezcan nuevas tarjetas, aunque la mayoría de ellas desaparecerán con rapidez.

Para el desarrollo de este proyecto se contemplaron las siguientes modelos de tarjetas de desarrollo:

- Arduino Mega 2560
- Raspeberry Pi B+
- Tessel 2

3.7.1 Arduino Mega 2560.

La tarjeta Arduino Mega [14] está basada en el microcontrolador Atmega2560 ofrece un mayor número de entradas/salidas digitales y de entradas analógicas que otros modelos, además de poseer una memoria de mayor capacidad dentro de la gama media de la línea

Arduino. La Tabla 10 muestra una comparación entre diferentes modelos de tarjetas de desarrollo de la marca Arduino.

Arduino	UNO	Mega 2560	Leonardo	Due
Microcontrolador	Atmega328	Atmega2560	Atmega32U4	AT91SAM3XE
Velocidad de reloj	16 MHz	16 MHz	16 MHz	84 MHz
Pines digitales de E/S	14	54	20	54
Entradas Analógicas	6	16	12	12
Salidas Analógicas	0	0	0	2(DAC)
Memoria de programa (Flash)	32 Kb	256 Kb	32 Kb	512 Kb
Memoria de Datos (SRAM)	2 Kb	8 Kb	2.5 Kb	96 Kb
Memoria Auxiliar	1 Kb	4 Kb	1 Kb	0 Kb
Precio en el mercado	~\$300.00 Pesos M.N.	~\$350.00 Pesos M.N.	~\$230.00 Pesos M.N.	~\$800.00 Pesos M.N.

Tabla 10: Comparativa de tarjetas Arduino de gama media.

Ventajas:

- Es multiplataforma, lo más importante es que ofrece soporte para el protocolo Firmata.
- Es una opción asequible en relación al costo.
- Posee muchos módulos para poder ampliar las prestaciones de la tarjeta los cuales son de fácil montaje.
- Es una tarjeta plug and play por lo que se pueden conectar los componentes y comenzar a programar.

Desventajas:

- Al ser solo un microcontrolador no posee el poder de cómputo para soportar un servidor web.

3.7.2 Raspberry Pi B+.

La tarjeta de desarrollo Raspberry Pi [15] en su versión B+ es más una minicomputadora que si bien no es capaz de sustituir a un PC de escritorio si es capaz de ejecutar distribuciones del sistema operativo Linux, sus especificaciones técnicas son:

- Un procesador Broadcom BCM2835.
- Memoria RAM de 512 MB SDRAM de 400 MHz.
- EL módulo de almacenamiento es por medio de una tarjeta microSD de 2 a 16 GB.

- Cuatro puertos USB.
- 40 Pines GPIO.

Su precio es de aproximadamente \$1200.00 Pesos M.N.

Ventajas:

- Al ser más una minicomputadora que solo una tarjeta de desarrollo nos permite ejecutar un sistema operativo por lo que puede albergar un servidor web.
- Posee una gran cantidad de accesorios para ampliar las prestaciones de la tarjeta.
- Es capaz de conectarse a internet por medio de un puerto Ethernet.

Desventajas:

- No es un dispositivo plug and play ya que requiere una previa configuración del sistema operativo que no siempre resulta sencilla.

3.7.3 Tessel 2.

La tarjeta Tessel 2 [16] es una tarjeta de desarrollo que surge a partir del enfoque que tiene Arduino, ser más familiar para los desarrolladores de software, su sistema operativo es un intérprete de JavaScript construido en torno al tiempo de ejecución de Lua, y es compatible con la API de Node.js; sus especificaciones técnicas son:

- Un procesador Mediatek MT7620n de 580 MHz.
- Memoria RAM de 64 MB DDR2.
- Memoria Flash de 32 MB.

Su costo es de aproximadamente \$700.00 Pesos M.N.

Ventajas:

- Tiene a Javascript como lenguaje nativo
- Posee conectividad Wifi.
- Es un dispositivo plug and play.

Desventajas:

- Es una tecnología sumamente reciente; nada asegura que se consolide en el mercado como la línea Arduino o Raspberry.

3.7.4 La elección de la tarjeta de desarrollo.

El modelo Raspberry Pi B+ es el ideal para el proyecto por el gran poder de cómputo que esta ofrece pudiendo mantener el control de los sensores, actuadores, el servidor y el sistema gestor de base de datos, todo encapsulado en el mismo modulo.

	Raspberry Pi B+	Tessel 2	Arduino Mega
Node.js	X	X	X
MongoDB	X		
Puerto Ethernet	X	X	
Web Service	X	X	
Puertos E/S	40	20	54

Tabla 11: Comparativa entre Tarjetas de desarrollo.

Por cuestiones de presupuesto se optó por dividir la funcionalidad que la tarjeta Raspberry Pi B+ ofrecería, entre una PC para alojar el servidor web, al módulo de control y una tarjeta Arduino Mega2560 que aunque presenta limitantes frente a una Raspberry Pi B+ como el alcance de la conexión USB es completamente funcional para llevar a cabo únicamente las funciones de los puertos GPIO (Entrada/Salida de Propósito General).

De esta manera a pesar de no contar con una tarjeta Raspberry Pi B+, usando en conjunto estos componentes se puede lograr los alcances de proyecto, además una de las ventajas del lenguaje Node.js es que se podría cambiar a una Raspberry Pi B+ sin mayores inconvenientes.

3.8 Costos del sistema.

El costo del sistema aquí reflejado, es solo una estimación que puede variar en el futuro por diversos factores, tales como la fluctuación en el precio del dólar, ya que ello afecta el precio de los componentes electrónicos para la producción de las partes del sistema.

Para calcular los costos totales del sistema [17], se ha dividido dichos costos en dos partes, la primera parte contempla el coste de la implementación del prototipo, el costo del diseño y de los materiales; la segunda parte contempla el costo del desarrollo del servidor y de la interfaz de usuario. Sumados ambos costos nos entregaran el costo total del sistema.

3.8.1 Coste de la implementación del prototipo.

Una cosa muy importante a mencionar es que el costo de un prototipo siempre es mayor al costo de producción en el mercado, debido a que el costo de los materiales al mayoreo suele reducir su valor.

3.8.1.1 Coste de Diseño.

En este apartado se toman en cuenta los diferentes aspectos que han sido necesarios para la creación del prototipo, en la Tabla 12, se desglosan las 260 horas que fueron necesarias para elaborarlo, así como el precio por hora de cada uno de los aspectos. Los precios están expresados en pesos mexicanos.

ASPECTO	Horas	Precio/hora	Total
Diseño del hardware y software	200	\$ 50 Pesos M.N.	\$ 10,000 Pesos M.N.
Depuración del software	5	\$ 50 Pesos M.N.	\$250 Pesos M.N.
Montaje del prototipo	20	\$ 40 Pesos M.N.	\$ 800 Pesos M.N.
Simulaciones y pruebas realizadas	35	\$ 35 Pesos M.N.	\$ 1225 Pesos M.N.
TOTAL			\$ 12275 Pesos M.N.

Tabla 12: Costes de diseño del prototipo.

3.8.1.2 Coste de materiales.

Para calcular los costos de los materiales necesarios se tomó en cuenta el costo de cada material utilizado para el módulo de control, de igual manera los precios están en pesos mexicanos.

La tabla fue elaborada tomando como módulo de control una Raspberry Pi B+ aunque para fines demostrativos la tarjeta utilizada será una Arduino 2560 y una PC para alojar al servidor.

MATERIAL	UNIDADES	PRECIO
Raspberry pi B+ con adaptador de puertos GPIO(General Ports Input-Output)	1	\$ 1350.00 Pesos M.N.
Memoria microSD de 8gb	1	\$ 150.00 Pesos M.N.
Sensor lm35	1	\$ 25.00 Pesos M.N.
Sonda YL38	1	\$ 45.00 Pesos M.N.
Módulo de dos Relés	1	\$ 75.00 Pesos M.N.
Cable	---	\$ 5.00 Pesos M.N.
Soldadura	---	\$ 5.00 Pesos M.N.
Placa Fenólica	---	\$ 15.00 Pesos M.N.
Gabinete de acrílico	1	\$ 281.00 Pesos M.N.
Electrovalvula 12V	1	\$ 200.00 Pesos M.N.
Ventilador	1	\$ 100.00 Pesos M.N.
TOTAL		\$ 2251.00 Pesos M.N.

Tabla 13: Costos del material para el Módulo de Control.

El costo total de la Implementación del Prototipo se obtiene al sumar el costo del Diseño del prototipo más el Costo de los materiales para el Módulo de Control el cual es de **\$16226.00 Pesos M.N.** (Tabla 14).

Tipo de Coste	PRECIO
Diseño del Prototipo	\$ 12275.00 Pesos M.N.
Material para el Módulo de Control	\$ 2251.00 Pesos M.N.
TOTAL (Implementación del prototipo)	\$16226.00 Pesos M.N.

Tabla 14: Costo de la Implementación del Prototipo.

3.8.2 Costo del desarrollo del servidor e interfaz de usuario.

Para el cálculo del costo del software utilice la técnica LOC [18] (líneas de código), la cual se basa en el número de líneas de código generadas para el sistema, así como el esfuerzo total por parte de los creadores del proyecto.

Para poder comenzar el cálculo de costos mediante LOC, es necesario analizar y descomponer el problema del sistema en partes las cuáles sean más fáciles de estimar; a pesar de que la programación para el sistema no requiere el uso de diferentes lenguajes para la parte del módulo de control y del servidor si es posible separarlos para una realizar mejor estimación.

Las secciones a considerar son las siguientes:

- Interfaz.
- Programación del módulo de control.
- Programación del servidor.

Lo siguiente es elaborar la tabla de LOC estimadas para este caso, la cual es la siguiente:

LOC ESTIMADAS				
MODULO	Optimista	Medio	Pesimista	Esperado
Interfaz	2000	2300	3500	2500
Programación del módulo de control.	150	200	350	250
Programación del servidor.	1500	1800	2200	2000
TOTAL				4750

Tabla 15: Estimación de Líneas de Código.

Para calcular el salario hombre-mes se obtiene en base a su salario real, sumando del costo de prestaciones.

En este caso en particular, el salario real se obtuvo en base a los siguientes criterios:

- Nivel de escolaridad: Licenciatura.
- Horas de trabajo: 6 diarias
- Puesto: programador
- Dificultad del sistema: media

A cada uno de estos criterios se le fue asignado un valor económico, los cuales dan como resultado el salario, cabe destacar que el dicho valor para cada uno de los criterios se obtuvo en base a una media de lo que hoy en día las empresas en el D.F. pagan a sus empleados en un puesto de programador. Los precios se encuentran en pesos mexicanos (M.N.).

CRITERIO	COSTO
Nivel de escolaridad	\$ 10000.00 Pesos M.N.
Horas de trabajo	\$ 2000.00 Pesos M.N.
Puesto	\$ 3000.00 Pesos M.N.
Dificultad del sistema	\$ 5000.00 Pesos M.N.
TOTAL	\$ 20000.00 Pesos M.N.

Tabla 16: Criterios de salarios.

De la Tabla 16 se obtiene un costo de \$ 20000.00 el cual sería un costo para un mes de trabajo. Estimando un tiempo de desarrollo de 4 meses, el costo total para el desarrollo de software es de **\$ 80000.00 M.N.**

Sumando el costo de la Implementación del Prototipo y el costo de la Interfaz de Usuario y el servidor se obtiene un costo total de **\$96226.00 Pesos M.N.** (Tabla

Tipo de Coste	PRECIO
Implementación del Prototipo	\$ 16226.00 Pesos M.N.
Desarrollo del servidor e Interfaz de usuario	\$ 80000.00 Pesos M.N.
TOTAL	\$96226.00 Pesos M.N.
(Costo Total)	

Tabla 17: Coste total del sistema.

Capítulo IV

4.1 Diagrama a bloques general del sistema.

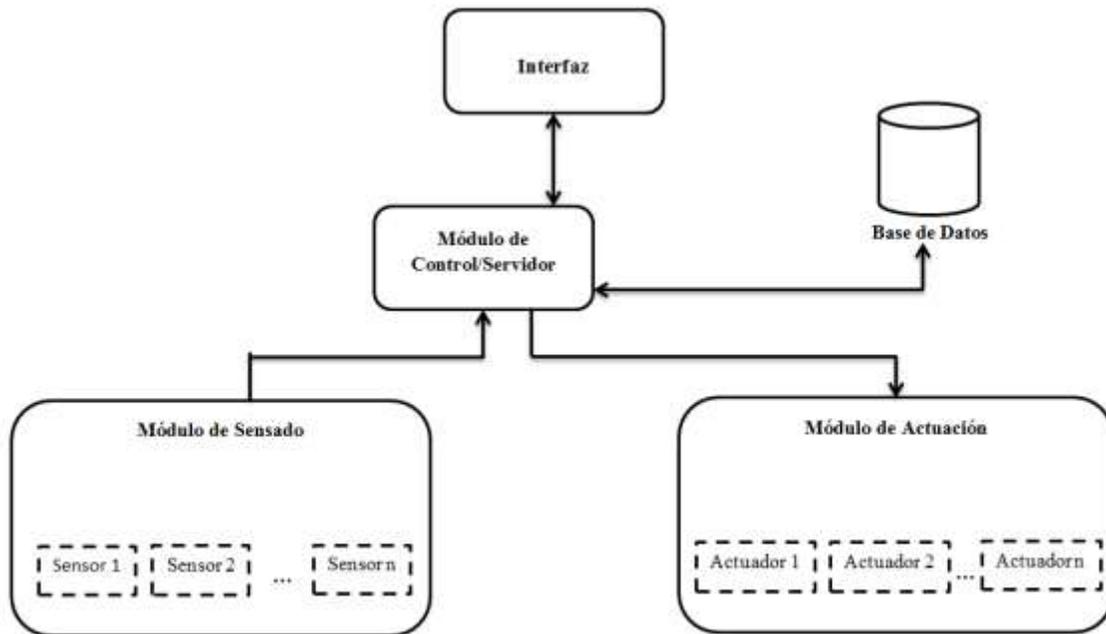


Figura 3: Diagrama a bloques del sistema.

El sistema como se puede apreciar consta de los siguientes bloques, los cuales se explican a continuación.

- **Módulo de Sensado:** Esta es una serie de bloques que comprende a los sensores encargados de muestrear datos de variables físicas para su posterior procesamiento.
- **Módulo de Actuación:** Esta serie de bloques comprenden los actuadores que serán controlados por el módulo de control según los requisitos de la hortaliza.
- **Interfaz:** Este será el módulo donde se mostrara los datos recopilados por los sensores, el estado de los actuadores así como su control.
- **Módulo de control/Servidor:** Este será el módulo acoplador de todos los demás bloques que comprenden en su totalidad al sistema, este será el encargado de almacenar los datos recopilados por los sensores en la base de datos, también es el encargado de evaluar dichos datos y activar los actuadores según los requisitos de la hortaliza, y alojara un servidor para poder desplegar la Interfaz de usuario.

4.2 Diagrama a bloques del Servidor/Módulo de control.

El servidor será el módulo encargado de todo el funcionamiento del sistema. Utilizar Node.js como lenguaje de programación nos permite realizar el control y monitoreo de los actuadores y sensores, sin la necesidad de crear un script por separado para realizar dicha función; además, brindar soporte a un servicio web para alojar la interfaz de usuario final y que el registro de las acciones del sistema pueda ser consultado a través de internet.

Una de las grandes ventajas de un servidor utilizando Node.js es que al tener toda la funcionalidad encapsulada en un único módulo, este puede ser migrado a otra tarjeta de desarrollo compatible sin mayores inconvenientes, únicamente adecuando algunas funcionalidades que dependerán de la tarjeta.

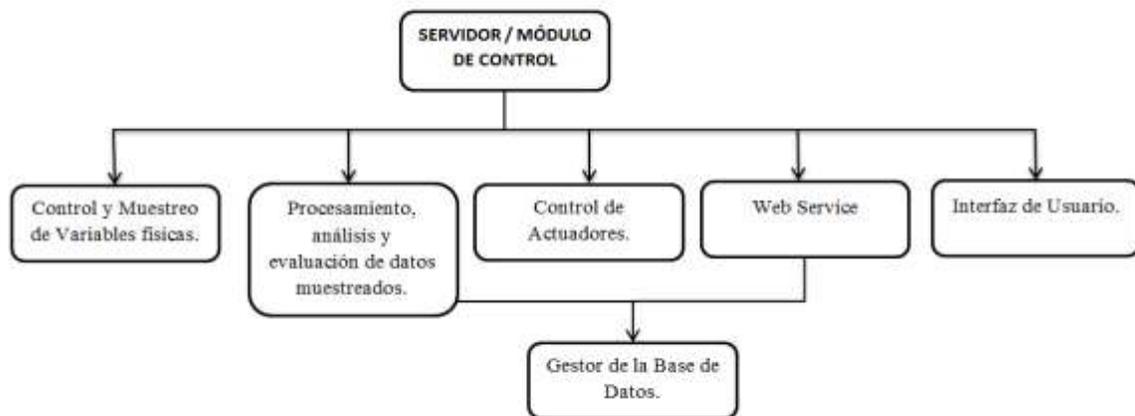


Figura 4: Diagrama a bloques del Servidor.

Descripción de cada uno de los bloques:

Control y Muestreo de Variables físicas: este bloque será el encargado de la recopilación de datos mediante el control directo de los sensores.

Procesamiento, análisis y evaluación de datos muestreados: este bloque será el encargado de realizar el procesamiento de los datos recopilados, posteriormente realizará un análisis y una evaluación con respecto a las condiciones necesarias para la hortaliza que está siendo monitoreada; por ello este bloque es capaz de acceder a la base de datos para obtener la información de la hortaliza. Emitirá una respuesta al bloque “Control de Actuadores” según sea conveniente; por ejemplo, si la temperatura es muy alta y sobrepasa el valor límite o el sistema determina que puede ser contraproducente a largo plazo se enviará una orden de activación al bloque “Control de Actuadores” para que este lleve a cabo dicha orden y active el actuador correspondiente a la temperatura. También se evaluarán ciertos casos especiales como el fallo en el suministro de agua.

Control de Actuadores: este bloque es el encargado del control directo de los actuadores según la orden emitida por el bloque “Procesamiento, análisis y evaluación de los datos muestreados”, ya sea activando o desactivando los actuadores según sea el caso.

Web Service: este será el bloque encargado de proveer de información al bloque “Interfaz de Usuario”, realizando para ello consultas en base de datos y comunicación directa con el bloque “Procesamiento, análisis y evaluación de los datos muestreados” para la obtención de datos muestreados. Siendo capaz de realizar comunicación con distintas plataformas.

Interfaz de Usuario: su función será desplegar la información que es consumida desde el bloque “Web Service”, la información general de la hortaliza que está siendo monitoreada los datos muestreados, históricos de las acciones realizadas por el sistema. Además de proveer un control manual sobre los actuadores y notificar al usuario sobre ciertos eventos especiales tales como el fallo en el suministro de agua.

Gestor de la Base de Datos: será el encargado de administrar y gestionar la información de las hortalizas, los datos muestreados y las ordenes generadas por el bloque “Procesamiento, análisis y evaluación de los datos muestreados”; datos que serán utilizados por el sistema.

4.3 Diagramas de Casos de Uso.

En este apartado, se utilizarán diagramas de casos de uso con la descripción de cada uno de ellos, para describir adecuadamente los casos de uso de la aplicación, se utilizarán tablas con el siguiente formato:

Identificador único del caso de uso	
Descripción corta	
Actores	Roles posible que pueden tomar los usuarios a la hora de interactuar con el sistema.
Objetivo	Servicio que el actor busca conseguir.
Precondiciones	Descripción del conjunto de estados del sistema anteriores a la ejecución del caso de uso.
Postcondiciones	Descripción del conjunto de estados del sistema posteriores a la ejecución del caso de uso.
Escenario Básico	Secuencia de acciones principales (información intercambiada) en la interacción del escenario básico.

Tabla 18: Caso de uso de ejemplo: descripción.

4.3.1 Diagramas de casos de uso para el actor Usuario.

En este diagrama se representan las funcionalidades y comportamientos del sistema ante su interacción con el usuario.

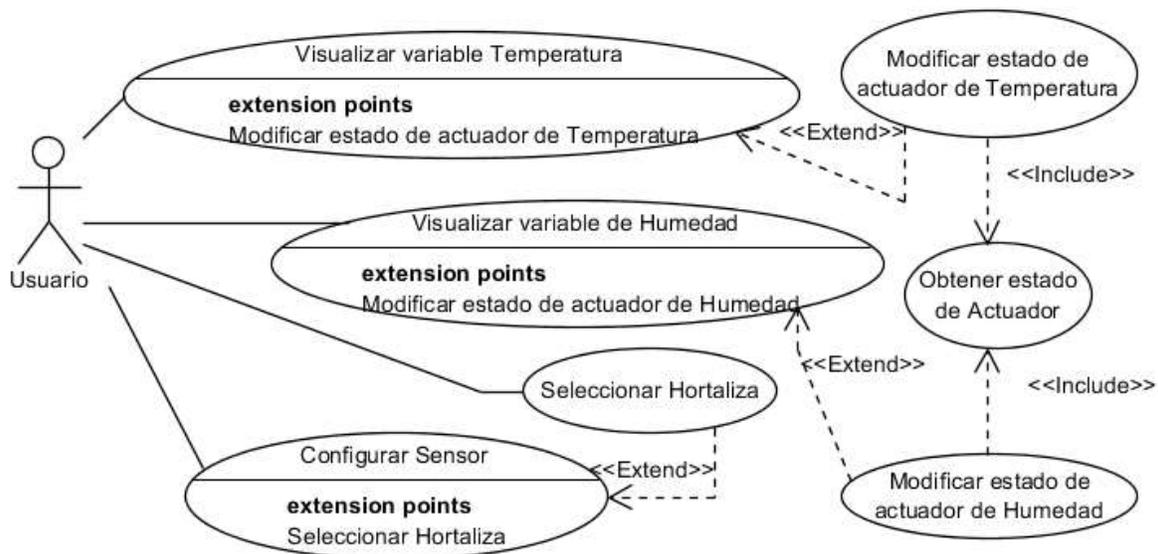


Figura 5: Diagrama de Casos de uso para el Usuario.

4.3.2 Diagrama de casos de uso para el usuario Módulo de control.

En este diagrama se representan las diferentes funcionalidades que realiza el módulo de control.

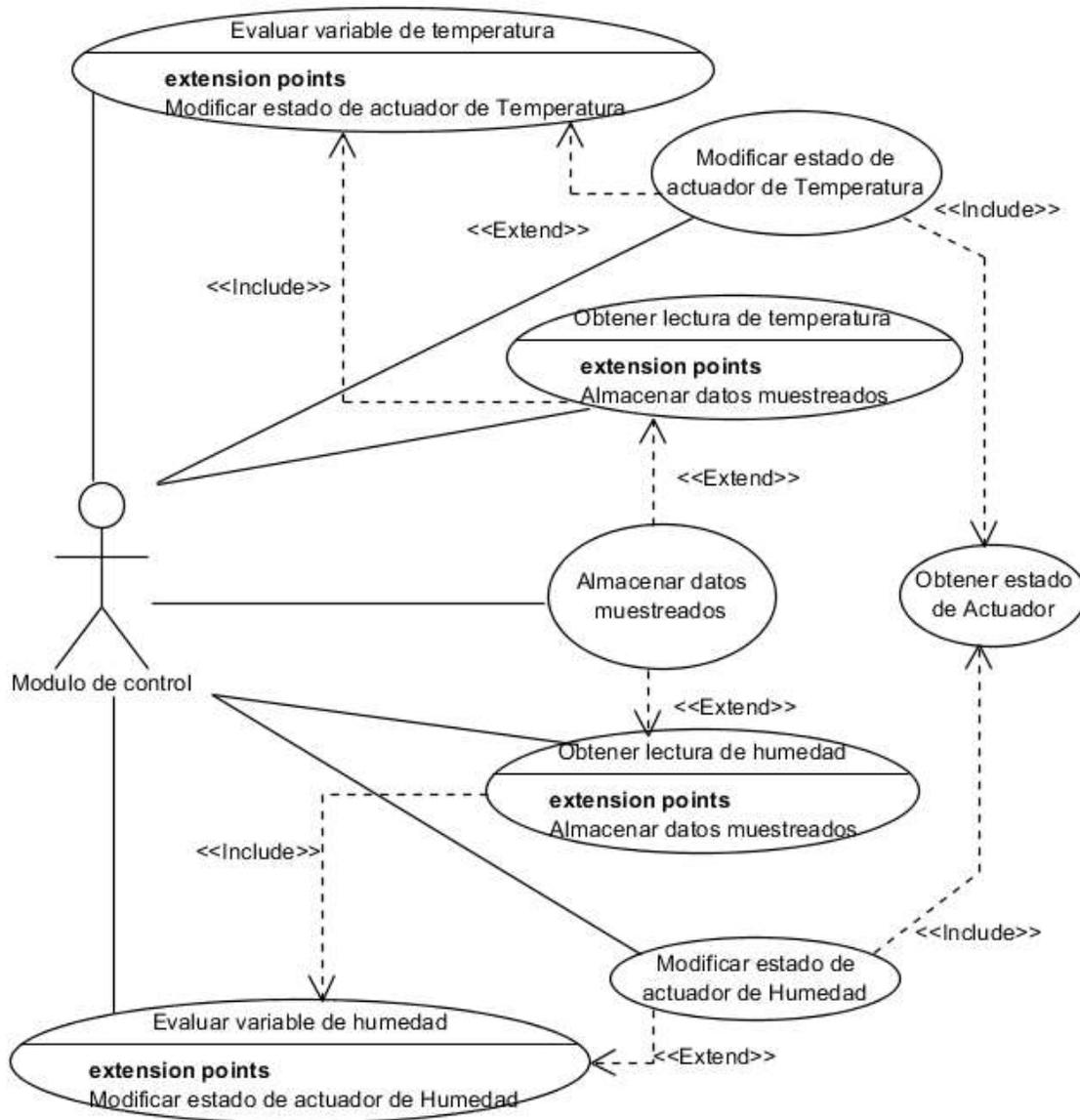


Figura 6: Diagrama de Casos de uso para el Módulo de Control.

4.3.3 Especificación de Casos de Uso.

A continuación se lista el conjunto de casos de uso según el formato de tabla anteriormente mostrado.

CU-01	
Visualizar variable Temperatura	
Actores	Usuario.
Objetivo	Visualizar el valor muestreado por el sensor de temperatura.
Precondiciones	---
Postcondiciones	El valor de la temperatura es mostrado en la interfaz de usuario.
Escenario Básico	<ul style="list-style-type: none">○ El usuario pulsa sobre el botón “Temperatura”○ La aplicación muestra una ventana con los valores muestreados por el sensor de temperatura.

Tabla 19: Caso de uso CU-01. Visualizar variable Temperatura.

CU-02	
Modificar el estado del actuador de Temperatura	
Actores	Usuario, Módulo de control.
Objetivo	Modificar el estado del actuador que controla la temperatura.
Precondiciones	El actuador de Temperatura debe estar conectado a una salida del Módulo de control para poder modificar su estado.
Postcondiciones	El estado del actuador se ha modificado según la petición del Usuario o del Módulo de control.
Escenario Básico	<ul style="list-style-type: none">○ El Usuario visualiza el valor actual de la temperatura.○ Se visualiza el estado del actuador.○ El Usuario cambia el estado del actuador. ○ El Módulo de control evalúa el valor muestreado por el sensor de temperatura con los valores de la hortaliza monitoreada.○ Según la evaluación el Módulo de control modifica el estado del actuador según la necesidad.

Tabla 20: Caso de uso CU-02. Modificar el estado del actuador de Temperatura.

CU-03	
Obtener el estado del Actuador	
Actores	Usuario, Módulo de control.
Objetivo	Obtener el valor estado actual del actuador.
Precondiciones	---
Postcondiciones	Se obtiene el estado del actuador, previo a su modificación.
Escenario Básico	<ul style="list-style-type: none"> ○ El Módulo de control solicita el estado del Actuador.

Tabla 21: Caso de uso CU-03. Obtener el estado del Actuador.

CU-04	
Visualizar variable Humedad	
Actores	Usuario.
Objetivo	Visualizar el valor muestreado por el sensor de humedad.
Precondiciones	---
Postcondiciones	El valor de la humedad es mostrado en la interfaz de usuario.
Escenario Básico	<ul style="list-style-type: none"> ○ El usuario pulsa sobre el botón “Humedad” ○ La aplicación muestra una ventana con los valores muestreados por el sensor de humedad.

Tabla 22: Caso de uso CU-04. Visualizar variable Humedad.

CU-05	
Modificar el estado del actuador de Humedad	
Actores	Usuario, Modulo de control.
Objetivo	Modificar el estado del actuador que controla la humedad.
Precondiciones	El actuador de Humedad debe estar conectado a una salida del Módulo de control para poder modificar su estado.
Postcondiciones	El estado del actuador se ha modificado según la petición del Usuario o del Módulo de control.
Escenario Básico	<ul style="list-style-type: none"> ○ El Usuario visualiza el valor actual de la humedad. ○ Se visualiza el estado del actuador. ○ El Usuario cambia el estado del actuador. ○ El Módulo de control evalúa el valor muestreado por el sensor de humedad con los valores de la hortaliza monitoreada. ○ Según la evaluación el Módulo de control modifica el estado del actuador según la necesidad.

Tabla 23: Caso de uso CU-05. Modificar el estado del actuador de Humedad.

CU-06	
Configurar Sensor	
Actores	Usuario
Objetivo	Cambiar parámetros de configuración del sensor.
Precondiciones	El sensor debe estar conectado a una entrada del módulo de control para poder modificar sus parámetros.
Postcondiciones	Los parámetros del sensor de temperatura serán modificados.
Escenario Básico	<ul style="list-style-type: none"> ○ El Usuario selecciona el sensor a configurar. ○ Los parámetros del sensor son mostrados al usuario. ○ El Usuario modifica los parámetros del Sensor.

Tabla 24: Caso de uso CU-06. Configurar Sensor.

CU-07	
Seleccionar Hortaliza	
Actores	Usuario
Objetivo	Seleccionar un tipo de hortaliza para monitorear.
Precondiciones	El sensor debe estar colocado en la hortaliza a monitorear.
Postcondiciones	El Módulo de control tendrá parámetros de Temperatura y Humedad de la hortaliza a monitorear.
Escenario Básico	<ul style="list-style-type: none"> ○ Se despliegan en la interfaz diferentes tipos de hortalizas. ○ El usuario selecciona una hortaliza. ○ El Módulo de control evalúa la hortaliza según sus parámetros de temperatura y humedad.

Tabla 25: Caso de uso CU-07. Seleccionar Hortaliza.

CU-08	
Obtener lectura de Temperatura	
Actores	Módulo de control.
Objetivo	El Módulo de control obtiene una lectura de temperatura para su posterior procesamiento.
Precondiciones	El sensor de Temperatura debe estar conectado para poder obtener una lectura.
Postcondiciones	---
Escenario Básico	<ul style="list-style-type: none"> ○ El Módulo de control solicita una lectura al sensor de temperatura.

Tabla 26: Caso de uso CU-08. Obtener lectura de Temperatura.

CU-09	
Evaluar variable de Temperatura	
Actores	Módulo de control.
Objetivo	Evaluar la lectura de la temperatura para posteriormente determinar el estado del actuador de Temperatura.
Precondiciones	El sensor de Temperatura debe estar conectado para poder obtener una lectura.
Postcondiciones	El Módulo de control guarda la lectura obtenida y delibera el estado del actuador de Temperatura.
Escenario Básico	<ul style="list-style-type: none"> ○ El Módulo de control evalúa el valor muestreado con los valores límites de la hortaliza que está monitoreando. ○ Se cambia el estado del actuador según el resultado de la evaluación.

Tabla 27: Caso de uso CU-09. Evaluar variable de Temperatura.

CU-10	
Almacenar datos muestreados	
Actores	Módulo de control.
Objetivo	Guardar el valor muestreado por el sensor en la base de datos.
Precondiciones	El sensor de Temperatura debe estar conectado para poder obtener una lectura.
Postcondiciones	El valor muestreado ha sido guardado en la base de datos.
Escenario Básico	<ul style="list-style-type: none"> ○ El Módulo de control obtiene el valor muestreado. ○ El valor muestreado es guardado en la base de datos.

Tabla 28: Caso de uso CU-10. Almacenar datos muestreados.

CU-11	
Obtener lectura de Humedad	
Actores	Módulo de control.
Objetivo	El Módulo de control obtiene una lectura de temperatura para su posterior procesamiento.
Precondiciones	El sensor de Humedad debe estar conectado para poder obtener una lectura.
Postcondiciones	---
Escenario Básico	<ul style="list-style-type: none"> ○ El Módulo de control solicita una lectura al sensor de humedad.

Tabla 29: Caso de Uso CU-11. Obtener lectura de Humedad

CU-12	
Evaluar variable de Humedad	
Actores	Módulo de control.
Objetivo	Evaluar la lectura de la temperatura para posteriormente determinar el estado del actuador de Humedad.
Precondiciones	El sensor de Humedad debe estar conectado para poder obtener una lectura.
Postcondiciones	El Módulo de control guarda la lectura obtenida y delibera el estado del actuador de Humedad.
Escenario Básico	<ul style="list-style-type: none"> ○ El Módulo de control evalúa el valor muestreado con los valores límites de la hortaliza que está monitoreando. ○ Se cambia el estado del actuador según el resultado de la evaluación.

Tabla 30: Caso de uso CU-12. Evaluar variable de Humedad.

4.4 Diagramas de Secuencia.

Para una comprensión del sistema más completa, se utilizaron diagramas de secuencia que ilustran la dinámica de las interacciones entre las distintas partes del sistema.

A continuación se presentan los diagramas de secuencia de las interacciones más importantes del usuario con el sistema. Los componentes implicados existen únicamente a nivel conceptual y no a nivel de implementación, lo que se busca es ofrecer una explicación clara y abstracta de las particularidades de la implementación.

4.4.1 Diagrama de Secuencia Visualizar Valor Muestreado.

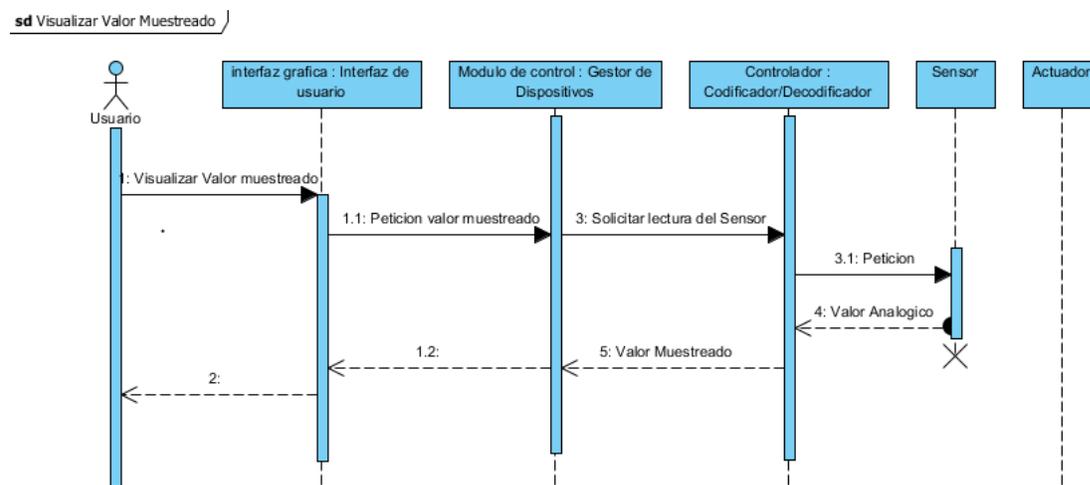


Figura 7: Diagrama de Secuencia Visualizar Valor Muestreado.

En este diagrama de secuencia el usuario estará en la pantalla para visualizar los valores muestreados por el sensor. Para llevar a cabo esto se creará una petición al módulo de control el cual obtendrá el valor analógico del sensor, realizará un procesamiento para convertirlo en un valor de temperatura o humedad y lo enviará a la interfaz de usuario para ser mostrado.

4.4.2 Diagrama de Secuencia Modificar Estado de Actuador.

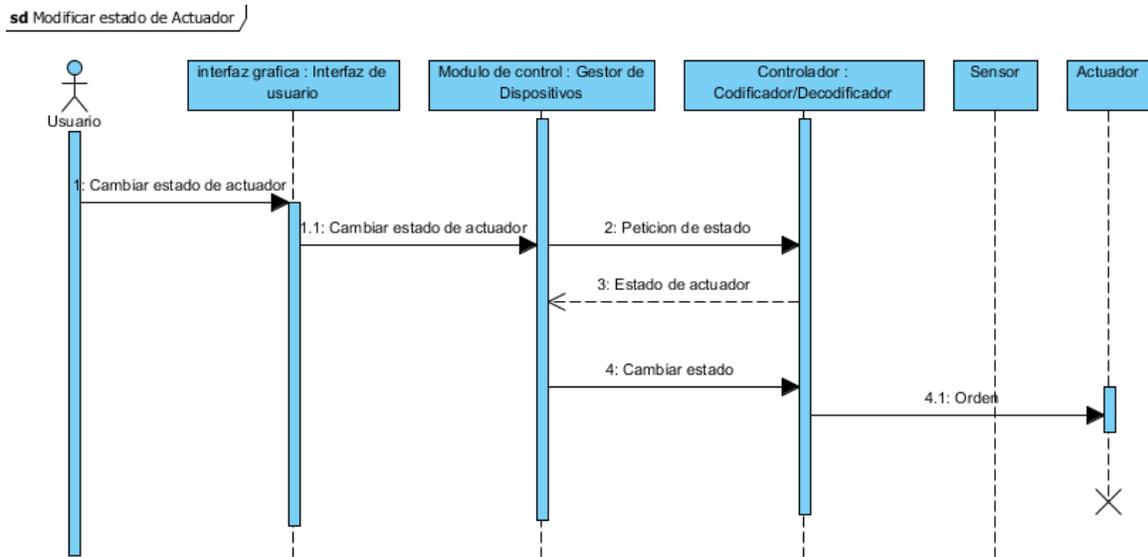


Figura 8: Diagrama de Secuencia Modificar Estado de Actuador.

En este diagrama de secuencia el usuario visualiza el estado de los actuadores conectados al sistema, y cambia el estado del actuador. Para ello se realizara una orden al módulo de control con el nuevo estado del actuador y este aplicara el cambio.

4.4.3 Diagrama de Secuencia Configurar Sensor.

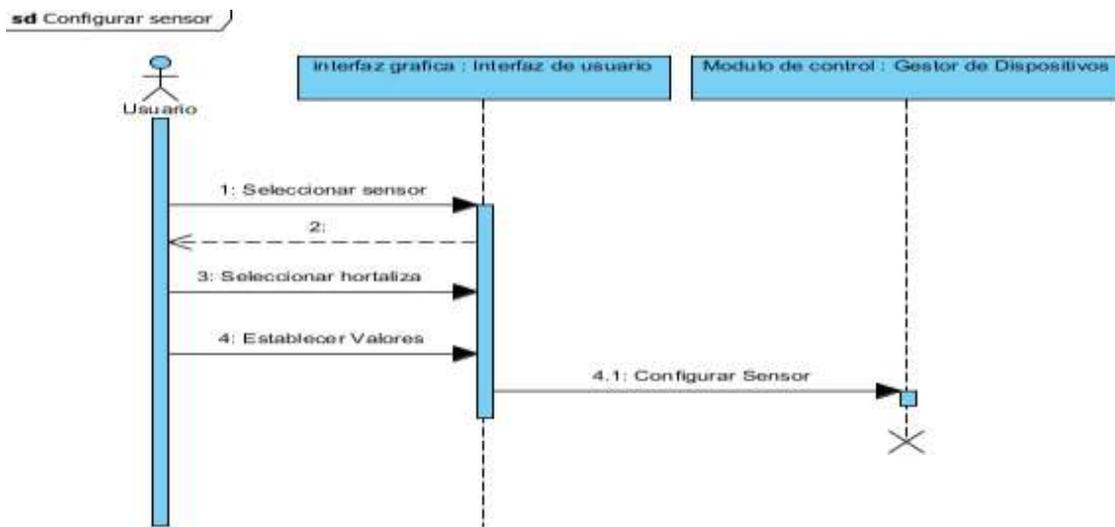


Figura 9: Diagrama de Secuencia Configurar Sensor.

En este diagrama de secuencia el usuario selecciona un sensor conectado y alguna de las hortalizas predeterminadas para establecer los valores máximos y mínimos que el módulo de control evaluara.

4.4.4 Diagrama de Secuencia Módulo de control.

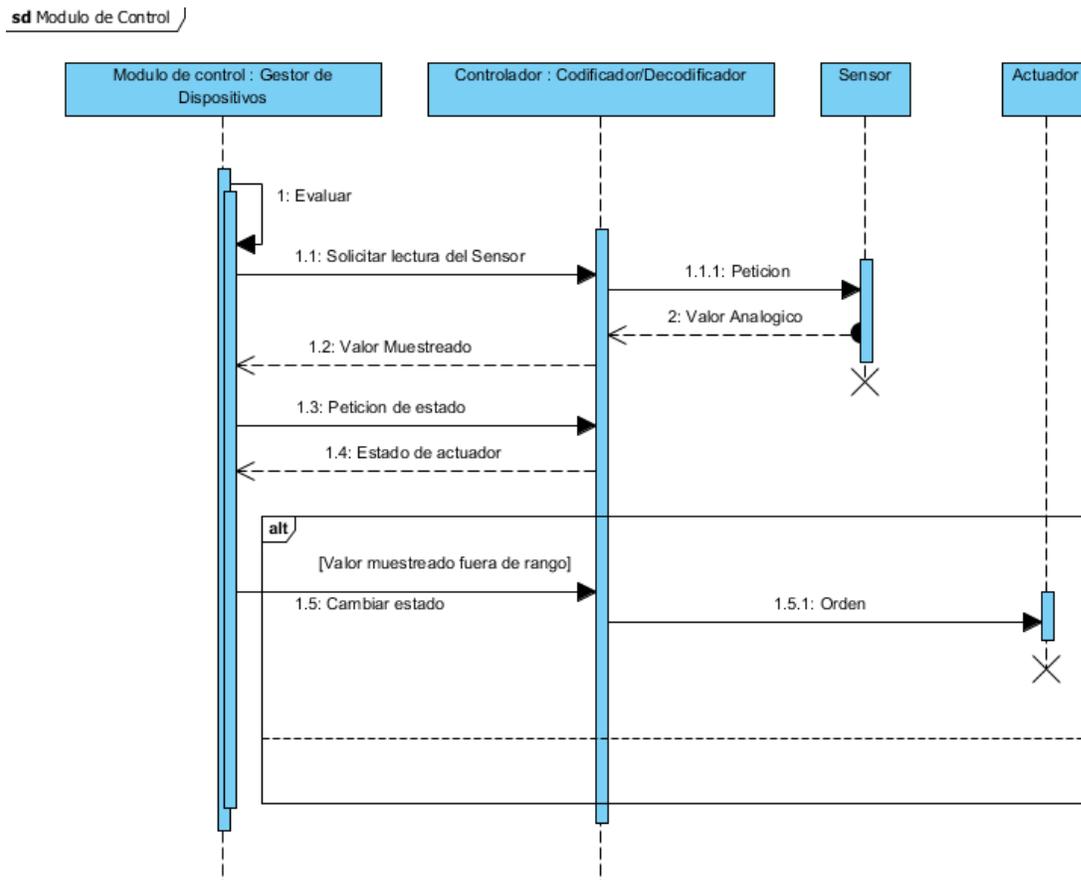


Figura 10: Diagrama de Secuencia para el Módulo de control

Este es el diagrama de secuencia para el módulo de control, se muestran las acciones que se ejecutaran de manera autónoma. El módulo de control evalúa los valores obtenidos por el sensor para que no superen los valores límite establecidos para cada hortaliza.

En caso que el valor supere los rangos establecidos el módulo de control generara una orden para cambiar el estado del actuador y controlar las variables físicas.

4.5 Diagramas de Estados.

4.5.1 Visualizar Variable de Temperatura.

El sistema recibe una petición para obtener el valor de temperatura, espera hasta recibir el valor del sensor, lo procesa y envía el valor muestreado a la interfaz.

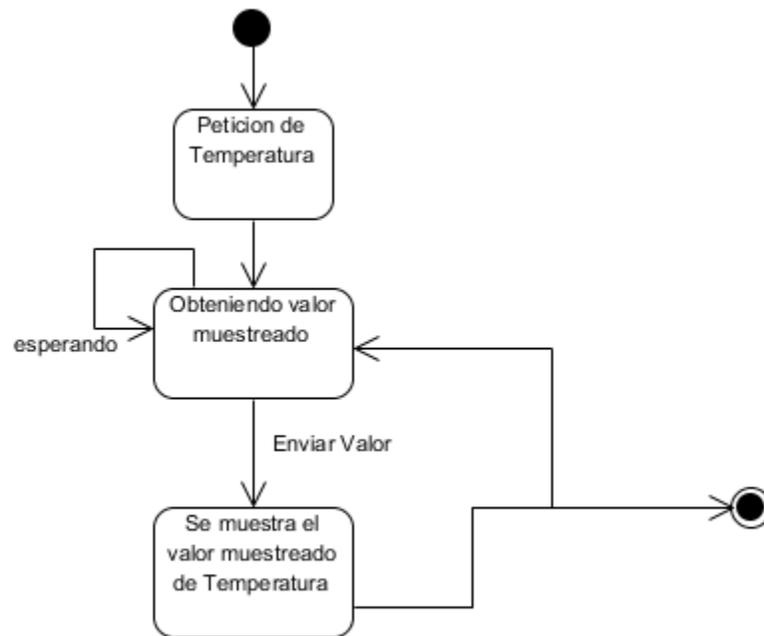


Figura 11: Diagrama de Estados Visualizar Variable de Temperatura.

4.5.2 Modificar Actuador de Temperatura.

El sistema se encuentra a la espera de una orden causada por algún valor de temperatura fuera de rango, cuando es recibida el sistema modifica el estado del actuador, y envía la orden al actuador para modificar su estado físicamente.

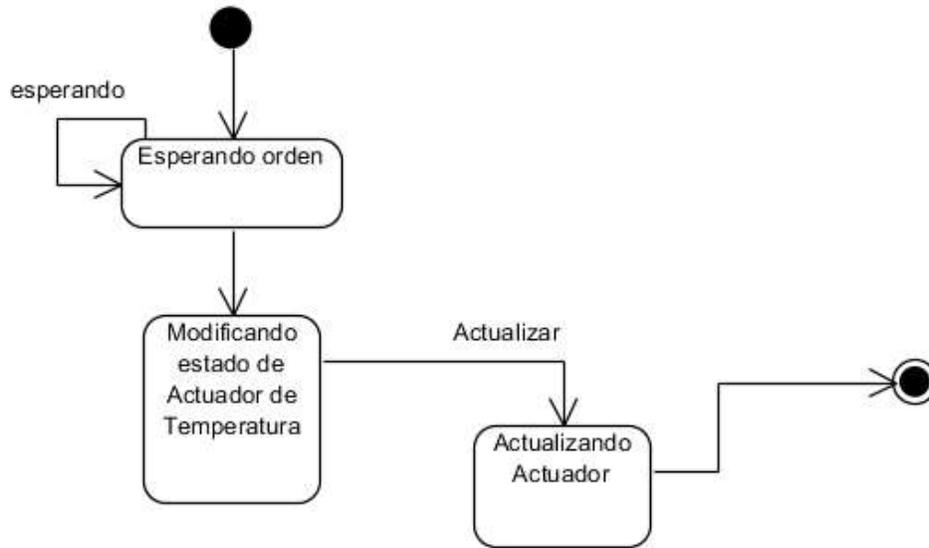


Figura 12: Diagrama de Estados Modificar Actuador de Temperatura.

4.5.3 Obtener Estado de Actuador.

El sistema verifica el estado actuador del actuador seleccionado y lo retorna para su posterior procesamiento.

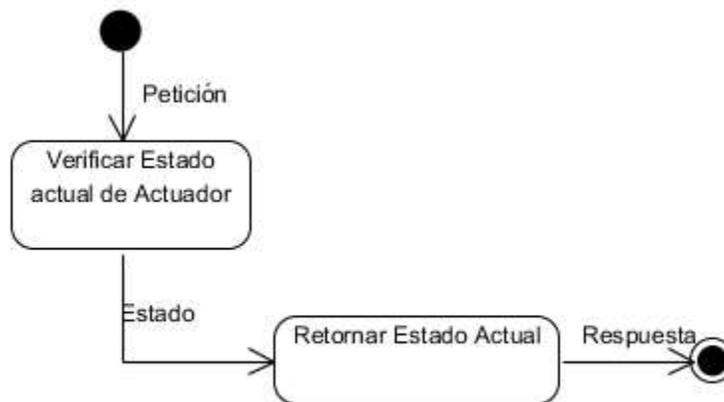


Figura 13: Diagrama de Estados Obtener Estado de Actuador

4.5.4 Visualizar Variable de Humedad.

El sistema recibe una petición para obtener el valor de humedad, espera hasta recibir el valor del sensor, lo procesa y envía el valor muestreado a la interfaz.

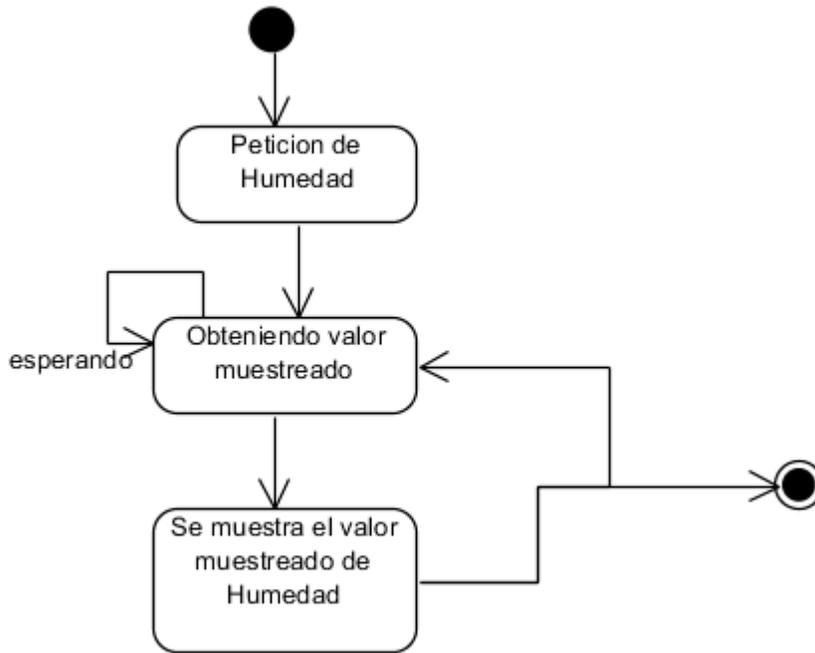


Figura 14: Diagrama de Estados Visualizar Variable de Humedad.

4.5.5 Modificar Estado de Actuador de Humedad.

El sistema se encuentra a la espera de una orden causada por algún valor de humedad fuera de rango, cuando es recibida el sistema modifica el estado del actuador, y envía la orden al actuador para modificar su estado físicamente.

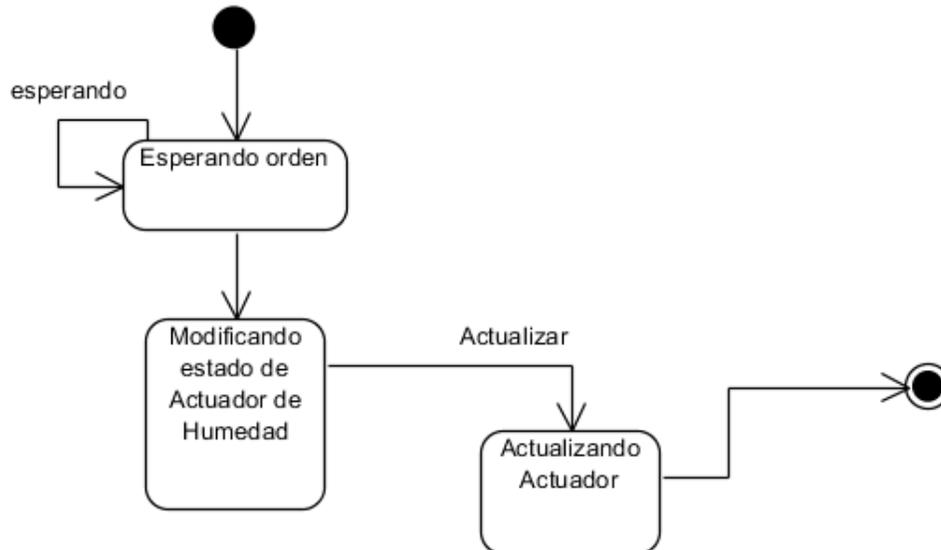


Figura 15: Diagrama de Estados Modificar Actuador de Humedad.

4.5.6 Modificar Sensor.

El sistema se encuentra a la espera que se seleccione un sensor. Luego que ha sido seleccionado se modifican sus atributos, y se valida que sean correctos. Si los atributos modificados son correctos se acepta la modificación. El sistema queda a la espera de otra modificación.

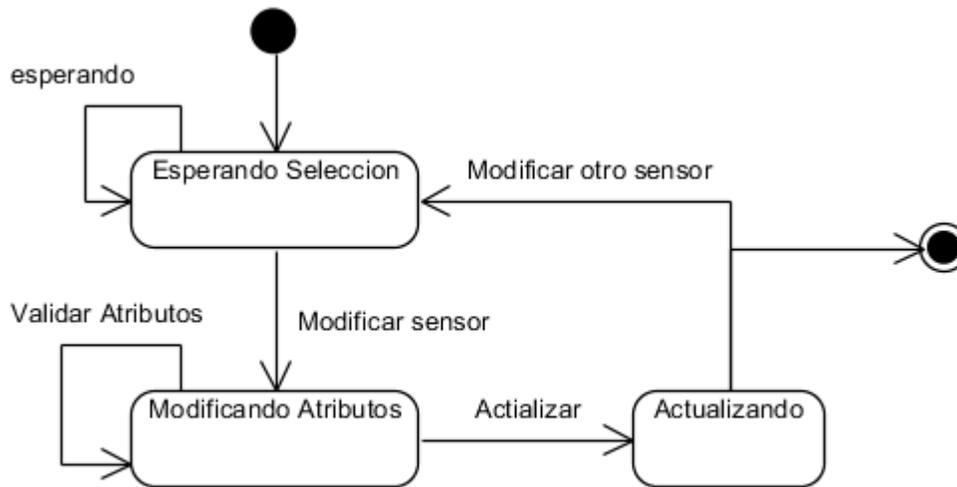


Figura 16: Diagrama de Estados Modificar Sensor.

4.5.7 Seleccionar Hortaliza.

El Sistema se encuentra en espera que se seleccione una hortaliza, se selecciona un grupo de sensores y se definen los parámetros límites según la hortaliza seleccionada, posteriormente se introduce el grupo de sensores al sistema.

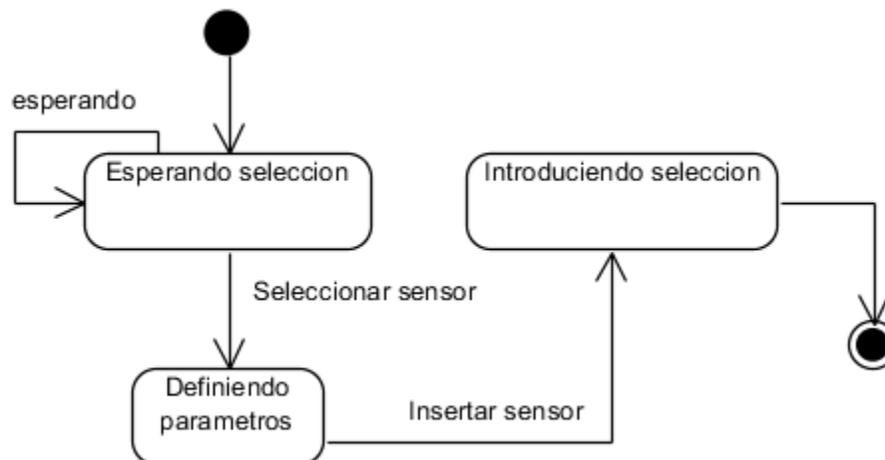


Figura 17: Diagrama de Estados Seleccionar Hortaliza.

4.5.8 Evaluar Variable de Temperatura.

El sistema recibe un valor muestreado por el sensor de temperatura, si el valor sobrepasa los límites se genera una orden la cual es enviada al actuador para cambiar su estado de manera física; si el valor no sobrepasa el rango establecido se evalúa un nuevo valor.

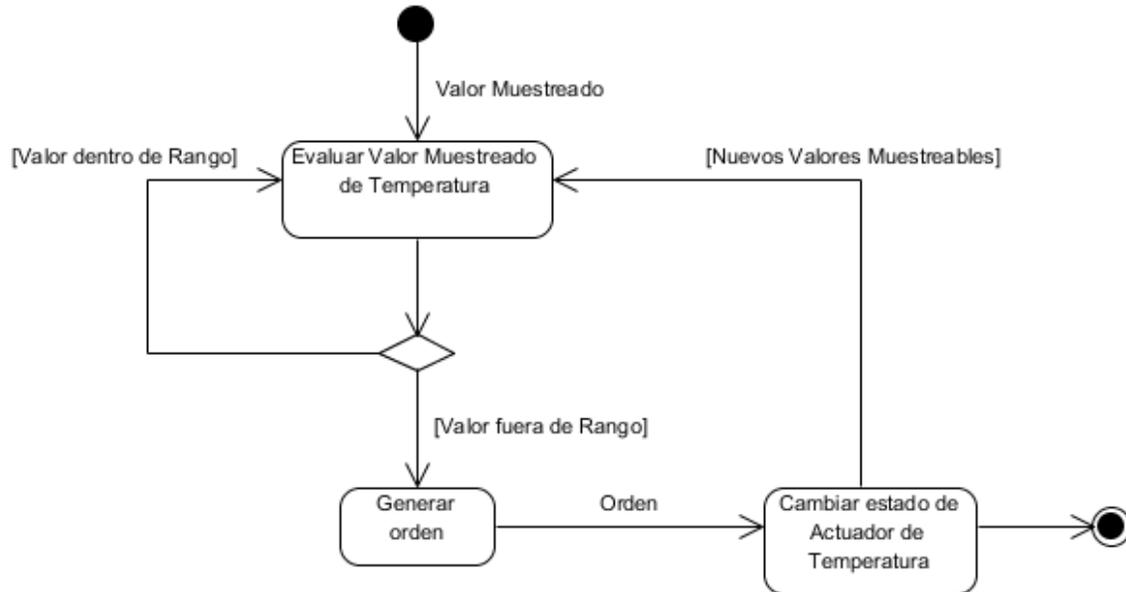


Figura 18: Diagrama de Estados Evaluar Variable de Temperatura.

4.5.9 Almacenar Datos Muestreados.

El sistema se encuentra en espera para recibir un valor muestreado por los sensores, se crea el schema para poder almacenar el dato en la base de datos. El dato se inserta en la base de datos para su posterior procesamiento.

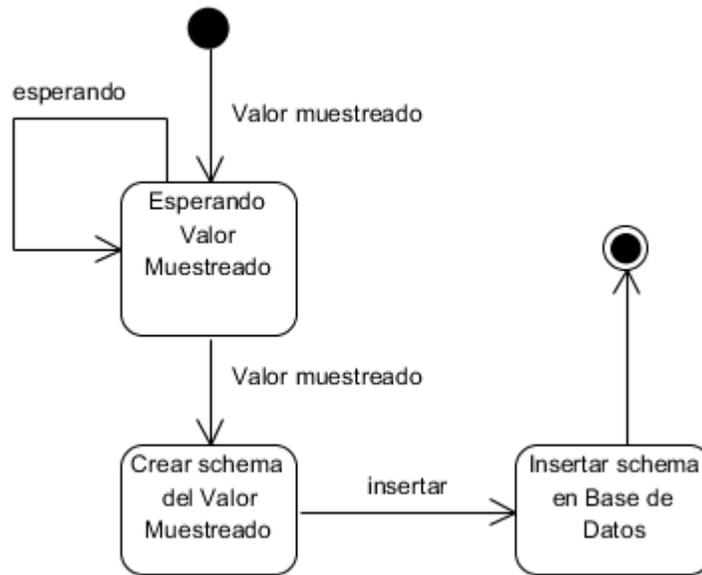


Figura 19: Diagrama de Estados Almacenar Datos Muestreados.

4.5.10 Evaluar Variable de Humedad.

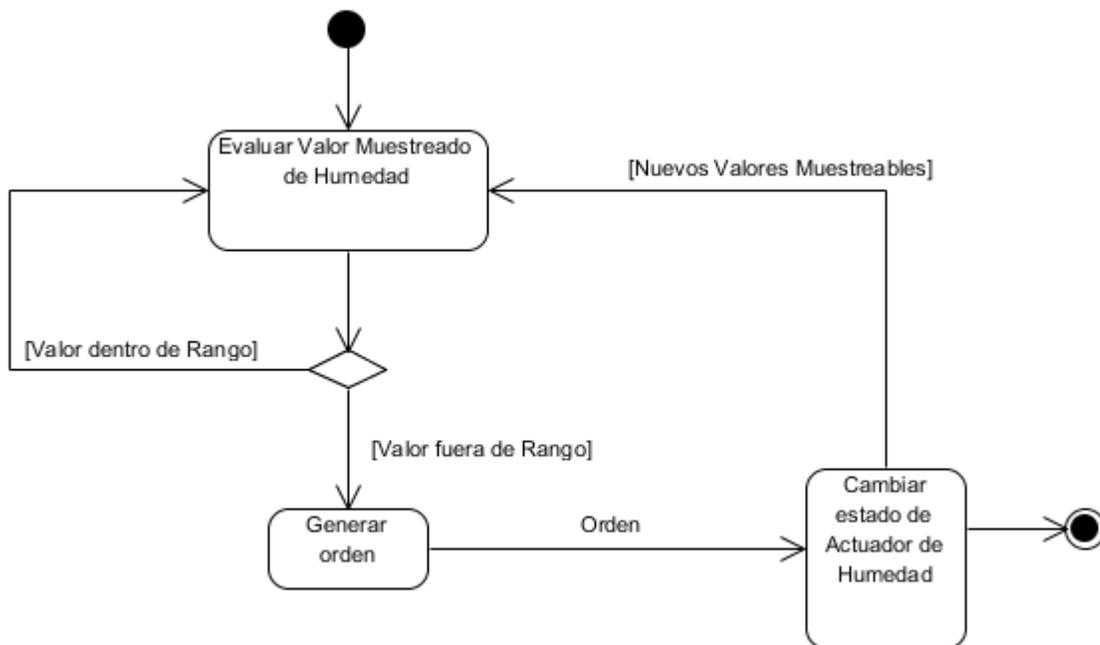


Figura 20: Diagrama de Estados Evaluar Variable de Humedad.

El sistema recibe un valor muestreado por el sensor de humedad, si el valor sobrepasa el los limites se genera una orden la cual es enviada al actuador para cambiar su estado de manera física; si el valor no sobrepasa el rango establecido se evalúa un nuevo valor.

4.6 Esquema de la base de datos.

A continuación se muestra el esquema para la base de datos; cabe aclarar que el sistema gestor MongoDB al tener un enfoque NoSQL sin esquema o con esquema dinámico nos permite enfocarnos en la aplicación y como esta accederá a la base de datos, por lo que las colecciones para la base de datos pueden ser embebidas dentro de otras colecciones de manera que favorezcan a la lectura de documentos en la base de datos.

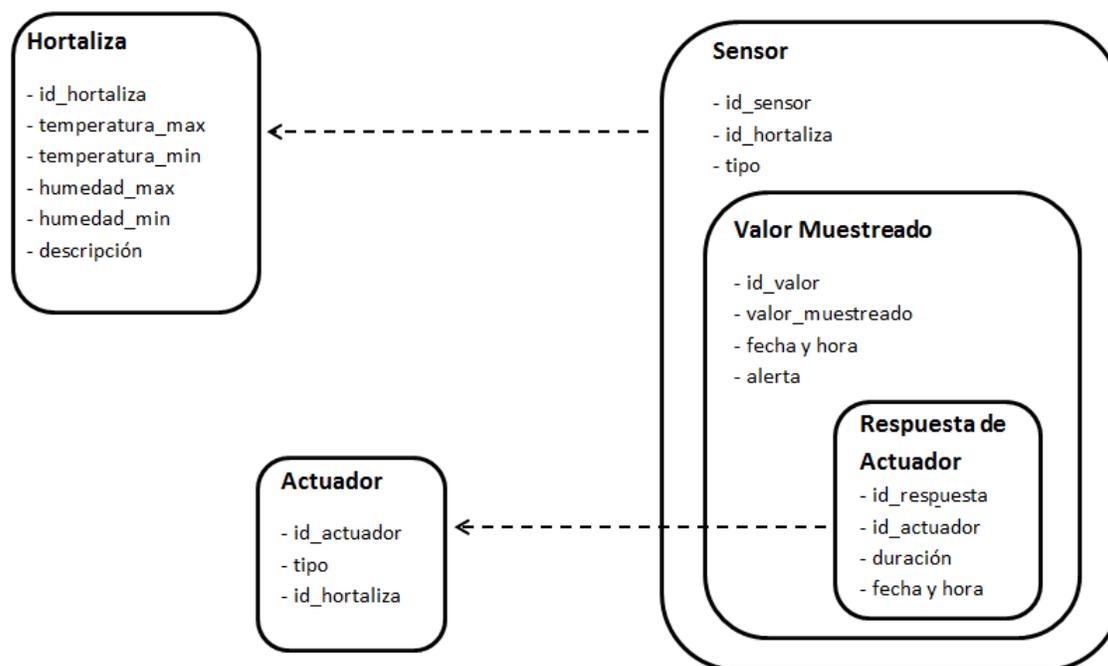


Figura 21: Esquema de la base de datos.

Dicho esto se describe el diagrama:

- Habrá una colección para los tipos de “Hortaliza”, ella contendrá la temperatura máxima y mínima, humedad máxima y mínima y una breve descripción de la misma.
- Una colección para los “Actuadores”, donde solo se almacenará el identificador de actuador, el tipo y el id de la hortaliza donde se encuentra.
- La colección más importante es donde son almacenados los “Sensores”, los atributos propios del sensor serán: el identificador del sensor, el tipo y el id de la hortaliza donde se encuentra; contendrá una colección embebida para los “Valores Muestreados” por el sensor cuyos atributos serán el identificador del valor muestreado, el valor propiamente, la fecha de muestreo y una alerta cuando se sobrepase un valor mínimo o máximo de la hortaliza que monitorea; esta colección a su vez contendrá otra colección embebida cuando una “Respuesta de Actuador”

sea generada cuyos atributos son: el identificador de respuesta, el id de actuador, la duración de la repuesta y la fecha en que fue realizada.

4.7 Diseño preliminar de la interfaz.

A continuación se presenta el diseño preliminar de la interfaz, cabe destacar que dicho diseño puede ser modificado en un futuro, en base a nuevos requerimientos encontrados.

4.7.1 Diseño de la pantalla inicial.

Esta pantalla contendrá las hortalizas que están siendo monitoreadas por el sistema, una descripción de la hortaliza y los valores límites de la misma.

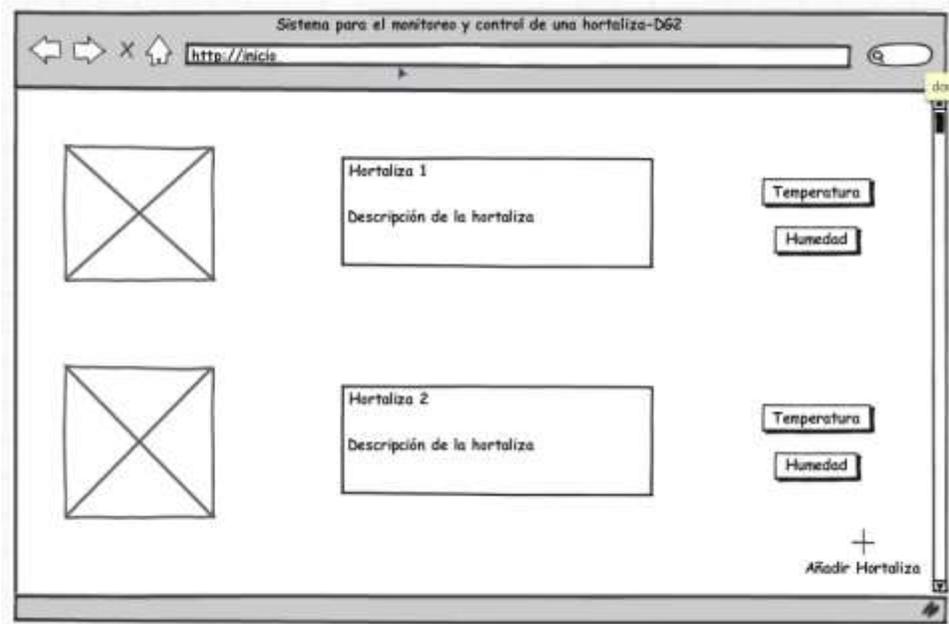


Figura 22: Pantalla Inicial.

4.7.2 Diseño de la pantalla de Temperatura de Hortaliza.

Al pulsar en el botón “Temperatura” de alguna hortaliza en la pantalla principal se desplegara la siguiente pantalla, la cual contiene una .grafica donde se mostraran los valores registrados por el sensor de temperatura, en otra tabla serán mostradas las respuestas del actuador ante valores fuera de rango.

Además habrá una opción para configurar el tiempo que tarda el sistema en tomar valores de temperatura para procesar y un botón para activar el actuador a consideración del usuario.

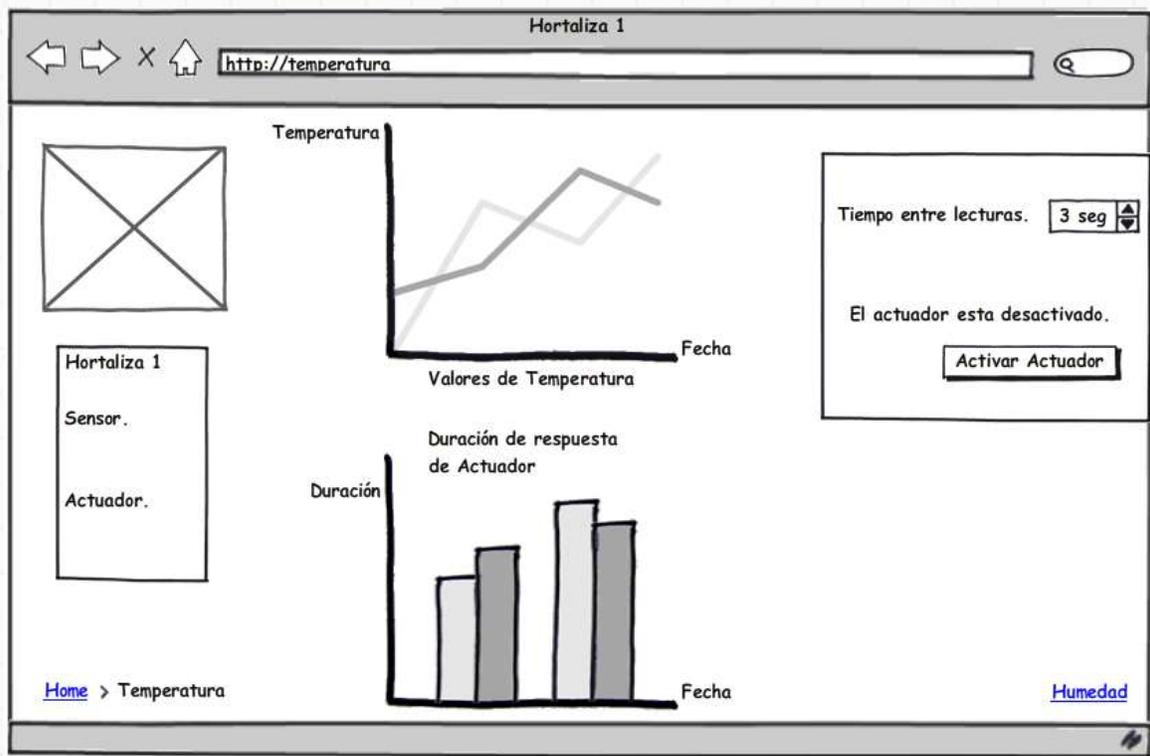


Figura 23: Pantalla Temperatura de Hortaliza.

4.7.3 Diseño de la pantalla de Humedad de la Hortaliza.

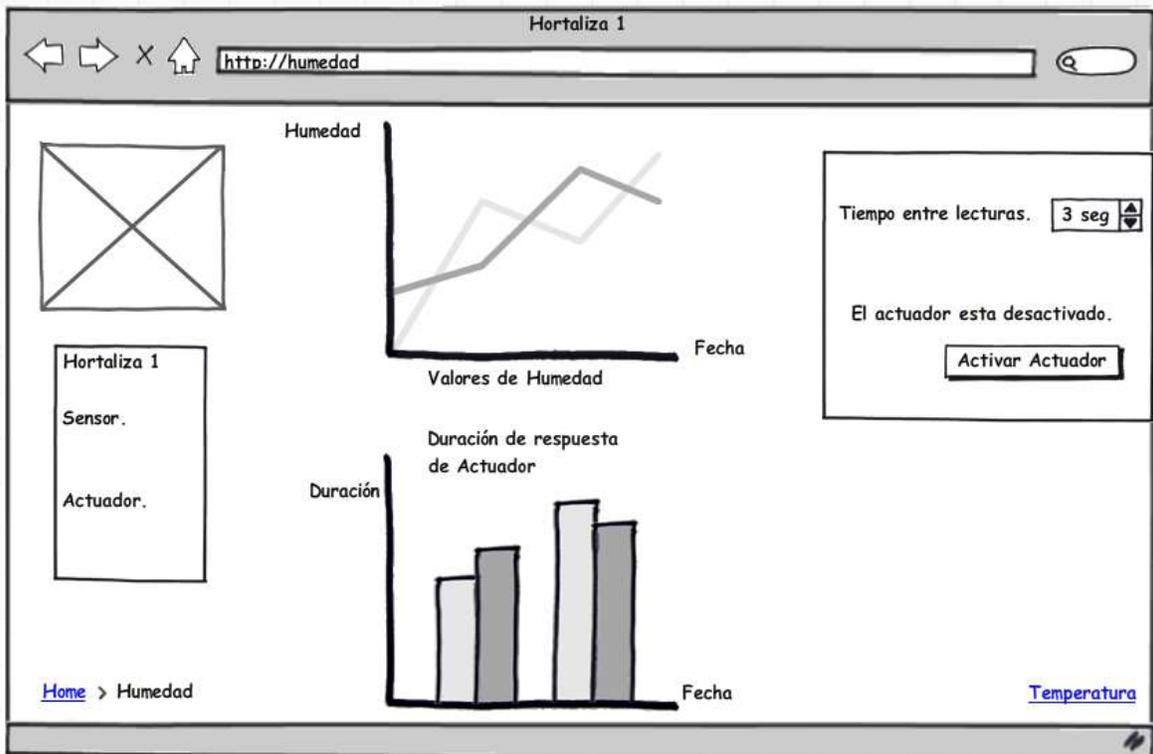


Figura 24: Pantalla Humedad de la Hortaliza.

De la misma manera que en la pantalla de Temperatura en la pantalla de humedad se despliegan dos graficas con los valores de humedad registrados y las respuestas del actuador.

4.7.4 Diseño de la pantalla Editar/Añadir Hortaliza.

En esta pantalla puede ser modificado el tipo de hortaliza que se está siendo monitoreada por el sensor, eligiendo de entre los tipos almacenados en la base de datos.

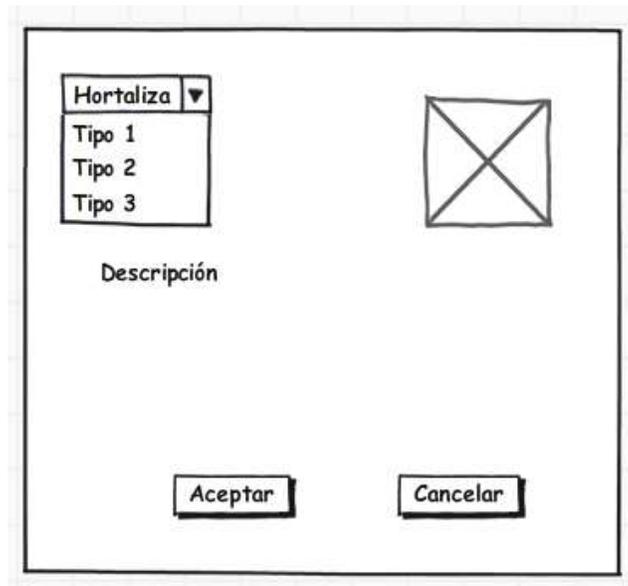


Figura 25: Pantalla Editar/Añadir Hortaliza.

4.8 Pruebas del sistema.

Las pruebas aplicadas al prototipo inicial fueron divididas en dos apartados, el primero contempla las pruebas desde los actuadores hacia el módulo de control y el segundo: las pruebas desde el módulo de control hacia los actuadores.

Las lecturas realizadas fueron hechas con un multímetro digital de 3 dígitos el cual posee una sensibilidad de 1mV.

4.8.1 Pruebas Sensor - Módulo de control.

El Convertidor analógico-digital que nuestra tarjeta Raspberry Pi B+ tiene una resolución de 10 bits y utiliza un valor de 5 volts como el valor de referencia más grande que puede recibir de un sensor, pero nuestro sensor entrega valores de entre 0 y 1.5 volts, esto nos hace perder un 80% del rango posible. Cambiar el valor de 5 volts a 1.5 nos asegurara la mejor resolución posible.

La ecuación utilizada para digitalizar el valor muestreado de temperatura es la siguiente:

$$temperatura = \frac{(5.0 * valor_muestreado * 100)}{1024}$$

Ecuación 1: Cálculo de Temperatura original.

Esta ecuación proviene de obtener el valor muestreado y determinar el porcentaje que tiene en el rango del ADC (1024), multiplicando este valor por el valor de referencia (5.0 volts) y dividiéndolo entre 10 mili volts que es el factor de escala por grado centígrado según la hoja de especificaciones del sensor.

Si utilizamos 1.1 volts como valor de referencia, y lo dividimos entre 1024 que es el rango del ADC obtendríamos que cada cambio en el valor muestreado equivale aproximadamente a 1.4648 mV. Si 10 mV equivalen a 1° C, $10 / 1.4648 \sim 6.079$. De esta manera obtenemos que cada cambio de 9.31 en el valor analógico, hay un cambio de 1° C de temperatura.

$$temperatura = \frac{valor_muestreado}{6.079}$$

Ecuación 2: Cálculo de Temperatura a 1.1v de Referencia.

Al usar el valor de 1.5 como valor de referencia limitamos el rango del sensor de 0 a 150 grados centígrados el cual es el valor máximo de temperatura que el sensor es capaz de medir.

Para las pruebas fueron realizadas 10 series con 10 mediciones cada una, dichas mediciones fueron del valor analógico entregado por el sensor y el valor registrado por el sistema, en la Tabla 31 se muestra la media de las 10 series de prueba y en la Figura 25 se muestra la gráfica de dichos valores.

Muestreo	Medición (mV)	Valor Registrado (°C)	Temperatura Teórica (°C)
1	52	5.2	5
2	99	9.9	10
3	151	15.0	15
4	199	19.8	20
5	249	25.0	25
6	303	30.4	30
7	356	35.3	35
8	402	40.3	40
9	449	44.9	45
10	500	50.1	50

Tabla 31: Pruebas de muestreo del sensor LM35.

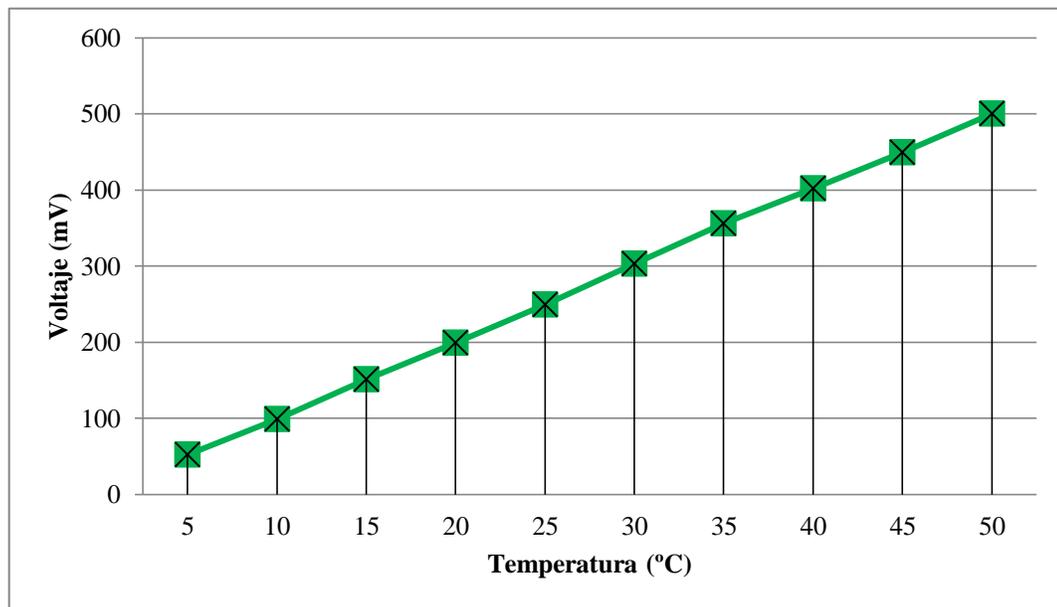
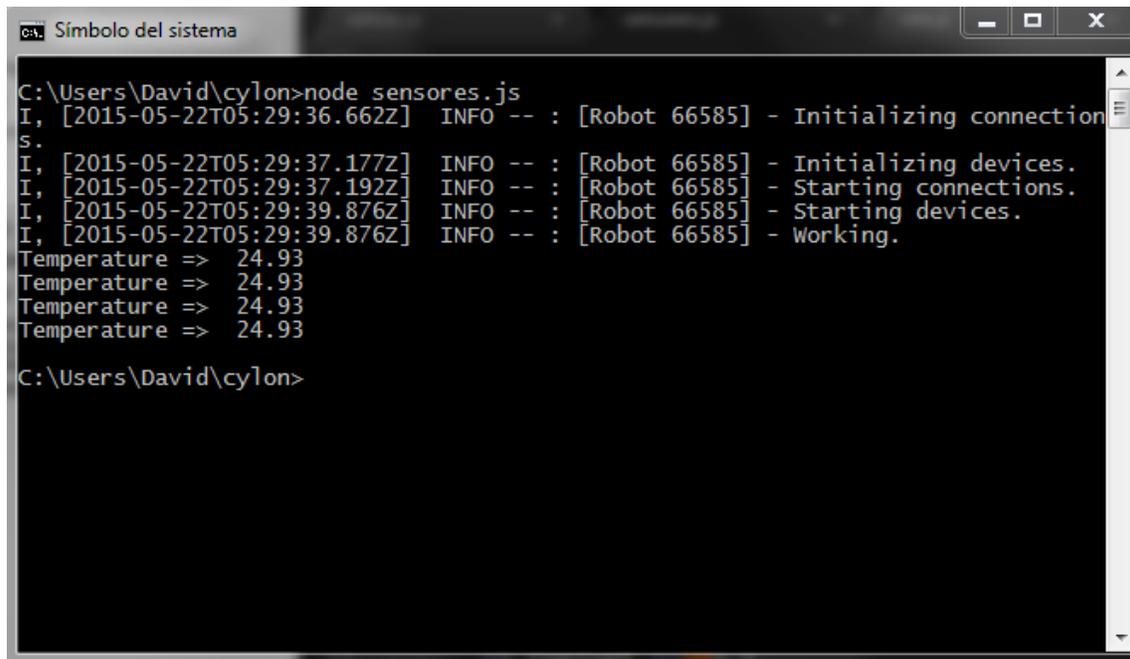


Figura 26: Gráfica de pruebas del sensor de temperatura.

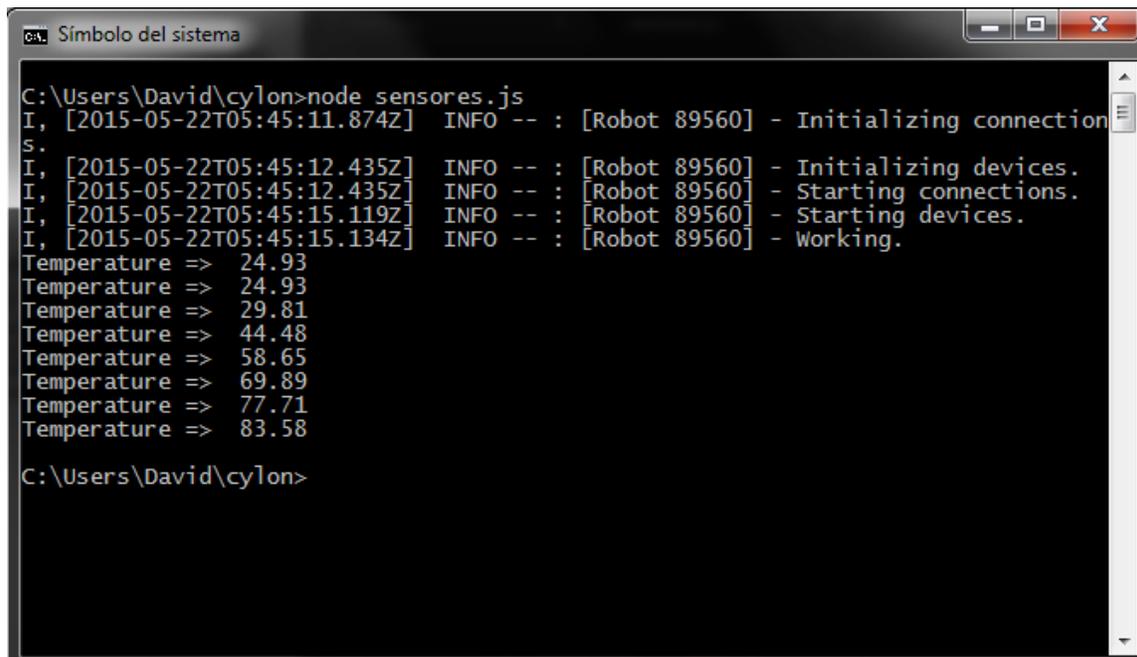
En las Figuras 26 y 27 se muestran capturas de pantalla del funcionamiento del sensor de temperatura desde la consola de comandos de Node.js.

En la Figura 26, se realiza el muestreo de variables a temperatura ambiente; la Figura 26 muestra las mediciones sometiendo el sensor a fuego directo con un intervalo de 2 segundos.



```
C:\Users\David\cylon>node sensores.js
I, [2015-05-22T05:29:36.662Z] INFO -- : [Robot 66585] - Initializing connection
s.
I, [2015-05-22T05:29:37.177Z] INFO -- : [Robot 66585] - Initializing devices.
I, [2015-05-22T05:29:37.192Z] INFO -- : [Robot 66585] - Starting connections.
I, [2015-05-22T05:29:39.876Z] INFO -- : [Robot 66585] - Starting devices.
I, [2015-05-22T05:29:39.876Z] INFO -- : [Robot 66585] - Working.
Temperature => 24.93
Temperature => 24.93
Temperature => 24.93
Temperature => 24.93
C:\Users\David\cylon>
```

Figura 27: Valores muestreados a temperatura ambiente.



```
C:\Users\David\cylon>node sensores.js
I, [2015-05-22T05:45:11.874Z] INFO -- : [Robot 89560] - Initializing connection
s.
I, [2015-05-22T05:45:12.435Z] INFO -- : [Robot 89560] - Initializing devices.
I, [2015-05-22T05:45:12.435Z] INFO -- : [Robot 89560] - Starting connections.
I, [2015-05-22T05:45:15.119Z] INFO -- : [Robot 89560] - Starting devices.
I, [2015-05-22T05:45:15.134Z] INFO -- : [Robot 89560] - Working.
Temperature => 24.93
Temperature => 24.93
Temperature => 29.81
Temperature => 44.48
Temperature => 58.65
Temperature => 69.89
Temperature => 77.71
Temperature => 83.58
C:\Users\David\cylon>
```

Figura 28: Valores muestreados a fuego directo.

El sensor de humedad cuenta con una salida analógica que entrega valores de entre 0 y 5 volts y una salida digital con resolución de 10 bits que entrega 1111111111 (1023) en su valor más bajo, cuando no existe presencia de agua y 0000000000 (0) cuando se encuentra sumergido totalmente en agua. Usando la salida digital, utilizamos la siguiente ecuación (**Ecuación 3**) para calcular la el porcentaje de humedad relativa:

$$humedad = \frac{(1023 - valor_muestreado) 100}{1023}$$

Ecuación 3: Calculo de Humedad Relativa.

Podemos hacer nuestra propia escala marcando pautas de 256 (0100000000) entre el valor de 1023 (1111111111) y 0 (0000000000), de igual manera que para el sensor de temperatura, la Tabla 33 muestra la media obtenida de 10 series de pruebas del valor analógico obtenido del sensor y el valor digital que registro el sistema.

Muestreo	Medición (V)	Valor Binario	Valor Decimal Registrado	Humedad Relativa Teórica (%)
1	0.952	1111111111	1023	0
2	1.111	1111001011	971	5
3	1.265	1110011000	920	10
4	1.418	1101100101	869	15
5	1.572	1100110010	818	20
6	1.725	1011111111	767	25
7	1.879	1011001100	716	30
8	2.032	1010011000	664	35
9	2.186	1001100101	613	40
10	2.339	1000110110	562	45
11	2.493	0111111111	511	50
12	2.646	0111001100	460	55
13	2.8	0110011001	409	60
14	2.953	0101100110	358	65
15	3.107	0100110010	306	70
16	3.26	0011111111	255	75
17	3.414	0011001100	204	80
18	3.567	0010011001	153	85
19	3.765	0001100110	102	90
20	3.965	0000110011	51	95
21	4.163	0000000000	0	100

Tabla 32: Serie de pruebas sensor de humedad.

La Figura 28 muestra la gráfica de la media obtenida de las 10 series de pruebas.

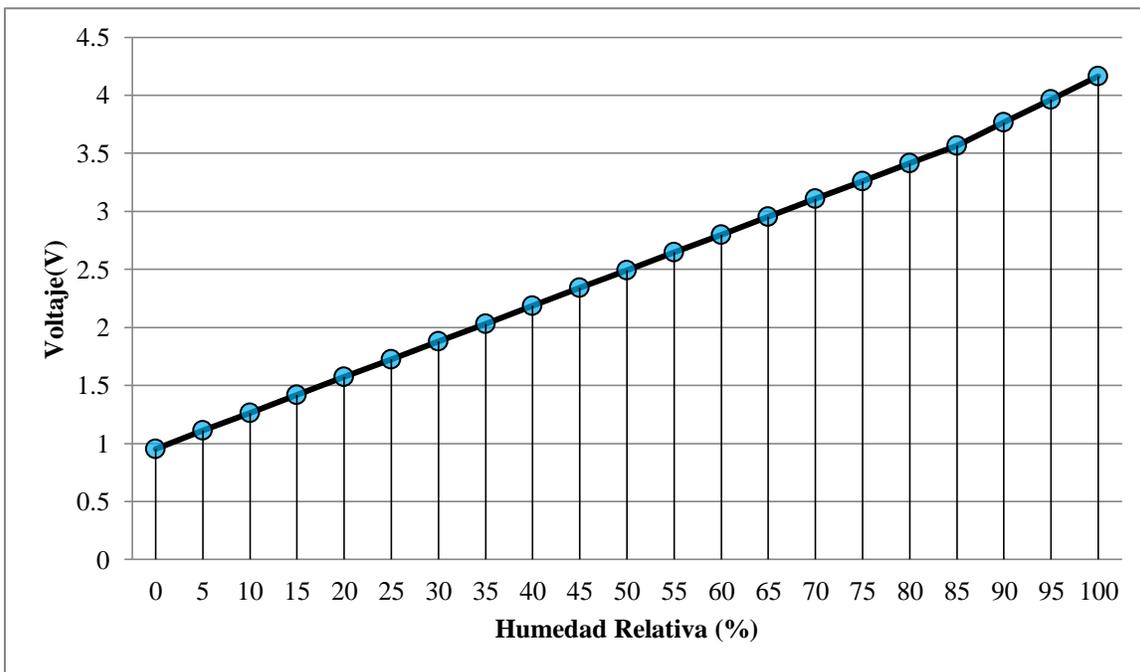


Figura 29: Grafica de pruebas del sensor de humedad.

En la Figura 29 se muestra el funcionamiento del sensor de humedad desde la consola de comandos de Node.js, el valor inicial corresponde al valor muestreado por el sensor cuando este se encuentra sin presencia absoluta de agua, posteriormente se puede observar un cambio drástico en el porcentaje entregado cuando el sensor es sumergido totalmente en agua, el tiempo entre la toma del valor muestreado es de 5 segundos.

```

C:\Users\David\cylon>node sensores.js
I, [2015-05-22T06:38:32.745Z] INFO -- : [Robot 88063] - Initializing connection
s.
I, [2015-05-22T06:38:33.135Z] INFO -- : [Robot 88063] - Initializing devices.
I, [2015-05-22T06:38:33.135Z] INFO -- : [Robot 88063] - Starting connections.
I, [2015-05-22T06:38:35.834Z] INFO -- : [Robot 88063] - Starting devices.
I, [2015-05-22T06:38:35.834Z] INFO -- : [Robot 88063] - Working.
Hummidity => 0.00
Hummidity => 0.00
Hummidity => 0.00
Hummidity => 100.00
Hummidity => 99.53
Hummidity => 99.42
Hummidity => 99.53
Hummidity => 99.42
C:\Users\David\cylon>

```

Figura 30: Valores muestreados de Humedad Relativa.

4.8.2 Módulo de control – Actuadores.

Como se especificó en el apartado de los relés estos son un módulo genérico útil para energizar cualquier actuador, sea de corriente alterna o corriente directa.

Las pruebas realizadas consistieron en calcular el tiempo de respuesta que tiene el relé al ser activado y el número de fallas que el relé presentó.

Se realizaron 20 activaciones y 20 desactivaciones para cada uno de los relés midiendo el tiempo de respuesta de cada uno; en la Tabla 33 se muestra el promedio de respuesta de cada uno de los relés.

	Tiempo de Activación	de Tiempo de desactivación	de No. De Fallas.
Actuador 1	0.4 segundos	0.5 segundos	0
Actuador 2	0.6 segundos	0.4 segundos	0

Tabla 33: Pruebas de tiempo de los Relés.

Capítulo V

5.1 Implementación de la Base de Datos.

La base de datos fue implementada con MongoDB, una base de datos no relacional que utiliza archivos JSON [19] (JavaScript Object Notation); cada uno de estos registros tiene un tamaño no mayor a 2 KB y el tamaño máximo para una colección de datos es de 16MB por lo que nos permite guardar aproximadamente 800,000 registros. Si el sistema crea un registro cada 10 minutos con los valores registrados, la base de datos creara 144 registros en un día; basándonos en este cálculo el sistema es capaz de almacenar los registros de 5555 días.

La finalidad de contar con una base de datos para el almacenamiento de registros es en gran medida para usarla como un histórico acerca de las mediciones y las acciones que el sistema registra y realiza para el cuidado adecuado de la hortaliza; de esta manera dichos registros pueden ser utilizados para replicar las acciones tomadas por el sistema y así generar de manera sistémica cuidados para una posterior generación de hortalizas en un entorno controlado donde factores ambientales tienen una menor fluctuación.

Un archivo JSON es un registro en formato de texto que es independiente de cualquier tipo de lenguaje de programación; está construido mediante una estructura clave valor, donde la primer estructura es una cadena de caracteres y la segunda puede ser desde una cadena de caracteres hasta un array de objetos JSON cada uno con su respectiva estructura.

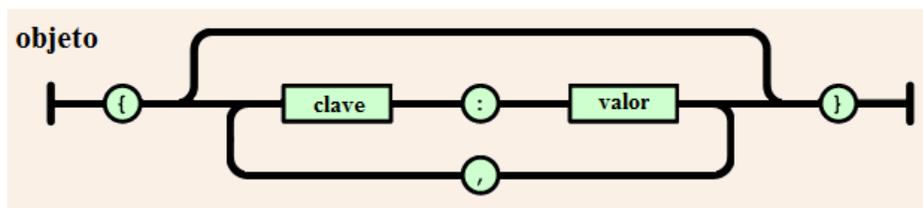


Figura 31: Estructura de un objeto JSON.

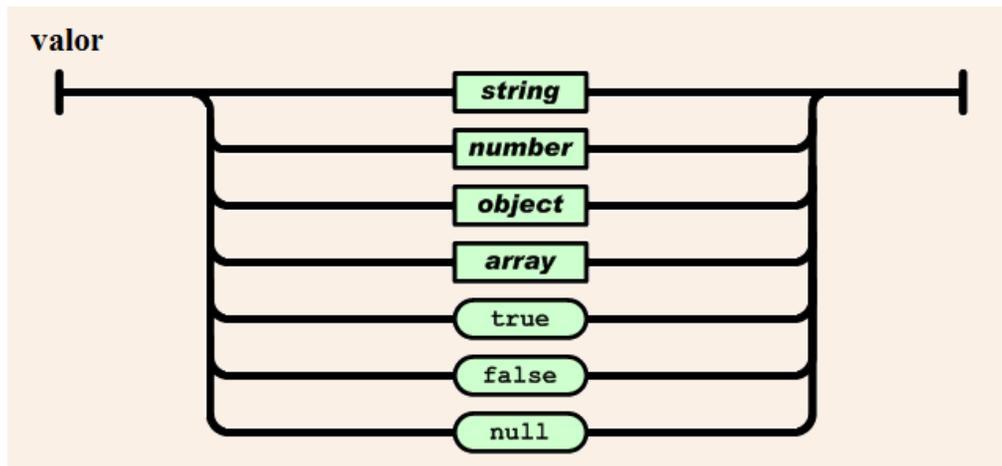


Figura 32: Tipos de valores admitidos.

Para el proyecto se utilizará el framework Mongoose [20] el cuál es un paquete de Node.js que permite crear los esquemas definidos en el capítulo 4.6 (Figura 21), la estructura para definirlo es la siguiente:

```
1
2 // Importación de Mongoose
3 var mongoose = require('mongoose');
4
5 // Conexión de Mongoose a MongoDB (ted/ted)
6 mongoose.connect
7 ('mongodb://ted:ted@ds061797.mongolab.com:61797/theenlighteneddeveloper',
8 function (error) {
9   if (error) {
10     console.log(error);
11   }
12 });
13
14 // Definición de Mongoose Schema
15 var Schema = mongoose.Schema;
16 var UserSchema = new Schema({
17   first_name: String,
18   last_name: String,
19   email: String
20 });
21
22 // Definición de un Modelo de Mongoose
23 var User = mongoose.model('users', UserSchema);
24
```

Figura 33: Definición de un esquema en Mongoose.

Cada uno de los atributos definidos en el esquema posee una clave y un valor que generará un objeto JSON en la base de datos.

Cada vez que sean ejecutados los archivos de esquema, se creará una colección de dicho tipo en la base de datos.

5.1.1 Modelos de Colecciones para la Base de Datos.

```
1 //models/index.js
2 if (!global.hasOwnProperty("db")) {
3
4   var mongoose = require("mongoose");
5   var dbName = "trabajoterminal"
6
7   global.db = {
8
9     mongoose: mongoose,
10    //models
11    Hortaliza: require("../hortaliza")(mongoose),
12    Actuador: require("../actuador"),
13    Sensor: require("../sensor"),
14    Value: require("../value"),
15    Activation: require("activation")
16  };
17 }
18
19 module.exports = global.db;
```

Figura 34: Colecciones para la Base de Datos.

En este archivo se definen todos los modelos (colecciones) que la base de datos creará; cada una tendrá su propia estructura en cuanto a atributos las cuales se muestran a continuación.

5.1.2 Colección Hortalizas.

```
1
2 module.exports = function(mongoose) {
3
4   var Schema = mongoose.Schema;
5   //mongoose model object
6   var HortalizaSchema = new Schema({
7
8     //name property : value
9     idHortaliza : String,
10    temp_max : int,
11    temp_min : int,
12    humm_max : int,
13    humm_min : int,
14    description : String,
15    plow : Date
16  });
17
18  return mongoose.model("Hortaliza", HortalizaSchema);
19 }
```

Figura 35: Esquema Hortaliza.

La colección Hortalizas, como se mencionó en el apartado 4.6 contendrá un identificador para poder ser diferenciada por los sensores y actuadores, así como valores máximos de temperatura y humedad que serán los parámetros por los cuales se realizará el monitoreo y control de la hortaliza, una breve descripción del tipo de hortaliza y la fecha en que se dio de alta.

5.1.3 Colección Actuador.

```
1 module.exports = function(mongoose) {
2
3   var Schema = mongoose.Schema;
4   //mongoose model object
5   var ActuadorSchema = new Schema({
6
7     //name property
8     idActuador : String,
9     tipo : String,
10    idHortaliza : String
11  });
12
13  return mongoose.model("Actuador", ActuadorSchema);
14 }
15 }
```

Figura 36: Esquema Actuador.

El esquema de actuador solo contendrá el identificador propio del sensor, el tipo de sensor que es y el identificador de la hortaliza donde se encuentra instalado.

5.1.4 Colección Sensor.

```
1 module.exports = function(mongoose) {
2
3   var Schema = mongoose.Schema;
4   //mongoose model object
5   var Sensorchema = new Schema({
6
7     //name property
8     idSensor : String,
9     idHortaliza : String,
10    type : String,
11    valueTaken : []
12  });
13
14  return mongoose.model("Sensor", SensorSchema);
15 }
```

Figura 37: Esquema Sensor.

La colección Sensor a diferencia de la colección Actuador contiene más parámetros ya que es junto con la colección Value las que tendrán arreglos de colecciones como atributos.

Esta colección contendrá el dos identificadores, uno propiamente para el sensor y otro para la hortaliza donde está instalado, su tipo y un arreglo de valores el cual será poblado con colecciones del tipo Value.

5.1.5 Colección Value.

```
1 module.exports = function(mongoose) {
2
3   var Schema = mongoose.Schema;
4   //mongoose model object
5   var ValueSchema = new Schema({
6
7     //name property
8     idSensor : String,
9     value : String,
10    type : String,
11    dateTaken : Date,
12    alert : boolean,
13    actuadorActivated : []
14  });
15
16  return mongoose.model("ValueSchema", ValueSchema);
17 }
```

Figura 38: Esquema Value.

La colección Value contiene el identificador del sensor donde fue generada, el valor y tipo de lectura, la fecha en la que fue generada, un campo del tipo boolean mediante el cual se determina si la lectura supera o no los valores límite de cada hortaliza. Esta colección también posee un arreglo de valores de Tipo Activation que se genera cuando un actuador es activado.

5.1.6 Colección Activation.

```
1 module.exports = function(mongoose) {
2
3   var Schema = mongoose.Schema;
4   //mongoose model object
5   var ActivationSchema = new Schema({
6
7     //name property
8     idRespuesta : String,
9     idActuador : String,
10    duration : int,
11    start : Date
12  });
13
14  return mongoose.model("ActivationSchema", ActivationSchema);
15 }
```

Figura 39: Esquema Activation.

El esquema Activation posee identificadores para la respuesta generada y el actuador que la generó atributo para determinar la duración de la respuesta y la fecha en la que fue generada.

5.2 Implementación del módulo de control.

5.2.1 Prototipo 1

El módulo de control fue desarrollado partiendo principalmente de la configuración de los pines de la tarjeta de desarrollo, con la ayuda de un protoboard para poder conectar los diferentes sensores y actuadores y así poder apreciar las acciones que el sistema realiza en base a la lectura que registra.

El alcance para el primer prototipo era realizar la conexión de los diferentes sensores de manera individual a la tarjeta de desarrollo, junto con la respuesta de los actuadores.

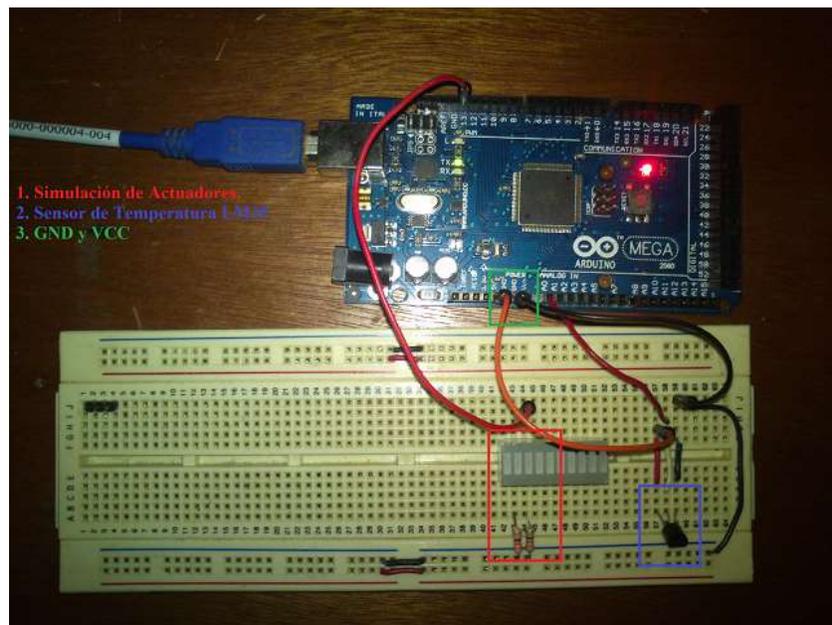


Figura 40: Prototipo 1 Simulado.

A continuación se presenta una simulación de este proceso mediante un sensor de temperatura; como se mencionó previamente en esta fase del proyecto los sensores y actuadores trabajan individualmente y no de manera conjunta como lo harán en el segundo prototipo, por lo que en esta simulación el actuador solamente se activa cuando la temperatura supera los 28°C.

```
38
Temperature => 18.57 °C
44
Temperature => 21.51 °C
44
Temperature => 21.51 °C
44
Temperature => 21.51 °C
50
Temperature => 24.44 °C
51
Temperature => 24.93 °C
55
Temperature => 26.88 °C
```

Figura 41: Registro de Temperatura.

En la **Figura 41** se puede observar el valor de la temperatura que registra el sistema cada 5 segundos y como esta no rebasa los 28°.

```
139
Temperature => 67.94 °C
Temperatura mayor a 28 grados, activando Actuador
110
Temperature => 53.76 °C
Temperatura mayor a 28 grados, activando Actuador
92
Temperature => 44.97 °C
Temperatura mayor a 28 grados, activando Actuador
81
Temperature => 39.59 °C
Temperatura mayor a 28 grados, activando Actuador
70
Temperature => 34.21 °C
Temperatura mayor a 28 grados, activando Actuador
62
Temperature => 30.30 °C
Temperatura mayor a 28 grados, activando Actuador
```

Figura 42: Simulación de Actuador.

En la **Figura 42** el valor de la temperatura supero los 28°C por lo que el sistema activo el actuador correspondiente.

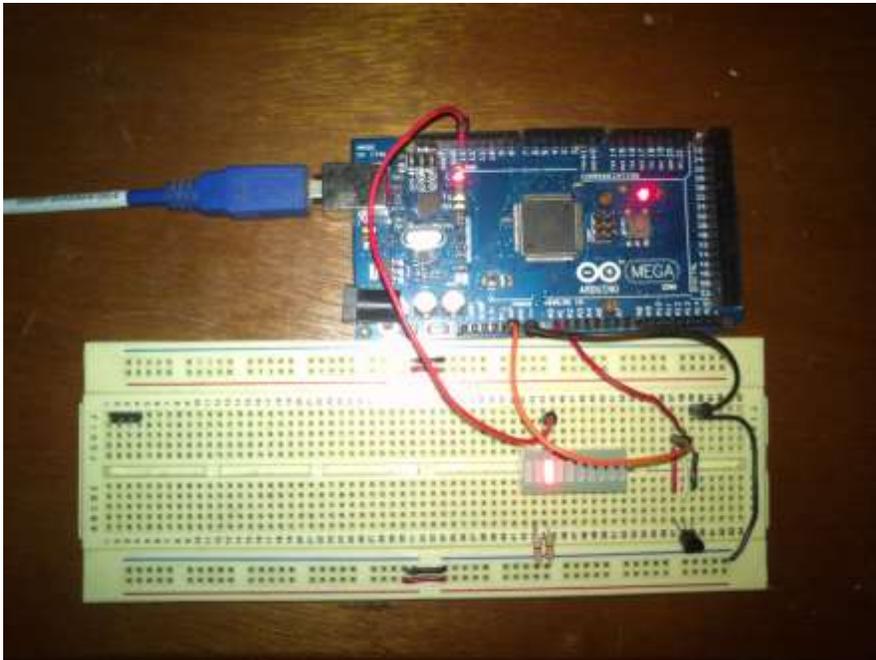


Figura 43: Prototipo 1 - Simulación de Actuador.

Cuando el valor de la temperatura regrese a un valor menor a 28° el sistema desactivará el actuador.

5.2.2 Prototipo 2

Para el prototipo 2, los sensores trabajan de manera conjunta y los actuadores se activan al cumplirse una serie de condiciones que son diferentes entre cada hortaliza, además el módulo de los sensores está elaborado en una placa fenólica.

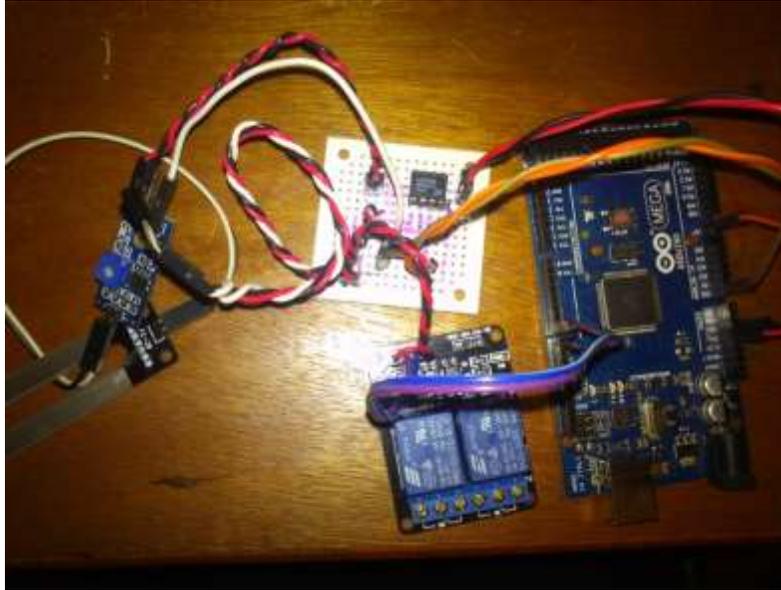


Figura 44: Prototipo 2

El sensor de Temperatura LM35 fue acoplado en un circuito de acondicionamiento para evitar interferencias con las lecturas de los otros sensores y además generar una salida constante entre 0 y 5 volts hacia la tarjeta de desarrollo. Para realizar dicho circuito de acondicionamiento se utilizó un amplificador operacional [21] LM358 en su configuración no inversora.

Se trata de un dispositivo electrónico que normalmente se presenta como circuito integrado que tiene dos entradas y una salida. La salida es la diferencia de las dos entradas multiplicada por un factor de ganancia.

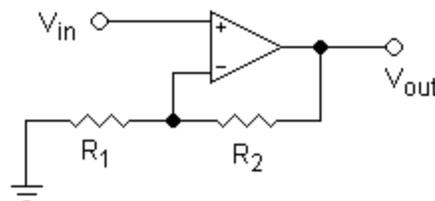


Figura 45: Amplificador Operacional - No Inversor.

5.3 Implementación de la Interfaz de Usuario.

Para la interfaz se ha optado por una interfaz intuitiva y de rápido acceso a todas las opciones posibles, esto con el fin de que sea más atractivo para los usuarios finales. La interfaz de usuario es únicamente un módulo de referencia donde se puede monitorear el funcionamiento del sistema. Al igual que el módulo de sensores y actuadores, el desarrollo de la interfaz está dividido en 2 prototipos.

5.3.1 Prototipo 1.

En este primer prototipo, se realizó una interfaz basada en los diseños preliminares mostrados en el **Capítulo 5.4**, la finalidad del primer prototipo fue para realizar todas las funcionalidades relacionadas con el monitoreo del sistema como conexión con la base de datos y despliegue de información; además de una gráfica que muestra el valor en tiempo real que está siendo sentido.

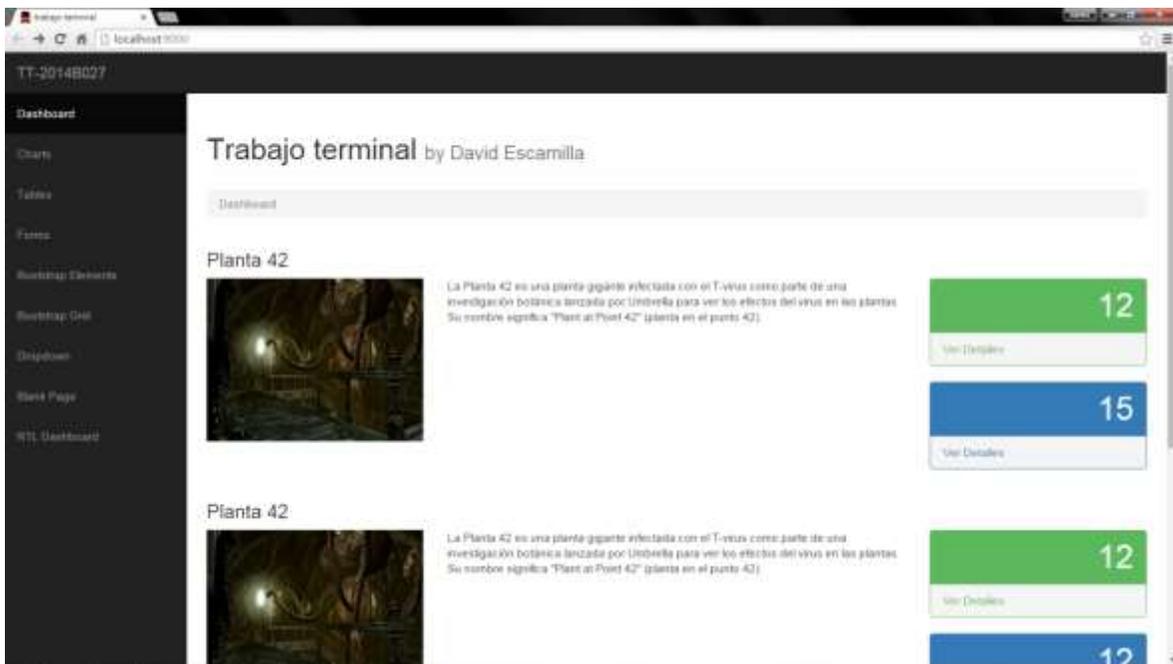


Figura 49: Prototipo 1 - Pantalla principal.

Como se puede observar, al iniciar el módulo, se presenta una lista con las diferentes hortalizas que están siendo monitoreadas por el sistema, una breve descripción y los parámetros límite en para cada una.

Además de dos bloques uno de color verde y uno de color azul correspondientes a la temperatura y a la humedad respectivamente, cada uno con un hipervínculo “Ver Detalles” que llevara al usuario a una nueva pantalla donde visualizar los datos de cada hortaliza.

Como este es un prototipo el menú lateral tiene una funcionalidad nula.

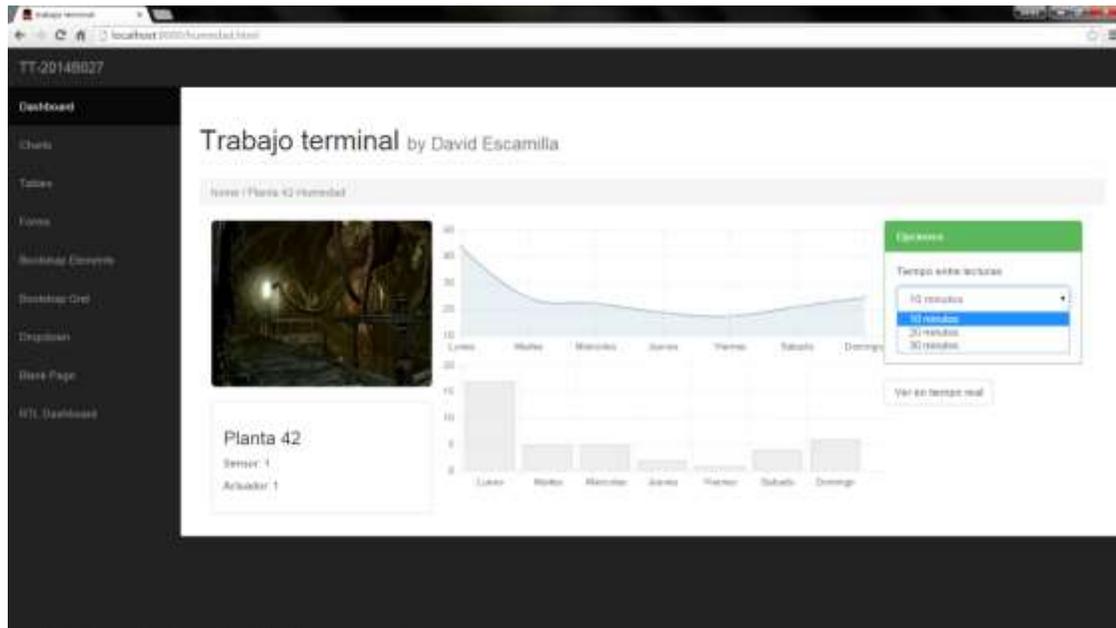


Figura 50: Prototipo 1 - Detalles de hortaliza.

En la **Figura 45** se puede observar la pantalla que despliega los detalles de temperatura, en ella se observan dos gráficas, la primera es una gráfica de línea que muestra el promedio de lecturas diarias que realizo el sistema.

La segunda, es una gráfica de barras donde se muestra el número de veces que el actuador vinculado a ese sensor fue activado.

Además se muestra un bloque donde se puede configurar el tiempo en que el sistema guarda un registro en la base de datos y un botón mediante el cual se genera un popup donde el usuario puede visualizar una gráfica de la lectura en tiempo real.

5.3.2 Prototipo 2

Ya que todas las funcionalidades primordiales de la interfaz de usuario fueron desarrolladas en el primer prototipo en la segunda fase de la implementación de la interfaz se elaboró un nuevo Look and Feel de todas la pantallas preliminares de la interfaz así como una navegación más intuitiva.



Figura 52: Prototipo 2 - Pantalla Principal

En la **Figura 47** se observa el rediseño de la interfaz de usuario, en esta etapa la base de datos posee una población de varios tipos de hortalizas, sin embargo solo la primera hortaliza tiene un juego de sensores y actuadores activo.



Figura 53: Prototipo 2 - Detalles de Hortaliza.

La pantalla que muestra los detalles de la hortaliza sigue mostrando una gráfica donde se muestra el promedio por día en una semana de las lecturas que el sistema registró; además se muestra unas tablas donde de igual manera de muestra el promedio de las lecturas que el sistema registró en una hora.

Capítulo VI

6.1 Trabajo a futuro.

Ya que el sistema es altamente escalable nuevos módulos de sensores y actuadores pueden ser instalados sin que el sistema sacrifique o limite la funcionalidad que brinda actualmente, de esta manera se pueden acondicionar e instalar nuevos sensores para poder brindar un mejor control sobre las variables del entorno; también se puede generar un recinto donde las variables físicas estén bajo un mayor control para que de esta manera se puedan proporcionar mejores condiciones de crecimiento.

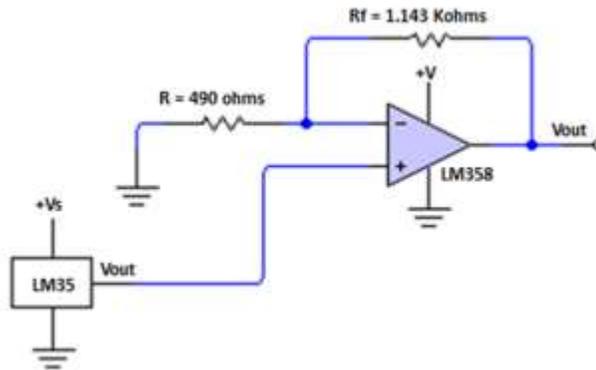
6.2 Conclusiones.

Se desarrolló un prototipo que es capaz de gestionar condiciones adecuadas para el desarrollo de una hortaliza. Durante el desarrollo del proyecto algunos requisitos fueron añadidos, sin embargo gracias a la metodología utilizada, dichas adiciones fueron implementadas de manera satisfactoria sin que el sistema sacrificara la funcionalidad que ofrecía hasta el momento. Por lo cual se puede concluir que la metodología utilizada para el desarrollo fue la correcta.

Los materiales utilizados para el prototipo entregable fueron elegidos como un suplemento funcional de los materiales óptimos.

Para desarrollar el proyecto se fueron necesarios gran parte de los conocimientos adquiridos a lo largo de carrera tanto del área de electrónica como del área de programación ya que este es un prototipo híbrido.

Anexo



Como se puede observar, el voltaje se aplica al pin positivo, ya que se conoce que la ganancia del amplificador operacional es muy grande, el voltaje en el pin positivo es igual al voltaje en el pin negativo y positivo, conociendo el voltaje en el pin negativo podemos calcular la relación que existe entre el voltaje de salida con el voltaje de entrada haciendo uso de un pequeño divisor de voltaje.

$$V_{out} = V_{in} \left(1 + \frac{R_2}{R_1}\right)$$

Ecuación 4: Voltaje de salida para un Amplificador en configuración no inversora.

Para poder calcular la ganancia de a aplicar en el amplificador operacional tenemos que:

$$Av = \frac{V_{out}}{V_{in}}$$

Ecuación 5: Ganancia de Voltaje.

Al despejar V_{in} de la Ecuación 4 obtenemos:

$$Av = \frac{V_{out}}{V_{in}} = \frac{R_2}{R_1} + 1$$

Ecuación 6: Ganancia de Voltaje.

El voltaje de salida máximo del circuito de acondicionamiento es de 5 volts ($V_{in} = 5$), mientras que el voltaje máximo entregado por el sensor es de 1.5 volts ($V_{out} = 5$) por lo tanto.

$$Av = \frac{V_{out}}{V_{in}} = \frac{5}{1.5} = 3.3333$$

Ecuación 7: Resultado de la Ganancia de Voltaje.

Al sustituir el resultado de la **Ecuación 7** en la **Ecuación 6** y despejar R_2 y proponiendo un valor de 499Ω para R_1 obtenemos el valor de ambas resistencias para así completar el circuito de acondicionamiento.

$$R_2 = (3.3333 - 1)(R_1)$$

$$R_2 = (3.3333 - 1)(490) = 1143.317 \Omega = 1.143 \text{ K}\Omega$$

Bibliografía

- [1] [En línea]. Available: http://ucanr.org/blogs/VictoryGrower_Blog/. Para ver un ejemplo publicitaria de la época acceder al vídeo <http://www.vegetablegardener.com/item/8121/video-victory-garden-1942-edition>.
- [2] SHARP, «SHARP,» 2013. [En línea]. Available: <http://sharp-world.com/corporate/news/130920.html>. [Último acceso: 25 Marzo 2015].
- [3] «The Food and Agricultural Organization,» [En línea]. Available: <http://www.fao.org/home/en/>.
- [4] «Parrot,» [En línea]. Available: <http://www.parrot.com/es/productos/flower-power/>.
- [5] R. S. Pressman, Ingeniería del Software Un enfoque practico, Connecticut: The McGraw-Hill, 2010.
- [6] J. MAROTO, Elementos de Horticultura General, Mundi-Prensa, 2000, p. 424.
- [7] Z. Serrano, Ministerio de Agricultura, Pesca y Alimentación., Madrid, 1990.
- [8] «Manual de Agricultura Urbana "Huerto en Casa",» [En línea]. Available: www.huertoencasa.mx.
- [9] «Compañía Mexicana proveedora de equipo de Instrumentacion y Telemetría,» [En línea]. Available: <http://www.ampere.com.mx/veris/relevador.php>.
- [10] A. P. Malvino, Principios de Electrónica., McGraw-Hill, 2000.
- [11] «Wiring.co,» [En línea]. Available: <http://www.wiring.org.co>.
- [12] «Node.js,» [En línea]. Available: <https://nodejs.org/about/>.
- [13] «Git Hub,» [En línea]. Available: <https://github.com/firmata/protocol>.
- [14] «Arduino.cc,» [En línea]. Available: <http://arduino.cc/en/Main/Products>.
- [15] «Raspberry Pi,» [En línea]. Available: <http://www.raspberrypi.org/products/model-b-plus/>.
- [16] «Teseel.io,» [En línea]. Available: <https://tessel.io/>.
- [17] Análisis y Diseño de Sistemas, McGraw Hill.
- [18] S. McConnell, Desarrollo y gestión de Proyectos Informáticos, Microsoft Press, 1997.

[19] «JSON,» [En línea]. Available: <http://www.json.org/>. [Último acceso: Julio 2015].

[20] «NPM,» [En línea]. Available: <https://www.npmjs.com/package/mongoose>. [Último acceso: Julio 2015].

[21] [En línea]. Available: http://www.electronics-tutorials.ws/opamp/opamp_1.html.