

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica



**Sistema electrónico para el control de riego y adquisición de datos en la
monitorización de temperatura, humedad y luminosidad de un
invernadero.**

**Informe de Proyecto de Graduación para optar por el título de
Ingeniero en Electrónica con el grado académico de Licenciatura**

Claudio Carvajal González

Cartago, Noviembre de 2015

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA ELECTRÓNICA

PROYECTO DE GRADUACIÓN

ACTA DE APROBACIÓN


Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. Juan Carlos Jiménez Robles

Profesor lector



Ing. Hayden Phillips Brenes

Profesor lector



Ing. Gabriela Ortiz León

Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Escuela Ingeniería Electrónica 20/11/2015
Lugar y fecha de la presentación

Declaratoria de autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, Costa Rica



Claudio Carvajal González

Céd: 206790934

Resumen

El Centro de Investigación en Biotecnología “CIB” es un grupo de investigación conformado por profesores del área de la biología. El centro se encuentra ubicado al frente de la Escuela de Ingeniería Ambiental dentro del Instituto Tecnológico de Costa Rica. Actualmente cuenta con un invernadero de investigación para el estudio de diversas especies, los profesores a cargo de estos proyectos se enfrentan con el problema de que no existe un sistema en el cual se puedan programar los intervalos de riego en las mesas/secciones de trabajo.

El CIB ha decidido realizar un sistema que permita programar el riego por mesas y secciones, además que genere un historial de las condiciones ambientales del lugar dadas por las variables temperatura, humedad y luminosidad. El prototipo desarrollado en este proyecto de graduación se va a implementar a futuro en un nuevo invernadero que se pretende construir atrás de las instalaciones del centro de investigación, ya se tiene establecido el terreno para tal fin.

Este sistema electrónico desarrollado toma en cuenta la cantidad de mesas a instalar en el nuevo invernadero para diseñar una red de sensores que se distribuyen por las 8 mesas y 2 secciones de trabajo, con el fin de monitorizar en más de un punto del invernadero la temperatura, humedad y luminosidad.

Palabras clave: BeagleBone Black, Humedad, Temperatura, Luminosidad, Sistema embebido, Linux, Monitorización, Base de Datos MySQL, Servidor web.

Abstract

The Biotechnology Research Center "CIB" is a research group formed by professors of biology. The center is located in front of the School of Environmental Engineering in the Technological Institute of Costa Rica. It currently has a research greenhouse for studying various species, teachers in charge of these projects are faced with the problem that there is no system which can be programmed watering intervals on tables working.

The CIB has decided to make a system that allows scheduling irrigation tables and sections, besides generating a history of environmental conditions at the place given by the variables temperature, humidity and light. The prototype developed in this graduation project will be implemented in the future in a new greenhouse to be constructed behind the facilities of the research center, it has set the stage for this purpose.

This electronic system developed taking into account the number of tables to be installed in the new greenhouse to design a network of sensors that are distributed throughout the 8 tables and two working sections, in order to monitor more than one point in greenhouse temperature, humidity and light.

Keywords: BeagleBone Black, Humidity, Temperature, Brightness, Embedded System, Linux, Monitoring, MySQL Database, Web Server.

Dedicatoria

A mis padres y a toda mi familia...

Agradecimiento

Primero que todo quiero agradecer a Dios por darme la oportunidad de estudiar y crecer como persona y profesional, él me guía en cada paso, en los buenos y malos momentos, por darme siempre la fortaleza que necesito y permitirme concluir mis estudios en una gran institución, siempre llevaré al TEC en mi corazón entregarme tanto aprendizaje a nivel profesional y personal.

Agradecer a toda mi familia por apoyarme en cada decisión que tomé, por siempre creer en mí, por todo ese amor brindado que me motivo a nunca bajar la cabeza y a siempre buscar más aprendiendo de mis errores. Especialmente agradezco a mis padres y a mi abuelita Tati, por todos los valores y principios, por enseñarme que en esta vida nada llega gratis y a luchar por mis sueños sin olvidar de dónde vengo. Sólo Dios sabe lo que representa mi familia en mi vida.

A mis amigos, han sido una parte muy importante de esta etapa en mi vida, por todos esos momentos de apoyo en los buenos y malos momentos, por estar cuando más los necesite. Esta etapa de la vida se cierra, pero la amistad continúa...

A los profesores, gracias por tantas enseñanzas a lo largo de estos años, por siempre enseñarnos a dar lo mejor de nosotros, a rebasar nuestros límites, por enseñarnos que con trabajo duro y esfuerzo todo es posible. Muchos son un gran ejemplo a seguir.

A los profesores miembros del tribunal evaluador, la Ing. Gabriela Ortiz, el Ing. Juan Carlos Jiménez y el Ing. Hayden Phillips, por su esfuerzo para contribuir en el presente proyecto, además al profesor Dr. Miguel Rojas Chaves, Ph.D por todo su apoyo brindado.

Gracias a todas esas personas que fueron parte de este proceso, se termina una etapa en mi vida, que me prepara para la que sigue.

Claudio Carvajal González

Cartago, 20 noviembre de 2015

ÍNDICE DE GENERAL

Declaratoria de autenticidad.....	3
Resumen.....	4
Abstract.....	5
Dedicatoria.....	6
Agradecimiento.....	7
ÍNDICE DE GENERAL.....	8
ÍNDICE DE FIGURAS.....	13
ÍNDICE DE TABLAS.....	16
TABLA DE NOMENCLATURA.....	17
Capítulo 1: Introducción.....	19
1.1 Contexto del problema.....	19
1.2 Descripción del sistema a innovar.....	20
1.3 Análisis de involucrados.....	20
1.4 Descripción del problema.....	21
1.5 Síntesis del problema.....	21
1.6 Enfoque de la solución.....	22
Capítulo 2: Meta y Objetivos.....	24
2.1 Meta.....	24
2.2 Objetivo general.....	24
2.3 Objetivos específicos.....	24
Capítulo 3: Marco Teórico.....	26
3.1 Red de Sensores.....	26
3.1.1 Nodos Activos.....	26
3.1.2 Nodos Pasivos.....	27
3.2 Sensores Eléctricos.....	27
3.2.1 Sensores de temperatura.....	27
3.2.1.1 Termopares.....	28
3.2.1.2 RTD.....	28
3.2.1.3 Termistores.....	28

3.2.1.4 Circuitos Integrados.....	28
3.2.2 Sensores de humedad	29
3.2.2.1 Circuitos Integrado HIH 4030	29
3.2.3 Sensores Luminosidad.....	30
3.2.3.1 Fotorresistencia LDR.....	30
3.2.3.1 Circuitos Integrados.....	30
3.3 Sistema de adquisición de datos	30
3.3.1 Unidad convertidor analógico digital	31
3.4. Sistemas de riego programado.....	31
3.5 Sistemas de procesamiento de datos.....	32
3.5.1 Microcontrolador.....	32
3.5.1.1 Plataformas de desarrollo Arduino	33
3.5.1.2 Características del Arduino UNO	34
3.5.1.3 Características del Arduino MEGA	34
3.5.3 Sistemas Embebido	35
3.5.3.1 PcDuino3	36
3.5.3.2 Raspberry Pi.....	37
3.5.3.4 BeagleBone.....	38
3.5.3.5 BeagleBone Black	39
3.6 Servidores Web.....	40
3.6.1 LAMP Server	40
3.6.1.1 GNU/Linux	40
3.6.1.2 Apache	41
3.6.1.3 MySQL.....	42
3.6.1.4 PHP	42
Capítulo 4: Procedimiento metodológico	44
4.1 Reconocimiento y definición del problema	44
4.2 Obtención y análisis de la información.....	45
4.3 Evaluación de las alternativas	45
4.4 Implementación de la solución.....	46
4.5 Reevaluación	46
Capítulo 5: Proceso de diseño	47

5.1 Descripción de la solución.....	47
5.1.1 Posibles soluciones	48
5.1.2 Selección de la solución.....	50
5.2 Selección de los sensores.....	51
5.2.1 Selección del sensor de temperatura.....	51
5.2.2 Selección del sensor de humedad	52
5.2.3 Selección del sensor de luminosidad	52
5.3 Caracterización de los sensores.....	53
5.3.1 Sensor LM35	54
5.3.2 Sensor HIH-4030	55
5.3.2 Sensor 1142	57
5.4 Sistema de procesamiento de datos.....	57
5.4.1 Selección de la unidad de procesamiento central CPU.....	59
5.4.2 Propuestas para la unidad de acondicionamiento de la red de sensores.....	60
5.4.3 Selección de la unidad de acondicionamiento de la red de sensores.....	61
5.5 Sistema de almacenamiento de datos	62
5.5.1 Unidad de acondicionamiento de la red de sensores.....	63
5.5.2 Sistema de adquisición de datos para la red de sensores.....	65
5.5.3 Desarrollo de la base de datos	68
5.5.3.1 Distribución de las tablas en el diagrama entidad relación DER	70
5.5.3.2 Procedimientos almacenados en la base de datos	72
5.5.4 Algoritmos de funcionamiento.....	73
5.6 Sistema de riego programado.....	79
5.6.1 Circuitos de potencia para la manipulación de las electroválvulas	80
5.6.2 Distribución de tiempos de riego.....	83
5.6.3 Algoritmo de funcionamiento	84
5.7 Diseño de la página web.....	85
5.7.1 Distribución de las pantallas	86
5.8 Diseño circuito impreso PCB	89
Capítulo 6: Resultados experimentales	92
6.1 Sistema de almacenamiento de datos	92
6.2 Página Web	96

6.2.1 Pestañas.....	97
6.2.2 Sistema riego programado.....	99
6.3 Circuito impreso PCB.....	101
Capítulo 7: Análisis de resultados	102
7.1 Sistema de almacenamiento de datos	102
7.2 Página web	103
7.2.1 Pestañas.....	103
7.2.2 Sistema de riego programado	104
7.3 Circuito impreso PCB.....	105
Capítulo 8: Conclusiones y Recomendaciones	106
8.1 Conclusiones.....	106
8.2 Recomendaciones.....	108
Bibliografía.....	109
Apéndices.....	113
Apéndice 1: Tablas utilizadas en el proceso de diseño.....	113
1.1 Sistema de almacenamiento de datos	113
1.2 Sistema de riego programado.....	115
Apéndice 2: Instalación de Debian GNU/Linux 7 en BeagleBone Black Rev C.....	118
Apéndice 3: Comandos de instalación de software para Debian GNU/Linux 7.....	119
3.1 Apache.....	119
3.2 MySQL Workbench	119
3.3 PHP.....	119
3.4 phpMyAdmin.....	119
Apéndice 4: Scripts.....	120
4.1 Inicializar config.sh.....	120
4.2 Script config.sh para configuración de GPIO y entradas analógicas.....	120
4.3 Configurar hora del sistema.....	123
Apéndice 5: Base de datos MySQL.....	124
5.1 Tablas creadas para la base de datos MySQL	124
5.2 Programas creados para la base de datos MySQL.....	128
5.2.1 CIBdb.sql	128
5.2.2 StoredProcedures.sql	132

5.2.3 LlenadoTablas.sql.....	136
Apéndice 6: Circuito impreso PCB.....	137
Apéndice 7: Prototipo desarrollado	138

ÍNDICE DE FIGURAS

Figura 1.1. Diagrama de bloques general para el sistema electrónico.....	23
Figura 5.1. Diagrama de bloques del sistema electrónico.	48
Figura 5.2. Primera propuesta para la distribución de red de sensores.....	49
Figura 5.3. Segunda propuesta para la distribución de red de sensores.....	49
Figura 5.4. Tercera propuesta para la distribución de red de sensores.	50
Figura 5.5. Gráficos tensión de salida vs grados Celsius para un canal analógico de 5V (derecha) y 1.8V (izquierda).....	55
Figura 5.6. Gráfico tensión de salida vs %RH a 25°C y 5V. [10]	56
Figura 5.7. Gráficos tensión de salida vs luminosidad para un canal analógico de 5V (derecha) y 1.8V (izquierda).....	57
Figura 5.8. Primera propuesta para la unidad de acondicionamiento de la red de sensores.....	60
Figura 5.9. Segunda propuesta para la unidad de acondicionamiento de la red de sensores.....	61
Figura 5.10. Circuito de acondicionamiento de la red de sensores.	64
Figura 5.11. Distribución de los pines por las cabeceras P8 y P9.....	65
Figura 5.12. Diagrama del sistema de adquisición de datos de la red de sensores.....	66
Figura 5.13. Divisores de tensión para acoplar la señal de los sensores.	67
Figura 5.14. Sistema de adquisición de datos con un trío de sensores conectados a una sección específica.....	68
Figura 5.15. Diagrama de Entidad Relación DER utilizado para desarrollar la base de datos.....	69
Figura 5.16. Diagrama de flujo principal utilizado para el sistema de almacenamiento.....	75
Figura 5.17. Diagrama de flujo utilizado para agregar información a la base de datos.	76
Figura 5.18. Diagrama de flujo utilizado para consultar información de la base de datos.....	77
Figura 5.19. Diagrama de flujo utilizado para eliminar información de la base de datos.	78
Figura 5.20. Distribución de los GPIO para cada mesa/sección del invernadero	79
Figura 5.21. Módulo de relés de 12V para 16 canales SainSmart.....	80
Figura 5.22. Circuito convertidor de tensión de 3.3V a 5V para la mesa 1 y sección 1.....	81
Figura 5.23. Acoples requeridos para el módulo de relés.	83
Figura 5.24. Diagrama de flujo principal para utilizado para el sistema de riego programado.	84
Figura 5.25. Diagrama del proceso de interacción de PHP con una base de datos MySQL.....	85
Figura 5.26. Diagrama del funcionamiento de las páginas web requeridas.....	86

Figura 5.27. Página de inicio de sesión.....	87
Figura 5.28. Página principal.....	87
Figura 5.29. Pestaña para configurar experimentos.....	88
Figura 5.30. Pestaña para consultar experimentos activos.....	89
Figura 5.31. Pestaña para eliminar experimentos activos.....	89
Figura 5.32. Esquemático del PCB.....	91
Figura 5.34. Tabla usuarios.....	93
Figura 5.35. Tablas tiposDeSensor.....	93
Figura 5.36. Tabla mesas.....	93
Figura 5.37. Tabla secciones.....	94
Figura 5.38. Tablas sensores.....	94
Figura 5.39. Tabla temperaturas.....	95
Figura 5.40. Tabla humedades.....	95
Figura 5.41. Tabla luminosidades.....	96
Figura 5.42. Tabla de experimentos activos.....	96
Figura 5.43. Configurando experimento en mesa 1 y sección 1.....	97
Figura 5.44. Mensaje de error para pestaña configurar.....	97
Figura 5.45. Resultado de consultar experimento en mesa 1 y sección 1.....	98
Figura 5.46. Resultado de consultar experimento en mesa 3 y sección 1.....	98
Figura 5.47. Resultado de eliminar experimento en mesa 4 y sección 1.....	98
Figura 5.48. Hora de riego para el experimento 17.....	100
Figura 5.49. Hora de riego para el experimento 19.....	100
Figura 5.50. PCB con las pistas y huecos.....	101
Figura 5.51. Montaje PCB.....	101
Figura a5.1. Tabla usuarios.....	124
Figura a5.2 Tabla mesas.....	124
Figura a5.3. Tabla secciones.....	125
Figura a5.4. Tabla sensores.....	126
Figura a5.5. Tabla tiposDeSensor.....	126
Figura a5.6. Tabla humedades.....	127
Figura a5.7 Tabla luminosidades.....	127
Figura a5.8. Tabla temperaturas.....	128

Figura a6.1. Plano de tierras y distribución de los componentes en el PCB.	137
Figura a7.1. Foto del prototipo implementado en este proyecto de graduación.	138

ÍNDICE DE TABLAS

Tabla 3.1. Principales características del Arduino UNO.....	34
Tabla 3.2. Principales características del Arduino MEGA.	35
Tabla 3.3. Principales características de la PcDuino3.....	36
Tabla 3.4. Principales características de la Raspberry Pi.	37
Tabla 3.5. Principales características de la BeagleBone.	38
Tabla 3.6. Principales características de la BeagleBone Black.....	39
Tabla 5.1. Información para calibrar el sensor a 5V y 25°C.	56
Tabla 5.2. Requerimientos para los sensores y el sistema de procesamiento de datos.	58
Tabla 5.3. Características valoradas para la selección de la unidad de acondicionamiento de la red de sensores.....	62
Tabla 6.1. Horas de riego para los experimentos seleccionados.....	99
Tabla a1.1. Distribuciones de los sensores de temperatura, humedad y luminosidad para las 16 entradas de cada multiplexor.....	113
Tabla a1.2. Combinaciones lógicas para seleccionar un sensor de temperatura, humedad y luminosidad en una mesa/sección específica.	114
Tabla a1.3. Distribución de los GPIO utilizados para el sistema de almacenamiento de datos de la red de sensores.....	114
Tabla a1.4. Distribución de los canales analógicos utilizados para el sistema de almacenamiento de datos de la red de sensores.....	115
Tabla a1.5. Distribución de los GPIO utilizados para el sistema de riego programado.....	115
Tabla a1.6 Distribución de las salidas de los circuitos convertidores en las entradas del módulo de relés.....	116
Tabla a1.7. Distribución de los tiempos de riegos por intervalos de minutos y horas para cada mesa y sección.....	117

TABLA DE NOMENCLATURA

<i>Abreviatura</i>	<i>Significado</i>
CIB	Centro de Investigación en Biotecnología
RTD	Resistance Temperature Resistor
%RH	Porcentaje de Humedad Relativa
LDR	Light Dependent Resistor
ADC	Analog Digital Converter
PIC	Peripheral Interface Controller
DSPIC	Digital Signal Peripheral Interface Controller
USB	Universal Serial Bus
PWM	Pulse Width Modulation
EEPROM	Electrically Erasable Programmable Read Only Memory
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver - Transmitter
CPU	Central Processing Unit
GPU	Graphics Processing Unit
DSP	Digital Signal Processor
DRAM	Dynamic Random Access Memory
SoC	System on a Chip
GPIO	General Purpose Input/Output
ARM	Advanced RISC Machine
RISC	Reduced Instruction Set Computer
HDMI	High Definition Multimedia Interface
SD	Secure Digital

eMMC	Embedded MultiMedia Card
WAMP	Windows Apache MySQL PHP
LAMP	Linux Apache MySQL PHP
MAMP	Macintosh Apache MySQL PHP
XAMP	X (operating system) Apache MySQL PHP
LXDE	GNU is Not Unix
GNU	Lightweight X11 Desktop Environment
HTTP	Hyper Text Transfer Protocol
SQL	Structure Query Language
PHP	HyperText Preprocessor
HTML	HyperText Markup Language
IP	Internet Protocol
URL	Uniform Resource Identifier
DER	Diagrama de Entidad Relación
PCB	Printed Circuit Board
CNC	Computer Numerical Control
DNS	Dynamic Name Server

Capítulo 1: Introducción

1.1 Contexto del problema

El Centro de Investigación en Biotecnología (CIB) forma parte de la unidad de investigación de la Escuela de Biología del Instituto Tecnológico de Costa Rica. Este centro de investigación se fundó en 1995 y se encuentra ubicado en la sede central de Cartago. Actualmente el CIB es coordinado por el profesor Dr. Miguel Rojas Chaves, Ph.D.

El CIB cuenta con diversas áreas de investigación, entre las que destacan biomédica, cultivo de tejidos vegetales, energías renovables, producción e investigación en biocontroladores, caracterización genética mediante aplicación de herramientas moleculares y microbiología. Este proyecto se relaciona con las áreas de energías renovables y microbiología.

El centro de investigación posee un invernadero pero se pretende construir otro con mejores condiciones en el terreno donde se construyó el nuevo edificio de la Escuela de Biotecnología. El sistema a desarrollar se pretende adaptar a las necesidades de diseño establecidas para este nuevo invernadero, la cantidad de mesas se definió en 8 y cada una de estas cuenta con 2 secciones de trabajo.

El invernadero actual ubicado a un costado de la Escuela de Ingeniería Ambiental cuenta con un sistema de riego que posee tuberías con válvulas de agua de cierre manual, que se extienden por cuatro mesas de trabajo. Esto limita a los profesores investigadores a tener que desplazarse al sitio para realizar el riego a las especies en investigación la cantidad de veces que se requiera. También posee un sistema que mide la temperatura y humedad en un solo lugar y almacena esta información en un registrador de datos.

El proyecto surge a causa de la necesidad de obtener procesos más eficientes para el riego de plantas según cada especie, además de contar con un sistema que permita estimar las condiciones ambientales y el comportamiento de las plantas dadas por la medición de temperatura, humedad y luminosidad dentro del invernadero. [9]

Se utilizan sensores para cada una de las tres variables a medir, en conjunto con un sistema eficiente para la adquisición y almacenamiento de los datos recibidos por los sensores, además se empleó un sistema de control de riego para que el usuario configure la cantidad de riegos por día dependiendo del tipo de planta que se necesite estudiar. [22]

1.2 Descripción del sistema a innovar

En este proyecto se diseñó una red de sensores distribuidos por las mesas y secciones de trabajo que registra los datos de temperatura, humedad y luminosidad medidos cerca del lugar donde crecen las plantas en el invernadero. [26]

Cuando un profesor investigador lleva a cabo un nuevo experimento se ve obligado a visitar el sitio para regar de forma manual las plantas. Para evitar esto y ahorrar tiempos a estas personas, se utilizan electroválvulas distribuidas por las mesas/secciones de trabajo, que manipulan el riego de manera programada según la necesidad de cada experimento, los riegos tardan 30 segundos activos. Un sistema de riego programado temporiza los intervalos de riego, según la configuración que el usuario define.

1.3 Análisis de involucrados

Se tiene como principal involucrado en el proyecto al Centro de Investigación en Biotecnología. El coordinador de este centro es el profesor Miguel Rojas Chaves, Ph.D que labora en conjunto con profesores de ingeniería en biotecnología en el ITCR. El profesor Jaime Brenes Madriz es el especialista en temas de invernaderos, pero además se cuenta con el respaldo de otros docentes de esta escuela que dan uso al invernadero.

El centro pretende construir un nuevo invernadero en el terreno que el ITCR les ha asignado, el cuál cuente con un sistema de riego programado y además monitorice las

variables temperatura, humedad y luminosidad para generar historiales, que permitan a los docentes estudiar las condiciones ambientales en las que crecen las plantas.

1.4 Descripción del problema

Actualmente el Centro de Investigación en Biotecnología cuenta con un invernadero para el estudio de diversas especies de plantas, este lugar posee cuatro mesas utilizadas para distribuir estas plantas.

El riego en el invernadero se realiza de forma manual, para la adquisición de datos se cuenta con único un sistema que almacena información de temperatura y humedad, no brinda valores de la intensidad lumínica del lugar. [11]

CIB aún no cuenta con un sistema electrónico, que permita registrar las variables temperatura, humedad y luminosidad por mesas y secciones dentro del invernadero, además de manipular válvulas de agua distribuidas por estas mesas y secciones de trabajo. El sistema es capaz de crecer en número de experimentos hasta llegar a abarcar las 16 secciones distribuidas por las 8 mesas, además en caso de finalizase algún experimento este debe ser eliminado reduciendo el número de experimentos activos.

1.5 Síntesis del problema

¿De qué forma se puede automatizar el riego programado de un invernadero en tiempo real y adquirir datos para monitorizar las variables temperatura, humedad y luminosidad?

1.6 Enfoque de la solución

Debido a la necesidad de obtener procesos más eficientes en el riego de las plantas, se diseñó un sistema electrónico que manipula el riego de una manera programada, para esto el usuario debe configurar a alguna de las 16 secciones de trabajo, la cantidad de riegos por día que se deben realizar. [7]

El nuevo sistema de adquisición de datos toma en cuenta la luminosidad en el invernadero, variable que el sistema actual no registra. Esto representa una ventaja debido a que la intensidad lumínica es un factor determinante en el crecimiento y desarrollo de las plantas. [8]

El sistema electrónico permite conocer el valor de la temperatura, humedad y luminosidad en más de un punto del invernadero, para esto se diseñó una red de sensores distribuidos por las mesas y secciones del lugar; además toda la información de los datos provenientes de la red de sensores se registra en una unidad de almacenamiento. [26]

Para la interfaz gráfica de usuario se implementó una página web que le permite al profesor investigador dar lectura de la información registrada en el sistema de almacenamiento y a su vez configurar el riego programado dentro del invernadero. También se diseñó en la página web un apartado para monitorizar el estado de riego, la temperatura, humedad y luminosidad para los experimentos activos desde la pantalla principal.

El diagrama de bloques del sistema electrónico que se desarrolló desde un modelo general se muestra en la figura 1.1. En esta figura se observa que el sistema recibe como entradas los datos provenientes de los sensores, que antes pasan por una unidad de acondicionamiento. También recibe los datos de entrada ingresados por el usuario para configurar un experimento en alguna de las secciones de trabajo. Las salidas del diagrama muestran que el sistema permite dar lectura al registro de datos almacenados y se encarga de realizar los respectivos riegos.

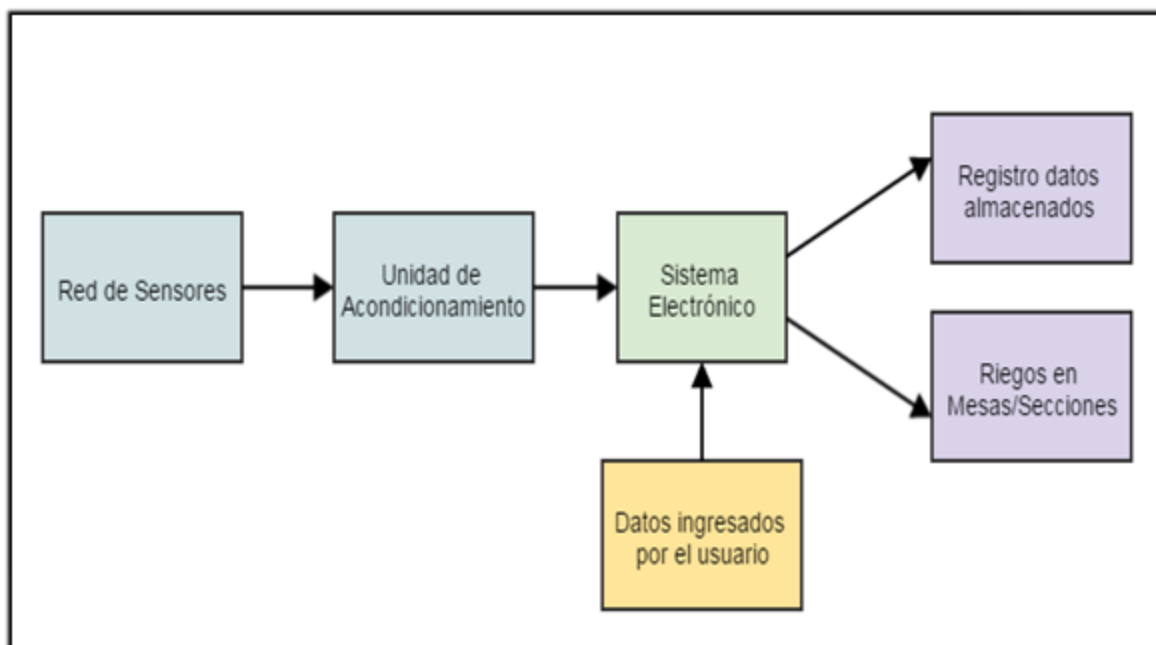


Figura 1.1. Diagrama de bloques general para el sistema electrónico.

Capítulo 2: Meta y Objetivos

2.1 Meta

Implementar un sistema electrónico de adquisición de datos para monitorizar las condiciones ambientales bajo las cuales crecen las plantas en el invernadero, tomando en cuenta las variables temperatura, humedad y luminosidad. Además el sistema se proyecta como una herramienta que le permita al docente programar el riego por separado en cada mesa/sección de trabajo, para desarrollar experimentos independientes de acuerdo a las necesidades de riego de las diferentes especies que se analizan.

2.2 Objetivo general

- Desarrollar un sistema electrónico que permita programar el riego y la adquisición de datos para la monitorización de la temperatura, humedad y luminosidad de un invernadero.

2.3 Objetivos específicos

- Diseñar una red de sensores de temperatura, humedad y luminosidad.
- Realizar la implementación un sistema de almacenamiento de los datos, provenientes de una red de sensores de temperatura, humedad y luminosidad.
- Desarrollar un sistema para el riego programado en cada una de las mesas del invernadero que se puede modificar por el usuario.

- Diseñar una página web que sirva de interfaz gráfica para el usuario, en la que se visualice la información que contiene el sistema de almacenamiento y a su vez permita configurar el riego en las mesas del invernadero.

Capítulo 3: Marco Teórico

En este capítulo se exponen los fundamentos teóricos necesarios para el diseño de la solución empleada. Se examinan diversos tipos de sensores eléctricos para medir temperatura, humedad y luminosidad, también se exponen conceptos acerca de sistemas de adquisición de datos basados en unidades convertidor analógico digital, sistemas de riego programado, diversos tipos de sistemas de procesamiento y servidores web junto a las herramientas requeridas para el desarrollo de una página web.

3.1 Red de Sensores

Para realizar la monitorización de las condiciones climáticas presentes en el invernadero, deben de existir experimentos activos de plantas en crecimiento. Se desea contar una red de sensores distribuida por las mesas/secciones de trabajo, para conocer la temperatura, humedad y luminosidad en más de un punto del invernadero. [26]

Las redes de sensores están compuestas por nodos, estos se clasifican en dos categorías según su función, los nodos activos y nodos pasivos. En estos nodos se acoplan los sensores a utilizar (temperatura, humedad y luminosidad), se pueden distribuir nodos por distintos lugares para recopilar la información cada tipo de sensor en más de un punto del invernadero. Los datos de estos nodos se pueden transmitir de manera inalámbrica o bien por cableado. [25]

3.1.1 Nodos Activos

Existen dos tipos de nodos activos, los medidores y los actuadores. Los nodos medidores conectan sensores e interpretan la información para transmitirla de acuerdo se programe y los nodos actuadores conectan interruptores, relés, entre otros que permiten

conmutar de un estado a otro, para actuar sobre los sensores y otros componentes de una manera establecida.

3.1.2 Nodos Pasivos

Los nodos pasivos se dividen en transmisores y concentradores. Los nodos concentradores recopilan los datos que suministran los sensores para almacenarlos y los nodos transmisores sólo se encargan de enviar la información de los sensores al receptor.

3.2 Sensores Eléctricos

En la actualidad existen gran variedad de sensores para realizar mediciones de temperatura, humedad y luminosidad. Por la diversidad del mercado existen casos de sensores que consisten en un simple transistor hasta los automatizados con pantallas donde es posible visualizar el resultado de la medición. [22]

3.2.1 Sensores de temperatura

Este tipo de sensores se utilizan para adquirir datos de temperatura presentes en el medio. La señal eléctrica de salida por lo general se representa por un valor analógico proporcional a la medición realizada según corresponda en la hoja específica de cada sensor. Es necesario contar con una etapa posterior que almacene y traduzca el valor obtenido por el sensor en una medida estándar para la variable temperatura, como los grados Celsius y grados Fahrenheit.

3.2.1.1 Termopares

Este tipo de sensores de temperatura son relativamente precisos y trabajan en un amplio rango de temperaturas. Están formados por dos metales que al juntarse el lugar de contacto produce una tensión de circuito abierto en función de la temperatura. Esta tensión es relativamente lineal para cambios pequeños de temperatura. [15]

3.2.1.2 RTD

Son sensores de temperatura contruidos en platino. Un RTD es un dispositivo hecho de bobinas. El principio de funcionamiento consiste en que la resistencia del metal al calentarse aumenta y al enfriarse disminuye, por lo tanto al hacer pasar una corriente eléctrica por el RTD se genera una tensión que se puede relacionar con la temperatura. Pueden medir temperaturas hasta los 850° Celsius. [15]

3.2.1.3 Termistores

El termistor consiste en una oblea semiconductor hecha de óxidos de metal sometida a altas temperaturas que después se recubre por lo general de vidrio. Como en el caso de los RTD se introduce una corriente que genera una tensión en el termistor que se relaciona con la temperatura. Presenta más sensibilidad comparado con el RTD pero un rango limitado de hasta 300° Celsius. [15]

3.2.1.4 Circuitos Integrados

Para el caso de los sensores de temperatura en circuitos integrados existen en el mercado cuatro categorías:

- Salida de tensión.

- Salida de corriente.
- Salida digital.
- Salida de resistencia.

En cuanto a los circuitos integrados con salida de tensión se encuentra el LM35 adaptado para medir grados Celsius y el LM34 adaptado para los grados Fahrenheit. Un circuito integrado conocido con salida de corriente es el AD590. En los integrados de salida digital se encuentran el LM56 y el LM57. Los de salida de resistencia son poco comunes. [16]

3.2.2 Sensores de humedad

Los sensores de humedad se basan en que el agua no es un material aislante sino que tiene una conductividad eléctrica, por lo tanto un par de cables ubicados muy cercanos pero sin hacer contacto eléctrico pueden conducir una pequeña cantidad de corriente si el ambiente está húmedo, si después se coloca una etapa que amplifique esa señal eléctrica se implementa un detector de humedad.

3.2.2.1 Circuitos Integrado HHH 4030

Este sensor presenta la ventaja de que viene integrado en una tarjeta. Mide la humedad relativa del medio %RH y entrega una salida de tensión analógica aproximadamente lineal de acuerdo al valor medido. La tensión aplicada a los pines de alimentación debe mantenerse entre un rango de 4v y 5.8v, se recomienda utilizar 5v. [10]

El sensor es de bajo consumo con una corriente típica de 200 μ A, además es sensible a la luz, por lo que se recomienda colocarlo de manera que no reciba la luz directamente en la superficie de contacto con el medio ambiente.

3.2.3 Sensores luminosidad

3.2.3.1 Fotorresistencia LDR

En una fotorresistencia la resistencia varía en función de la luz que incide sobre su superficie. A mayor luz incidiendo en la superficie del LDR (“Light-Dependent Resistor”) menor será la resistencia y a menor luz incidiendo la resistencia aumenta. [23]

3.2.3.1 Circuitos Integrados

En cuanto a circuitos integrados para sensores de luz se presentan el sensor 1142 y el 1143. La diferencia entre estos es que 1142 posee un alcance de 1000 lux y se utiliza para medir luz en interiores y en exterior cuando se está nublado; por otra parte el sensor 1143 presenta un alcance de hasta 70000 lux por lo que es recomendado para medir la luz en exteriores, dado que la luminosidad en un día despejado al medio día alcanza valores de 100000 lux. [17]

Estos dispositivos pueden presentar diferencias de acuerdo al tipo de luz que se está midiendo, la diferencia entre la luz fluorescente y la incandescente es de un factor de 1,2. Para un uso adecuado de estos sensores, se debe colocar el dispositivo de manera que la fotocelda que viene integrada quede en una buena posición para detectar la luz, ya que de lo contrario la medición puede presentar errores o no ser la esperada.

3.3 Sistema de adquisición de datos

Para realizar un sistema de adquisición de datos de las condiciones climáticas presentes dadas por la temperatura, humedad y luminosidad, se requiere de una unidad convertidor analógico digital que permite que los datos ingresen al sistema.

3.3.1 Unidad convertidor analógico digital

Las unidades convertidor analógico digital ADC son utilizadas para convertir una señal analógica en una señal digital representada por un número binario. La resolución del ADC define la exactitud de la conversión y está ligada a la cantidad de bits de salida del ADC, la ecuación 1 muestra la relación que existe entre la resolución del ADC, la lectura en el convertidor, la tensión analógica medida y la tensión de alimentación V_{cc} . [24]

$$\frac{\text{Resolución del ADC}}{V_{cc}} = \frac{\text{Lectura del ADC}}{\text{Tensión Analógica Medida}} \quad (1)$$

La resolución del ADC se calcula con la siguiente ecuación 2, donde n es la cantidad de bit de salida del convertidor.

$$\text{Resolución del ADC} = 2^n \quad (2)$$

La tensión de alimentación por lo general es de 5v o 3,3v. Para la lectura de los bits de salidas en el ADC, se ingresa la señal analógica en el puerto de entrada, el resultado está entre un rango definido por la resolución del convertidor. Un ejemplo de esto es un ADC de 10 bits de salida, presenta una resolución de 1024 y los valores de salida al ingresar la tensión analógica de entrada estarán entre el rango de 0 y 1024.

3.4. Sistemas de riego programado

Para realizar el riego programado dentro de un invernadero, se deben adaptar tuberías con electroválvulas que permitan manipular de una forma eléctrica el flujo de agua por los conductos. [3]

Las electroválvulas están compuestas básicamente por dos partes, la primera consiste en una cabeza magnética que contiene una bobina en un tubo y la segunda el cuerpo con los orificios de calibrado que aseguran el cierre en la electroválvula. El cierre y la apertura se

realizan a través de un campo magnético generado por la bobina que atrae un émbolo. Para proveer la presión al agua en las tuberías se debe de instalar una electrobomba centrífuga. [21]

3.5 Sistemas de procesamiento de datos

El procesamiento de datos consiste en tomar los datos que ingresan por un puerto de entrada y darles algún tratamiento deseado. Según criterios dados por el programador, estos datos entrantes se pueden seleccionar y almacenar para utilizarlos dentro de la lógica del sistema.

Actualmente se encuentran en el mercado una extensa gama de sistemas de procesamientos de datos, existen los microcontroladores (PIC y DSPIC), las plataformas Arduino (UNO, Mega y Leonardo) y las tarjetas de desarrollo embebidas (PC Duino, Raspberry PI, BeagleBoard Xm, BeagleBone y BeagleBone Black, entre otros). [1], [2], [5], [6], [12] y [20]

3.5.1 Microcontrolador

El microcontrolador se conoce como uno de los primeros sistemas embebidos en fabricarse tomando en cuenta que está compuesto de la integración de un microprocesador, una memoria y una unidad para dispositivos periféricos de entrada/salida, permite por medio de instrucciones de programa propias de cada dispositivo, la ejecución de algún código específico para solucionar un problema definido.

En el presente documento se analizará el microcontrolador ATMEGA328P y ATMEGA2560 de la compañía Arduino.

3.5.1.1 Plataformas de desarrollo Arduino

Las plataformas de desarrollo Arduino son tarjetas prototipo para desarrollar código abierto basado en una combinación de hardware y software, que se ejecuta por medio del ambiente de programación “Arduino Software (IDE)”. Esta plataforma además de programarse por medio del lenguaje de Arduino puede soportar funciones básicas de C y algunas definidas de C++.

Este entorno de desarrollo presenta la ventaja de que se puede correr desde sistemas operativos como Windows, Macintosh OSX y en múltiples distribuciones de Linux. El dispositivo puede programarse mediante un adaptador USB.

Existen muchas librerías a disposición para desarrollar proyectos en estas plataformas de una manera gratuita, ya que al ser de código abierto se ha vuelto una gran comunidad de desarrolladores que dejan sus aportes al público general. Si se desea agregar una nueva librería al entorno de programación Arduino (IDE), se debe ir a la carpeta donde se ha instalado el software y en la carpeta lib agregarla, al finalizar se debe reiniciar el programa.

Estas tarjetas de desarrollo integran en la mayoría de los casos un microcontrolador ATmega, que presenta pines de propósito general de entrada/salida tanto analógicos como digitales. Los pines se extienden por la placa hasta unas salidas hembra para poder utilizarlos de una manera sencilla, además la plataforma cuenta con un conector USB para realizar la programación del microcontrolador, una memoria flash para almacenar datos, un reloj de sistema generado por un oscilador local, un botón de reinicio, reguladores de tensión, entre otras características. [4]

Los pines de entrada/salida digitales operan en un rango de 3.3V a 5V, estos pines le permiten al microcontrolador comunicarse con dispositivos externos. También los puertos analógicos sirven para que el Arduino interactúe con señales analógicas como las producidas por sensores electrónicos.

3.5.1.2 Características del Arduino UNO

Este microcontrolador está diseñado con una arquitectura de tipo AVR, desarrollada por Atmel con un conjunto de 131 instrucciones que pueden ser ejecutadas, pero gracias al Arduino Software (IDE) es posible usar lenguaje de más alto nivel (el lenguaje de Arduino está basado en C) que a la hora de compilarse se convierte en un conjunto de instrucciones que el ATmega328 puede interpretar. Las principales características del Arduino UNO se detallan en la tabla 3.1. [1] y [4]

Tabla 3.1. Principales características del Arduino UNO.

<i>Unidad</i>	<i>Características</i>
Microcontrolador	ATMega328P
Memoria Flash	32kB 2kB asignados al gestor de arranque
Memoria EEPROM	1kB
Memoria SRAM	2kB
Frecuencia de reloj	Oscilador local de 16MHz
Alimentación	Tensión entre un rango de 7V y 12V
Entradas analógicas	6 pines
Entradas/Salidas digitales	6 pines

3.5.1.3 Características del Arduino MEGA

Por otra parte el Arduino MEGA comparado con el Arduino UNO presenta una mayor cantidad de pines digitales de entrada/salida con 54 y mayor número de entradas

análogas con 16. Las principales características del Arduino MEGA se observan en la tabla 3.2. [2]

Tabla 3.2. Principales características del Arduino MEGA.

<i>Unidad</i>	<i>Características</i>
Microcontrolador	ATMega2560
Memoria Flash	256kB 8kB asignados al gestor de arranque
Memoria EEPROM	4kB
Memoria SRAM	8kB
Frecuencia de reloj	Oscilador local de 16MHz
Alimentación	Tensión entre un rango de 7V y 12V
Entradas analógicas	16 pines
Entradas/Salidas digitales	54 pines

3.5.3 Sistemas Embebido

Los sistemas embebidos deben el nombre a que presentan integrados en un solo chip una gran cantidad de unidades funcionales, como son el caso de la unidad de procesamiento central CPU, unidad de procesamiento gráfica GPU, unidad de procesamiento digital de señales DSP, memoria RAM, unidad de puertos entrada/salida, entre otras. Este chip donde convergen estas unidades funcionales se conoce como “Sistema en un Chip”, SoC en inglés por sus siglas. Existen en el mercado muchas plataformas de desarrollo embebidas las cuales trabajan como pequeñas computadoras, que por lo general funcionan con sistemas operativos Linux y presentan en su hardware puertos USB, Ethernet, pines entrada/salida de propósito general (GPIO), salida de video, entre otros para que el usuario utilice a conveniencia.

3.5.3.1 PcDuino3

La placa de desarrollo embebida PcDuino3 puede ejecutar sistemas operativos como Ubuntu Linux y Android 4.0. Esta tarjeta es compatible con las plataformas de Arduino conocidas en inglés como “Arduino Shields”. Soporta códigos en lenguaje C, C++. También es compatible con programas en Python y lenguaje Java con el estándar de paquetes Android para el desarrollo de sistemas. Trabaja con un SoC AllWinner A20 con arquitectura ARM. [12]

El PcDuino3 contiene 14 pines digitales de entrada/salida donde dos de estos se pueden utilizar como salidas PWM, además de 6 entradas analógicas y un puerto UART. En la siguiente tabla 3.3 se muestran más características de este sistema embebido.

Tabla 3.3. Principales características de la PcDuino3.

<i>Unidad</i>	<i>Características</i>
Procesador CPU	ARM Cortex A7 Dual - Core
Procesador gráfico GPU	ARM Mali400MP2
DRAM	1GB
Memoria interna	Memoria flash de 4GB
Memoria externa	Entrada de tarjeta microSD
Salida de Video	HDMI
Alimentación	Adaptador 5V, 2A
Puertos USB	1 USB Host, 1 USB OTG
Interfaz de red	Ethernet 10M/100Mbps y Wi Fi incorporado

3.5.3.2 Raspberry Pi

La plataforma de desarrollo embebida Raspberry Pi puede poner en marcha sistemas operativos Linux y Android 4.0, para esto utiliza una tarjeta de memoria micro SD donde almacena la información para arrancar el sistema operativo. Opera con un SoC Broadcom BCM2835 que funciona con arquitectura ARM.

La gran desventaja de este dispositivo es que no integra en su hardware pines de entradas analógicas, lo cual limita realizar una gran variedad de proyectos. Sus principales características se observan en la siguiente tabla 3.4. [20]

Tabla 3.4. Principales características de la Raspberry Pi.

<i>Unidad</i>	<i>Características</i>
Procesador CPU	ARM1176JZFS 700MHz
SDRAM	512MB
Memoria externa	Entrada de tarjeta microSD
Salida de Video	HDMI
Alimentación	Adaptador 5V, 2A
Puertos USB	4 USB 2.0 Host
Interfaz de red	Ethernet 10M/100Mbps, RJ45
Principales conexiones	27 GPIO y puerto UART

3.5.3.4 BeagleBone

Este dispositivo embebido como los anteriores expuestos presenta arquitectura ARM, por lo que es posible utilizar sistemas operativos Android 4.0 y distintas distribuciones de Linux. El SoC integrado en esta placa es el Sitara ARM Cortex A8.

Una de las ventajas de este sistema es la cantidad de pines de múltiples propósitos a cada extremo de la plataforma, posee un total de 92 pines donde 65 son GPIO (“Entrada/Salida de Propósito General”), 7 pines para entradas analógicas y lo demás pines para tierras GND, tensiones de 5V y 3,3V, entre otros. Las principales características de este sistema embebido se observan en la tabla 3.5. [5]

Tabla 3.5. Principales características de la BeagleBone.

<i>Unidad</i>	<i>Características</i>
Procesador CPU	AM3359 ARM Cortex A8 /720MHz
Memoria DRAM	256MB DDR2 / 400MHz
Memoria externa	Entrada de tarjeta microSD
Salida de Video	Capa de complementos “Cape add-ons”
Salida de Audio	Capa de complementos “Cape add-ons”
Alimentación	Adaptador 5V, 2A
Puertos USB	1 USB 2.0 Host 1 USB 2.0 Client: Alimentación y depuración
Interfaz de red	Ethernet 10M/100Mbps, RJ45
Principales conexiones	65 GPIO y 7 entradas analógicas Se debe utilizar 3,3V para todas las señales de E/S

3.5.3.5 BeagleBone Black

La plataforma embebida BeagleBone Black (Rev C) al igual que la BeagleBone trabaja con un SoC Sitara ARM Cortex A8. Este dispositivo electrónico comparado con la BeagleBone posee de más en su hardware un puerto de salida microHDMI, el doble de memoria RAM al doble de frecuencia de operación y una memoria interna extra eMMC de 4GB. [6]

Una ventaja que presenta es que se pueden cargar sistemas operativos tanto de la memoria micro SD como de una memoria interna eMMC. Las principales características de la BeagleBone Black Rev C se muestran en la tabla 3.6.

Tabla 3.6. Principales características de la BeagleBone Black.

<i>Unidad</i>	<i>Características</i>
Procesador CPU	AM3359 ARM Cortex A8 1GHz
DRAM	512MB DDR3 800MHz
Memoria interna	4GB 8 bits eMMC
Memoria externa	Entrada de tarjeta microSD
Salida de Video	microHDMI, "cape add-ons"
Salida de Audio	microHDMI, "cape add-ons"
Alimentación	Adaptador 5V, 2A
Puertos USB	1 USB 2.0 Host 1 USB 2.0 Client: Alimentación y depuración
Interfaz de red	Ethernet 10M/100Mbps, RJ45.
Principales conexiones	65 GPIO y 7 entradas analógicas Se debe utilizar 3,3V para todas las señales de E/S

3.6 Servidores Web

Los servidores web pueden trabajar bajo diferentes sistemas operativos. En el caso de “WAMP Server” se utiliza Windows de la empresa Microsoft como sistema operativo, además de las herramientas Apache2 como servidor web, MySQL para gestionar las bases de datos y PHP como lenguaje de programación para desarrollar las páginas web dinámicas. El nombre “WAMP Server” se debe a la letra inicial de cada una de estas herramientas Windows, Apache2, MySQL y PHP. [14] y [18]

Para sistemas operativos que operan bajo distribuciones de Linux se pueden emplear un sistema análogo conocido como “LAMP Server”. En sistemas operativos Macintosh el servidor se conoce como “MAMP Server”. Existe también un servidor independiente de software libre conocido como “XAMP Server” disponible para sistemas operativos de Microsoft, distribuciones de GNU/Linux y MacOS. [13]

3.6.1 LAMP Server

Estos servidores presentan la ventaja de que son de acceso libre, sencillos de configurar y muy robustos. Por contar con una gran comunidad de desarrolladores estas herramientas están en constante desarrollo y mejora al recibir sugerencias de sus usuarios.

3.6.1.1 GNU/Linux

El sistema GNU/Linux no es el producto de una sola compañía como es el caso de Microsoft (Windows) y Apple (Macintosh), sino que es la consecuencia de la contribución de múltiples compañías y grupos de personas. Linux es un componente central o núcleo (kernel), cuya función es asignar los recursos de hardware a los programas que el usuario utiliza. Este es un componente esencial para un sistema operativo pero no puede funcionar

por sí mismo, sino que debe emplearse en conjunto de algún sistema operativo que normalmente para Linux se utiliza GNU (“GNU is Not Unix”) desarrollado por el proyecto GNU. Este proyecto se inició en enero 1984 y abarca una gran colección de programas como aplicaciones, librerías y paquetes de GNU. [19]

Existen una gran cantidad de distribuciones de GNU/Linux, las cuales se pueden categorizar por estabilidad, seguridad y requerimientos de hardware. En cuanto a las distribuciones más estables destacan Red Hat Linux, OpenSUSE, Fedora, Debian, Arch Linux. Por otra parte las distribuciones que contienen mejoras en la parte de seguridad se conocen Kali Linux, BlackArch Linux y Arch Assault.

Otros sistemas operativos como Puppy Linux, Lubuntu y LXLE son compatibles con equipos de bajos recursos de hardware. También existen distribuciones que suelen presentar un entorno gráfico más apto para personas no familiarizadas con los sistemas operativos GNU/Linux como el caso de Ubuntu, Linux Mint y Elementary OS, estas distribuciones son recomendadas ejecutarlas en equipos con suficientes recursos de hardware.

3.6.1.2 Apache

Apache es un proyecto de código abierto y de multiplataforma debido a que existen versiones para muchos sistemas operativos entre los que destacan Microsoft Windows, Mac OS y distribuciones de GNU/Linux. El protocolo más común utilizado para ver las páginas web es el HTTP que debe el nombre a las siglas en inglés “*Hyper Text Transfer Protocol*”. El servidor web Apache se encarga de gestionar las peticiones HTTP, que provienen de visitantes (equipos cliente) que acceden a la página web alojada en el servidor, para dar su acceso o denegación, también informa al visitante de los errores en las páginas web no encontradas. El archivo principal de configuración se llama `apache2.conf` ubicado en el directorio `/etc/apache2/apache2.conf` para servidores LAMP.

3.6.1.3 MySQL

MySQL es un sistema de administración de bases de datos de código abierto. Se encarga de gestionar bases de datos relacionales por lo que utiliza múltiples tablas para manipular el flujo de información. Este software se desarrolló en los lenguajes de programación C y C++ lo que le permite una gran adaptación para interactuar con otros entornos de desarrollo y programas como PHP, Perl y Java.

Existen múltiples herramientas visuales para diseñar bases de datos a manera de diagramas que contempla desarrollo de software, para distribuciones GNU/Linux como Debian el programa Workbench se presenta como una gran herramienta para el desarrollo de bases de datos. Workbench permite modelar las bases de datos en forma de tablas y diagramas, desarrollar código en lenguaje MySQL para diseñar los procedimientos almacenados que se implementan en la base de datos, también presenta herramientas de administración de usuarios, configuración del servidor, copias de seguridad, entre otros. [14]

3.6.1.4 PHP

El lenguaje de programación PHP "*HyperText Preprocessor*" es de código abierto adecuado para el desarrollo de páginas web, una característica muy importante de este lenguaje es que puede trabajar códigos en HTML "*HyperText Markup Language*", por lo que en un mismo archivo se puede combinar código PHP y HTML siguiendo ciertas reglas. HTML es utilizado para desplegar el contenido de los datos de una página web.

PHP es utilizado para diseñar páginas web dinámicas, ya que por lo general trabajan con bases de datos desarrolladas en MySQL, que permiten que el contenido almacenado aumente o cambie, lo que las vuelve dinámicas en la manipulación de datos. PHP como lenguaje de programación se ejecuta en el servidor que recibe la petición y reúne la información consultando las bases de datos, otras páginas web, servidores, entre otros para devolver como respuesta al equipo cliente una página web del tipo estática, pero su creación ha sido dinámica por el proceso de consultas. [18]

Un detalle importante de este proceso es que el equipo cliente no tiene posibilidad de determinar el código que el servidor le ha enviado después de la petición, lo que aumenta la seguridad de las páginas web. PHP permite ejecutar contenido en código Javascript, Java, C, C++, MySQL, el mencionado HTML, entre otros.

Existe un proyecto también de código abierto en PHP que permite a través de una interfaz web gestionar y administrar las bases de datos implementadas en MySQL. Esta herramienta es conocida como phpMyAdmin y permite acceder a todas las funciones básicas de la base de datos desarrollada en MySQL. Esta aplicación web consiste en un conjunto de archivos escritos en PHP que al instalarse el programa phpMyAdmin se instalan en nuestro servidor web, de manera que al ingresar a estas páginas podemos encontrar las bases de datos que se han desarrollado en código MySQL. Para ingresar a este proyecto se debe ingresar desde un buscador web, ya sea Google Chrome, Mozilla Firefox o Internet Explorer la dirección <http://localhost/phpmyAdmin/index.php>, el localhost dado por una dirección IP “*Internet Protocol*” y el número de puerto (80), se puede sustituir por un URL “*Uniform Resource Identifier*” definido.

Capítulo 4: Procedimiento metodológico

En este capítulo se muestran los mecanismos utilizados para el procedimiento metodológico basados en el diseño de ingeniería que estructura como primer paso el reconocimiento y definición del problema, como segundo apartado la obtención y análisis de la información que junto al tercer paso de evaluación de las posibles alternativas se vuelven iterativos hasta seleccionar la opción que mejor se adapte. El siguiente paso fue la implementación de la solución y se finalizó con la reevaluación.

4.1 Reconocimiento y definición del problema

Para definir el problema se contactó con profesores de la Escuela de Biotecnología, especialmente a los profesores que le dan uso al actual invernadero con el fin de conocer las necesidades específicas que poseen para el cultivo de las plantas que investigan. Después de varias reuniones donde se debatió y se realizaron las preguntas necesarias para definir el alcance del proyecto, se logró determinar que se requiere de un sistema de almacenamiento para poder registrar los valores de temperatura, humedad y luminosidad presentes en el invernadero, en conjunto con un sistema de riego programado que permita definir los intervalos de tiempo que se desea regar una sección de mesa definida. Esto permitiría monitorizar las variables climáticas y a su vez realizar de forma automática el riego por cada sección de mesa según el intervalo de riego establecido.

Definido el problema a resolver, se procedió a establecer los objetivos específicos que permitan segmentar en diferentes etapas la solución, iniciando con la unidad de acondicionamiento para la red de sensores, un sistema de almacenamiento de información y un sistema de riego programada, además de toda la programación requerida para implementar un servidor con sus respectivas páginas web que dan el funcionamiento al programa a desarrollar.

4.2 Obtención y análisis de la información

Para conocer las necesidades a solventar con el sistema electrónico se realizaron una serie de consultas generales al profesor Dr. Miguel Rojas Chaves, Ph.D coordinador del Centro de Investigación en Biotecnología, y para las preguntas más específicas sobre temas relacionados al invernadero como el tipo de riego, intervalo de tiempo para riegos, variables climáticas a medir, entre otros aspectos a los profesores Jaime Brenes Madriz y Karol Jiménez Quesada, encargados de darle uso al invernadero actual.

4.3 Evaluación de las alternativas

Para la selección de los diferentes criterios de diseño se implementaron una serie de propuestas en cada caso, con el fin de analizarlas y escoger la que más se adapte a las necesidades del proyecto, teniendo presente los objetivos a cumplir. Se evaluaron 3 diferentes alternativas para la selección de la distribución de sensores para diseñar la red, la primera con una cantidad de 12 sensores, la segunda distribuye 24 unidades y la tercera un total de 48 sensores. También se evaluaron 2 alternativas para definir el tipo de unidad de acondicionamiento para la red de sensores a emplear, la primera utiliza 3 unidades de Arduino MEGA y la segunda 3 multiplexores analógicos 4067B.

Para seleccionar la unidad de procesamiento central a utilizar se evaluaron las características de las plataformas detalladas en el marco teórico con el fin de escoger la que más se adapte a las necesidades del proyecto. Un aspecto muy importante para esta selección es la cantidad de puertos digitales de entrada/salida.

4.4 Implementación de la solución

En el desarrollo del sistema electrónico se segmentó el problema por objetivos específicos con el propósito de trabajarlos de forma simultánea para aprovechar de una mejor manera el tiempo establecido.

Para la red de sensores se estudió el comportamiento de cada tipo de sensor, con el propósito de caracterizar cada uno de estos para adaptarlos al sistema de procesamiento central empleado. En cuanto a la unidad de almacenamiento fue necesario realizar un estudio del lenguaje SQL y los programas dedicados para generar diagramas de entidad relación.

El riego programado se empleó mediante un algoritmo que se encarga de temporizar los intervalos de riego de cada mesa / sección, lo que permite manipular las electroválvulas en las horas acordadas.

4.5 Reevaluación

Durante el proceso de diseño se realizaron pruebas de manera repetitiva con el fin de verificar el correcto funcionamiento de los sistemas implementados, además se efectuaron las pruebas finales con la totalidad del sistema electrónico desarrollado puesto en ejecución.

Capítulo 5: Proceso de diseño

5.1 Descripción de la solución

Para la solución del presente proyecto se diseñó un sistema de adquisición de datos, un sistema de riego programado por intervalos de tiempo, un circuito de acople de potencia para manipular electroválvulas y uno para acondicionar sensores de temperatura, humedad y luminosidad. Todo esto debe poderse desarrollar en un invernadero que contará con ocho mesas de trabajo divididas en dos secciones cada una.

Dado a los requisitos del problema se implementó un sistema de procesamiento, una base de datos, electroválvulas, sensores de temperatura, humedad y luminosidad, además de un circuito de acople de potencia y otro de acondicionamiento de la red de sensores. El diseño se desarrolló para 8 mesas y 2 secciones, lo que permite un máximo de 16 experimentos activos.

En la figura 5.1 se observa el diagrama de bloque que describe la solución del problema, la red de sensores está compuesta por sensores de temperatura, humedad y luminosidad que son enviados a una unidad de acondicionamiento. Esta unidad se encarga de enviar los datos de cada tipo de sensor a 3 entradas analógicas del ADC en el sistema de procesamiento central, en este se almacena la información que puede ser consultada. Además el sistema electrónico se encarga de recibir los datos de entrada que selecciona el usuario para configurar cada experimento y por medio de una unidad de riego programado realiza los riegos por las 16 secciones de trabajo.

En la siguiente sección 5.1.1 se presentan tres posibles soluciones para la ubicación de los sensores por las mesas y secciones de trabajo en el invernadero, estas varían según la cantidad y la distribución de los tipos de sensores.

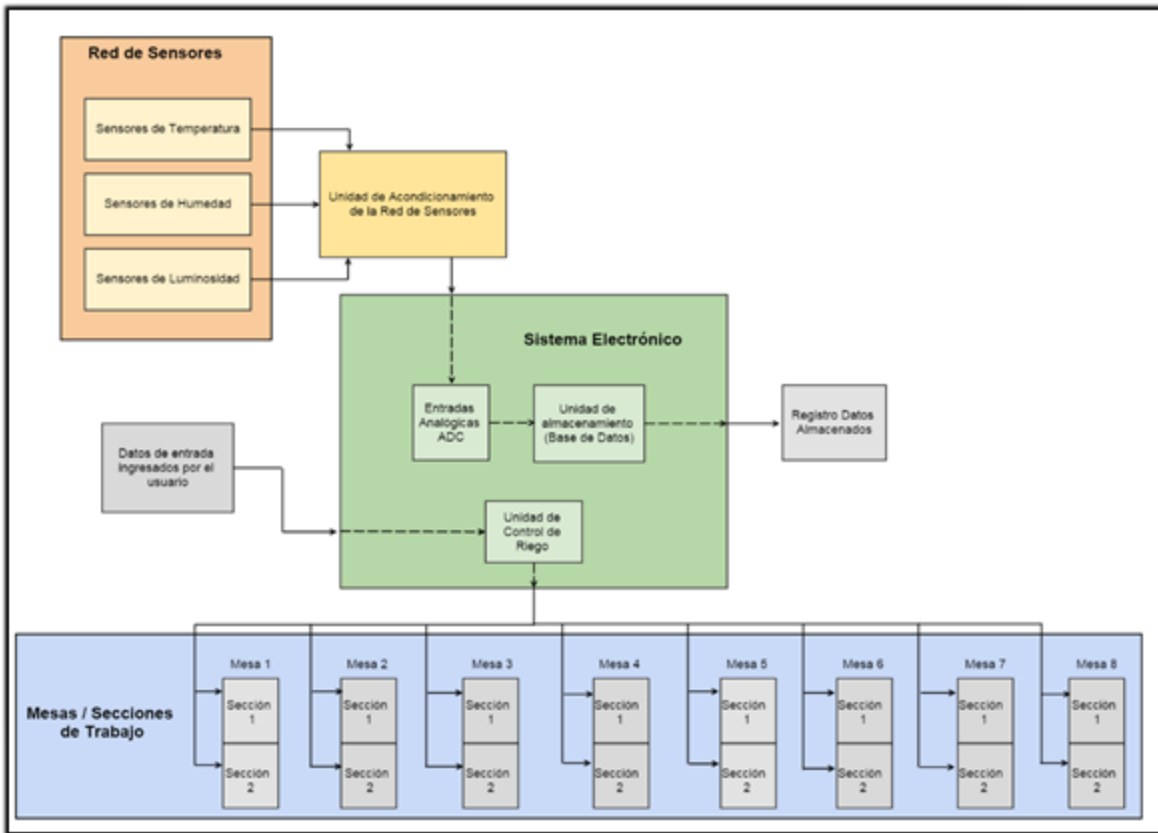


Figura 5.1. Diagrama de bloques del sistema electrónico.

5.1.1 Posibles soluciones

En este apartado se plantean distintas distribuciones de los sensores de temperatura, humedad y luminosidad ubicados por las ocho mesas que contará el invernadero, de manera que se muestren los puntos a favor de la opción a utilizar para diseñar la red de sensores.

Con 12 sensores se puede implementar una red en la que un trío de sensores de temperatura, humedad y luminosidad, se asigna a dos mesas de trabajo en el invernadero como se muestra en la figura 5.2. Tomando en cuenta que los experimentos se realizan en cada sección y no en cada mesa, con este método cada trío de sensores (nodo pasivo) se configura para cuatro secciones.

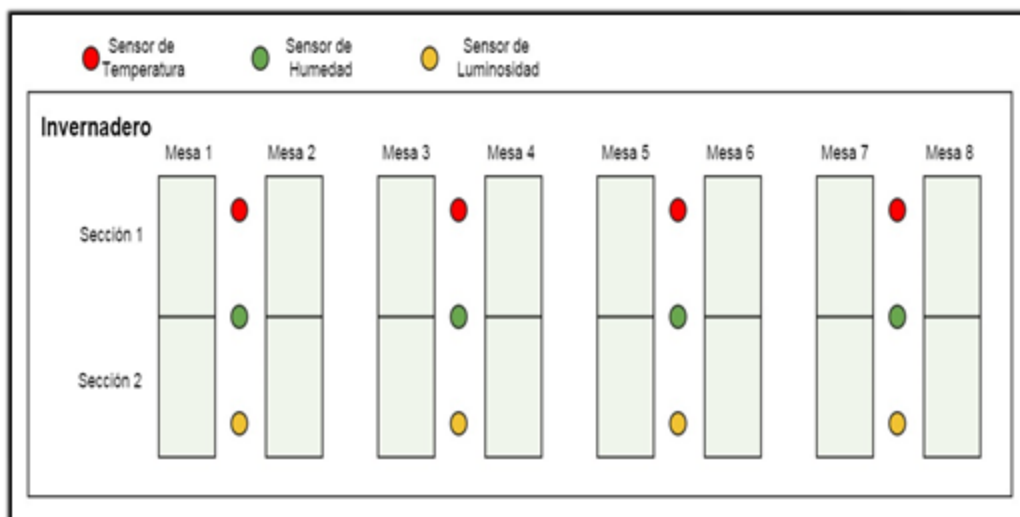


Figura 5.2. Primera propuesta para la distribución de red de sensores.

Al incrementar el número de sensores de la red a 24, se pueden distribuir un trío de sensores por mesa como se muestra en la figura 5.3. Con este nuevo mecanismo la cantidad de secciones asignada a los tríos de sensores es de dos.

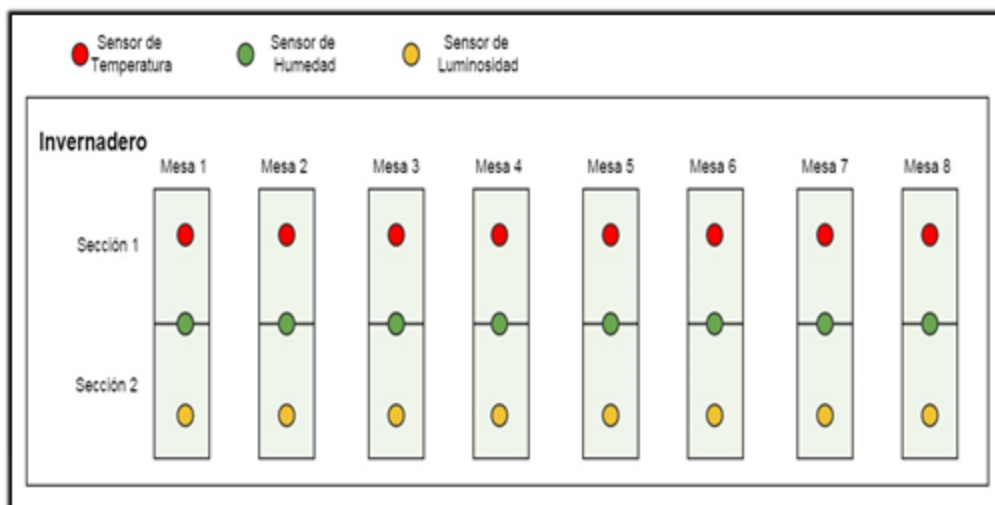


Figura 5.3. Segunda propuesta para la distribución de red de sensores.

Si se estipula un trío de sensores para cada sección, la red aumenta a un total de 48 sensores distribuidos por el invernadero como se observa en la siguiente figura 5.4.

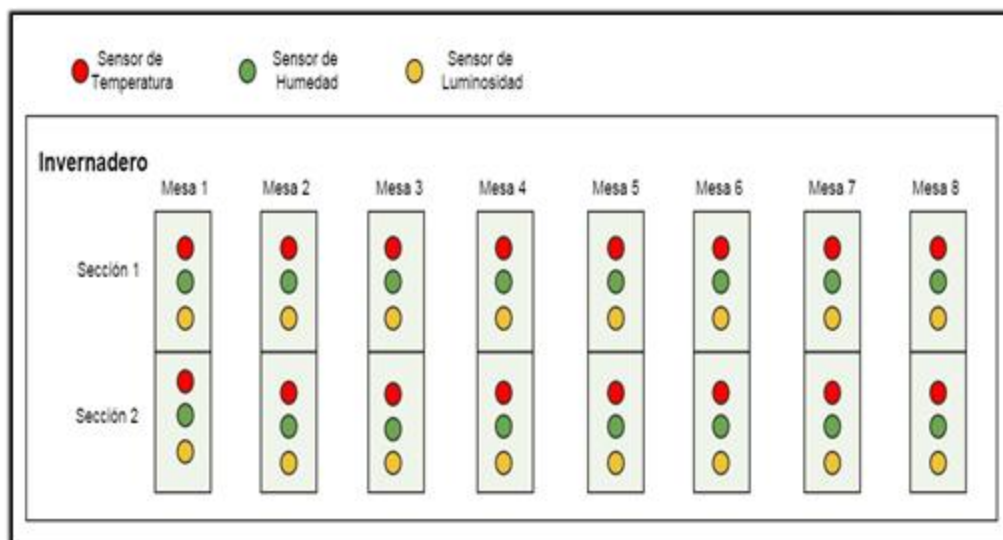


Figura 5.4. Tercera propuesta para la distribución de red de sensores.

5.1.2 Selección de la solución

Después de analizar las tres propuestas mostradas en la sección 5.1.1, se observa que para la primera la cantidad de sensores para distribuir en la red es de 12. Esta distribución presenta una desventaja y es que la luminosidad se mide solo en un extremo de las mesas y esta medición depende de la posición con la que inciden los rayos de luz el invernadero, según la hora del día. Además si se considera el efecto del viento que incide sobre el invernadero, en el extremo de incidencia la temperatura llega a variar con respecto al otro, ya sea que aumente por vientos cálidos o disminuya por vientos fríos o alisios de acuerdo a la estación del año.

Para la segunda propuesta la cantidad de sensores aumenta a 24 pero sigue con la desventaja que los sensores de temperatura se encuentran en las secciones 1 y los de luminosidad en las secciones 2, a pesar de esto cuenta con la ventaja de que distribuye un trío de sensores independientes para cada mesa de trabajo.

Se elige la tercera propuesta de distribución para la red de sensores, con 48 unidades se puede establecer un trío de sensores con cada variable a medir en las 16 secciones de

trabajo. Con esta selección cada experimento que se configura por sección puede mantener total independencia con las demás secciones respecto a la adquisición de datos, además al contar con mayor número de unidades el alcance de los sensores respecto a la ubicación de las plantas por secciones permite monitorizar de cerca el comportamiento y crecimiento de cada especie.

5.2 Selección de los sensores

Para la selección de los sensores a utilizar se consideraron aspectos como la linealidad de la señal de salida, rangos de medición, tensiones de operación, entre otros, con el fin de utilizar los que se ajusten mejor al sistema electrónico y a las condiciones del entorno donde se van a implementar.

5.2.1 Selección del sensor de temperatura

El sensor LM35 fabricado por National Semiconductor es el elegido para medir temperatura en el invernadero, está diseñado para calibrarse en grados Celsius. Puede operar de 4V a 20V, además presenta un factor de escala de $10\text{mV}/^{\circ}\text{C}$ y un rango completo de operación de -55°C a 150°C . Estas características junto al bajo costo económico justifican la selección del LM35. [16]

Un aspecto importante a resaltar es el hecho de que el valor de tensión del canal analógico donde se va a acoplar el sensor limita el rango de operación del LM35, según el factor de escala. En el caso de los microcontroladores PIC y para las placas de Arduino la tensión de los puertos analógicos es de 5V, lo que representa un rango máximo de 500°C , pero para dispositivos con arquitectura ARM como los sistemas embebidos se tiene que la tensión de referencia para el canal analógico es de 1.8V, que simbolizan un valor máximo de 180°C según el factor de escala.

5.2.2 Selección del sensor de humedad

En cuanto a la variable humedad se selecciona el sensor HIH-4030 fabricado por la compañía Honeywell. Un criterio considerado es la linealidad en la tensión de salida respecto al valor de %RH (porcentaje de humedad relativa) medido. Presenta un mejor rendimiento si se utiliza con 5V de alimentación. En la hoja de datos que ofrece el fabricante se brindan datos en la tabla 2 para calibrar el sensor a 25°C. [10]

El rango de operación de este sensor por tratarse de porcentajes varía entre 0 y 100, además el HIH-4030 es un sensor sensible a la luz, por lo que se debe ubicar el dispositivo de manera que la parte superior del circuito integrado no reciba luminosidad para mejor rendimiento.

5.2.3 Selección del sensor de luminosidad

El sensor de luz Phidget 1142 se implementa en el proyecto debido a que presenta gran precisión para la medición de luminosidad en interiores, esto considerando que el invernadero debe llevar un techo de material plástico para evitar que ingrese el agua cuando llueve. Todo esto provoca que la medición no sea en el exterior a plena luz de día, sino que se considera de interior pero a pesar de eso debe presentar un amplio rango de trabajo como el caso de los 1000 lux que tiene de alcance este sensor. [17]

Posee una tensión de salida lineal para representar el rango de medición de 1 lux (Luz de la Luna) a 1000 lux (Iluminación de estudio de televisión o exterior en día nublado). Trabaja con una tensión de alimentación de 2.4V a 5.5V, el valor idóneo donde el sensor presenta el mejor rendimiento son los 5V.

Además posee un rango de operación entre una temperatura mínima de -40°C y una máxima de 85°C. La impedancia de salida del sensor es de 1k Ω y la corriente máxima de consumo es de 5mA. Presenta un tiempo de respuesta máximo de 2ms.

5.3 Caracterización de los sensores

Para traducir la tensión de salida de cada sensor y obtener el valor de temperatura humedad o luminosidad, se deben conocer detalles del puerto analógico como son la tensión y la resolución en bits con el fin de procesar de manera correcta el dato medido por cada sensor.

En el caso de los microcontroladores de Arduino la tensión de entrada analógica se trabaja a 5V y cuenta con una resolución de 10 bits lo que representa 1024 valores distintos para traducir los 5V, pero en el caso de plataformas embebidas como la BeagleBone Black cada puerto analógico opera a 1.8V con una resolución de 12 bits, que simbolizan 4096 distintos valores para representar los 1.8V.

Conociendo los detalles del ADC se procede a utilizar la siguiente ecuación 3 para calcular la tensión que representa cada lectura realizada en el puerto analógico.

$$Tensión = \frac{Valor\ Canal\ ADC * Vcc}{Resolución\ del\ ADC} \quad (3)$$

Para el caso una entrada de 5V y 10 bits de resolución se deriva de la ecuación 3 la siguiente ecuación 4.

$$Tensión = \frac{Valor\ Canal\ ADC * 5}{1024} \quad (4)$$

En el caso de un entrada de 1.8V con 12 bit de resolución de obtiene de la ecuación 3 la siguiente ecuación 5.

$$Tensión = \frac{Valor\ Canal\ ADC * 1.8}{4096} \quad (5)$$

Con la magnitud de tensión que representa cada lectura en el ADC, se procedió a consultar las hojas de datos para buscar la relación tensión/variable específica para cada sensor. El valor de la variable a medir con cada sensor se calcula como se muestra a continuación en la ecuación 6.

$$\text{Variable Sensor} = \text{Tensión} * \frac{\text{RelaciónTensión}}{\text{Variable}} \quad (6)$$

5.3.1 Sensor LM35

Para calcular el valor de temperatura en el LM35 debe utilizarse el factor de escala que viene en la hoja de datos, el cual nos da una relación de tensión con respecto a grados centígrados que se debe de utilizar en la ecuación 6. Este factor de escala es de 10mV/°C, que representa para 1V un valor de 100°C, la relación tensión/variable a utilizar se selecciona de 100. [15]

Como se muestra en la figura 5.5, el factor de escala limita el rango de temperatura que se puede medir de acuerdo al valor del canal analógico a utilizar. Para el caso de utilizar un puerto de 5V se podría medir hasta un rango de 500°C y para el caso de 1.8V se tiene un valor máximo de 180°C, esto es imposible dado a lo que representan esas magnitudes, valores mayores a 35°C se consideran elevados para presentarse en el invernadero, además de que el circuito integrado no resiste temperaturas extremas de 500°C o 180°C, se fundiría antes de realizar la medición.

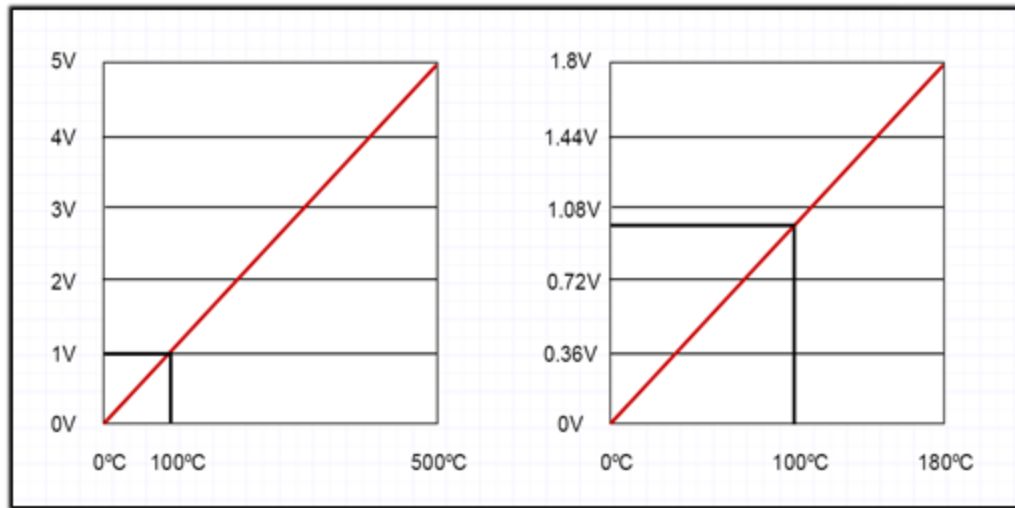


Figura 5.5. Gráficos tensión de salida vs grados Celsius para un canal analógico de 5V (derecha) y 1.8V (izquierda).

Las ecuaciones 7 y 8 detallan el cálculo de temperatura para una entrada analógica de 5V a 10 bits (placas Arduino) y otra de 1.8V a 12 bits (plataformas de desarrollo embebidas) respectivamente.

$$Temperatura = \frac{Valor\ Canal\ ADC * 5}{1024} * 100 \quad (7)$$

$$Temperatura = \frac{Valor\ Canal\ ADC * 1.8}{4096} * 100 \quad (8)$$

5.3.2 Sensor HHH-4030

El valor de humedad %RH (porcentaje de humedad relativa) se obtiene mediante el sensor HHH-4030. Para esto la hoja de datos de este sensor establece una tabla para calibrarlo a 5V y 25°C con un factor de escala de 30.680mV/%RH. La tabla se detalla a continuación y la variación lineal de la tensión de salida se muestra en la figura 5.6.

Tabla 5.1. Información para calibrar el sensor a 5V y 25°C.

Porcentaje de humedad relativa	Valores obtenidos a 5V
Porcentaje de humedad relativa	Tensión de Salida Vout
0%RH	0.958V
75.3%RH	3.268V

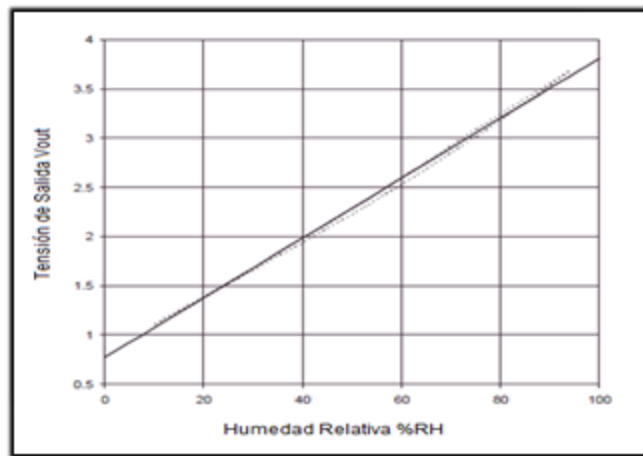


Figura 5.6. Gráfico tensión de salida vs %RH a 25°C y 5V. [10]

Las ecuaciones 9 y 10 detallan el cálculo de humedad para una entrada analógica de 5V a 10 bits como es el caso de las placas de Arduino y microcontroladores PIC. Cabe destacar que para el cálculo de la ecuación 10 se necesita conocer primero la temperatura en grados centígrados.

$$Sensor\ RH = \left(\frac{Valor\ Canal\ ADC * 5}{1024} - 0.958 \right) / 0.0307 \quad (9)$$

$$Humedad = \frac{Sensor\ RH}{1.0546 - 0.00216 * T} \quad (10)$$

Para hacer uso del sensor en plataformas embebidas que trabajan con 1.8V a 12bits las entradas analógicas, se debe de ajustar la ecuación 9 como se muestra en la siguiente ecuación 11.

$$Sensor\ RH = \left(\frac{Valor\ Canal\ ADC * 1.8}{4096} - 0.3348 \right) / 0.0110448 \quad (11)$$

5.3.2 Sensor 1142

Con el sensor de luz Phighet 1142 se posee un alcance máximo de 1000 lux, por lo tanto la relación tensión/variable varía según el valor de entrada del puerto analógico como se observa en la figura 5.7.

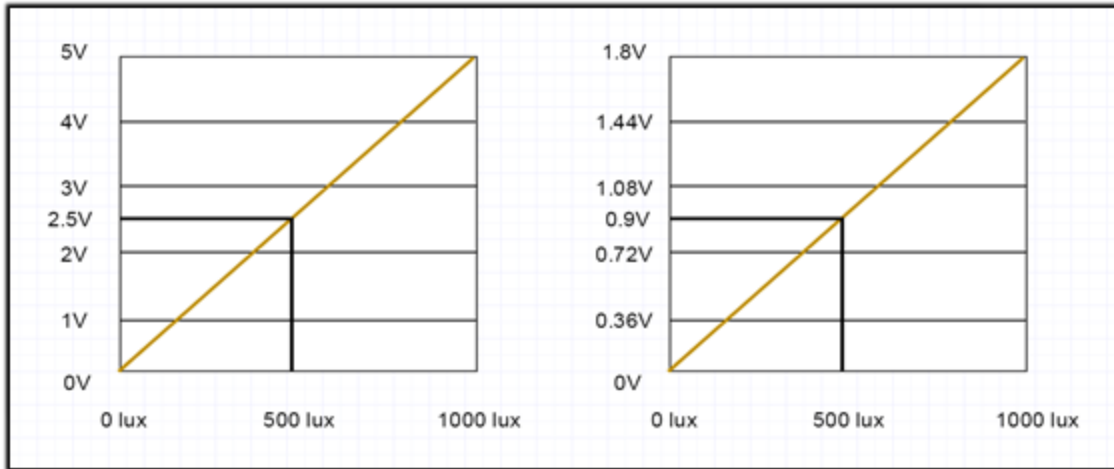


Figura 5.7. Gráficos tensión de salida vs luminosidad para un canal analógico de 5V (derecha) y 1.8V (izquierda).

El cálculo para determinar la luminosidad se muestra en las siguientes ecuaciones 11 y 12 para un puerto de 5V a 10 bits y otro de 1.8V a 12 bits respectivamente.

$$Luminosidad = \frac{Valor\ Canal\ ADC * 5}{1024} * \frac{500}{2.5} \quad (12)$$

$$Luminosidad = \frac{Valor\ Canal\ ADC * 1.8}{4096} * \frac{500}{0.9} \quad (13)$$

5.4 Sistema de procesamiento de datos

Debido a las necesidades del futuro invernadero de contar con 8 mesas de trabajo divididas en 2 secciones, se requieren 16 salidas digitales de control para manipular (abrir /

cerrar) las electroválvulas, por lo que la cantidad de pines digitales de entrada / salida con los que debe contar el sistema de procesamiento central preferiblemente deben ser mínimo de 16.

Otro aspecto a considerar es la cantidad de entradas analógicas que se requieren para acoplar los 48 sensores de la red distribuidos por las secciones. La cantidad de 48 puertos analógicos no se encuentra para la mayoría de microcontroladores PIC, sistemas embebidos y tarjetas de Arduino. El único caso de un sistema de procesamiento expuesto en el marco teórico en la sección 3.5.1.3 que cuente con 16 entradas analógicas es el Arduino MEGA.

Para contrarrestar esta limitación es posible implementar un circuito acondicionador de sensores que disminuya a 3 la cantidad de entradas analógicas, con el fin de dar lectura a tres sensores a la vez, uno por cada variable a medir. En caso de implementar el circuito acondicionar con unidades de multiplexado, se necesitan 4 bits de control que aumentan el número de pines digitales de entrada / salida que requiere el sistema de procesamiento.

La tabla 5.2 detalla las cantidades de sensores a utilizar para cada tipo de variable, además de las cantidades requeridas para las salidas digitales y las entradas analógicas.

Tabla 5.2. Requerimientos para los sensores y el sistema de procesamiento de datos.

<i>Requerimiento</i>	<i>Especificación</i>
Sensores de Humedad	16 unidades
Sensores de Temperatura	16 unidades
Sensores de Luminosidad	16 unidades
Entradas analógicas	3 entradas analógicas para dar lectura a cada tipo de sensor
Salidas digitales	<ul style="list-style-type: none"> ▪ Mínimo 16 pines ▪ Máximo 20 pines

5.4.1 Selección de la unidad de procesamiento central CPU

Considerando los aspectos de diseño detallados en la tabla 5.2 y los sistemas de procesamiento abarcados en el marco teórico del presente documento, se determina que la tarjeta de desarrollo embebida BeagleBone Black Rev C de la compañía BeagleBoard.org se adapta a las necesidades funcionales y económicas de este proyecto.

El sistema operativo Debian GNU/Linux 7.0 con un escritorio LXDE es el elegido para trabajar en la BeagleBone Black por su estabilidad y por contener el mínimo software pre instalado; otras distribuciones de GNU/Linux como Ubuntu por ejemplo vienen con un centro de software y muchos otros programas que no se van a utilizar en este proyecto. Para instalar los programas que si se van a necesitar se debe de recurrir a una consola terminal para introducir los comandos de instalación.

Esta plataforma trabaja con un procesador AM3359 ARM Cortex A8 con una frecuencia de operación de 1GHz, además de componentes de hardware integrados como puertos USB, interfaz de red, salida de video microHDMI, 512MB de memoria DRAM, memoria interna de 4GB, entre otros.

Cabe destacar que en esta placa embebida es posible cargar un sistema operativo desde una memoria microSD lo que brinda una gran versatilidad para implementar la base de datos y el servidor local que se emplea para el desarrollo de la página web. También la plataforma cuenta con 96 pines para una variedad de funciones, entre los cuales 65 son GPIO (“Entrada/Salida de Propósito General” por sus siglas en Inglés) y 7 entradas analógicas. Esto permite una variedad de opciones para desarrollar el sistema de adquisición de datos y el sistema de riego programado.

5.4.2 Propuestas para la unidad de acondicionamiento de la red de sensores

Para complementar las funciones específicas externas al sistema de procesamiento central, se presentan dos propuestas de diseño que se pueden emplear para desarrollar el sistema electrónico. La primer propuesta consiste en utilizar tres Arduino MEGA como unidad de acondicionamiento de los sensores, este dispositivo al contar con 16 entradas analógicas se configura para recibir de entrada los 16 sensores de temperatura distribuidos por las 8 mesas con doble sección, de igual manera se realiza el mismo procedimiento para los 16 sensores de humedad y para los 16 sensores de luminosidad, con 3 de estas unidades es posible acondicionar los 48 sensores de la red en el invernadero. En la siguiente figura 5.8 se muestra el diagrama para acondicionar los sensores.

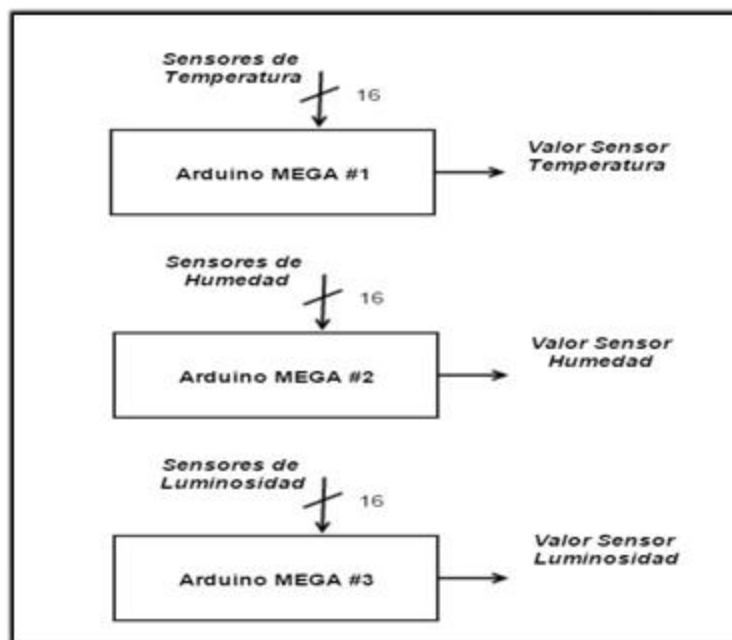


Figura 5.8. Primera propuesta para la unidad de acondicionamiento de la red de sensores.

La otra propuesta consiste en diseñar un circuito de selección basado en circuitos digitales de multiplexado que poseen canales de entrada analógicos, esto da la posibilidad de seleccionar con señales digitales las entradas analógicas de los sensores de la red por

temperatura, humedad y luminosidad. La salida de esta unidad de multiplexado es de 3 líneas que se deben conectar a los puertos analógicos del sistema de procesamiento.

Se debe de utilizar un multiplexor de 16 entradas y 4 bits de control para los 16 sensores de temperatura y otros dos más para los de humedad y luminosidad como se ejemplifica en la figura 5.9.

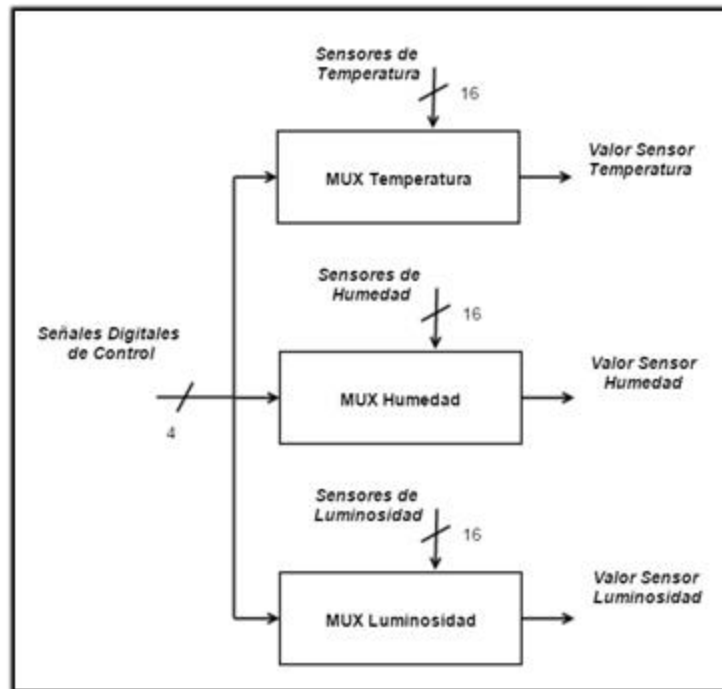


Figura 5.9. Segunda propuesta para la unidad de acondicionamiento de la red de sensores.

5.4.3 Selección de la unidad de acondicionamiento de la red de sensores

Una de las ventajas que presenta la primer propuesta es que al contar con 3 Arduino MEGA se tiene 3 unidades de procesamiento más, las cuales se pueden comunicar por los puertos serie para el proceso de envío de los datos de los sensores. Su limitación es el costo económico asociado a la función que se está cumpliendo con estas herramientas.

La segunda propuesta es la elegida debido a la versatilidad que presenta para acoplar un gran número de sensores. En este caso la limitación es que el sistema de procesamiento

central debe de disponer de 4 pines digitales de salida GPIO para asignar a las señales de control utilizadas para seleccionar los sensores en los multiplexores.

Como en este diseño se especificó que el invernadero dispondrá de 8 mesas divididas en 2 secciones (16 secciones en total), es suficiente trabajar con multiplexores de 16 entradas y 4 bits de control. Las 3 salidas de los multiplexores se deben conectar a 3 canales analógicos de la plataforma embebida BeagleBone Black que dispone de un total de 7.

La tabla 5.3 resume la características valoradas para la selección de la unidad de acondicionamiento de la red de sensores, en esta se presenta el porcentaje asociado a cada característica y el valor asignado a cada una de las dos propuestas de solución. La segunda solución al ser la elegida es la que presenta la mayor ponderación.

Tabla 5.3. Características valoradas para la selección de la unidad de acondicionamiento de la red de sensores.

<i>Características</i>	<i>Porcentaje (%)</i>	<i>Solución 1</i>	<i>Solución 2</i>
Funcionalidad	50	50	50
Escalabilidad	15	13	12
Bajo costo económico	25	10	23
Tiempo de desarrollo	10	5	10
Ponderación	100	78	95

5.5 Sistema de almacenamiento de datos

El sistema de almacenamiento de datos se puede dividir por etapas como se hace referencia en el presente apartado. La unidad de acondicionamiento para la red de sensores acopla la distribución de los sensores por mesas y secciones, el sistema de adquisición de datos se encarga de definir las combinaciones lógicas para seleccionar los sensores en el circuito de acondicionamiento y a su vez de acoplar las salidas de esta unidad en los puertos

analógicos para su respectiva lectura, por último se tiene la base de datos que es la que se encarga de almacenar toda la información tomada por los sensores en las secciones donde se han configurado experimentos.

5.5.1 Unidad de acondicionamiento de la red de sensores

Esta unidad utiliza un circuito de 3 multiplexores CMOS 4067B para seleccionar un trío de sensores conformado por las variables temperatura, humedad y luminosidad, asociados a una sección de trabajo en específico del invernadero. Esto se puede efectuar dado que el diseño acopla todos los sensores de temperatura en un mismo multiplexor y realiza el mismo mecanismo para los sensores de humedad y luminosidad con el fin de tener un control por tipo de sensor.

Cada multiplexor contiene 16 canales de entrada donde se asignan los sensores para cada variable a medir. La distribución de los sensores para las entradas de esta unidad se muestra en la tabla a1.1 del apéndice 1. La simbología “ST_XX” hace referencia a “Sensor Temperatura _ #Mesa #Sección”, por poner un ejemplo “ST_42” se relaciona con “Sensor Temperatura Mesa 4 Sección 2”. Este mismo procedimiento se emplea para nombrar los sensores de humedad SH_XX y los sensores de luminosidad SL_XX

La salida de esta unidad serán 3 señales analógicas que representan los valores de temperatura, humedad y luminosidad medidos en la sección consultada por medio de los 4 bits de control A, B, C, D. Como se ha comentado anteriormente con este mecanismo es posible adaptar una red de 48 sensores para monitorizar por el sistema electrónico, donde estos sensores se reparten en 3 segmentos de 16 conectados en las entradas de los multiplexores según el tipo de sensor.

A nivel gráfico, se muestra en la figura 5.10 la unidad de acondicionamiento con los bits de control para los 4 selectores que se conectan en paralelo entre las señales de los 3 multiplexores, con el fin de seleccionar las mismas entradas en los 3 dispositivos en cada consulta. Cabe resaltar que en la figura se muestran todas las conexiones de los

multiplexores, a excepción de la polarización de los chips, para el caso de los CMOS 4067B el pin para la tensión VCC es el número 24 y el de la referencia GND el número 12.

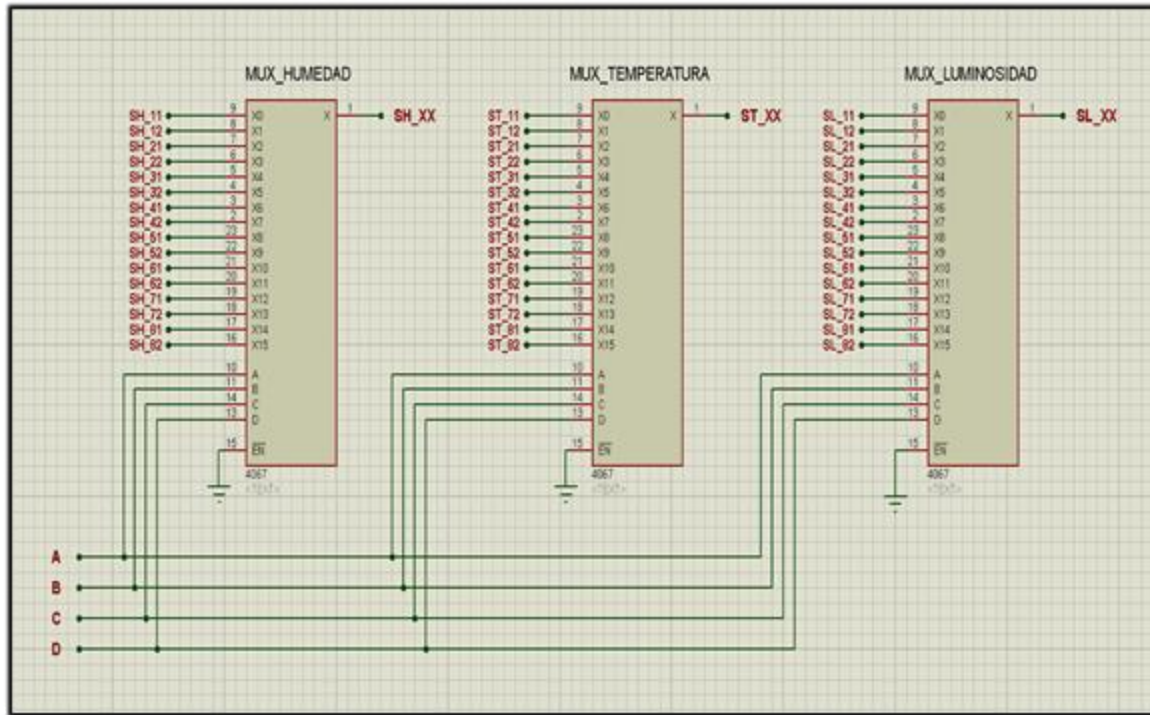


Figura 5.10. Circuito de acondicionamiento de la red de sensores.

Las 16 combinaciones lógicas para seleccionar los sensores dentro de la unidad de acondicionamiento se muestran en la tabla a1.2 en el apéndice 1. Las señales digitales de control A, B, C y D son las entradas de esta unidad de acondicionamiento.

El bit de control representado por la señal A es el menos significativo y el de la señal D el más significativo. Para cada multiplexador las entradas con número par se asignan a las mesas con sección 1 y las entradas con número impar se establecen a las mesas con sección 2.

5.5.2 Sistema de adquisición de datos para la red de sensores

El sistema de adquisición de datos consiste en la unidad de acondicionamiento de la red de sensores junto al sistema de procesamiento, que se encarga de asignar las 4 señales digitales de control según la ejecución del programa principal. El sistema puede acceder a los sensores asignados para cada sección de trabajo y almacenar en variables los datos que estos entregan.

El sistema de procesamiento BeagleBone Black posee las conexiones de los GPIO a cada extremo de la tarjeta llamados P8 y P9, es de suma importancia tener esto presente para evitar conectar señales en puertos erróneos. Para diferenciarlos la placa posee escritos estos nombres. En la siguiente figura 5.11 se presenta un esquemático de una vista superior a la plataforma con la distribución de los pines por cada cabecera P8 y P9, se puede observar que cada cabecera consiste en dos líneas de pines donde a una se le asignan los nombres de pines par de izquierda a derecha en orden ascendente y a la otra los pines impar de igual manera.

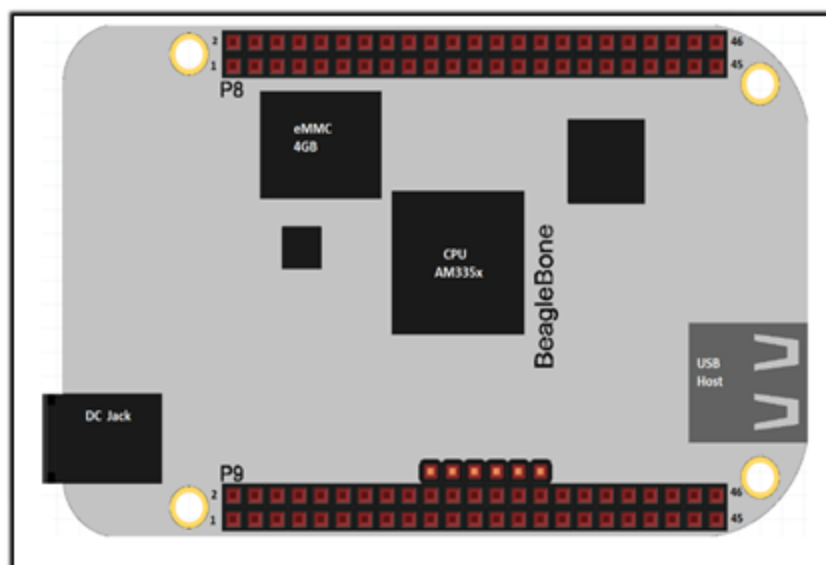


Figura 5.11. Distribución de los pines por las cabeceras P8 y P9.

Para cada bit de control en los selectores de los multiplexores, se asigna un número de GPIO como se muestra en la tabla a1.3 del apéndice 1. En la figura 5.12 se observan las

señales digitales de control A, B, C y D de color verde que salen de la BeagleBone Black y entran a la unidad de acondicionamiento, además de las salidas de esta unidad (líneas naranjadas) que se conectan a las entradas analógicas en los puertos AN0, AN1 y AN2. La asignación de los pines para estos puertos analógicos se detalla en la tabla a1.4 en el apéndice 1.

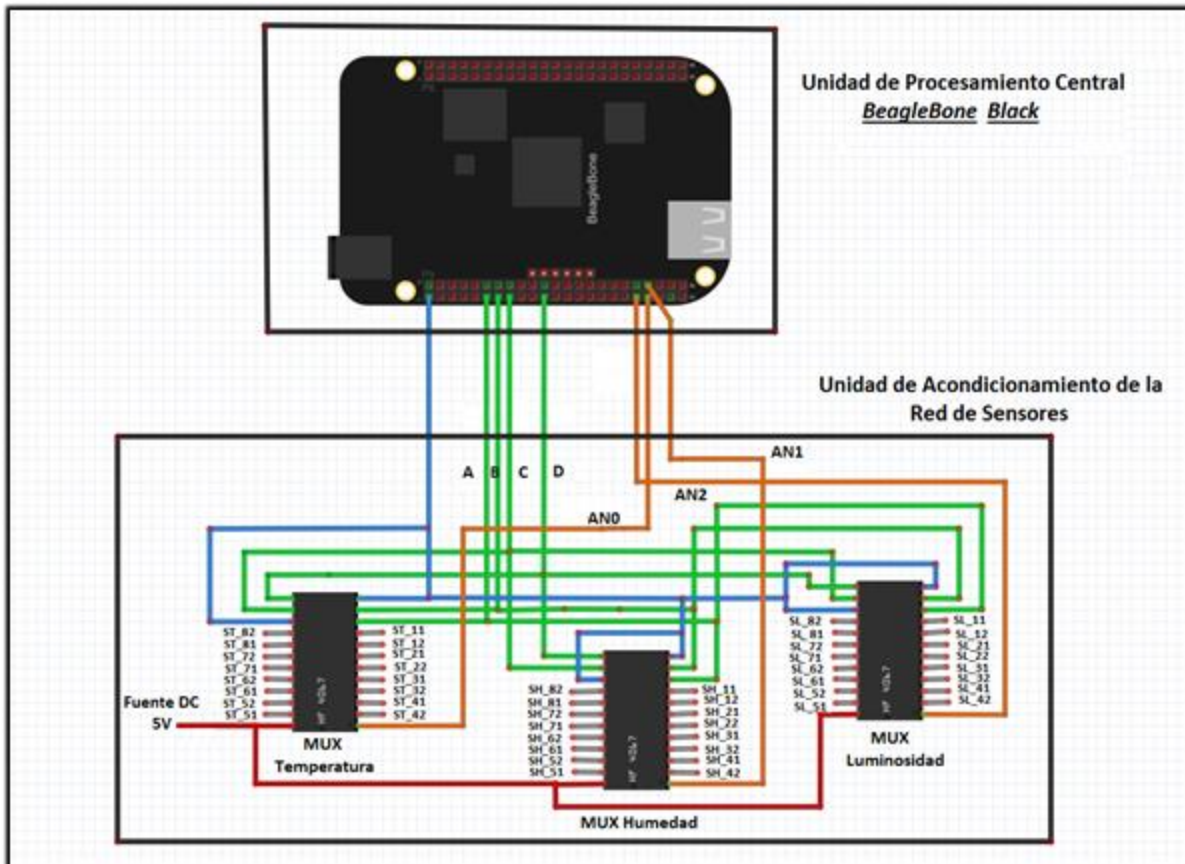


Figura 5.12. Diagrama del sistema de adquisición de datos de la red de sensores.

Los sensores a utilizar presentan un rango de tensión de entrada en el que pueden funcionar, sin embargo 5V es la tensión recomendada para trabajar con estos 3 tipos de sensores, esto implica que la salida de los multiplexores debe pasar por un circuito divisor de tensión, para reducir la magnitud a un valor de 1.8V máximo a conectar en los canales analógicos de la BeagleBone Black.

Para los sensores de temperatura a pesar de polarizarse a 5V, no es necesario implementar un circuito divisor debido a que su factor de escala es de 1mV/°C, esto representa para valores de salida de 1V una medición de 100°C, que se considera extrema para presentarse dentro del invernadero, por lo que el rango de medición está bien cubierto con una tensión de 1.8V (180°C). Los circuitos divisores para las señales de salida de los multiplexores de humedad y luminosidad se muestran en la siguiente figura 5.13.

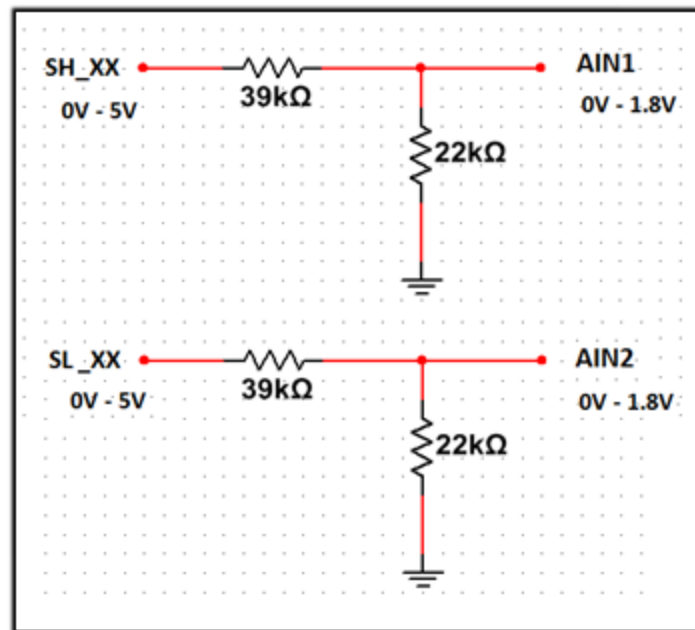


Figura 5.13. Divisores de tensión para acoplar la señal de los sensores.

Los valores de 39kΩ y 22kΩ seleccionados para las resistencias, sirven para dividir los 5V máximos de entrada que salen de cada multiplexor a 1.8V, con esto se puede abarcar todo el rango del canal analógico ante las variaciones de entrada de 0V hasta 5V. La ecuación 13 detalla el cálculo de la tensión de salida del circuito divisor.

$$AIN(1/2) = (0V - 5V) * \frac{22k\Omega}{22k\Omega + 39k\Omega} \quad (13)$$

La siguiente figura 5.14 presenta el diagrama del sistema de adquisición de datos con conexiones a los sensores de la mesa 1 y sección 1 en específico, también ejemplifica el acople de los circuitos divisores en los multiplexores de humedad y luminosidad.

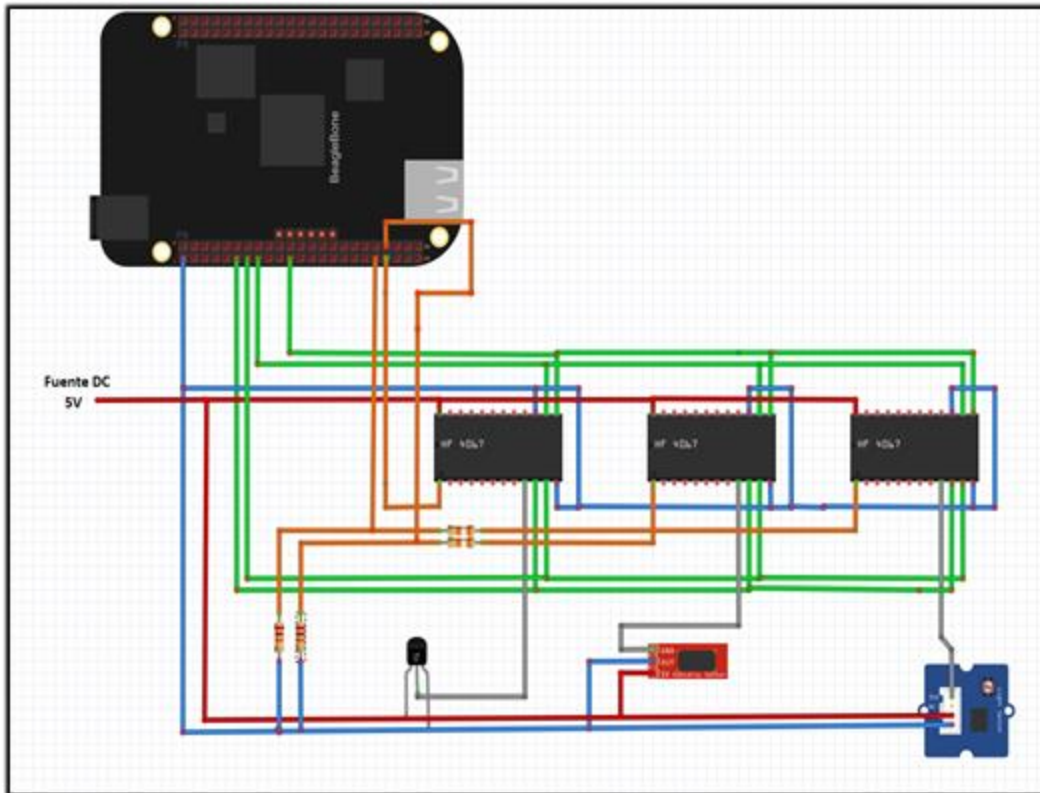


Figura 5.14. Sistema de adquisición de datos con un trío de sensores conectados a una sección específica.

5.5.3 Desarrollo de la base de datos

Para el desarrollo de la base de datos se utiliza el software llamado MySQL Workbench, el cual permite el modelado de datos, desarrollo de código SQL, administración de usuarios, configurar y administrar las bases de datos con el servidor web, entre otras funciones. El comando de instalación se encuentra en el apéndice 3.2, en este apéndice 3 se encuentran los comandos de instalación de programas para Debian GNU/Linux 7 utilizados en el proyecto.

Una de las razones por la selección de este programa es que cuenta con versiones para distribuciones de Linux, también por ser una herramienta visual que ofrece diseñar Diagramas de Entidad Relación “DER” para modelar los datos de un sistema de información. En estos diagramas se implementan tablas con que poseen ciertas relaciones asociadas a los requerimientos de la base de datos. Cada tabla posee atributos donde se pueden almacenar datos del tipo carácter, entero, flotante, entre otros para administrar el tipo de información que se registra. En la figura 5.15 se observa el diagrama entidad relación que se utiliza para implementar la base de datos del sistema de almacenamiento.

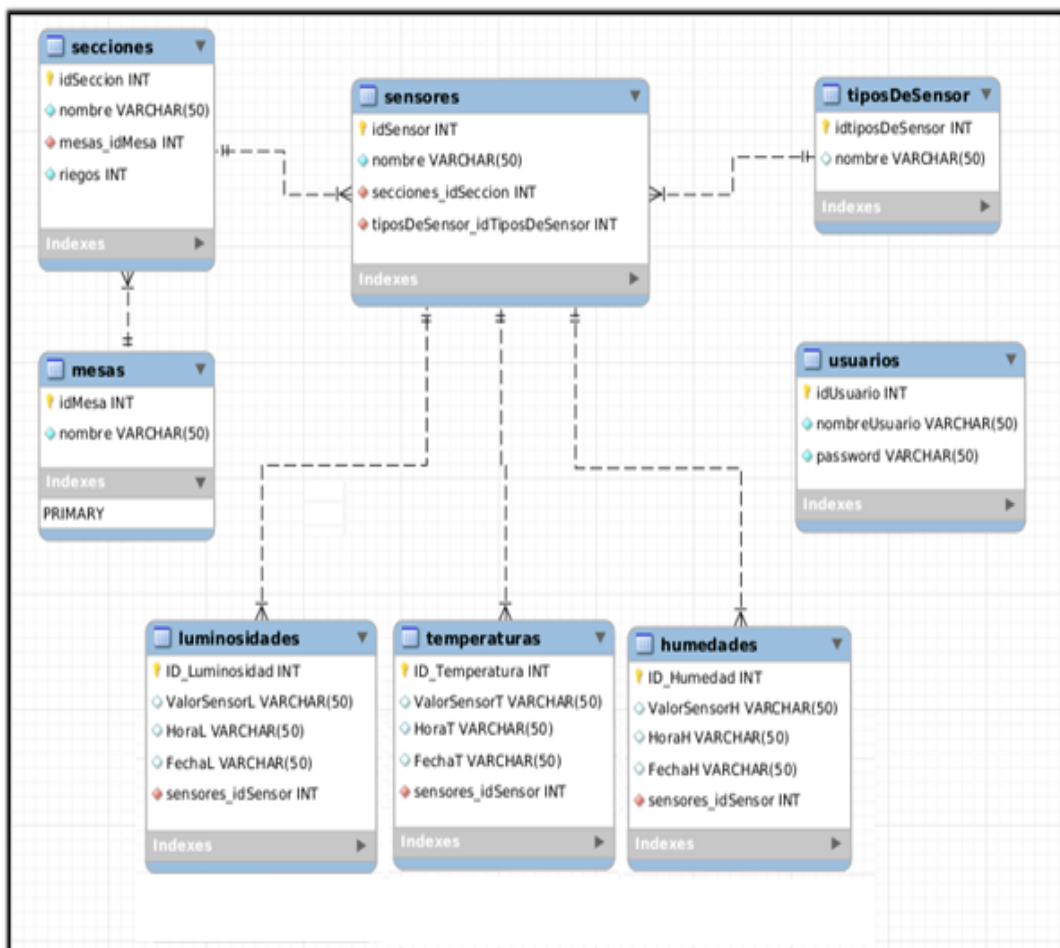


Figura 5.15. Diagrama de Entidad Relación DER utilizado para desarrollar la base de datos.

5.5.3.1 Distribución de las tablas en el diagrama entidad relación DER

Para implementar la base de datos se diseñaron ciertas relaciones en las tablas utilizadas en el DER, esto con el fin de cumplir las necesidades y requerimientos para almacenar la información.

El sistema de almacenamiento debe ser capaz de registrar un valor de temperatura, humedad y luminosidad por cada una de las 16 secciones de trabajo, para esto se implementaron diversos métodos en cada tabla que se detallan a más adelante. Cabe destacar que las tablas usuarios, mesas y tiposDeSensor deben de registrar la información desde la base de datos en el programa MySQL Workbench, para esto se debe utilizó el código mostrado en el apéndice 5.2.3. Las demás tablas registran la información desde la página web que se desarrolla en el apartado 5.7 de este documento. En el apéndice 5.1 se detallan todos los componentes y atributos de estas tablas.

- **Tabla usuarios:** Esta tabla se utiliza para almacenar un usuario con una contraseña, se debe de registrar cada nuevo usuario desde la base de datos MySQL.

- **Tabla mesas:** En la tabla mesas se registran el nombre de 8 mesas distintas desde la base de datos MySQL. Deben de crearse estas 8 mesas para que el sistema funcione correctamente.

- **Tabla secciones:** En esta tabla se almacena la información del nombre de la sección, el indicador de la mesa a la cual se va a asociar la sección y la cantidad de riegos que se realizan por día.

- **Tabla sensores:** La tabla sensores se encarga de generar 3 distintos nombres de acuerdo al tipo de sensor y al número de mesa y sección. Con esto a cada sensor del trío de las 16 secciones se le asignan nombres distintos.

- **Tabla tiposDeSensor:** La tabla tiposDeSensor se utiliza para agregar los nombres “temperatura”, “humedad” y “luminosidad” que permiten identificar el tipo de sensor, se deben de registrar desde la base de datos MySQL como las tablas usuarios y mesas.

- **Tabla humedades:** En la tabla humedades se registra el valor de humedad junto a hora y fecha a la que se toma el dato.

- **Tabla luminosidades:** En la tabla luminosidades se almacena el valor de luminosidad con la hora y fecha a la que se registra el dato.

- **Tabla temperaturas:** En la tabla temperaturas se registra el valor de temperatura además de la hora y fecha a la que se almacena el dato.

5.5.3.2 Procedimientos almacenados en la base de datos

Los procedimientos almacenados conocidos por sus siglas en inglés SP “Stored Procedure” se utilizan para que el manejo y el flujo de datos que se realizan dentro de la base de datos desarrollada. En otras palabras los procedimientos almacenados son tipos de funciones que se agregan a la base de datos con el fin de ser llamados para manipular la información de cada tabla. Se implementan en lenguaje SQL y se deben agregar en la base de datos a trabajar. A las tablas de la base de datos se les puede agregar y eliminar la información por medio de los procedimientos almacenados.

Para el diseño de la base de datos se utilizan 9 procedimientos almacenados, de los cuales 8 son para agregar datos en las 8 tablas mesas, secciones, sensores, tiposDeSensor, temperaturas, humedades, luminosidades y usuarios, además de un último procedimiento para eliminar toda información en las 8 tablas asociada a un experimento de una sección que sea indicado. El código implementado para la elaboración de estos procedimientos se observa en el apéndice 5.2.2.

A continuación se describe la función general de cada uno de los procedimientos almacenados utilizados en la base de datos del sistema de almacenamiento.

- **insertarUsuario:** agrega un nombre de usuario con una contraseña en la tabla usuarios de la base de datos.
- **insertarMesa:** agrega un nombre a cada mesa en la tabla mesas.
- **sp_crearSeccion:** agrega un nombre a cada sección relacionada a un nombre de mesa en la tabla secciones. En este procedimiento almacenado por ser el encargado de agregar un el nombre de la sección donde se va a implementar un nuevo experimento, se decidió agregar un atributo extra “riegos” que se encarga de registrar el intervalo de tiempo para regar la respectiva sección. Este atributo riegos es de tipo entero y es requerido en el sistema de riego que se detalla más adelante.

- **sp_agregarSensores:** agrega un trío de nombres para cada sensor de temperatura, humedad y luminosidad, asociados a un nombre de sección y a un nombre de mesa.
- **sp_agregarTemperatura:** agrega un valor del sensor de temperatura con la hora y fecha que se dio la medición en la tabla temperaturas, relacionado a un nombre de sensor, un nombre de sección y un nombre de mesa.
- **sp_agregarHumedad:** agrega un valor del sensor de humedad con la hora y fecha que se dio la medición en la tabla humedades, relacionado a un nombre de sensor, un nombre de sección y un nombre de mesa.
- **sp_agregarLuminosidad:** agrega un valor del sensor de luminosidad con la hora y fecha que se dio la medición en la tabla luminosidades, relacionado a un nombre de sensor, un nombre de sección y un nombre de mesa.
- **sp_borrarSeccion:** elimina un experimento activo que se encuentre registrado en la base de datos. Para esto el presente procedimiento almacenado se encarga de eliminar los datos asociados a la sección que se indica para borrar en cada una de las 7 tablas en las que se almacena la información. En la tabla usuarios no se agregan datos de los experimentos activos por sección.

5.5.4 Algoritmos de funcionamiento

Se desarrollan diversos algoritmos utilizados para agregar, eliminar y consultar información a la base de datos con el fin de manipular posteriormente el flujo de datos desde el programa principal. También se diseña el algoritmo principal para el sistema de almacenamiento.

En el diagrama principal del sistema de almacenamiento se emplean dos variables principales de control, \$cont registra la cantidad de experimentos activos por sección y

\$cont2, se utiliza como un contador para ir agregando la información en las tablas temperaturas, humedades y luminosidades de una manera creciente hasta que se llegue al número final de experimento activo.

Además para los algoritmos de agregado, consulta y eliminado de información de la base de datos, se implementan los procedimientos almacenados que por medio de llamadas del tipo “call” manipulan la información por tablas desde la base de datos implementada. Esto permite trabajar sobre la base de datos con una simple llamada a algún procedimiento haciendo referencia al mismo con la cantidad de atributos que posee.

Para dar un ejemplo el procedimiento `sp_crearSeccion` (`nombreSeccion varchar(50)`, `nombreMesa varchar(50)`, `numeroRiegos int`) posee tres atributos, por lo que para poder realizar una correcta llamada se debe escribir “call `sp_crearSeccion('Mesa 1','Seccion2','6')`,” que almacena en la base de datos específicamente en la tabla `secciones` a la mesa 1 asociada a la sección 2 y con una cantidad de 6 riegos por día.

En el algoritmo de agregar información se toma la variable del tipo entero “`riegos`”, que indica la cantidad de riegos a realizar por día en la mesa y sección especificada. Para consultar la información se crea la variable `$ms`, que almacena los índices del número de mesa y sección para poder nombres posteriormente las variables `$tms`, `$hms` y `$lms`, estas almacenan los nombres de temperatura, humedad y luminosidad con los correspondientes índices a consultar.

Se muestran en las siguientes 4 figuras los algoritmos implementados en el sistema de almacenamiento de datos. La figura 5.16 detalla el algoritmo principal, en la figura 5.17 se observa el algoritmo para agregar información, las figura 5.18 y 5.19 ejemplifica el diagrama de flujo para consultar y eliminar información respectivamente.

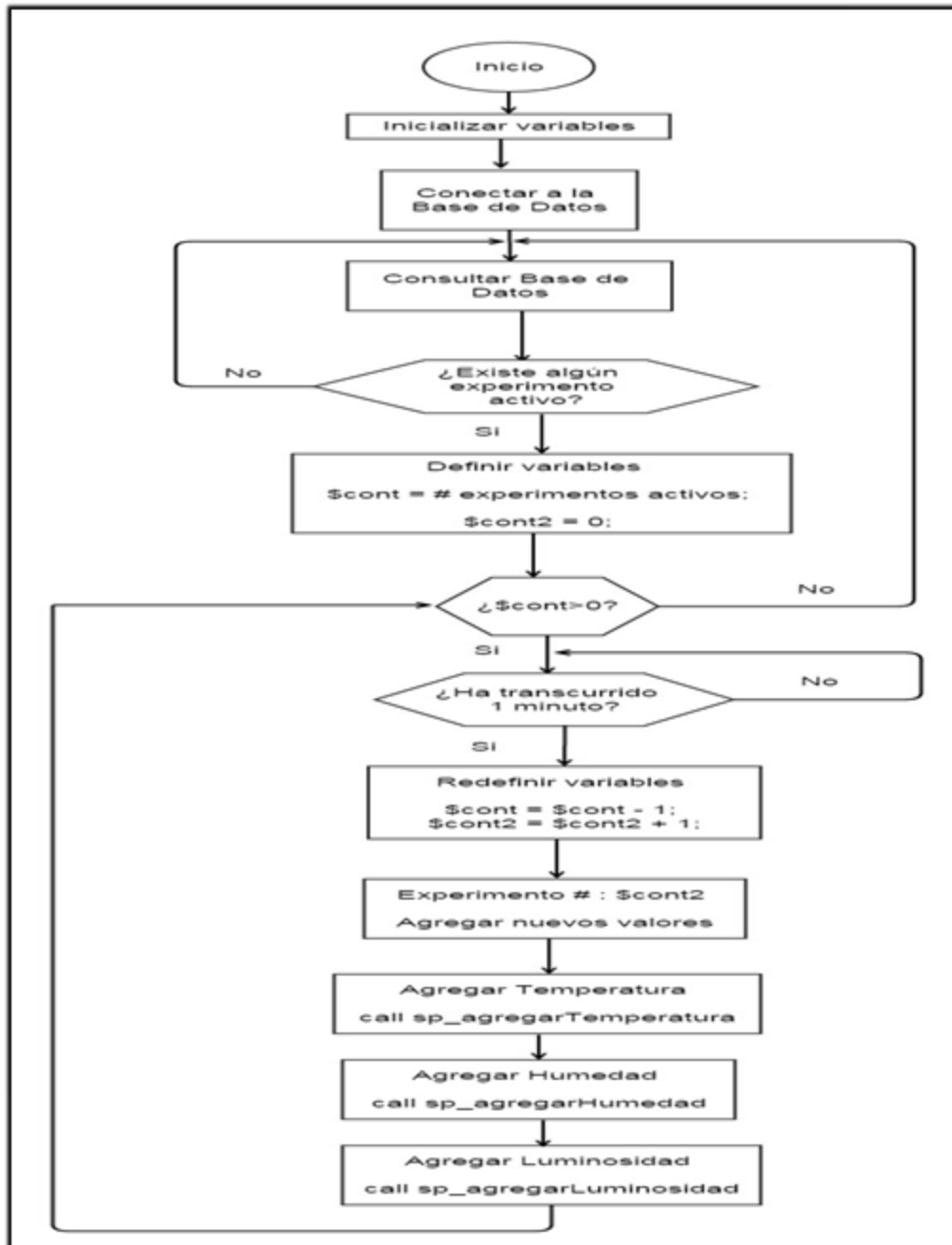


Figura 5.16. Diagrama de flujo principal utilizado para el sistema de almacenamiento.

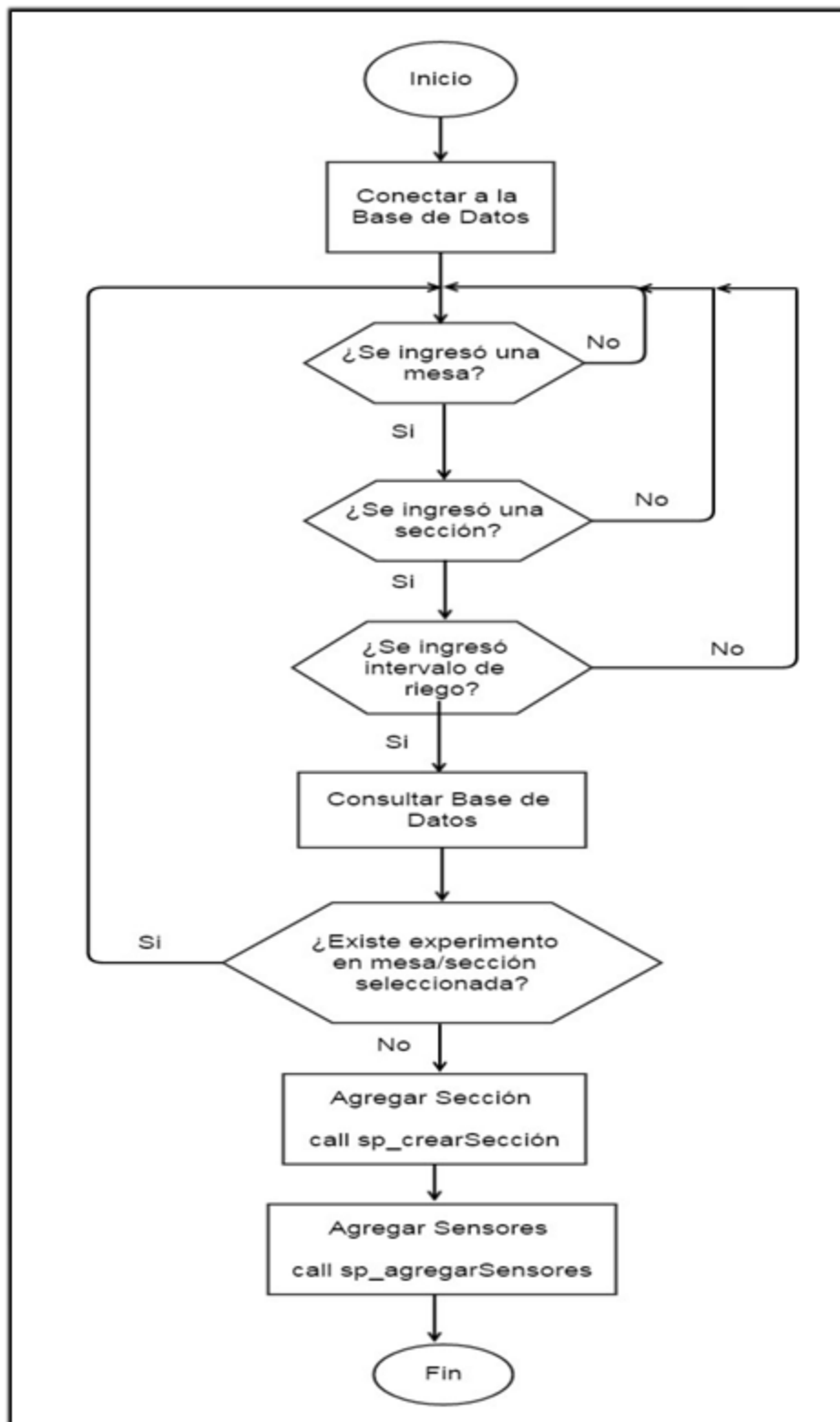


Figura 5.17. Diagrama de flujo utilizado para agregar información a la base de datos.

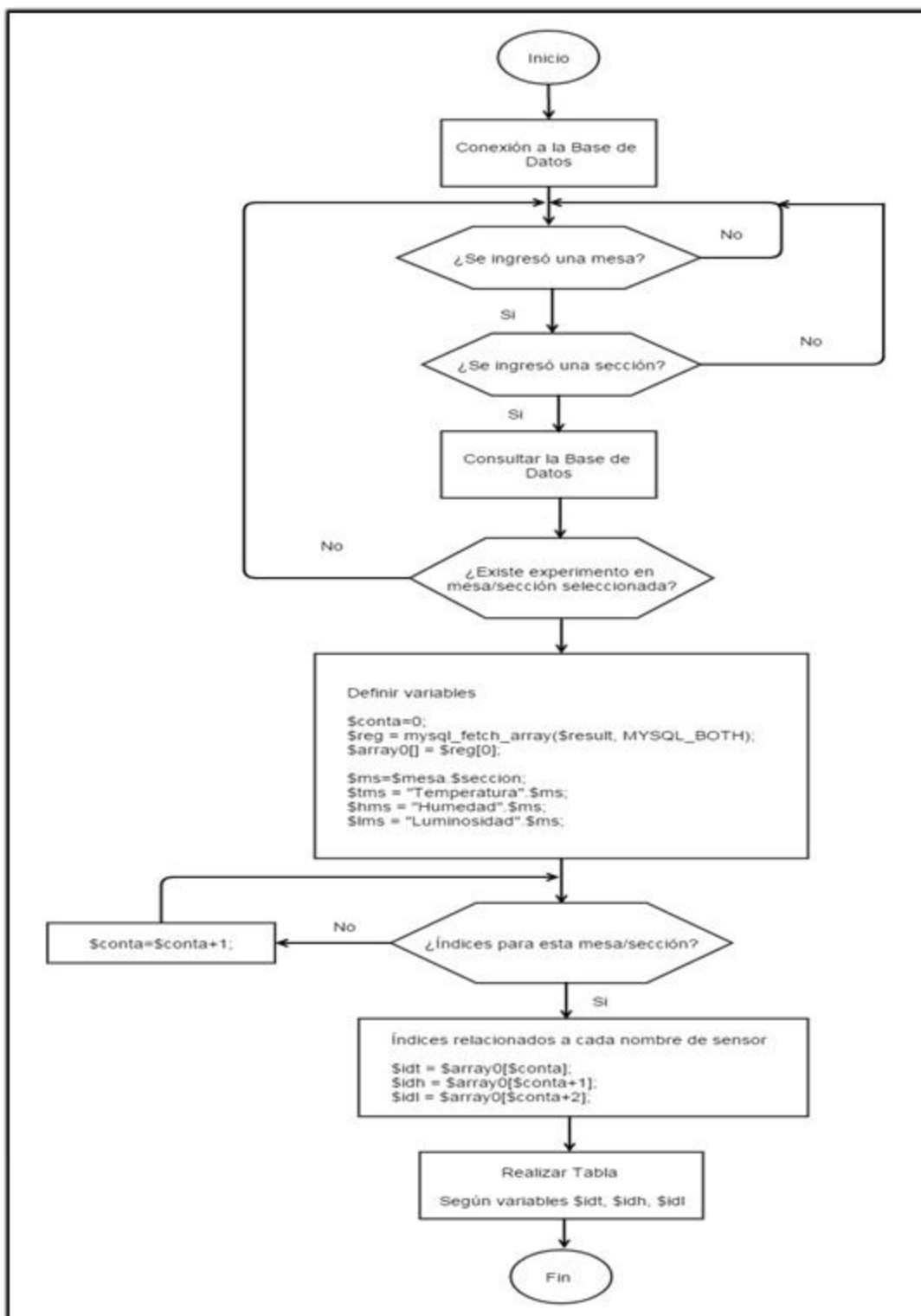


Figura 5.18. Diagrama de flujo utilizado para consultar información de la base de datos.

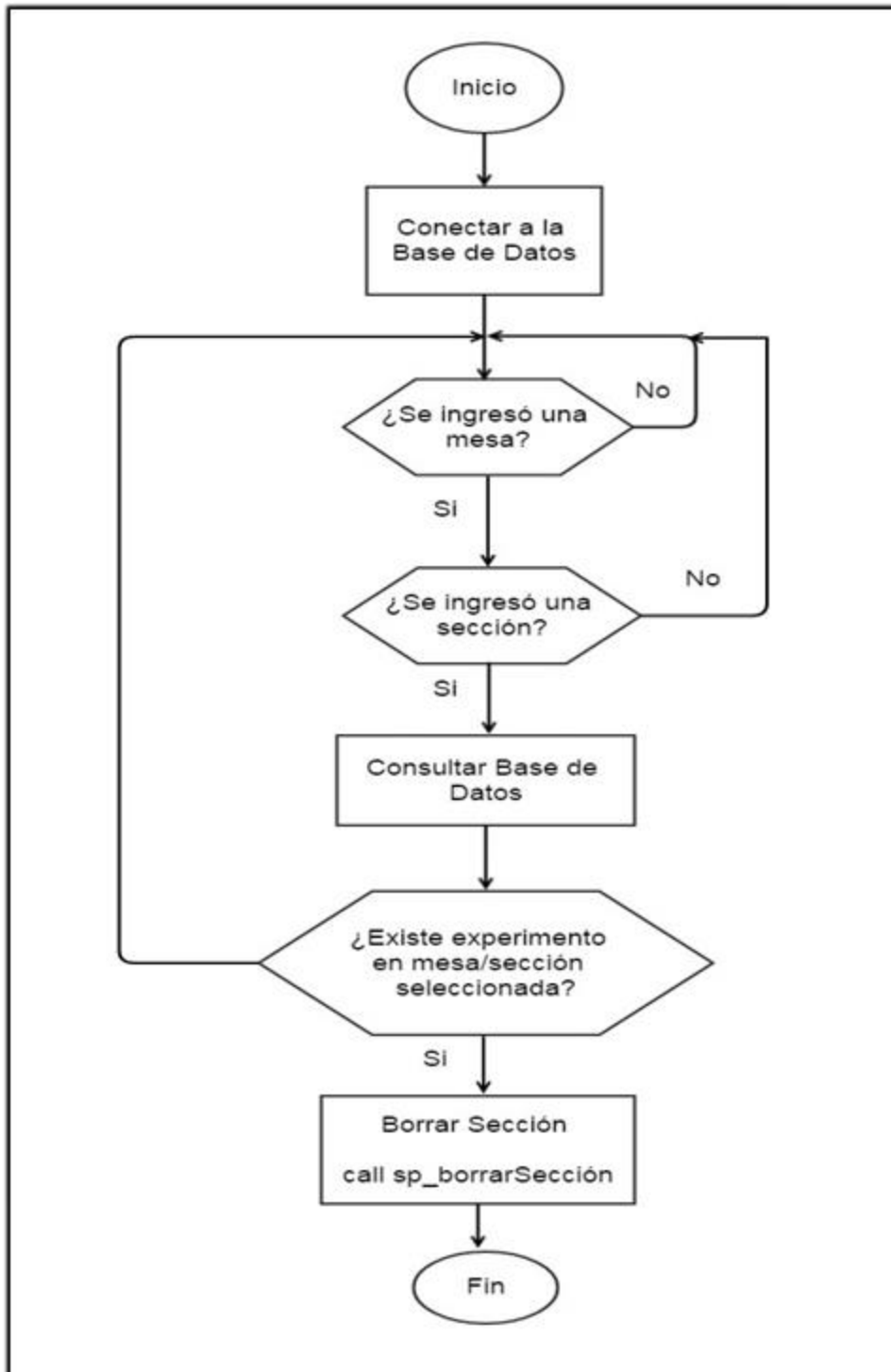


Figura 5.19. Diagrama de flujo utilizado para eliminar información de la base de datos.

5.6 Sistema de riego programado

Las electroválvulas a implementar en el presente proyecto son del tipo DC (corriente directa) y trabajan a 12 V. Las piezas a utilizadas son de la compañía “U.S SOLID”. Una de las razones para escoger las válvulas solenoides DC es su simplicidad a la hora de adaptarse a los relé de 12V, dispositivos de conmutación que permiten activar y desactivar el paso de corriente eléctrica a las electroválvulas por medio de una señal de control.

Para implementar los riegos en el invernadero, la BeagleBone Black Rev C dispone de 16 GPIO asignados como salidas digitales, que sirven como señales de control para activas las electroválvulas distribuidas por la totalidad de secciones de trabajo. La tabla 3.4 en el marco teórico señala que los GPIO funcionan a una tensión de 3.3V, por lo que se debe de adaptar un circuito de potencia para manipular las electroválvulas debido a que estas operan a 12V y los 3.3V de cada “entrada/salida de propósito general” son insuficientes tanto de tensión como de corriente eléctrica.

En la tabla a1.5 del apéndice 1 se muestran la información de los GPIO utilizados en el sistema de riego y en la figura 5.20 se observa la distribución de estos pines en la plataforma embebida BeagleBone Black por cada mesa y sección.

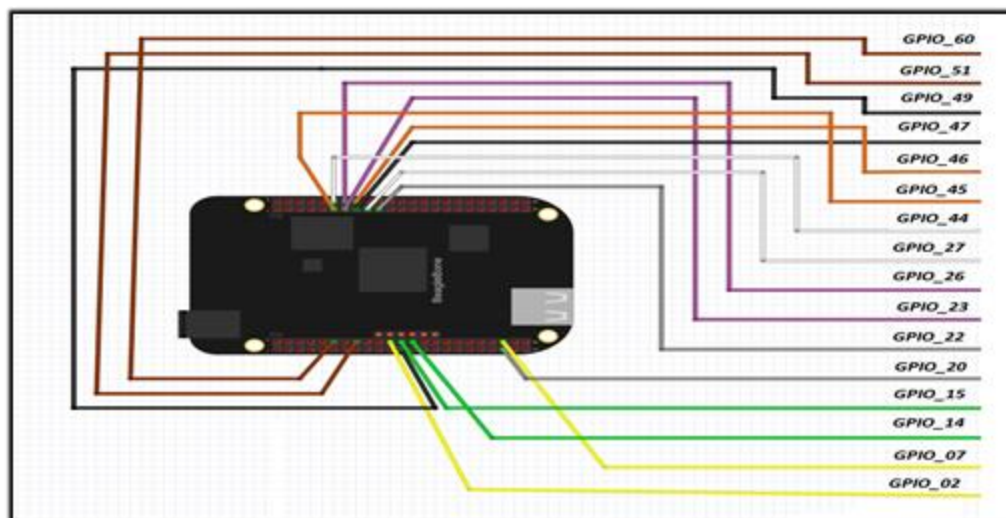


Figura 5.20. Distribución de los GPIO para cada mesa/sección del invernadero

Cabe destacar que 6 de los GPIO disponibles para utilizar como entradas y salidas digitales quedan disponibles. Esto representa una ventaja a la hora de que el sistema electrónico deba ser capaz de trabajar con más de 8 mesas y 2 secciones, dado que puede asignar más salidas de control para manipular válvulas solenoides en nuevas secciones.

5.6.1 Circuitos de potencia para la manipulación de las electroválvulas

En el desarrollo de los circuitos de potencia para las válvulas solenoides se utilizó un módulo de relés de 12V para 16 canales, de tal forma que a las 16 secciones que se van a disponer en el invernadero, se les asignan un relé específico que acopla los 12V a cada electroválvula. Los dispositivos a implementar son de modo normalmente cerrado, por lo que requieren una polarización de 12V para abrir el paso de agua. Este módulo SainSmart se observa en la figura 5.21.

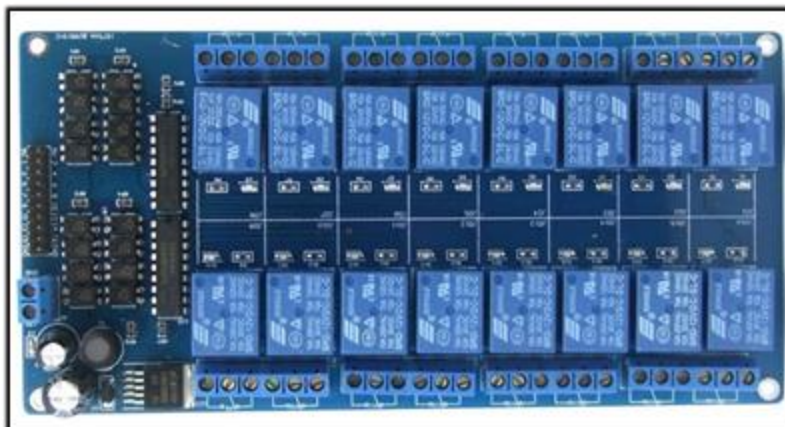


Figura 5.21. Módulo de relés de 12V para 16 canales SainSmart.

El módulo además posee 2 circuitos integrados Toshiba ULN2803A que sirven para la interconexión entre circuitos digitales de bajo nivel lógico y corrientes/tensiones de mayor magnitud requeridas por lámparas, relés o cargas similares. En este caso se emplean para interconectar las señales de los GPIO con los relés y electroválvulas que trabaja a 12V. También dispone de un regulador de tensión LM25765 integrado en la plataforma, el cuál funciona con una tensión de salida de 5V.

Un aspecto a considerar es que las salidas de 3.3V de los GPIO se deben de convertir a 5V por medio de circuitos con transistores NPN 2N4401, dado a que las entradas de bajo nivel del módulo de 16 relés así lo requiere. Estas entradas trabajan con lógica inversa por lo que requieren de 0V para activar los relés y 5V para desactivarlos.

En la figura 5.22 se detalla el circuito para convertir a 5V la señales de los 16 GPIO, se deben de conectar 16 circuitos de estos en cada GPIO, en la figura 49 sólo se ejemplifica sólo la conexión para la mesa 1 y sección 1 en el GPIO_02. El circuito consta de un transistor NPN 2N4401, una resistencia de 1k Ω en la base del transistor y otra resistencia de 120 Ω en el colector conectada en el otro extremo a los 5V de VCC.

Cuando la entrada GPIO_02 está en 0V no fluye corriente por la base del transistor lo que provoca que este no se polarice y tampoco fluya corriente por el colector, todo esto causa que la salida en el colector se conecte directo a VCC. Para el caso en que la señal GPIO_02 cambie a 3.3V la base del transistor se polariza y fluye una corriente por el colector que provoca que la salida del circuito sea el valor de V_{CE} (Colector-Emisor). La magnitud de V_{CE} es menor a 1V.

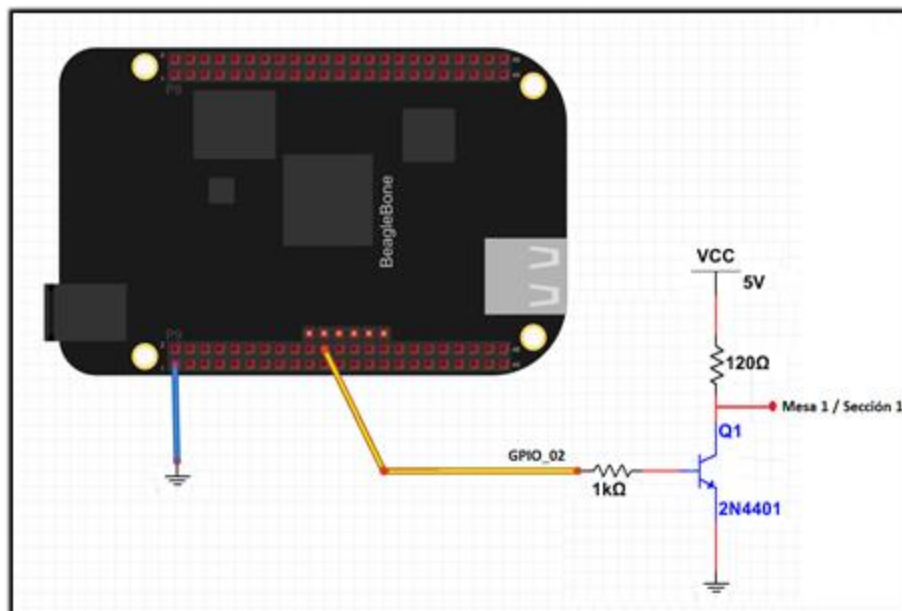


Figura 5.22. Circuito convertidor de tensión de 3.3V a 5V para la mesa 1 y sección 1.

Las 16 salidas de los circuitos convertidores de tensión se conectan a las entradas de bajo nivel del módulo de 16 canales de relés SainSmart, la tabla a1.6 en el apéndice 1 muestra la distribución para estas conexiones. Un aspecto a resaltar es que la referencia GND de las entradas de bajo nivel y la referencia de la entrada de alimentación de 12V deben de estar conectas, de lo contrario el funcionamiento de los relés no va a ser el esperado.

Después de conectar todas las salidas de los circuitos convertidores, queda por acoplar las 16 válvulas solenoides, para esto se utilizan dos de los tres pines de salida de cada relé. Cada relé tiene dos modos de funcionamiento, si se utiliza el pin del centro junto al pin L se tiene el modo de operación de circuito abierto, en el caso contrario de conectarlo al pin R se obtiene el modo circuito cerrado. Para este proyecto se utiliza el modo de funcionamiento de circuito abierto, dado a que se ocupa tener desconectada la electroválvula y cuando la señal de control actúa se conecta para ejecutar un riego en el invernadero.

La siguiente figura 5.23 muestra las conexiones para el módulo de 16 relés. La plataforma debe de polarizarse con 12V en corriente directa DC, las 16 entradas de bajo nivel se acoplan como se designa en la tabla a1.6, los relés utilizan los pines L directo a los 12V y los pines del centro se conectan en serie con las válvulas solenoides, que continúan en serie hacia la referencia. Cada relé posee dos leds indicadores para los modos de funcionamiento, para la configuración de circuito abierto empleada se tienen los indicadores con la letra L y el número de entrada a la cual pertenece el relé.

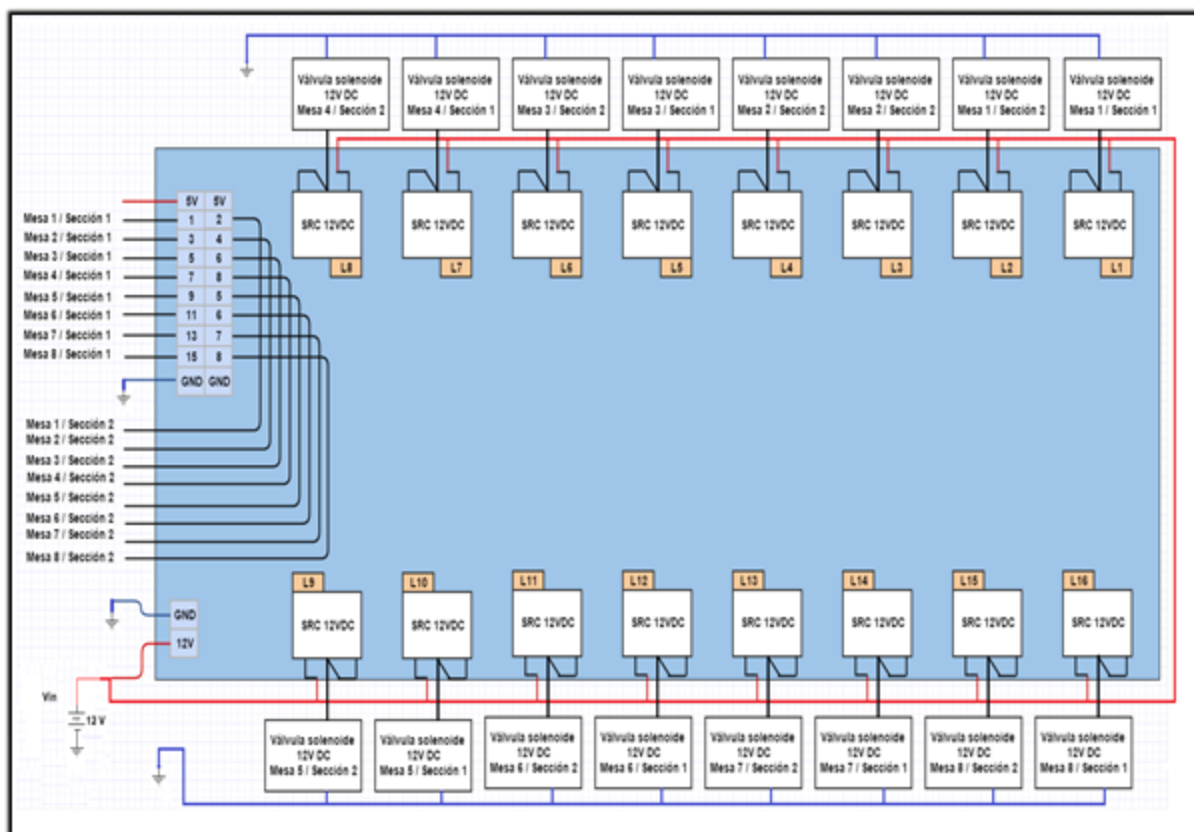


Figura 5.23. Acoples requeridos para el módulo de relés.

5.6.2 Distribución de tiempos de riego

La distribución de los tiempos de riego para cada sección de trabajo en el invernadero, se realiza por intervalos de minutos y horas. Para los intervalos por minuto se asigna a cada mesa y sección los mismo tiempos de riego a excepción de la mesa 1 y la mesa 2 que presentan diferentes tiempos de riegos para las secciones. En cuanto a los intervalos de riego por horas se diseña de manera que cada una de las 16 secciones tenga diferentes combinaciones de tiempos de riego. Todas estas combinaciones para tiempos de riego se realizan para evitar que en casos de uso máximo, no se tengan que abrir todas las electroválvulas a la vez debido a que puede causar problemas de presión en las tuberías. La tabla a1.7 del apéndice 1 presenta la distribución de los tiempos de riego para todas las secciones de trabajo.

5.6.3 Algoritmo de funcionamiento

El algoritmo para el sistema de riego programado utiliza una variable \$conta para almacenar la cantidad de experimentos activos y realizar los respectivos riegos durante 30 segundos. En la figura 5.24 se observa el diagrama de este algoritmo.

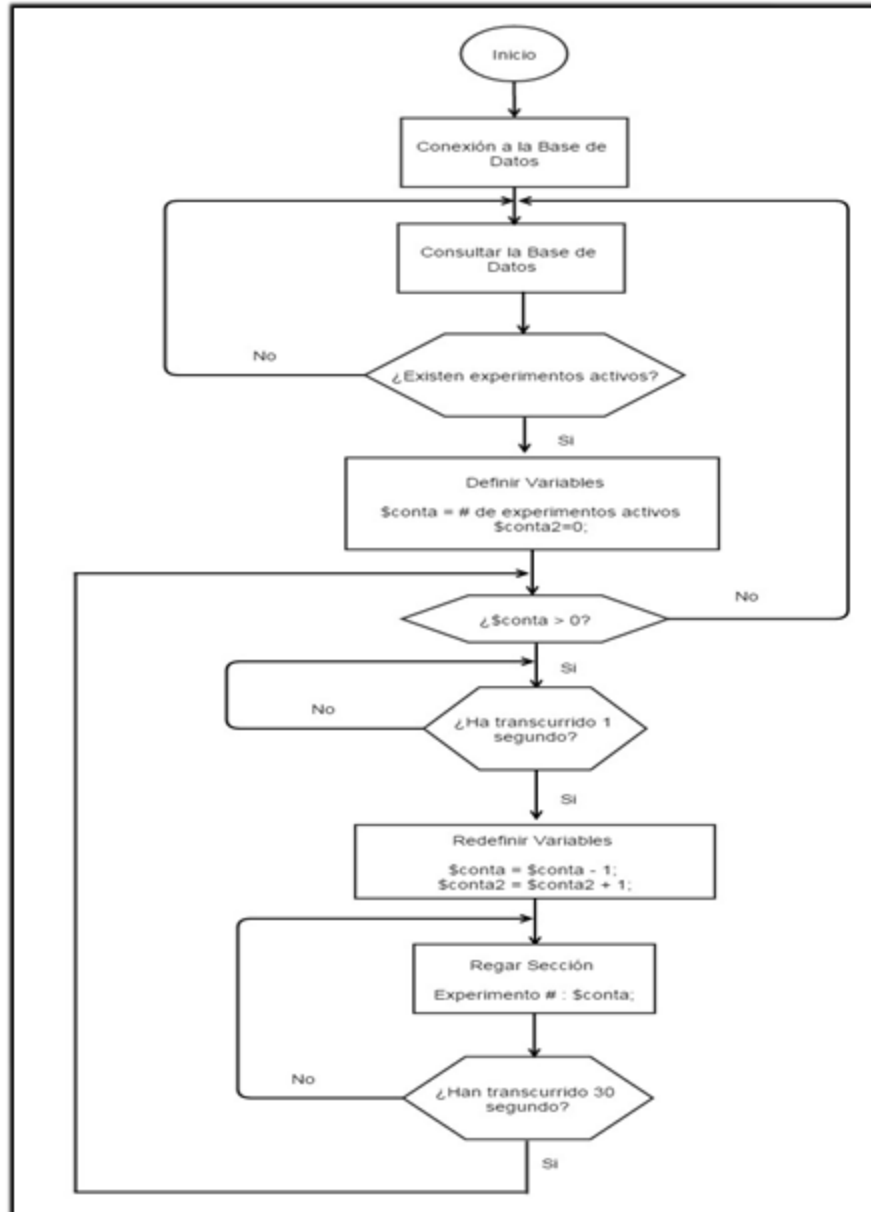


Figura 5.24. Diagrama de flujo principal para utilizado para el sistema de riego programado.

5.7 Diseño de la página web

Una página web termina siendo un conjunto de archivos de diversas extensiones, que se alojan en un servidor web (LAMP), para esto se debe instalar en el sistema operativo Debian GNU/Linux 7 las herramientas Apache y PHP como se muestra en el apéndice 3. Una vez instalados se pueden alojar archivos en la carpeta con ubicación `/var/www/`. El servidor web se puede conectar con otros equipos por medio del puerto 80, estos pueden acceder a la página índice editando desde un buscador web la dirección `http://X.X.X.X:8080/`. Las X se deben a que la dirección IP varía de acuerdo la red donde se trabaje.

Se requiere de una base de datos por la necesidades del proyecto de ir aumentando en número de experimentos o disminuyendo sea el caso, además para almacenar toda la información proveniente de la red de sensores. Para interactuar con la base de datos el servidor web recibe una solicitud de un equipo web cliente en forma de un archivo PHP, este archivo pasa al intérprete de PHP que lo procesa y realiza las modificaciones o consultas a la base de datos necesarias, también envía de vuelta al cliente web una respuesta en forma de página web dinámica al haber manipulado la información en la base de datos. La siguiente figura 5.25 muestra el proceso para recibir páginas web dinámicas.

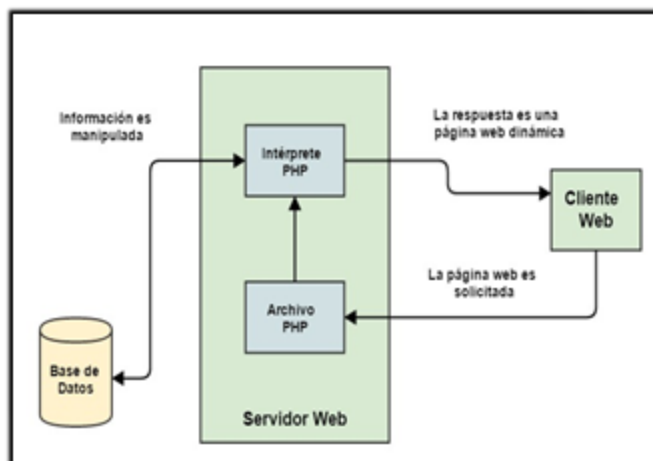


Figura 5.25. Diagrama del proceso de interacción de PHP con una base de datos MySQL.

5.7.1 Distribución de las pantallas

Para el diseño de la página web se implementa una primera página en el índice principal del servidor web para el inicio de sesión. Esta página se encarga de consultar en la base de datos si la información que ingresa el usuario existe, para dar la autorización de acceder a la página principal o realizar una denegación, esta pantalla se muestra en la figura 5.27.

En la página principal “dashboard” se pretende disponer de pestañas en las cuales se redireccione a otras páginas cuando se configura, consulta o elimina un experimento para dar el aviso correspondiente y la acción que conlleva en la base de datos desarrollada en MySQL. También en esta pantalla principal se diseñó una sección para mostrar la tabla de experimentos activos, además del control de riego y la monitorización de temperatura, humedad y luminosidad. La página principal se observa en la figura 5.28. El funcionamiento de estas pantallas y la interacción con la base de datos se muestran en un diagrama en la siguiente figura 5.26.

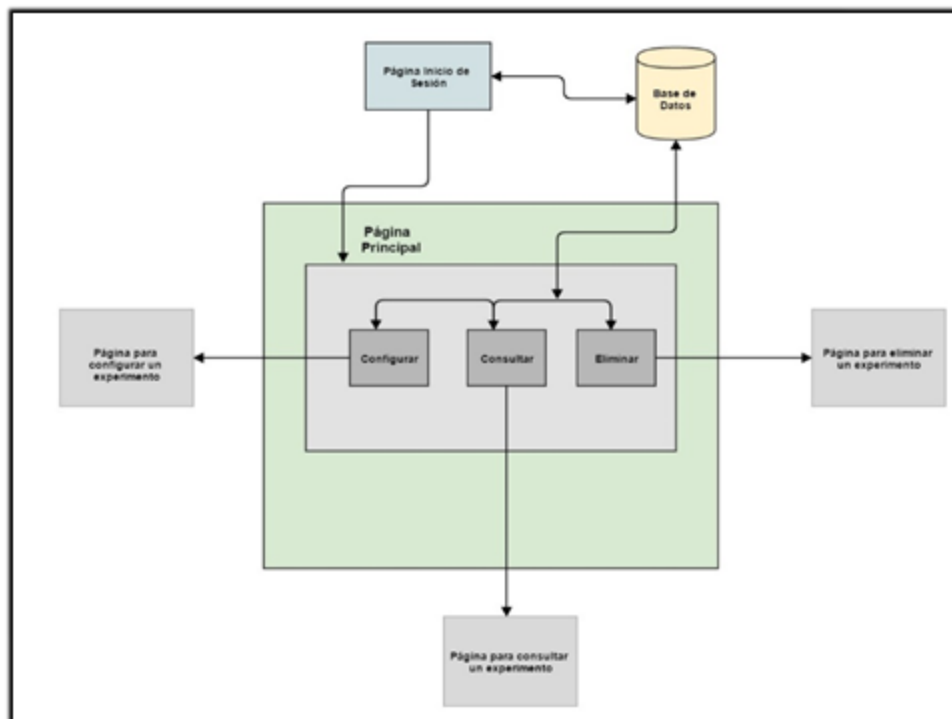


Figura 5.26. Diagrama del funcionamiento de las páginas web requeridas.

Instituto Tecnológico de Costa Rica
Centro de Investigación en Biotecnología
Invernadero de Investigación

Iniciar Sesión

Usuario
Administrador

Contraseña
.....

Ingresar

Figura 5.27. Página de inicio de sesión.

TEC | Tecnológico de Costa Rica

Centro de Investigación en Biotecnología
Invernadero de Investigación

21:14:46

Acción Configurar Consultar Eliminar



El Centro de Investigación en Biotecnología CIB es la unidad de investigación de la Escuela de Biología del Tecnológico de Costa Rica. El centro de investigación se fundó en 1990 y se encuentra ubicado en la sede central de Cartago.

Tabla de Experimentos Activos

Experimento	Mesa	Sección	Intervalo de Riego	Riego por Día
No se han configurado experimentos.				

Figura 5.28. Página principal.

En la página principal en el apartado de las pestañas se realizan las acciones de configurar, consultar y eliminar experimentos, estas manipulan o consultan la información en la base de datos y despliegan otras pantallas con los resultados obtenidos. En la pestaña configurar se puede seleccionar un número de mesa y sección, además de un intervalo de tiempo de riego por minutos o horas definidos en el programa. Si el número de mesa y sección no están registrados en la base de datos, se puede crear un nuevo experimento con los datos seleccionados, en caso de que el usuario no seleccione datos en las 3 viñetas el sistema no configura un nuevo experimento. La figura 5.29 presenta la pestaña configurar, la figura 5.30 la pestaña consultar y la figura 5.31 la pestaña eliminar.

Las pestañas consultar y eliminar se diseñan para que el usuario pueda seleccionar un número de mesa y sección, si los datos seleccionados representan un experimento activo se muestra una tabla con la información de los sensores o se elimina el experimento de acuerdo a la pestaña elegida.

Un aspecto que se consideró como restricción de diseño es que la selección de una mesa, sección e intervalo de tiempo para el riego, sólo le permite al usuario marcar una casilla de cada tipo. Con esto se asegura que no van a ingresar datos erróneos en el momento que el sistema realiza las llamadas a los procedimientos almacenados en la base de datos desarrollada en MySQL.



Figura 5.29. Pestaña para configurar experimentos.



Figura 5.30. Pestaña para consultar experimentos activos.



Figura 5.31. Pestaña para eliminar experimentos activos.

5.8 Diseño circuito impreso PCB

Se decidió implementar un PCB (“Printed Circuit Board”) en una placa de cobre para distribuir los circuitos convertidores de tensión de 3.3V a 5V y las entradas de alimentación para los 48 sensores, que utiliza la red en el invernadero. Para fabricar el PCB se utiliza la fresadora de comandos CNC (“Computer Numerical Control”), controlada bajo el software ISO Pro. El circuito acondicionador de la red de sensores se lleva a cabo en una “ferro board” debido a que en el diseño se deben cruzar pistas, lo que provoca que sean necesarias

técnicas de fabricación fotosensibles para producir el montaje superficial a doble capa en el PCB y este no es el caso para la fresadora de comandos CNC.

El software utilizado para el diseño del circuito impreso es el Eagle 7.3.0. En este se realizó el esquemático y la distribución de los componentes basados en las dimensiones de placa de cobre, se trabajó con una de 12cm * 10cm debido a que el tamaño final del diseño abarco un total de 104.14cm * 74.6cm, de ser necesario o recomendado la placa se puede cortar para reducir las secciones sobrantes.

Con el programa ISO Pro (T-Tech) instalado en la computadora del taller de mantenimiento de Ingeniería en Electrónica, se carga el diseño realizado en el software Eagle 7.3.0, se deben ajustar los bordes de la placa con el ISO Pro para esto se debe dirigir la fresadora a cada uno de los 4 extremos y verificar que estén dentro de las dimensiones de la placa de cobre. Este programa funciona como un sistema que permite perforar, fresar y rutear la placa de circuito.

Como se observa en la figura 5.32 el esquemático del PCB presenta 3 parejas de 16 “pin header” para alimentar los 16 sensores de temperatura, los 16 de humedad y los 16 de luminosidad, estas líneas se conectan a otros dos pines de alimentación VCC y GND. Las dos líneas de 8 pines son las entradas de los 16 circuitos convertidores de 3.3V a 5V, las salidas de estos circuitos se agrupan en parejas según el número de mesa, el pin superior se asigna a las secciones 1 y el inferior a las secciones 2. La figura a6.1 en el apéndice 6 muestra el plano de tierras y la distribución de los componentes por la placa con sus respectivas pistas.

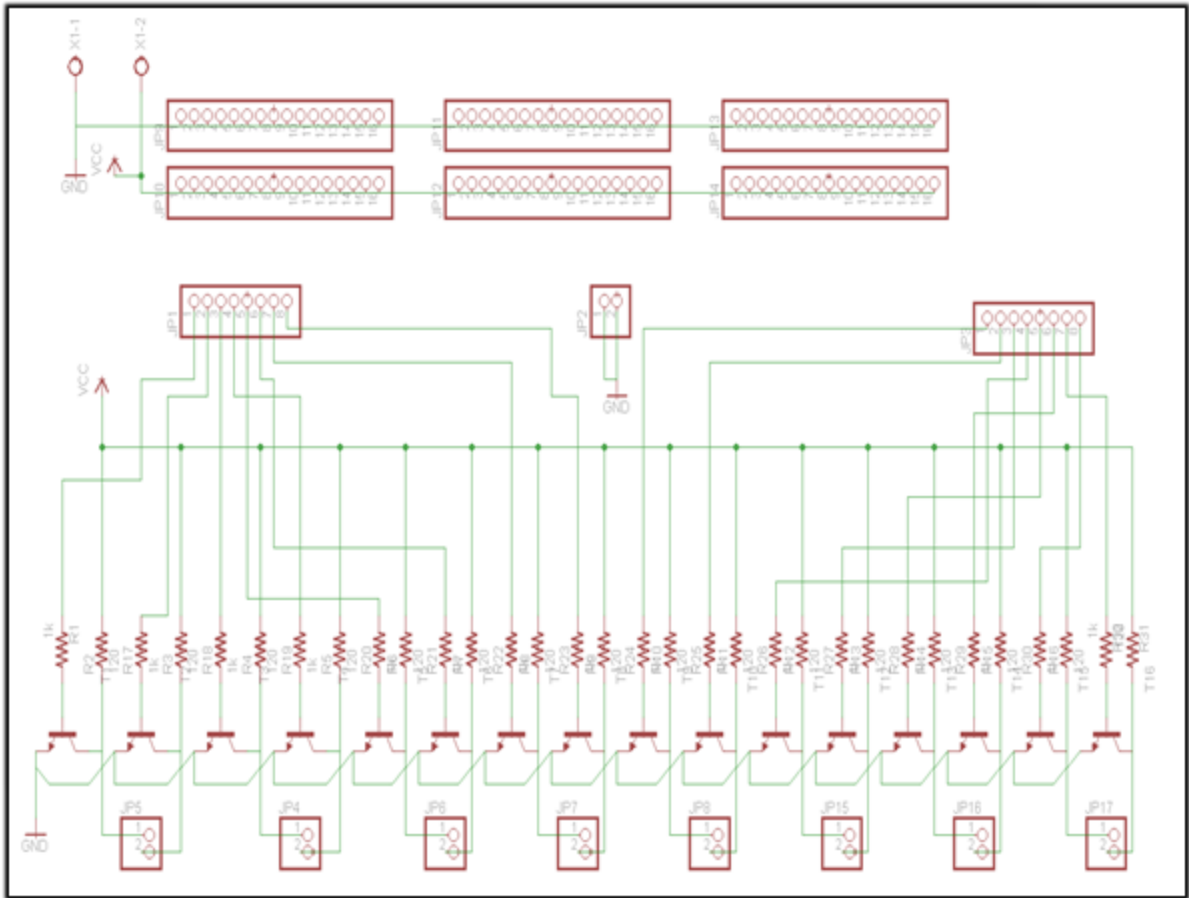


Figura 5.32. Esquemático del PCB.

Capítulo 6: Resultados experimentales

En esta etapa se muestran los resultados de las pruebas que se le realizaron al sistema electrónico descrito en la etapa de diseño, para mostrar los resultados del sistema de almacenamiento se utilizó la herramienta phpMyAdmin, que permite acceder a todas las funciones de la base de datos MySQL. Los resultados obtenidos para la página web se tomaron del navegador Google Chrome.

Para realizar las pruebas pertinentes no fue posible contar con todos los 48 sensores que requiere la red, se dispuso de 3 sensores de luminosidad, 3 de temperatura y 1 de humedad. Además para realizar los riegos programados se contó con 2 válvulas solenoides, estas no se acoplaron con algún tipo de tuberías sino que se verificó su funcionamiento pasando aire de un extremo al otro. En la figura a7.1 se observa una foto del prototipo desarrollado en este proyecto de graduación.

6.1 Sistema de almacenamiento de datos

Se partió del diseño propuesto para la red de sensores de temperatura, humedad y luminosidad; además del sistema de almacenamiento de datos y la página web desarrollados para obtener los resultados que se observan en las figuras 5.34, 5.35, 5.36, 5.37, 5.38, 5.39, 5.40 y 5.41.

Las 3 primeras tablas se llenan localmente desde la base de datos en MySQL Workbench, para la tabla de usuarios se ingresó correctamente el nombre “Administrador” y la contraseña “CIBbiotec.invernadero2015”, lo mismo para la tabla tiposDeSensor con los nombres “Temperatura”, “Humedad” y “Luminosidad” y para la tabla mesas con 8 nombres distintos según el número de mesa. Los resultados se detallan en las siguientes figuras 5.34, 5.35 y 5.36.

idUsuario	nombreUsuario	password
1	Administrador	CIBbiotec.invernadero2015

Figura 5.34. Tabla usuarios.

idtiposDeSensor	nombre
1	Temperatura
2	Humedad
3	Luminosidad

Figura 5.35. Tablas tiposDeSensor.

idMesa	nombre
1	Mesa 1
2	Mesa 2
3	Mesa 3
4	Mesa 4
5	Mesa 5
6	Mesa 6
7	Mesa 7
8	Mesa 8

Figura 5.36. Tabla mesas.

Desde la página web se procede a configurar 4 distintos experimentos el primero en la mesa 7 y sección 2 con un riego por día, el segundo en la mesa 1 y sección 1 con 240 riegos por día, el tercero en la mesa 4 y sección 1 con 48 riegos por día y el cuarto en la mesa 6 y sección 2 con una cantidad de 24 riegos por día. Los datos obtenidos se muestran en la figura 5.37, en esta se puede observar como el atributo idSeccion para el primer experimento es 1, esto se debe a que se habían creado 15 experimentos con anterioridad, lo que provocó que este atributo con característica de autoincremento aumentará su número de identificador.

idSeccion	nombre	mesas_idMesa	riegos
16	Seccion 2	7	1
17	Seccion 1	1	240
18	Seccion 1	4	48
19	Seccion 2	6	24

Figura 5.37. Tabla secciones.

La figura 5.38 detalla los resultados obtenidos para la tabla sensores, esta tabla almacena unos nombres definidos para cada sensor de acuerdo al tipo y la ubicación de mesa y sección. La información se registra cuando el usuario configura un nuevo experimento.

En la siguiente tabla el idSensor no empieza en 1 sino que en 46 debido a que se habían configurado 15 experimentos con anterioridad, lo que causó que se agregaran 45 distintos nombres por tratarse de tríos de sensores. Esto evidencia el crecimiento dinámico del sistema de almacenamiento diseñado.

idSensor	nombre	secciones_idSeccion	tiposDeSensor_idtiposDeSensor
46	Temperatura72	16	1
47	Humedad72	16	2
48	Luminosidad72	16	3
49	Temperatura11	17	1
50	Humedad11	17	2
51	Luminosidad11	17	3
52	Temperatura41	18	1
53	Humedad41	18	2
54	Luminosidad41	18	3
55	Temperatura62	19	1
56	Humedad62	19	2
57	Luminosidad62	19	3

Figura 5.38. Tablas sensores.

Las figuras 5.39, 5.40 y 5.41 muestran los datos almacenados en las tablas temperaturas, humedades y luminosidades asociadas al experimento #17 configurado para la mesa 1 y sección 1. El tiempo de muestreo fue de 1 minuto, en las figuras se observan los 15 valores obtenidos durante 15 minutos entre las horas 12:44 pm y 12:59 pm del día 14 de

Octubre del 2015. La prueba se realizó en los laboratorios de electrónica que se encuentran aproximadamente a 100 metros de la estación meteorológica de ITCR en Cartago. Los valores de temperatura están traducidos en grados centígrados, los de humedad representan porcentaje de humedad relativa y los de luminosidad se asocian con la unidad lux.

ID_Temperatura	ValorSensorT	HoraT	FechaT	sensores_idSensor
83	23.9	12:44:16	2015-10-14	49
84	23.9	12:45:16	2015-10-14	49
85	23.86	12:46:16	2015-10-14	49
86	23.9	12:47:16	2015-10-14	49
87	23.99	12:48:17	2015-10-14	49
88	23.99	12:49:17	2015-10-14	49
89	23.86	12:50:17	2015-10-14	49
90	23.95	12:51:18	2015-10-14	49
91	23.86	12:52:18	2015-10-14	49
92	23.86	12:53:18	2015-10-14	49
93	23.9	12:54:18	2015-10-14	49
94	23.81	12:55:18	2015-10-14	49
95	23.86	12:56:18	2015-10-14	49
96	23.86	12:57:18	2015-10-14	49
97	23.86	12:58:18	2015-10-14	49
98	23.81	12:59:18	2015-10-14	49

Figura 5.39. Tabla temperaturas.

ID_Humedad	ValorSensorH	HoraH	FechaH	sensores_idSensor
83	67.7	12:44:16	2015-10-14	50
84	67.78	12:45:16	2015-10-14	50
85	67.9	12:46:16	2015-10-14	50
86	68.61	12:47:16	2015-10-14	50
87	68.25	12:48:17	2015-10-14	50
88	68.37	12:49:17	2015-10-14	50
89	68.25	12:50:17	2015-10-14	50
90	68.49	12:51:18	2015-10-14	50
91	67.54	12:52:18	2015-10-14	50
92	67.86	12:53:18	2015-10-14	50
93	68.21	12:54:18	2015-10-14	50
94	68.25	12:55:18	2015-10-14	50
95	68.84	12:56:18	2015-10-14	50
96	68.45	12:57:18	2015-10-14	50
97	68.21	12:58:18	2015-10-14	50
98	67.58	12:59:18	2015-10-14	50

Figura 5.40. Tabla humedades.

ID_Luminosidad	ValorSensorL	HoraL	FechaL	sensores_idSensor
83	279.29	12:44:16	2015-10-14	51
84	276.36	12:45:16	2015-10-14	51
85	279.29	12:46:16	2015-10-14	51
86	276.12	12:47:16	2015-10-14	51
87	277.09	12:48:17	2015-10-14	51
88	276.85	12:49:17	2015-10-14	51
89	279.54	12:50:17	2015-10-14	51
90	279.29	12:51:18	2015-10-14	51
91	278.8	12:52:18	2015-10-14	51
92	282.47	12:53:18	2015-10-14	51
93	278.8	12:54:18	2015-10-14	51
94	285.15	12:55:18	2015-10-14	51
95	285.15	12:56:18	2015-10-14	51
96	284.17	12:57:18	2015-10-14	51
97	283.69	12:58:18	2015-10-14	51
98	276.12	12:59:18	2015-10-14	51

Figura 5.41. Tabla luminosidades.

6.2 Página Web

Para la página web se consideran los resultados obtenidos a la hora de utilizar las pestañas para configurar, consultar y eliminar los experimentos con los que se trabajó en el apartado anterior 6.1 y que se observan en la figura 5.42. Además para verificar el funcionamiento del sistema de riego programado se detallan las horas a regar cada experimento.

Experimento	Mesa	Sección	Intervalo de Riego	Riegos por Día
16	7	2	24 h	1
17	1	1	10 min	240
18	4	1	30 min	48
19	6	2	1 h	24

Figura 5.42. Tabla de experimentos activos.

6.2.1 Pestañas

Los resultados de configurar el experimento en la mesa 1 y sección 1 con un intervalo de riego de 10 minutos se observan en la figura 5.43, cuando se configuró este experimento no existía ninguno otro en la misma mesa y sección por lo que el sistema procedió a configurar uno nuevo.



Figura 5.43. Configurando experimento en mesa 1 y sección 1.

Se volvió a intentar configurar un nuevo experimento con los datos anteriores, pero para este caso el sistema si devolvió un mensaje de error, debido a que ya estaba en uso esa mesa y sección. La figura 5.44 muestra este mensaje de error.



Figura 5.44. Mensaje de error para pestaña configurar.

Se consultó el experimento #17 ubicado en la mesa 1 y sección 1 después de 10 minutos para obtener los resultados que se detallan en la figura 5.45. Si se pretende consultar un experimento que no ha sido configurado como es el caso de la mesa 3 y sección 1, los resultados que se obtienen se observan en la figura 5.46.

Consultando Experimento					
Mesa: 1 / Sección: 1					
Número de Muestra	Temperatura [°C]	Humedad [%RH]	Luminosidad [Lux]	Hora [h:ms:cs]	Fecha [año-mes-día]
1	23.95	68.05	285.64	12:43:16	2015-10-14
2	23.9	67.7	279.29	12:44:16	2015-10-14
3	23.9	67.78	276.36	12:45:16	2015-10-14
4	23.86	67.9	279.29	12:46:16	2015-10-14
5	23.9	68.61	276.12	12:47:16	2015-10-14
6	23.99	68.25	277.09	12:48:17	2015-10-14
7	23.99	68.37	276.85	12:49:17	2015-10-14
8	23.86	68.25	279.54	12:50:17	2015-10-14
9	23.95	68.49	279.29	12:51:18	2015-10-14
10	23.86	67.54	278.8	12:52:18	2015-10-14

Figura 5.45. Resultado de consultar experimento en mesa 1 y sección 1.



Figura 5.46. Resultado de consultar experimento en mesa 3 y sección 1.

Para eliminar un experimento este debe de haberse creado con anterioridad, el experimento #18 en la mesa 4 y sección 1 se elimina y los resultados obtenidos se muestran en la figura 5.47. Para que el experimento se elimine correctamente se debe de aceptar las condiciones las cuales indican que la información registrada para el experimento se perderá.



Figura 5.47. Resultado de eliminar experimento en mesa 4 y sección 1.

6.2.2 Sistema riego programado

Para realizar las pruebas al sistema de riego programado, se utilizaron los mismos experimentos que se detallan en la figura 5.42 con los correspondientes intervalos de riego. Se monitorizaron las horas de riego para cerciorar que se efectuaran todos los riegos durante 3 horas para los experimentos # 17, 18 y 19, el experimento #16 al contar con un solo riego por día no se tomó en cuenta.

La hora de partida fue a la 1:41 pm y se terminó de monitorizar los riegos a las 4:41pm. En la siguiente tabla 6.1 se detallan las horas de riego en ese período de tiempo para los experimentos a verificar.

Tabla 6.1. Horas de riego para los experimentos seleccionados.

# Experimento	Mesa	Sección	Intervalo de Riego	Horas de Riego
17	1	1	10min	1:50, 2:00, 2:10, 2:20, 2:30, 2:40, 2:50, 3:00, 3:10, 3:20, 3:30, 3:40, 3:50, 4:00, 4:10, 4:20, 4:30 y 4:40
18	4	1	30min	2:05, 2:35, 3:05, 3:35, 4:05 y 4:35
19	6	2	1h	2:11, 3:11 y 4:11

Todos los riegos se realizaron correctamente, las electroválvulas estuvieron abiertas un tiempo de 30 segundos por cada riego, la diferencia entre el momento en que la válvula solenoide fue abierta/cerrada con respecto a la hora del servidor fue de máximo 3 segundos para algunos riegos.

La figura 5.48 detalla cómo se observa en la página web cuando un riego se encuentra activo, para este caso se realizó una captura de pantalla a la 1:50:08 pm en donde la mesa 1 y sección 1 del experimento #17 debía encontrarse efectuando un riego. De igual manera se selecciona la hora 2:11:21 pm para tomar una segunda captura de pantalla y el resultado se muestra en la figura 5.49, que demuestra que el riego se efectuó para la mesa 6 y sección 2 acorde a la tabla 5.2.

**Centro de Investigación en Biotecnología
Invernadero de Investigación**

01:50:08

[Acerca](#) [Configurar](#) [Consultar](#) [Eliminar](#)

El Centro de Investigación en Biotecnología CIB es la unidad de investigación de la Escuela de Biología del Tecnológico de Costa Rica. El centro de investigación se fundó en 1995 y se encuentra ubicado en la sede central de Cartago.

Tabla de Experimentos Activos

Experimento	Mesa	Sección	Intervalo de Riego	Riegos por Día
16	7	2	24 h	1
17	1	1	10 min	240
18	4	1	30 min	48
19	6	2	1 h	24

Control de Riego

- Experimento 16
Mesa 7 / Sección 2
Estado Riego: -
- Experimento 17
Mesa 1 / Sección 1
Estado Riego: Activo
- Experimento 18
Mesa 4 / Sección 1
Estado Riego: -
- Experimento 19
Mesa 6 / Sección 2
Estado Riego: -

Figura 5.48. Hora de riego para el experimento 17.

**Centro de Investigación en Biotecnología
Invernadero de Investigación**

02:11:21

[Acerca](#) [Configurar](#) [Consultar](#) [Eliminar](#)

El Centro de Investigación en Biotecnología CIB es la unidad de investigación de la Escuela de Biología del Tecnológico de Costa Rica. El centro de investigación se fundó en 1995 y se encuentra ubicado en la sede central de Cartago.

Tabla de Experimentos Activos

Experimento	Mesa	Sección	Intervalo de Riego	Riegos por Día
16	7	2	24 h	1
17	1	1	10 min	240
18	4	1	30 min	48
19	6	2	1 h	24

Control de Riego

- Experimento 16
Mesa 7 / Sección 2
Estado Riego: -
- Experimento 17
Mesa 1 / Sección 1
Estado Riego: -
- Experimento 18
Mesa 4 / Sección 1
Estado Riego: -
- Experimento 19
Mesa 6 / Sección 2
Estado Riego: Activo

Figura 5.49. Hora de riego para el experimento 19.

6.3 Circuito impreso PCB

El resultado que se obtuvo después de que la fresadora termino de fabricar el circuito impreso en la placa de cobre se observa en la figura 5.50. A la placa se le debieron realizar los huecos con un taladro para poder conectar los componentes. El montaje de todos los componentes en el PCB se detalla en la figura 5.51. Se realizaron pruebas de continuidad a la placa de cobre para asegurar que todas las pistas y tierras diseñadas quedaron de manera correcta, esta prueba se realizó dos veces la primera apenas terminada la fabricación de PCB y la segunda después de soldar todos los componentes que se integraron en el circuito impreso.

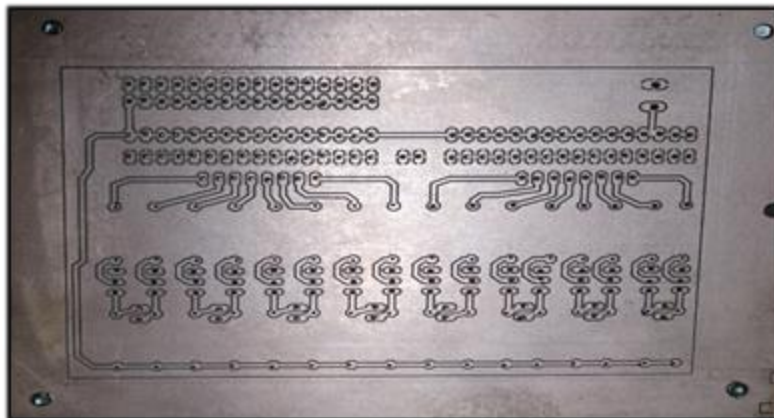


Figura 5.50. PCB con las pistas y huecos.



Figura 5.51. Montaje PCB.

Capítulo 7: Análisis de resultados

En este capítulo se examinan los resultados obtenidos en el capítulo 6 para justificar el funcionamiento de las diversas unidades que conforma el sistema electrónico desarrollado.

7.1 Sistema de almacenamiento de datos

Para el sistema de almacenamiento de datos empleado en el software MySQL Workbench, se obtuvo 100% de integridad en los datos registrados con respecto a la información ingresada. Esto supone que cada llamada a los procedimientos almacenados se efectuó correctamente para cada una de las 8 tablas del diagrama de entidad relación, además esto implica que las relaciones diseñadas para estas tablas en el diagrama surgieron efecto y le dieron el dinamismo esperado a la base de datos.

También en las tablas temperaturas, humedades y luminosidades se obtuvieron valores cercanos a los datos que proporciono la estación meteorológica del ITCR en Cartago con diferencias menores al 10%. La información de esta estación se consultó vía internet en el sitio web que se actualiza periódicamente.

La información registrada en estas tablas se almaceno el día 14 de Octubre del 2015 en los laboratorios de la escuela de Ingeniería en Electrónica, durante 15 minutos entre las horas 12:44 pm y 12:59 pm, en este intervalo de tiempo la temperatura leída en la página web adquirió valores entre 24.5°C y 24.8°C y la humedad varió entre 67%RH y 70%RH. La página web no brinda valores de luminosidad.

Como se puede apreciar en la figura 5.39 los valores de temperatura registrados en el sistema de almacenamiento durante el intervalo de 15 minutos se mantuvieron constantes en 23°C con un valor mínimo de 23.81°C y un dato máximo de 23.99°C. En cuanto a los datos de humedad que se muestran en la figura 5.40 el mínimo registrado fue de 67.54%RH y el máximo de 68.84%RH.

Si se compara la información leída en la página web de la estación meteorológica con la registrada en el sistema de almacenamiento, se determina que el porcentaje de error promedio para los datos de temperatura fue de un 3% y para los valores de humedad de un 2%.

7.2 Página web

Se analiza el funcionamiento de la página web a partir de los resultados obtenidos en las figuras 5.42, 5.43, 5.44, 5.45, 5.46, 5.47, 5.48 y 5.49, estas imágenes fueron tomadas al sitio web durante las pruebas realizadas al sistema electrónico. En la figura 5.42 se visualiza que la tabla de experimentos activos contiene las 4 distintas configuraciones ingresados al sistema de almacenamiento de datos, esta información es consistente con la registrada en la tabla secciones que se detalla en la figura 5.37

7.2.1 Pestañas

Se examinaron los resultados obtenidos después de configurar, consultar y eliminar un experimento haciendo uso de las pestañas en la página web, en la figura 5.43 se observa el resultado de configurar un experimento en la mesa 1 y sección 1 con un intervalo de riego de 10 minutos, un mensaje indica que se ha configurado el experimento de manera correcta. La figura 5.44 muestra el mensaje de error que el sistema entregó al intentar configurar un nuevo experimento en la mesa 1 y sección 1, que se encontraban en uso.

Los datos registrados para el experimento en la mesa 1 y sección 1 se consultaron por medio de la pestaña consultar y los resultados obtenidos se muestran en la figura 5.45, esta información concuerda con la registrada en el sistema de almacenamiento para los valores de temperatura, humedad y luminosidad presentados en las figuras 5.39, 5.40 y 5.41. Se intentó consultar los datos de un experimento no creado en este caso la mesa 3 y sección 1 y el resultado obtenido en el sitio web fue un mensaje de error que se detalla en la figura 5.46.

La pestaña eliminar como se observa en la figura 5.31, contiene un apartado para aceptar unas condiciones que indican que la información en caso de no respaldarse se pierde, debe de seleccionarse para proceder a eliminar el experimento en la mesa / sección deseada. Se eliminó el experimento en la mesa 4 y sección y los resultados obtenidos se detallan en la figura 5.47, la página web envió el mensaje de que el experimento se eliminó de forma satisfactoria.

7.2.2 Sistema de riego programado

Los experimentos # 17, 18 y 19 se monitorizaron desde el sitio web durante 3 horas, para analizar el resultado de los riegos efectuados en este período sólo se consideró que la válvula solenoide se abriera y cerrara, dado a que por limitaciones no se conectaron tuberías de agua al sistema. Para la mesa 1/sección 1 se realizaron los 18 riegos según la tabla 5.2, además para la mesa 4/ sección 1 se efectuaron un total de 6 riegos y en la mesa 6/sección 2 se dieron sólo 3 riegos.

Para los 3 experimentos activos se realizaron la cantidad de riegos esperados acorde con la configuración especificada en la figura 5.42, además en pocos riegos el instante en que la electroválvula se abría/cerraba no se dio en el momento justo según el reloj del sitio web y la tabla 5.2 sino que varió en 2 y 3 segundos. Esto se debe a varios factores entre el que se destaca la velocidad de internet con la que trabaja el servidor web, la página web debe de actualizarse cada segundo para mantener sincronizados los estados de riego. A pesar de esto las diferencias de 2 y 3 segundos se presentan como valores aceptables tomando en cuenta que los intervalos de riego se emplean en minutos y horas.

Los riegos se realizaron en los primeros 30 segundos de cada minuto como se estableció. La figura 5.48 muestra el estado de riego activo para la mesa 1/sección 1 a la 1:50 pm con 8 segundos, en ese momento según la tabla 5.2 se debía estar realizando un riego. También la figura 5.49 muestra el momento de un riego para la mesa 6/ sección 2 a las 2:11pm con 21 segundos, el cual concuerda con lo indicado en la tabla 5.2.

7.3 Circuito impreso PCB

En las dos pruebas de continuidad que se le realizaron al PCB se utilizó un multímetro digital, en la primera apenas fabricado el circuito impreso por la fresadora de comandos CNC, se midió la continuidad por las pistas diseñadas que se observan en la figura 5.51 tomando en cuenta la distribución de las tierras por la placa, el resultado fue el esperado descartando problemas de fabricación. Para la segunda prueba de continuidad se montaron anteriormente los componentes, en el montaje se necesitó soldar las terminales de todos los componentes para fijarlos en la placa de cobre, el resultado de la prueba con los componentes integrados en la tarjeta fue el correcto.

Al finalizar las dos pruebas de continuidad fue necesario realizar una última prueba al PCB para verificar su funcionamiento con la fuente de alimentación, para estas mediciones se utilizó el multímetro digital para medir las diferencias de potencial por la tarjeta. Se conectaron 5V de entrada en el conector verde de alimentación que se observa en la figura 5.51, además de las 16 entradas para los circuitos convertidores de tensión y el resultado para cada uno de estos circuitos fue el esperado, convirtiendo los 3.3V de entrada en 5V de salida.

Capítulo 8: Conclusiones y Recomendaciones

8.1 Conclusiones

- La base de datos desarrollada en MySQL Workbench es capaz de registrar la información proveniente de la red de sensores de temperatura, humedad y luminosidad; además mediante llamadas a los procedimientos almacenados desde la página web se manipula la información almacenada en las tablas de la base de datos diseñada.
- Las unidades de multiplexado acondicionan al sistema electrónico más sensores, se requiere de una entrada analógica por cada multiplexor. Utilizando 3 multiplexores de 16 entradas se diseñó una red de 48 sensores de temperatura, humedad y luminosidad.
- Mediante un divisor de tensión que utiliza una resistencia de $39k\Omega$ y otra de $22k\Omega$, se implementó un circuito de ajuste de tensión requerido en las unidades de multiplexado de sensores de humedad y luminosidad, para convertir la tensión de 5V a 1.8V.
- El sensor de luminosidad presenta un rango máximo de medición de 1000 lux y el sensor de humedad trabaja por porcentajes de humedad relativa.

- El circuito convertidor de tensión de 3.3V a 5V se desarrolló con transistores NPN 2N4401 y con una configuración de emisor común que utiliza una resistencia de 120 Ω en el colector y otra de 1k Ω en la base.
- El módulo de relés mantiene el circuito de potencia generado por la fuente DC de 12V acoplado a los 16 canales de relés, se activan cuando las entradas de bajo nivel reciben una tensión de 0V, en caso de ser 5V se desactivan. Este mecanismo permite manipular las electroválvulas conectadas en las salidas de los relés para el funcionamiento del sistema de riego programado.
- Al implementar una pantalla de inicio de sesión se mantiene el acceso al sitio web restringido, para que los docentes del centro de investigación sean las personas que le den uso al sistema electrónico. Las pestañas en la pantalla principal le permiten al usuario configurar un nuevo experimento, además de consultar o eliminar un experimento registrado.
- El lenguaje de programación PHP permite dar lectura a los datos en las entradas analógicas y manipular el funcionamiento de los puertos GPIO del sistema embebido, además puede incorporar otros lenguajes de programación requeridos como HTML y JavaScript. Mediante funciones en JavaScript y PHP, se desarrolló un método que toma la hora del sistema y la compara cada segundo con las horas de riegos establecidas, esto permite el correcto funcionamiento del sistema de riego programado.

8.2 Recomendaciones

- Modificar la configuración del servidor web para asignar un cliente DNS (“Domain Name Server”) a una dirección IP definida y de esta forma poder acceder al sitio web desde el buscador, ingresando un dominio URL en lugar de la dirección IP junto al número de puerto, que es el caso actual.
- Implementar un script que permite actualizar la hora del sistema cada vez que el sistema electrónico se enciende o se reinicia, por el momento se deben de ingresar unos comandos de forma manual.
- Mejorar el diseño de la página web.
- Diseñar circuitos impresos PCB, para agrupar cada trío de sensores de temperatura, humedad y luminosidad en las 16 secciones de trabajo.

Bibliografía

- [1] Arduino. (2015). Arduino UNO. Recuperado de <https://www.arduino.cc/en/Main/ArduinoBoardUno>
- [2] Arduino. (2015). Arduino mega 2560. Recuperado de <https://www.arduino.cc/en/Main/ArduinoBoardMega2560>
- [3] Asco Numatics. (2014). Tecnología de electroválvulas y válvulas. Recuperado de http://www.asconumatics.eu/images/site/upload/_es/pdf1/00005es.pdf
- [4] Atmel. (2014). *ATmega48a/pa/88a/pa/168a/pa/328/p Datasheet*. Recuperado de http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Summary.pdf
- [5] Coley, G. (2012). BeagleBone Rev A6 System Reference Manual. Recuperado de http://beagleboard.org/static/beaglebone/latest/Docs/Hardware/BONE_SRM.pdf
- [6] Coley, G. (2013). BeagleBone Black Rev A5.6 System Reference Manual. Recuperado de http://www.soselectronic.com/a_info/resource/c/BB-BBLK-000.pdf
- [7] Calvo, C. (2008). Riego automatizado: Comparación de dos métodos de riego: Temporizador y bandeja de riego por demanda, en cultivo de lechuga sembrada en invernadero. (Tesis de licenciatura). Instituto Tecnológico de Costa Rica, Escuela de Ingeniería Agrícola, Cartago, Costa Rica.

- [8] Crespo, F. (1997). Necesidades de riego en invernadero. Recuperado de <http://lan.inea.org:8010/web/materiales/web/riego/anuncios/trabajos/Necesidades%20de%20riego%20en%20invernadero.pdf>
- [9] Gómez, N. (2003). Determinación del coeficiente de cultivo, la evapotranspiración real y potencial del cultivo de chile bajo invernadero. (Tesis de Licenciatura). Instituto Tecnológico de Costa Rica, Escuela de Ingeniería Agrícola, Cartago, Costa Rica
- [10] Honeywell. (2008). HIH-4030/31 series humidity sensor. Recuperado de http://sensing.honeywell.com/index.php/ci_id/51625/la_id/1/document/1/re_id/0
- [11] López, Y. (2005). Control climático en invernaderos. Recuperado de <http://www.interempresas.net/Horticola/Articulos/77307-Control-climatico-en-invernaderos.html>
- [12] LinkSprite. (2015). PcDuino3. Recuperado de http://www.linksprite.com/?page_id=812
- [13] Linux Community. (2010). Easy LAMP Server Installation. Recuperado de <https://www.linux.com/learn/tutorials/288158-easy-lamp-server-installation>
- [14] MySQL. (2015). MySQL Workbench. Recuperado de <http://dev.mysql.com/doc/workbench/en/>

- [15] National Instruments. (2012). Sensores de temperatura. Recuperado de <http://www.ni.com/white-paper/10635/es/>
- [16] National Semiconductor. (2015). LM53/LM35A/C/LM35CA/LM35D precision centigrade temperature sensors. Recuperado de <http://pdf.datasheetcatalog.net/datasheet/nationalsemiconductor/DS005516.PD>
- [17] Phidgets. (2012). Light Sensor 1000 Lux. Recuperado de http://www.phidgets.com/products.php?category=1&product_id=1142_0
- [18] PHP. (2015). Documentation. Recuperado de <https://secure.php.net/docs.php>
- [19] Proyecto GNU. (2015). El sistema operativo GNU. Recuperado de <http://www.gnu.org/gnu/linux-and-gnu.html>
- [20] Raspberry Pi. (2015). Raspberry Pi model B. Recuperado de <https://www.raspberrypi.org/products/model-b-plus>
- [21] Rivera, A.L. (2011). Rediseño de la tubería principal para un sector con ocho invernaderos de la Finca Innovaplant, Sarchí, Alajuela. (Tesis de Licenciatura). Instituto Tecnológico de Costa Rica, Escuela de Ingeniería Agrícola, Cartago, Costa Rica.
- [22] Roberto, L. (2008). Diseño integrado de un sistema de riego por goteo autocompensado y control de temperatura mediante un sistema de nebulización, para el cultivo de kalanchoe en invernadero. (Tesis de licenciatura). Instituto Tecnológico de Costa Rica, Escuela de Ingeniería Agrícola, Cartago, Costa Rica.

[23] RS Components. (1997). Light Dependent Resistor. Recuperado de http://www.biltek.tubitak.gov.tr/gelisim/elektronik/dosyalar/40/LDR_NSL19_M51.pdf

[24] Sparkfun. (2015). Analog to digital conversion. Recuperado de <https://learn.sparkfun.com/tutorials/analog-to-digital-conversion>.

[25] Tloom. (2013). Definición de nodo o mota de una red de sensores inalámbrica. Recuperado de <http://www.tloom.com/definicion-de-nodo>.

[26] Vernavá, J. P. (2010). Interfaz de adquisición de datos de una red de sensores por medio de un sistema embebido basado en microcontrolador PIC32 con comunicación USB. (Tesis de licenciatura). Instituto Tecnológico de Costa Rica, Escuela de Ingeniería Electrónica, Cartago, Costa Rica.

Apéndices

Apéndice 1: Tablas utilizadas en el proceso de diseño

1.1 Sistema de almacenamiento de datos

Tabla a1.1. Distribuciones de los sensores de temperatura, humedad y luminosidad para las 16 entradas de cada multiplexor.

<i>Entrada MUX</i>	<i>MUX Temperatura</i>	<i>MUX Humedad</i>	<i>MUX Luminosidad</i>	<i>Mesa #</i>	<i>Sección#</i>
0	ST_11	SH_11	SL_11	<i>Mesa 1</i>	<i>Sección 1</i>
1	ST_12	SH_12	SL_12	<i>Mesa 1</i>	<i>Sección 2</i>
2	ST_21	SH_21	SL_21	<i>Mesa 2</i>	<i>Sección 1</i>
3	ST_22	SH_22	SL_22	<i>Mesa 2</i>	<i>Sección 2</i>
4	ST_31	SH_31	SL_31	<i>Mesa 3</i>	<i>Sección 1</i>
5	ST_32	SH_32	SL_32	<i>Mesa 3</i>	<i>Sección 2</i>
6	ST_41	SH_41	SL_41	<i>Mesa 4</i>	<i>Sección 1</i>
7	ST_42	SH_42	SL_42	<i>Mesa 4</i>	<i>Sección 2</i>
8	ST_51	SH_51	SL_51	<i>Mesa 5</i>	<i>Sección 1</i>
9	ST_52	SH_52	SL_52	<i>Mesa 5</i>	<i>Sección 2</i>
10	ST_61	SH_61	SL_61	<i>Mesa 6</i>	<i>Sección 1</i>
11	ST_62	SH_62	SL_62	<i>Mesa 6</i>	<i>Sección 2</i>
12	ST_71	SH_71	SL_71	<i>Mesa 7</i>	<i>Sección 1</i>
13	ST_72	SH_72	SL_72	<i>Mesa 7</i>	<i>Sección 2</i>
14	ST_81	SH_81	SL_81	<i>Mesa 8</i>	<i>Sección 1</i>
15	ST_82	SH_82	SL_82	<i>Mesa 8</i>	<i>Sección 2</i>

Tabla a1.2. Combinaciones lógicas para seleccionar un sensor de temperatura, humedad y luminosidad en una mesa/sección específica.

# Combinación Lógica	D	C	B	A	Entradas MUX	Mesa #	Sección#
0	0	0	0	0	0	Mesa 1	Sección 1
1	0	0	0	1	1	Mesa 1	Sección 2
2	0	0	1	0	2	Mesa 2	Sección 1
3	0	0	1	1	3	Mesa 2	Sección 2
4	0	1	0	0	4	Mesa 3	Sección 1
5	0	1	0	1	5	Mesa 3	Sección 2
6	0	1	1	0	6	Mesa 4	Sección 1
7	0	1	1	1	7	Mesa 4	Sección 2
8	1	0	0	0	8	Mesa 5	Sección 1
9	1	0	0	1	9	Mesa 5	Sección 2
10	1	0	1	0	10	Mesa 6	Sección 1
11	1	0	1	1	11	Mesa 6	Sección 2
12	1	1	0	0	12	Mesa 7	Sección 1
13	1	1	0	1	13	Mesa 7	Sección 2
14	1	1	1	0	14	Mesa 8	Sección 1
15	1	1	1	1	15	Mesa 8	Sección 2

Tabla a1.3. Distribución de los GPIO utilizados para el sistema de almacenamiento de datos de la red de sensores.

# Bit Control	GPIO	Pin	P8	P9	Propósito
0	GPIO_30	11	-	X	Señal Digital de Control A
1	GPIO_31	13	-	X	Señal Digital de Control B
2	GPIO_48	15	-	X	Señal Digital de Control C
3	GPIO_03	21	-	X	Señal Digital de Control D

Tabla a1.4. Distribución de los canales analógicos utilizados para el sistema de almacenamiento de datos de la red de sensores.

# Puerto	Nombre	Pin	P8	P9	Variable
0	AIN_0	39	-	X	Temperatura
1	AIN_1	40	-	X	Humedad
2	AIN_2	37	-	X	Luminosidad

1.2 Sistema de riego programado

Tabla a1.5. Distribución de los GPIO utilizados para el sistema de riego programado.

GPIO	Pin	P8	P9	Propósito
GPIO_02	22	-	X	Mesa 1 – Sección 1
GPIO_07	42	-	X	Mesa 1 – Sección 2
GPIO_14	26	-	X	Mesa 2 – Sección 1
GPIO_15	24	-	X	Mesa 2 – Sección 2
GPIO_20	41	-	X	Mesa 3 – Sección 1
GPIO_22	19	X	-	Mesa 3 – Sección 2
GPIO_23	13	X	-	Mesa 4 – Sección 1
GPIO_26	14	X	-	Mesa 4 – Sección 2
GPIO_27	17	X	-	Mesa 5 – Sección 1
GPIO_44	12	X	-	Mesa 5 – Sección 2
GPIO_45	11	X	-	Mesa 6 – Sección 1
GPIO_46	16	X	-	Mesa 6 – Sección 2
GPIO_47	15	X	-	Mesa 7 – Sección 1
GPIO_49	23	-	X	Mesa 7 – Sección 2

<i>GPIO_51</i>	16	-	X	<i>Mesa 8 – Sección 1</i>
<i>GPIO_60</i>	12	-	X	<i>Mesa 8 – Sección 2</i>
<i>GPIO_61</i>	26	X	-	
<i>GPIO_65</i>	18	X	-	
<i>GPIO_66</i>	07	X	-	
<i>GPIO_67</i>	08	X	-	
<i>GPIO_68</i>	10	X	-	
<i>GPIO_69</i>	09	X	-	

Tabla a1.6 Distribución de las salidas de los circuitos convertidores en las entradas del módulo de relés.

<i>Salidas Circuitos Convertidor Mesa / Sección</i>	<i># Entrada Bajo Nivel</i>
<i>Mesa 1 / Sección 1</i>	<i>1</i>
<i>Mesa 1 / Sección 2</i>	<i>2</i>
<i>Mesa 2 / Sección 1</i>	<i>3</i>
<i>Mesa 2 / Sección 2</i>	<i>4</i>
<i>Mesa 3 / Sección 1</i>	<i>5</i>
<i>Mesa 3 / Sección 2</i>	<i>6</i>
<i>Mesa 4 / Sección 1</i>	<i>7</i>
<i>Mesa 4 / Sección 2</i>	<i>8</i>
<i>Mesa 5 / Sección 1</i>	<i>9</i>
<i>Mesa 5 / Sección 2</i>	<i>10</i>
<i>Mesa 6 / Sección 1</i>	<i>11</i>
<i>Mesa 6 / Sección 2</i>	<i>12</i>
<i>Mesa 7 / Sección 2</i>	<i>13</i>

<i>Mesa 7/ Sección 2</i>	14
<i>Mesa 8/ Sección 2</i>	15
<i>Mesa 8/ Sección 2</i>	16

Tabla a1.7. Distribución de los tiempos de riegos por intervalos de minutos y horas para cada mesa y sección.

<i>Mesa / Sección</i>	<i>Intervalos de Minutos</i>			<i>Intervalos de Horas Cada X hora</i>
	<i>10min</i>	<i>15min</i>	<i>30min</i>	
<i>Mesa 1/ Sección 1</i>	0, 10, 20, 30, 40 y 50	0, 15, 30 y 45	0 y 30	<i>Hora X : Minuto 0</i>
<i>Mesa 1/ Sección 2</i>	1, 11, 21, 31, 41 y 51	1, 16, 31 y 46	1 y 31	<i>Hora X : Minuto 1</i>
<i>Mesa 2/ Sección 1</i>	2, 12, 22, 32, 42 y 52	2, 17, 32 y 47	2 y 32	<i>Hora X : Minuto 2</i>
<i>Mesa 2/ Sección 2</i>	3, 13, 23, 33, 43 y 53	3, 18, 33 y 48	3 y 33	<i>Hora X : Minuto 3</i>
<i>Mesa 3/ Sección 1</i>	4, 14, 24, 34, 44 y 54	4, 19, 34 y 49	4 y 34	<i>Hora X : Minuto 4</i>
<i>Mesa 3/ Sección 2</i>	4, 14, 24, 34, 44 y 54	4, 19, 34 y 49	4 y 34	<i>Hora X : Minuto 5</i>
<i>Mesa 4/ Sección 1</i>	5, 15, 25, 35, 45 y 55	5, 20, 35 y 50	5 y 35	<i>Hora X : Minuto 6</i>
<i>Mesa 4/ Sección 2</i>	5, 15, 25, 35, 45 y 55	5, 20, 35 y 50	5 y 35	<i>Hora X : Minuto 7</i>
<i>Mesa 5/ Sección 1</i>	6, 16, 26, 36, 46 y 56	6, 21, 36 y 41	6 y 36	<i>Hora X : Minuto 8</i>
<i>Mesa 5/ Sección 2</i>	6, 16, 26, 36, 46 y 56	6, 21, 36 y 41	6 y 36	<i>Hora X : Minuto 9</i>
<i>Mesa 6/ Sección 1</i>	7, 17, 27, 37, 47 y 57	7, 22, 37 y 42	7 y 37	<i>Hora X : Minuto 10</i>
<i>Mesa 6/ Sección 2</i>	7, 17, 27, 37, 47 y 57	7, 22, 37 y 42	7 y 37	<i>Hora X : Minuto 11</i>
<i>Mesa 7/ Sección 2</i>	8, 18, 28, 38, 48 y 58	6, 21, 36 y 41	8 y 38	<i>Hora X : Minuto 12</i>
<i>Mesa 7/ Sección 2</i>	8, 18, 28, 38, 48 y 58	6, 21, 36 y 41	8 y 38	<i>Hora X : Minuto 13</i>
<i>Mesa 8/ Sección 2</i>	9, 19, 29, 39, 49 y 59	9, 24, 39 y 54	9 y 39	<i>Hora X : Minuto 14</i>
<i>Mesa 8/ Sección 2</i>	9, 19, 29, 39, 49 y 59	9, 24, 39 y 54	9 y 39	<i>Hora X : Minuto 15</i>

Apéndice 2: Instalación de Debian GNU/Linux 7 en BeagleBone Black Rev C

- **Paso 1:** Descargar la imagen de Debian para la tarjetas microSD de la página oficial beagleboard.org en el apartado de últimas imágenes.
- **Paso 2:** Por lo general el archivo descargado que contiene la imagen viene en formato .xz, para este paso se debe descomprimir este archivo con algún programa, se recomienda 7-zip que se encuentra gratuito en la web para una gran variedad de sistemas operativos.
- **Paso 3:** Con el archivo descomprimido en formato .img se debe de instalar otro programa para quemar este archivo en la tarjeta microSD, se recomienda utilizar Windows y el programa “Win32 Disk Imager” que se encuentra también de forma gratuita en la web. Se debe conectar la tarjeta en la computadora y en el programa se selecciona la imagen en formato .img y la unidad de disco donde se encuentra la tarjeta microSD, en esta se graba la imagen.
- **Paso 4:** Retirar la tarjeta microSD del ordenador una vez finalizado el proceso de escritura.
- **Paso 5:** El paso final es conectar la tarjeta microSD en la ranura de la BeagleBone Black para verificar su funcionamiento, para esto se debe mantener pulsado el botón de “boot” en la plataforma antes de conectar la alimentación, esto carga el sistema que se encuentra en la tarjeta de memoria microSD.

Apéndice 3: Comandos de instalación de software para Debian GNU/Linux 7

Permisos de administrador:

- `sudo su`

Actualizar sistema:

- `apt-get update`
- `apt-get upgrade`

3.1 Apache

- `apt-get install apache2`

3.2 MySQL Workbench

- `apt-get install mysql-server`
- `apt-get install mysql-workbench`

3.3 PHP

- `apt-get install php5-common libapache2-mod-php5`

3.4 phpMyAdmin

- `apt-get install phpMyAdmin`

Apéndice 4: Scripts

Permisos de administrador:

- `sudo su`

4.1 Inicializar `config.sh`

- Abrir el archivo `config.sh`: `nano /etc/rc.local`

```
#!/bin/bash
#
#By default this script does nothig.

sh /var/www/CIBinvernadero/public/scripts/config.sh

exit 0
```

- Ejecutar el archivo para realizar cambios: `/etc/rc.local`

4.2 Script `config.sh` para configuración de GPIO y entradas analógicas

- Abrir el archivo `config.sh`:
`nano /var/www/CIBinvernadero/public/scripts/config.sh`

```
#!/bin/bash

echo 51 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio51/value
chmod 666 /sys/class/gpio/gpio51/direction
echo out > /sys/class/gpio/gpio51/direction

echo 30 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio30/value
chmod 666 /sys/class/gpio/gpio30/direction
echo out > /sys/class/gpio/gpio30/direction

echo 31 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio31/value
chmod 666 /sys/class/gpio/gpio31/direction
echo out > /sys/class/gpio/gpio31/direction
```



```
echo 60 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio60/value
chmod 666 /sys/class/gpio/gpio60/direction
echo out > /sys/class/gpio/gpio60/direction

echo 48 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio48/value
chmod 666 /sys/class/gpio/gpio48/direction
echo out > /sys/class/gpio/gpio48/direction

echo 3 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio3/value
chmod 666 /sys/class/gpio/gpio3/direction
echo out > /sys/class/gpio/gpio3/direction

echo 2 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio2/value
chmod 666 /sys/class/gpio/gpio2/direction
echo out > /sys/class/gpio/gpio2/direction

echo 49 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio49/value
chmod 666 /sys/class/gpio/gpio49/direction
echo out > /sys/class/gpio/gpio49/direction

echo 15 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio15/value
chmod 666 /sys/class/gpio/gpio15/direction
echo out > /sys/class/gpio/gpio15/direction

echo 14 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio14/value
chmod 666 /sys/class/gpio/gpio14/direction
echo out > /sys/class/gpio/gpio14/direction

echo 20 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio20/value
chmod 666 /sys/class/gpio/gpio20/direction
echo out > /sys/class/gpio/gpio20/direction

echo 7 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio7/value
chmod 666 /sys/class/gpio/gpio7/direction
echo out > /sys/class/gpio/gpio7/direction

echo 66 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio66/value
chmod 666 /sys/class/gpio/gpio66/direction
echo out > /sys/class/gpio/gpio66/direction
```

```
echo 67 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio67/value
chmod 666 /sys/class/gpio/gpio67/direction
echo out > /sys/class/gpio/gpio67/direction
```

```
echo 69 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio69/value
chmod 666 /sys/class/gpio/gpio69/direction
echo out > /sys/class/gpio/gpio69/direction
```

```
echo 68 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio68/value
chmod 666 /sys/class/gpio/gpio68/direction
echo out > /sys/class/gpio/gpio68/direction
```

```
echo 45 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio45/value
chmod 666 /sys/class/gpio/gpio45/direction
echo out > /sys/class/gpio/gpio45/direction
```

```
echo 44 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio44/value
chmod 666 /sys/class/gpio/gpio44/direction
echo out > /sys/class/gpio/gpio44/direction
```

```
echo 23 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio23/value
chmod 666 /sys/class/gpio/gpio23/direction
echo out > /sys/class/gpio/gpio23/direction
```

```
echo 26 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio26/value
chmod 666 /sys/class/gpio/gpio26/direction
echo out > /sys/class/gpio/gpio26/direction
```

```
echo 47 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio47/value
chmod 666 /sys/class/gpio/gpio47/direction
echo out > /sys/class/gpio/gpio47/direction
```

```
echo 46 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio46/value
chmod 666 /sys/class/gpio/gpio46/direction
echo out > /sys/class/gpio/gpio46/direction
```

```
echo 27 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio27/value
chmod 666 /sys/class/gpio/gpio27/direction
```

```
echo out > /sys/class/gpio/gpio27/direction

echo 65 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio65/value
chmod 666 /sys/class/gpio/gpio65/direction
echo out > /sys/class/gpio/gpio65/direction

echo 22 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio22/value
chmod 666 /sys/class/gpio/gpio22/direction
echo out > /sys/class/gpio/gpio22/direction

echo 61 > /sys/class/gpio/export
chmod 666 /sys/class/gpio/gpio61/value
chmod 666 /sys/class/gpio/gpio61/direction
echo out > /sys/class/gpio/gpio61/direction

echo cape-bone-iio > /sys/devices/bone_capemgr.9/slots
```

- Ejecutar el archivo /etc/rc.local para realizar cambios:
/etc/rc.local

4.3 Configurar hora del sistema

Para fijar la fecha y hora se deben ingresar los valores requeridos, aquí se presentan los comandos utilizando un ejemplo.

- `date --set "2015-10-30 16:25"`
- `hwclock --set --date="2015-10-30 16:25"`

Apéndice 5: Base de datos MySQL

5.1 Tablas creadas para la base de datos MySQL

- **Tabla usuarios:** Esta tabla contiene 3 atributos. El primero el idUsuario es de tipo entero y posee la característica de autoincremento, nombreUsuario una variable del tipo varchar que almacena el nombre de un usuario al sistema y password que también es del tipo varchar que registra una contraseña a cada usuario. La tabla usuarios es independiente de las demás tablas por lo que no presenta relación con ninguna otra tabla. La tabla usuarios se observa en la figura a5.1.

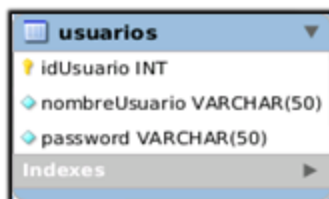


Figura a5.1. Tabla usuarios.

- **Tabla mesas:** La tabla mesas emplea dos atributos, idMesa de tipo entero que posee la cualidad autoincremento y la variable nombre de tipo varchar utilizada para registrar un nombre a cada mesa a agregar. En esta tabla se deben de agregar las 8 mesas a utilizar con los nombres Mesa 1, Mesa 2 y así sucesivamente. La tabla mesas no depende de relaciones para almacenar información, dando total independencia para agregar las mesas deseadas en la base de datos. La tabla mesas se muestra en la figura a5.2.



Figura a5.2 Tabla mesas.

- **Tabla secciones:** Para almacenar los datos asociados a una sección se deben de crear 4 atributos a la tabla secciones, idSeccion de tipo entero con característica de autoincremento, un varchar nombre para guardar el nombre de cada sección, un identificador mesas_idMesa de tipo entero para relacionar con idMesa de la tabla mesas, con el fin de asociar un nombre de sección a cada mesa creada; y por último otra variable entera riegos que se encarga de almacenar la cantidad a regar por día cada sección. Esta tabla se detalla a continuación en la figura a5.3.



Figura a5.3. Tabla secciones.

- **Tabla sensores:** Un criterio de diseño para la base de datos consistió en almacenar en una tabla sensores un nombre específico para cada sensor de la red para asociarlos a los nombre temperatura, humedad o luminosidad dado sea el caso, por lo que esta tabla sensores se presenta como guía para nombrar cada sensor según sea su ubicación por mesas/secciones en el invernadero y asociarlo a algún tipo de sensor.

Para realizar todo esto la tabla posee 4 atributos, un entero idSensor con propiedad de autoincremento, un varchar nombre que se encarga de guardar el nombre para cada sensor, otro entero de tipo índice secciones_idSeccion que se encarga de mantener una relación con idSeccion de la tabla secciones, y un último entero tiposDeSensor_idtiposDeSensor que se relaciona con idtiposDeSensor de la tabla tiposDeSensor. A estas relaciones se le conocen como llaves foráneas (*foreign key*) y se utilizan para mantener vínculos entre tablas. La tabla sensores se visualiza en la figura a5.4.



Figura a5.4. Tabla sensores.

- **Tabla tiposDeSensor:** Para asociar el nombre específico de sensor de la tabla sensores a un tipo de sensor (Temperatura, Humedad o Luminosidad), se crea la tabla tiposDeSensor con 2 atributos. El primero es de tipo entero idtiposDeSensor con la propiedad de autoincremento y el segundo un varchar nombre para almacenar el nombre de cada tipo de sensor. Esta tabla se muestra a continuación en la figura a5.5.

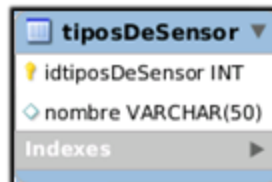
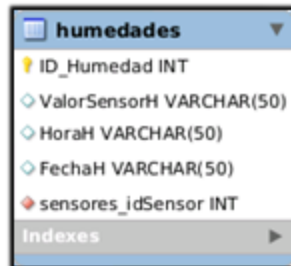


Figura a5.5. Tabla tiposDeSensor.

- **Tabla humedades:** La tabla humedades se encarga de registrar los datos provenientes de los sensores de este tipo en cada mesa/sección activa, para esto se debe utilizar un atributo entero con el nombre sensores_idSensor para relacionarlo con el idSensor de la tabla sensores.

Este mecanismo permite almacenar los datos de todos los sensores de humedad distribuidos por las mesas/secciones del invernadero en una misma tabla, pero con un cierto idSensor que diferencia la mesa/sección específica a la que pertenece cada sensor.

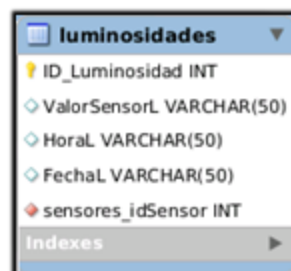
La tabla como todas las demás debe de contener un atributo entero con la propiedad de autoincremento que para este caso se llama ID_Humedad, una variable ValorSensorH de tipo varchar que guarda el dato tomado por el sensor, además de otras 2 variables HoraH y FechaH para almacenar la hora y fecha a la que el sensor registra la información. La tabla humedades se observa en la figura a5.6.



humedades	
ID_Humedad	INT
ValorSensorH	VARCHAR(50)
HoraH	VARCHAR(50)
FechaH	VARCHAR(50)
sensores_idSensor	INT
Indexes	

Figura a5.6. Tabla humedades.

- **Tabla luminosidades:** Como en el caso anterior la tabla luminosidades contiene 5 atributos que presentan las mismas funciones de la tabla humedades, pero nombrados ID_Luminosidad, ValorSensorL, HoraL, FechaL y sensores_idSensor. Esta tabla se encarga de almacenar los datos provenientes de los sensores de luminosidad de todas las mesas/secciones del invernadero. Esta tabla se detalla en la siguiente figura a5.7.



luminosidades	
ID_Luminosidad	INT
ValorSensorL	VARCHAR(50)
HoraL	VARCHAR(50)
FechaL	VARCHAR(50)
sensores_idSensor	INT
Indexes	

Figura a5.7 Tabla luminosidades.

- **Tabla temperaturas:** Al igual que en la tablas humedades y luminosidades, la tabla temperaturas contiene 5 atributos denominados ID_Temperatura, ValorSensorT, HoraT, FechaT y sensores_idSensor. Se utiliza para registrar la información de los

sensores de temperatura de todas las mesas/secciones del invernadero. La tabla temperaturas se visualiza en la figura a5.8.

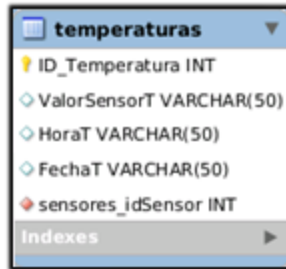


Figura a5.8. Tabla temperaturas.

5.2 Programas creados para la base de datos MySQL

5.2.1 CIBdb.sql

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

DROP SCHEMA IF EXISTS `CIBdb` ;
CREATE SCHEMA IF NOT EXISTS `CIBdb` DEFAULT CHARACTER SET latin2
;
USE `CIBdb` ;

-----
-- Table `CIBdb`.`mesas`
-----
CREATE TABLE IF NOT EXISTS `CIBdb`.`mesas` (
  `idMesa` INT NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(50) NOT NULL ,
  PRIMARY KEY (`idMesa`) )
ENGINE = InnoDB;

-----
-- Table `CIBdb`.`secciones`

```



```

-----
CREATE TABLE IF NOT EXISTS `CIBdb`.`secciones` (
  `idSeccion` INT NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(50) NOT NULL ,
  `mesas_idMesa` INT NOT NULL ,
  `riegos` INT NOT NULL ,
  PRIMARY KEY (`idSeccion`) ,
  CONSTRAINT `fk_secciones_mesas`
    FOREIGN KEY (`mesas_idMesa`)
      REFERENCES `CIBdb`.`mesas` (`idMesa`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE INDEX `idx_secciones_mesas_idMesa` ON `CIBdb`.`secciones`
(`mesas_idMesa` ASC) ;

-----
-- Table `CIBdb`.`tiposDeSensor`
-----
CREATE TABLE IF NOT EXISTS `CIBdb`.`tiposDeSensor` (
  `idtiposDeSensor` INT NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(50) NULL DEFAULT NULL ,
  PRIMARY KEY (`idtiposDeSensor`) )
ENGINE = InnoDB;

-----
-- Table `CIBdb`.`sensores`
-----
CREATE TABLE IF NOT EXISTS `CIBdb`.`sensores` (
  `idSensor` INT NOT NULL AUTO_INCREMENT ,
  `nombre` VARCHAR(50) NOT NULL ,
  `secciones_idSeccion` INT NOT NULL ,
  `tiposDeSensor_idtiposDeSensor` INT NOT NULL ,
  PRIMARY KEY (`idSensor`) ,
  CONSTRAINT `fk_sensores_secciones1`
    FOREIGN KEY (`secciones_idSeccion`)
      REFERENCES `CIBdb`.`secciones` (`idSeccion`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_sensores_tiposDeSensor1`
    FOREIGN KEY (`tiposDeSensor_idtiposDeSensor`)
      REFERENCES `CIBdb`.`tiposDeSensor` (`idtiposDeSensor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

CREATE INDEX `idx_sensores_secciones_idSeccion` ON
`CIBdb`.`sensores` (`secciones_idSeccion` ASC) ;

CREATE INDEX `idx_sensores_tiposDeSensor_idTipoDeSensor` ON
`CIBdb`.`sensores` (`tiposDeSensor_idtiposDeSensor` ASC) ;

```

```

-----
-- Table `CIBdb`.`humedades`
-----

```

```

CREATE TABLE IF NOT EXISTS `CIBdb`.`humedades` (
  `ID_Humedad` INT NOT NULL AUTO_INCREMENT ,
  `ValorSensorH` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraInicioH` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraFinalH` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraH` VARCHAR(50) NULL DEFAULT NULL ,
  `FechaH` VARCHAR(50) NULL DEFAULT NULL ,
  `sensores_idSensor` INT NOT NULL ,
  PRIMARY KEY (`ID_Humedad`) ,
  CONSTRAINT `fk_humedades_sensores1`
    FOREIGN KEY (`sensores_idSensor`)
      REFERENCES `CIBdb`.`sensores` (`idSensor`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin2;

```

```

CREATE INDEX `idx_humedades_sensores_idSensor` ON
`CIBdb`.`humedades` (`sensores_idSensor` ASC) ;

```

```

-----
-- Table `CIBdb`.`luminosidades`
-----

```

```

CREATE TABLE IF NOT EXISTS `CIBdb`.`luminosidades` (
  `ID_Luminosidad` INT NOT NULL AUTO_INCREMENT ,
  `ValorSensorL` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraInicioL` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraFinalL` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraL` VARCHAR(50) NULL DEFAULT NULL ,
  `FechaL` VARCHAR(50) NULL DEFAULT NULL ,
  `sensores_idSensor` INT NOT NULL ,
  PRIMARY KEY (`ID_Luminosidad`) ,
  CONSTRAINT `fk_luminosidades_sensores1`
    FOREIGN KEY (`sensores_idSensor`)
      REFERENCES `CIBdb`.`sensores` (`idSensor`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB

```

```

DEFAULT CHARACTER SET = latin2;

CREATE INDEX `idx_luminosidades_sensores_idSensor` ON
`CIBdb`.`luminosidades` (`sensores_idSensor` ASC) ;

```

```

-----
-- Table `CIBdb`.`temperaturas`
-----

```

```

CREATE TABLE IF NOT EXISTS `CIBdb`.`temperaturas` (
  `ID_Temperatura` INT NOT NULL AUTO_INCREMENT ,
  `ValorSensorT` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraInicioT` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraFinalT` VARCHAR(50) NULL DEFAULT NULL ,
  `HoraT` VARCHAR(50) NULL DEFAULT NULL ,
  `FechaT` VARCHAR(50) NULL DEFAULT NULL ,
  `sensores_idSensor` INT NOT NULL ,
  PRIMARY KEY (`ID_Temperatura`) ,
  CONSTRAINT `fk_temperaturas_sensores1`
    FOREIGN KEY (`sensores_idSensor`)
    REFERENCES `CIBdb`.`sensores` (`idSensor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin2;

```

```

CREATE INDEX `idx_temperaturas_sensores_idSensor` ON
`CIBdb`.`temperaturas` (`sensores_idSensor` ASC) ;

```

```

-----
-- Table `CIBdb`.`usuarios`
-----

```

```

CREATE TABLE IF NOT EXISTS `CIBdb`.`usuarios` (
  `idUsuario` INT NOT NULL AUTO_INCREMENT ,
  `nombreUsuario` VARCHAR(50) NOT NULL ,
  `password` VARCHAR(50) NOT NULL ,
  PRIMARY KEY (`idUsuario`) )
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

```

5.2.2 StoredProcedures.sql

- sudo su
- nano /var/www/CIBinvernadero/db/StoredProcedures.sql

➤ Procedimiento para agregar un usuario:

```
create procedure insertarUsuario (nombreUsuario varchar (50),
password varchar (50))
begin
    insert into usuarios(nombreUsuario, password)
    values(nombreUsuario, password);
end
```

➤ Procedimiento para agregar una mesa:

```
create procedure insertarMesa (nombre varchar(50))
begin
    insert into mesas(nombre)
    values(nombre);
end
```

➤ Procedimiento para agregar una sección:

```
create procedure sp_crearSeccion(nombreSeccion varchar(50),
nombreMesa varchar(50), numeroRiegos int)
begin
    declare mesa_id int;
    select idMesa into mesa_id from mesas m
    where m.nombre = nombreMesa;
    -- Insertamos en la tabla secciones.
    insert into secciones(nombre,mesas_idMesa,riegos)
    values(nombreSeccion,mesa_id,numeroRiegos);
end
```

➤ Procedimiento para agregar un trío de sensores:

```
create procedure sp_agregarSensores(nombreMesa varchar(50),
nombreSeccion varchar(50), nombreSensorT varchar(50),
nombreSensorH varchar(50), nombreSensorL varchar(50))
begin
    declare mesa_id int;
```

```

declare seccion_id int;
declare sensorTempID int;
declare sensorHumID int;
declare sensorLumID int;

select idMesa into mesa_id from mesas m
where m.nombre = nombreMesa;

select idSeccion into seccion_id from secciones s
where s.mesas_idMesa= mesa_id and s.nombre=nombreSeccion ;

-- Seleccionamos el id del sensor tipo temperatura
select ts.idTiposDeSensor into sensorTempID from
tiposDeSensor ts
where ts.nombre = 'Temperatura';
-- Seleccionamos el id del sensor tipo humedad
select ts.idTiposDeSensor into sensorHumID from
tiposDeSensor ts
where ts.nombre = 'Humedad';
-- Seleccionamos el id del sensor tipo luminosidad
select ts.idTiposDeSensor into sensorLumID from
tiposDeSensor ts
where ts.nombre = 'Luminosidad';
-- Insertamos en la tabla sensores.
insert into sensores(nombre,
secciones_idSeccion,tiposDeSensor_idTiposDeSensor)
values(nombreSensorT,seccion_id,
sensorTempID), (nombreSensorH,seccion_id,
sensorHumID), (nombreSensorL,seccion_id, sensorLumID);

end

```

➤ *Procedimiento para agregar valor de temperatura:*

```

create procedure sp_agregarTemperatura(nombreSensor varchar(50),
ValorSensor varchar(50))
begin
declare sensor_id int;
select idSensor into sensor_id from sensores s
where s.nombre = nombreSensor;
-- Insertamos en la tabla temperaturas.
insert into temperaturas(ValorSensorT, HoraT,
FechaT,sensores_idSensor)
values (ValorSensor, curTime(), curDate(), sensor_id);
end

```

➤ Procedimiento para agregar valor de humedad:

```
create procedure sp_agregarHumedad(nombreSensor varchar(50),
ValorSensor varchar(50))
begin
    declare sensor_id int;
    select idSensor into sensor_id from sensores s
    where s.nombre = nombreSensor;
    -- Insertamos en la tabla humedades.
    insert into humedades(ValorSensorH, HoraH, FechaH,
sensores_idSensor)
    values (ValorSensor, curTime(), curDate(), sensor_id);
end
```

➤ Procedimiento para agregar valor de luminosidad:

```
create procedure sp_agregarLuminosidad(nombreSensor varchar(50),
ValorSensor varchar(50))
begin
    declare sensor_id int;
    select idSensor into sensor_id from sensores s
    where s.nombre = nombreSensor;
    -- Insertamos en la tabla luminosidades.
    insert into luminosidades(ValorSensorL, HoraL, FechaL,
sensores_idSensor)
    values (ValorSensor, curTime(), curDate(), sensor_id);
end
```

➤ Procedimiento para borrar toda la información asociada una sección:

```
create procedure sp_borrarSeccion(mesa varchar(50), seccion
varchar(50))
begin
    declare seccionID int;
    declare sensorID int;
    declare sensorTipoID int;
    declare mesaID int;

    -- Seleccionamos el id de la seccion a eliminar
    select s.idSeccion into seccionID from secciones s
```

```

inner join mesas m on m.idMesa = s.mesas_idMesa
where s.nombre = seccion and m.nombre = mesa;

-- Seleccionar el id del sensor tipo temperatura
select ts.idTiposDeSensor into sensorTipoID from
tiposDeSensor ts
where ts.nombre = 'Temperatura';
-- Seleccionar el id del sensor de temperatura
correspondiente a la seccion deseada
select idSensor into sensorID from sensores sn
where sn.secciones_idSeccion = seccionID and
      sn.tiposDeSensor_idTiposDeSensor = sensorTipoID ;
-- Borrarnos de la tabla temperauras los datos recolectados
por el sensor de temperatura
delete from temperaturas where sensores_idSensor = sensorID;
delete from sensores where secciones_idSeccion = seccionID
and tiposDeSensor_idtiposDeSensor = sensorTipoID;

-- Seleccionamos el id del sensor tipo humedad
select sn.idTiposDeSensor into sensorTipoID from
tiposDeSensor sn
where sn.nombre = 'Humedad';
-- Seleccionar el id del sensor de humedad correspondiente a
la seccion de deseada
select idSensor into sensorID from sensores sn
where sn.secciones_idSeccion = seccionID and
      sn.tiposDeSensor_idTiposDeSensor = sensorTipoID ;
-- Borrarnos de la tabla humedades los datos recolectados por
el sensor de humedad
delete from humedades where sensores_idSensor = sensorID;
delete from sensores where secciones_idSeccion = seccionID
and tiposDeSensor_idtiposDeSensor = sensorTipoID;

-- Seleccionamos el id del sensor tipo luminosidad
select sn.idTiposDeSensor into sensorTipoID from
tiposDeSensor sn
where sn.nombre = 'Luminosidad';
-- Seleccionar el id del sensor de humedad correspondiente a
la seccion de deseada
select idSensor into sensorID from sensores sn
where sn.secciones_idSeccion = seccionID and
      sn.tiposDeSensor_idTiposDeSensor = sensorTipoID ;
-- Borrarnos de la tabla luminosidades los datos recolectados
por el sensor de luminosidad
delete from luminosidades where sensores_idSensor =
sensorID;

```

```

delete from sensores where secciones_idSeccion = seccionID
and tiposDeSensor_idtiposDeSensor = sensorTipoID;

-- Borramos tabla secciones
select idMesa into mesaID from mesas m
where m.nombre = mesa;
select idSeccion into seccionID from secciones s
where s.mesas_idMesa= mesaID and s.nombre = seccion ;
delete from secciones where mesas_idMesa= mesaID and nombre
= seccion ;
end

```

5.2.3 LlenadoTablas.sql

- sudo su
- nano /var/www/CIBinvernadero/db/LlenadoTablas.sql

➤ Tablas mesas:

```

-- se insertan las mesas
insert into mesas(nombre)
values('Mesa 1'), ('Mesa 2'), ('Mesa 3'), ('Mesa 4'), ('Mesa 5'),
('Mesa 6'), ('Mesa 7'), ('Mesa 8');

```

➤ Tabla tiposDeSensor:

```

-- Se inserta el nombre de cada sensor
insert into tiposDeSensor(nombre)
values('Temperatura'), ('Humedad'), ('Luminosidad');

```

➤ Tabla usuarios:

```

-- se insertan los usuarios
insert into usuarios(nombreUsuario,password)
values ('Administrador', 'CIBbiotec.invernadero2015');

```


Apéndice 6: Circuito impreso PCB

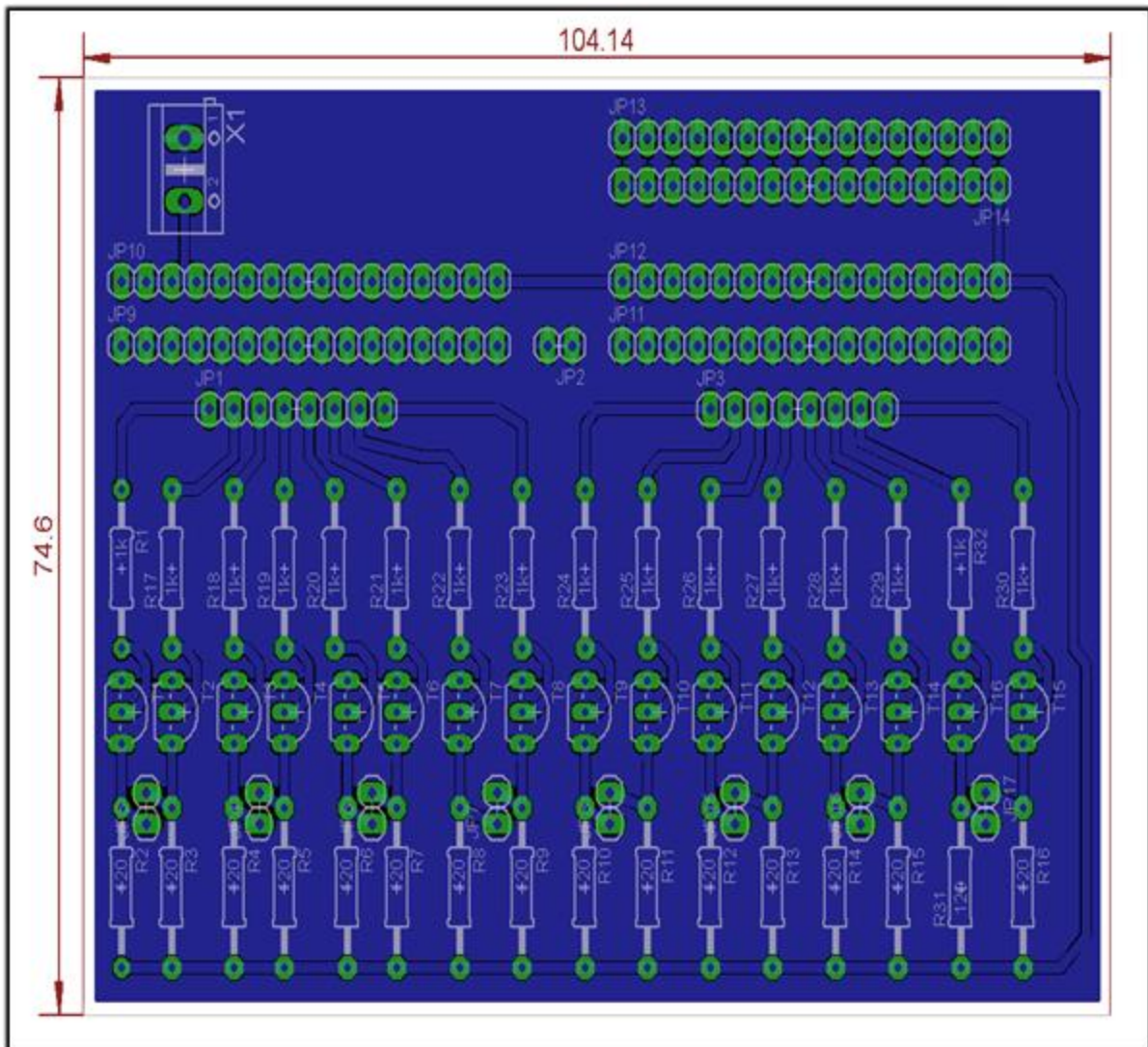


Figura a6.1. Plano de tierras y distribución de los componentes en el PCB.

Apéndice 7: Prototipo desarrollado

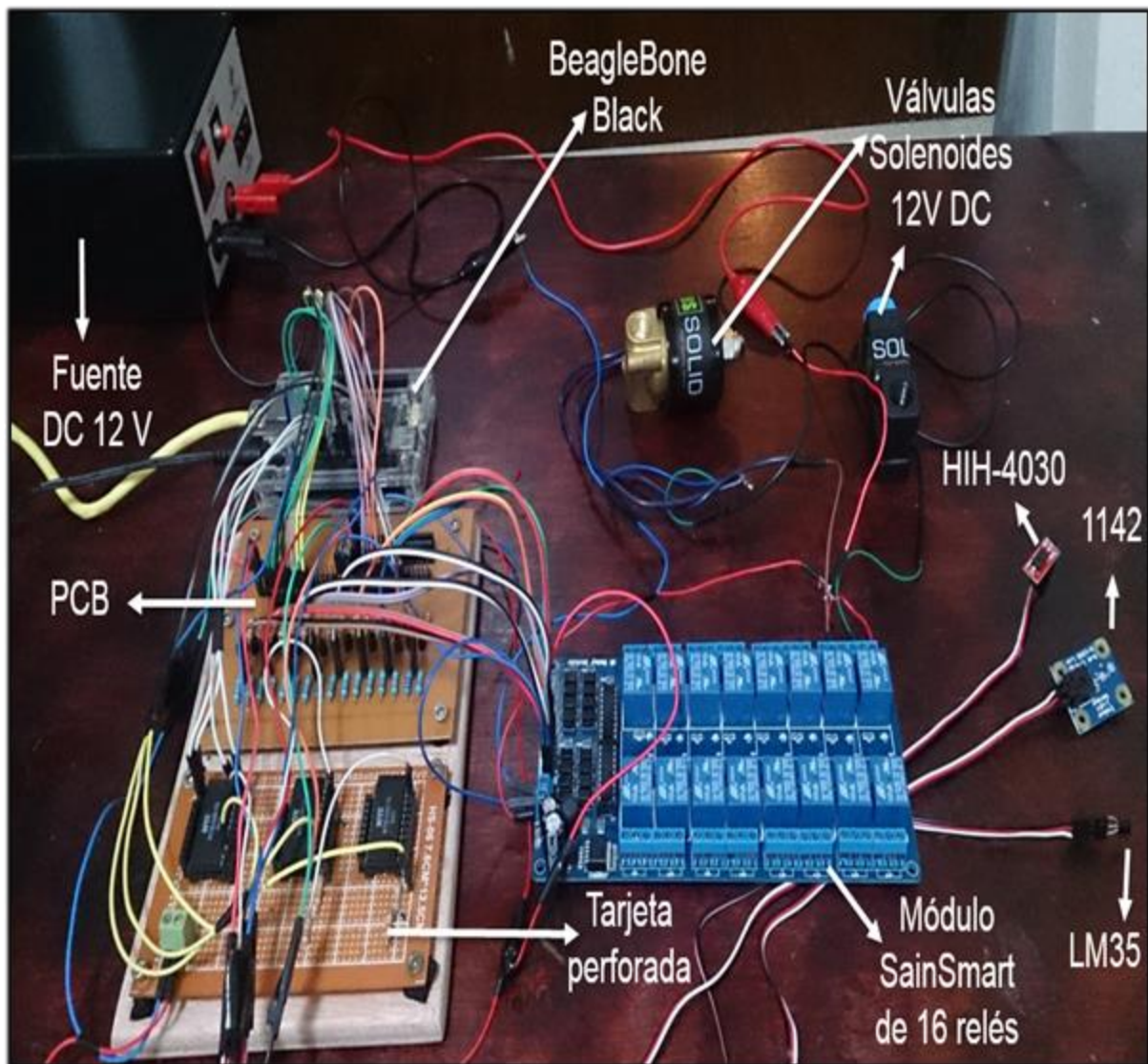


Figura a7.1. Foto del prototipo implementado en este proyecto de graduación.