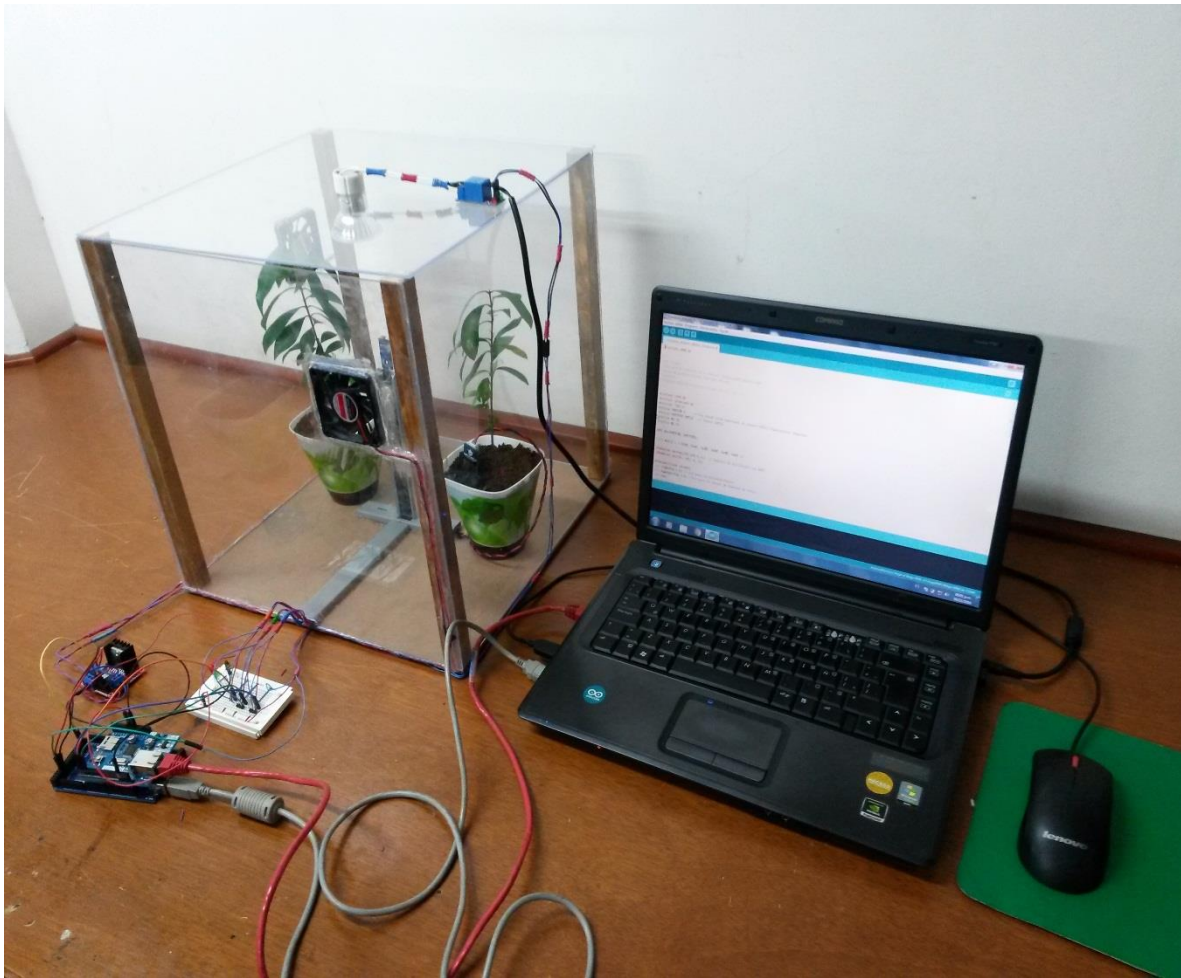


DISEÑO DE UN SISTEMA DE MONITOREO, REGISTRO Y CONTROL DE TEMPERATURA Y HUMEDAD PARA UN CULTIVO DE INVERNADERO.



Jhonny Wbeimar Perea Palacios

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA ELÉCTRICA

Estudiante:

JHONNY WBEIMAR PEREA PALACIOS

Código: 82'260.122.

Modalidad:

Trabajo de Investigación Formativa

Título:

DISEÑO DE UN SISTEMA DE MONITOREO, REGISTRO Y CONTROL DE
TEMPERATURA Y HUMEDAD PARA UN CULTIVO DE INVERNADERO.

Pereira, Noviembre de 2016

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

Programa de Ingeniería Eléctrica.

Grupo de Investigación en Ingeniería Electrónica



DISEÑO DE UN SISTEMA DE MONITOREO, REGISTRO Y CONTROL DE
TEMPERATURA Y HUMEDAD PARA UN CULTIVO DE INVERNADERO.

© Jhonny Wbeimar Perea Palacios

Co-Director: Ing. M.Sc(C). Arley Bejarano Martínez

Director: Ing. M.Sc. Andrés Felipe Calvo Salcedo

Pereira, Noviembre de 2016

Programa de Ingeniería Eléctrica.

Universidad Tecnológica de Pereira

La Julita. Pereira (Colombia)

TEL: (+57) (6)3137122

www.utp.edu.co

Versión web disponible en: <http://recursosbiblioteca.utp.edu.co/tesis/index.html>

Agradecimientos

Primeramente, a Dios, a mis padres y hermanos por el apoyo incondicional durante éste arduo y dificultoso trayecto. Gracias por el soporte, la paciencia y el amor.

Deseo hacer una mención especial para el Ingeniero Andrés Felipe Calvo Salcedo por la acogida y confianza para la realización de éste proyecto quien junto al Ingeniero Arley Bejarano Martínez fueron de vital importancia para la culminación exitosa.

Al grupo de docentes y funcionarios del programa de Ingeniería Eléctrica por la enseñanza y disposición brindada, en especial a Clara Inés ortega Vinasco por la ayuda y el apoyo incondicional, así como los profesores Alfonso Álzate y Ricardo Hincapié por los consejos, por ser una guía y principalmente un ejemplo a seguir durante todo el proceso de mi formación académica y personal.

A mis compañeros, amistades y en general a el grupo de personas que de un modo u otro aportaron para lograr ésta meta.

RESUMEN

El presente proyecto de grado consiste en el desarrollo de una aplicación la cual permite monitoreo, registro y control de un cultivo en invernadero a través de la implementación de diversos dispositivos (microcontrolador, sensores y actuadores), para brindar al usuario un historial de las mediciones de las variables físicas más relevantes (Temperatura, humedad relativa, luminosidad y humedad del suelo) en todo el proceso de desarrollo del cultivo.

Para lograr dichas medidas se utilizarán diferentes tipos de sensores los cuales arrojaran medidas calibradas de las variables de interés sensadas de forma iterada en el transcurso del tiempo, estos sensores estarán conectados a un dispositivo microcontrolador de la familia Arduino, el cual recibirá los datos y los enviara a través de Ethernet a un servidor, para posteriormente realizar una comunicación con la aplicación desarrollada en PHP para almacenar las medidas en una base de datos SQL, luego de esto se realizara una representación gráfica detallada en intervalos de tiempo y rangos de medida, para poder observar el comportamiento de las variables a través del tiempo, además le permite al usuario poder descargar gráficos en tiempo real y archivos en Excel de la base de datos para realizar análisis estadísticos .

Es importante denotar que el control implementado es uno tipo on-off, con una histéresis de $1[^\circ\text{C}]$ para la temperatura, permitiendo mantener en un rango predefinido las variables de interés para el óptimo desarrollo del cultivo.

Tabla de contenido

1.INTRODUCCIÓN	9
1.1. PLANTEAMIENTO DEL PROBLEMA	10
2. OBJETIVOS	11
2.1. Objetivo General	11
2.2. Objetivos Específicos	11
3. METODOLOGÍA.....	12
3.1. Etapas metodológicas	12
3.2. Criterios de validez	13
4. ESTADO DEL ARTE.....	14
4.1. Importancia del cultivo bajo invernaderos	14
4.2. Problemática que afrontan los invernaderos no automatizados.....	15
4.3. Situación actual del cultivo en invernaderos en Colombia	15
5. DESARROLLO	17
5.1. Sistema microcontrolador	17
5.2. Arduino	18
5.3. Sensores	20
5.4. Desarrollo metodológico	24
5.5. Circuito implementado.....	26
5.5.1. Medición de las variables físicas (Temperatura, humedad relativa, luminosidad y humedad del suelo) con Arduino	28
5.5.2. Conexión Arduino a base de datos MySQL	28
5.5.3. Interfaz gráfica.....	29
5.6. Diagramas sistema de control y registro.....	30
5.6.1. Diagrama sistema de control	31
5.6.2. Diagrama sistema de captura y registro	32

5.7. Sistema desarrollado	33
6. RESULTADOS.....	35
7. CONCLUSIONES, APORTES Y RECOMENDACIONES	42
7.1. Conclusiones.....	42
7.2. Aportes	42
7.3. Recomendaciones	42
8. BIBLIOGRAFÍA	44
Anexos	46

Índice de figuras

1. Arduino Mega 2560	17
2. DTH22, Sensor de temperatura y Humedad	21
3. Higrómetro, módulo SKU (SEN0114)	22
4. LDR (Sensor de luminosidad)	23
5. Circuito implementado	26
6. Interfaz gráfica	29
7. Diagrama sistema de control	31
8. Diagrama sistema de captura y registro	32
9. Sistema implementado	33
10. Visualización del COM Arduino.....	35
11. Visualización de la base de datos en MySQL.....	36
12. Visualización interfaz gráfica.....	37
13. Visualización grafica Temperatura Ambiente.....	37
14. Visualización grafica Humedad del Ambiente.....	38
15. Visualización grafica Luminosidad.....	39
16. Visualización grafica Humedad del Suelo.....	39
17. Visualización grafica control de temperatura.....	40
18. Visualización grafica control de humedad relativa del ambiente	41

1.INTRODUCCIÓN

Colombia es un país con grandes recursos naturales, su economía en el transcurso de la historia ha dependido de la explotación de dichos recursos. La globalización de los mercados exige una mayor competitividad, esto es, cumplir con procesos de producción y calidad a fin de lograr comercializar sus productos en mercados tanto locales como internacionales; por tal motivo todo lo que se produzca en el país debe tener los mejores estándares de calidad y eficiencia que le permitan competir en esta nueva economía.

Nuestro sector agrícola se explota u aprovecha en un bajo porcentaje por lo que la tecnificación y automatización de la producción agrícola es prioritaria si se desea competir en el ámbito económico mundial [1].

El agricultor es reacio a cambios de paradigma, ya que asocia tecnología con dificultad de manejo y altos costos, es por eso que se busca diseñar e implementar un sistema fiable de bajo costo que cumpla con las normativas vigentes.

En nuestro país se han realizado estudios y aplicaciones con invernaderos los cuales se han abordado de manera local y requieren la constante intervención por parte del hombre, por ende, se busca un sistema autónomo y controlable desde cualquier parte del mundo [2].

Un invernadero es una estructura cerrada, cubierta por materiales transparentes, dentro de la cual es posible obtener unas condiciones artificiales de microclima controlado, y con ello realizar cosechas en condiciones controladas obteniendo producciones de mayor calidad y mejores precios en cualquier instante del año [4]; El desarrollo tecnológico de los cultivos permite llevar un control en sus diferentes fases de crecimiento. La calidad de vida de una cosecha se logra con un microclima estable y acorde a las condiciones del producto a cultivar, por tal motivo es indispensable censar y manipular las variables climáticas prioritarias.

Después de controlar el microclima del invernadero, se almacenan en una base de datos las variables y detalles más importantes para garantizar un historial de las variables de estado que el agricultor desea analizar. Con esta base de datos, se tiene una herramienta para optimizar las cosechas y garantizar control predictivo [5].

1.1. PLANTEAMIENTO DEL PROBLEMA

La principal función de un invernadero es la de controlar y mantener un espacio delimitado de terreno bajo condiciones climáticas ideales dependiendo del cultivo, con el fin de mantener estables las variables de interés: Humedad relativa del aire, temperatura ambiente interna, luminosidad entre otras. Sin embargo, en nuestro país se habla mucho de automatización, pero se aplica poco, ya que la mayoría de invernaderos son controlados por descarte de clima, mas no por sensores calibrados que indiquen los diferentes niveles para su optimización y medida, y son controlados total o parcialmente por operadores en el mismo sitio del cultivo, por lo que se busca dar una solución a dicho inconveniente.

Una alternativa para solucionar esta situación ha sido la instrumentación, registro y control de los invernaderos con el fin de mejorar cada proceso implementando tecnología liviana y de bajo costo que permita el adecuado cuidado y desarrollo que se requiere en cada producción, apoyándose de sistemas controlados de forma remota. Esta instrumentación permite mejorar los estándares de calidad al reducir la probabilidad de error, al mismo tiempo que se aumenta la capacidad y velocidad de producción del producto.

En vista del rápido crecimiento de la tecnología y los elementos de comunicación desarrollados, se piensa en la posibilidad de adaptar parte de la tecnología existente para mejorar el control y monitoreo de un cultivo bajo invernadero.

Conjunto a lo anterior Tecnoparque SENA requiere de la automatización de un cultivo de invernadero en el corregimiento Arabia del municipio de Pereira (Risaralda) con el fin de implementar tecnológicamente dichas instalaciones. Este requerimiento se logra por medio del convenio interinstitucional SENA-UTP entre el Grupo de Investigación de Ingeniería Electrónica y Tecnoparque SENA.

2. OBJETIVOS

2.1. Objetivo General

Desarrollar un sistema automático que permita el monitoreo de las variables físicas, temperatura, humedad relativa, humedad del suelo y luminosidad, así como el control de la temperatura y la humedad relativa de un sistema de invernadero con registro de variables en una base de datos online con lenguaje SQL

2.2. Objetivos Específicos

- Diseñar un sistema de instrumentación y medida para variables de Temperatura, humedad relativa y humedad del suelo y luminosidad.
- Diseñar un sistema de gestión de datos que permita el almacenamiento y consulta en internet del historial de las medidas.
- Realizar pruebas de control con el fin de verificar el correcto funcionamiento del registro de las variables medidas utilizando el sistema de gestión de datos.
- Validar estadísticamente el sistema de instrumentación y medida.

3. METODOLOGÍA

3.1. Etapas metodológicas

El objetivo de este proyecto es trabajar en la automatización, monitoreo y control de un cultivo invernadero.

Los objetivos específicos guiarán el desarrollo del trabajo; la realización de este proyecto se desarrollará en 5 etapas metodológicas enunciadas a continuación:

Etapa I: Establecer el estado del arte en el tema de estudio, recopilando información de los diferentes trabajos realizados en este campo.

Etapa II: Realizar el diseño de un sistema de instrumentación y medida para variables de Temperatura, humedad relativa y humedad del suelo.

- Obtener los sensores requeridos
- Calibrar dichos sensores mediante los equipos patrón y la asesoría de la empresa Metromedics S.A.S. con la cual el grupo de investigación tiene convenio.

Etapa III: Diseñar un sistema de gestión de datos que permita el almacenamiento y consulta online del historial de las medidas.

- Escoger sistema embebido para la adquisición de datos.
- Diseñar y construir el enlace hombre máquina para lectura y almacenamiento de datos.
- Ingresar a la base de datos.

Etapa IV: Diseñar el sistema de automatización que permita el control de las variables: temperatura, humedad relativa y humedad del suelo

- Seleccionar el entorno de programación que facilite la realización del código de automatización.
- Seleccionar el control adecuado a implementar.

Etapas V: Validar estadísticamente el sistema de automatización y control

- Por medio del sistema microcontrolado capturar las medidas y registrar la información en la base de datos.
- Extraer registros de la base de datos, graficarlos y permitir su descarga.
- Verificar que ante perturbaciones climáticas el sistema opera en los rangos preestablecidos.

3.2. Criterios de validez

- Para la elaboración de este proyecto se toman en cuenta los criterios soportados en los documentos de la referencia bibliográfica número [1] hasta la número [16].
- Validación de la metodología mediante la medición de la temperatura, humedad relativa, humedad del suelo y luminosidad; consulta en la base de datos y control del sistema ante fluctuaciones climáticas no deseadas.

4. ESTADO DEL ARTE

Actualmente diversos sistemas y procesos se encuentran automatizados, es decir, se han aplicado sistemas mecánicos y electrónicos para controlar el proceso y no requerir intervención humana para su funcionamiento. La automatización se ha desarrollado rápidamente, se han creado nuevos avances tecnológicos y técnicas de control, que son aplicados en diversas áreas como la industria, medicina, aeronáutica, entre otros.

En el área de la agronomía, la automatización cumple un rol muy importante debido a que optimiza los procesos, incrementando la productividad y mejora la calidad de los productos, esto ayuda a satisfacer las exigencias del mercado. Por otro lado, los avances dentro del control climático han permitido la mejora de sistemas como es el caso de los invernaderos.

Diferentes métodos y modelos de control son aplicados para controlar el comportamiento de las variables climáticas que afectan el entorno del invernadero. Se usan desde sistemas de control usando lógica clásica hasta sistemas de control inteligente, que usan lógica difusa, redes neuronales, entre otros. Además, se han desarrollado diferentes dispositivos y mecanismos que permiten medir y variar las variables de interés dentro del invernadero [13].

4.1. Importancia del cultivo bajo invernaderos

Los consumidores demandan productos agrícolas de alta calidad y diversidad durante todo el año, esto no es posible debido a los cambios climáticos que se producen, es por ello que el uso de invernaderos juega un papel muy importante para enfrentar este problema.

Los invernaderos permiten obtener condiciones climáticas adecuadas para el desarrollo y crecimiento de las plantas dentro de un espacio cerrado. Por lo tanto, se crea una barrera física entre los cultivos dentro del invernadero y el clima exterior, esto ayuda a proteger a las plantas de los fenómenos atmosféricos como la caída de granizas y heladas que dañan los cultivos, así como también de los fuertes vientos, las lluvias y elevadas temperaturas. Por otro lado, facilita el control de plagas y enfermedades las cuales pueden ocasionar grandes pérdidas en la producción.

En consecuencia, la implementación de invernaderos permite al agricultor obtener cosechas durante todo el año, incluso en épocas de invierno, ayuda a reducir el tiempo y obtener productos agrícolas de alta calidad.

4.2. Problemática que afrontan los invernaderos no automatizados

La mayoría de agricultores que optan por construir invernaderos para proteger sus cultivos del clima externo y crear un microclima dentro de este espacio cerrado, no toman en cuenta la implementación de un sistema automatizado que los ayude a lograr establecer las mejores condiciones climáticas de acuerdo a la necesidad de cada tipo de cultivo.

Las variables climáticas como la temperatura, humedad, CO₂ y la radiación solar influyen en el crecimiento y desarrollo de las plantas. Un mal manejo de estos parámetros puede afectar considerablemente a la planta. Por ejemplo, un exceso de humedad en el interior del invernadero puede generar enfermedades en los cultivos.

Por otro lado, existe un mal manejo de recursos naturales como el agua, cuando el invernadero no cuenta con un sistema que permita monitorear continuamente las variables climáticas como temperatura y humedad, no se establece un tiempo de riego adecuado y se dificulta mantener la temperatura en las condiciones ideales para el cultivo, ya que no se sabe cuánta agua se necesita o con qué frecuencia se debe efectuar el riego. Los invernaderos operados manualmente por el agricultor, ayudan a proteger los cultivos de los fenómenos climáticos y crear un microclima; sin embargo, no brinda las ventajas o eficiencias que se logran obtener con un invernadero automatizado.

4.3. Situación actual del cultivo en invernaderos en Colombia

En muchos países la automatización de invernaderos se encuentra muy desarrollada.

Estos invernaderos están caracterizados por aplicar diversas tecnologías de control y comunicación como buses de campo, redes inalámbricas de sensores, control inteligente, entre otros.

En Colombia, muchos departamentos utilizan invernaderos para mejorar la calidad de sus cultivos y aumentar la producción, así como también para protegerlos del clima, que en los últimos años ha ido cambiando debido al calentamiento global. Sin embargo, la mayoría de estos invernaderos son operados manualmente por los agricultores, no poseen ningún sistema automatizado que controle los parámetros físicos de interés, tampoco se cuenta con un sistema de monitoreo y recopilación de datos, los cuales llegarían a ser una fuente de datos muy importante para mejorar el manejo de cultivos en invernaderos y así llevar un registro detallado del comportamiento del cultivo.

Así mismo, no se cuenta con sensores que midan las variables como temperatura y humedad del invernadero de forma calibrada, los agricultores no pueden saber con exactitud la humedad relativa del ambiente y/o la temperatura durante la noche, entre otros. No se darían cuenta si se produce un cambio en la humedad que podría causar problemas o enfermedades en las plantas.

Por último, la variación de cualquier variable climática dentro del invernadero se realiza de forma manual, por simple experiencia, ya que no se cuenta con sistemas de calefacción, ventilación, importantes para establecer condiciones de humedad o temperatura adecuadas para la planta. Esto ocasiona un uso ineficiente de recursos como el agua y la energía eléctrica.

El cultivo en Colombia bajo invernadero desde un principio y hasta la fecha se ha centrado básicamente en la producción de flores de corte para exportación, las cuales se ubican en la zona de la sabana de Bogotá, el altiplano Cundiboyasense y la región de Rionegro Antioquia. Sin embargo, el control del clima se limita básicamente a la protección de los cultivos a las bajas temperaturas, en épocas de heladas y de lluvias excesivas, generando altos costos por las continuas aplicaciones de químicos, retraso en la producción, calidades deficientes para su mercado.

Durante 25 años de desarrollo de esta industria, la infraestructura en invernaderos casi no ha cambiado desde el punto de vista de la climatización: Los invernaderos en Colombia son de bajo costo comparados con la infraestructura en otras latitudes, pero no ofrecen un control climático. Esta situación contrasta con los principios de control de los procesos biológicos en horticultura, donde se trata de optimizar el micro clima por medio de invernaderos, equipos y mecanismos de control apropiados.

5. DESARROLLO

Una vez establecido el estado del arte en el tema de estudio y recopilando información de los diferentes trabajos realizados en este campo se procedió a implementar el sistema de sensado y automatización.

5.1. Sistema microcontrolador

Por todas las características que nos brinda sumado al bajo precio de mercado y a su versatilidad para realizar montajes electrónicos de control, registro y censado de variables físicas se optó por microcontrolador Arduino Mega 2560.

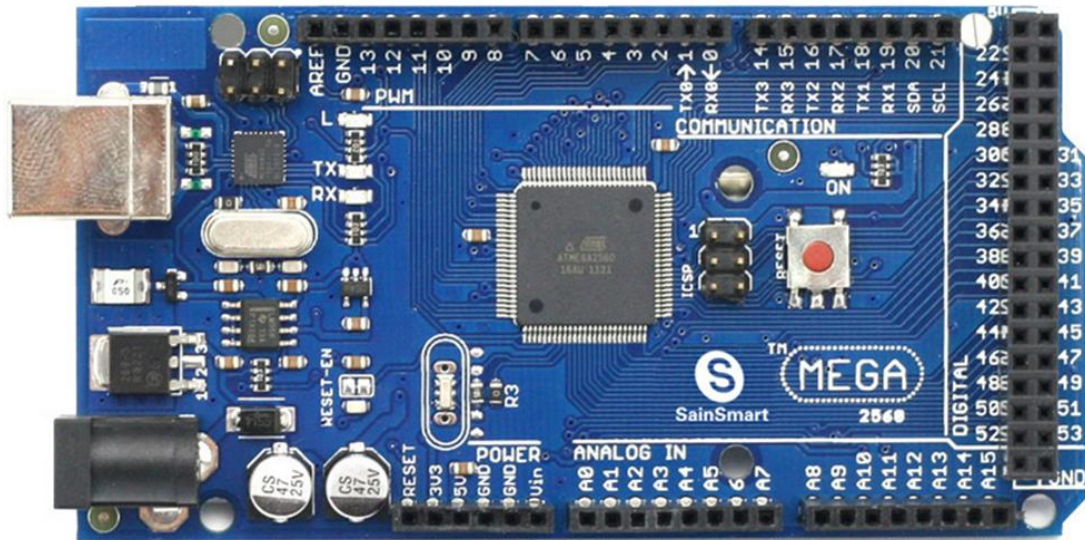


Figura 1. Arduino Mega 2560

5.2. Arduino

Arduino es una plataforma de prototipos electrónicos de código abierto (open-source) basado en hardware y software flexibles y fáciles de usar. Está pensado para estudiantes, diseñadores, y para cualquiera interesado en crear objetos o entornos interactivos.

Arduino puede interpretar el entorno mediante la recepción de sus entradas desde una variedad de sensores y puede afectar su entorno mediante el control de transductores, luces, motores y otros artefactos. El Microcontrolador de la placa se programa usando el “Arduino Programming Language” (basado en Wiring1) y el “Arduino Development Environment” (basado en Processing2). Los proyectos de Arduino pueden ser autónomos o se pueden comunicar con software en ejecución en un ordenador, tableta o equipo móvil.

Las placas se pueden ensamblar a mano o encargarse preensambladas; el software se puede descargar gratuitamente. Los diseños de referencia del hardware están disponibles bajo licencia open-source, por lo que eres libre de adaptarlas a tus necesidades.

Hay muchos otros microcontroladores y plataformas microcontroladoras disponibles para computación física. Microcontrolador PIC, Parallax Basic Stamp, Netmedia's BX-24, Phidgets, MIT's Handyboard, Raspberry Pi y muchas otras ofertas de funcionalidad similar. Todas estas herramientas toman los desordenados detalles de la programación de Microcontrolador y la encierran en un paquete fácil de usar. Arduino también simplifica el proceso de trabajo con microcontroladores, pero ofrece algunas ventajas para profesores, estudiantes y aficionados interesados sobre otros sistemas:

Las placas Arduino son relativamente baratas comparadas con otras plataformas microcontroladoras. La versión menos cara del módulo Arduino puede ser ensamblada a mano, e incluso los módulos de Arduino preensamblados cuestan menos de 50\$.

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

El entorno de programación de Arduino es fácil de usar para principiantes, pero suficientemente flexible para que usuarios avanzados puedan aprovecharlo también. Para profesores, está convenientemente basado en el entorno de programación Processing, de manera que estudiantes aprendiendo a programar en ese entorno estarán familiarizados con el aspecto y la imagen de Arduino.

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

El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia

Creative Commons, por lo que diseñadores experimentados de circuitos pueden hacer su propia versión del módulo, extendiéndolo y mejorándolo. Incluso usuarios relativamente inexpertos pueden construir la versión de la placa del módulo para entender cómo funciona y ahorrar dinero.

Arduino Mega 2560 es una versión ampliada de la tarjeta original de Arduino y está basada en el Microcontrolador Atmega2560. Dispone de 54 entradas/salidas digitales, 14 de las cuales se pueden utilizar como salidas PWM (modulación de anchura de pulso). Además, dispone de 16 entradas analógicas, 4 UARTs (puertas series), un oscilador de 16MHz, una conexión USB, un conector de alimentación, un conector ICSP y un pulsador para el reset. Para empezar a utilizar la placa sólo es necesario conectarla al ordenador a través de un cable USB, o alimentarla con un adaptador de corriente AC/DC. También, para empezar, puede alimentarse mediante una batería.

Una de las diferencias principales de la tarjeta Arduino MEGA 2560 es que no utiliza el convertidor USB-serie de la firma FTDI. Por lo contrario, emplea un Microcontrolador Atmega8U2 programado como actuar convertidor USB a serie.

Esta placa debido a su gran poder es utilizada para grandes proyectos, entre los más importantes se encuentran los de DOMOTICA e IMPRESORAS 3D.

El Arduino MEGA2560 es compatible con la mayoría de los shield o tarjetas de aplicación/ampliación disponible para las tarjetas Arduino UNO original.

Características del Arduino Mega 2560

- Microprocesador ATmega2560
- Tensión de alimentación (recomendado) 7-12V
- Integra regulación y estabilización de +5Vcc
- 54 líneas de Entradas/Salidas Digitales (14 de ellas se pueden utilizar como salidas PWM)
- 16 Entradas Analógicas
- Máxima corriente continua para las entradas: 40 mA
- Salida de alimentación a 3.3V con 50 mA
- Memoria de programa de 256Kb (el bootloader ocupa 8Kb)
- Memoria SRAM de 8Kb para datos y variables del programa
- Memoria EEPROM para datos y variables no volátiles
- Velocidad del reloj de trabajo de 16MHz
- Reducidas dimensiones de 100 x 50 mm

5.3. Sensores

Para la realización de éste proyecto se van a implementar diferentes tipos de sensores los cuales se denotarán a continuación:

DHT22

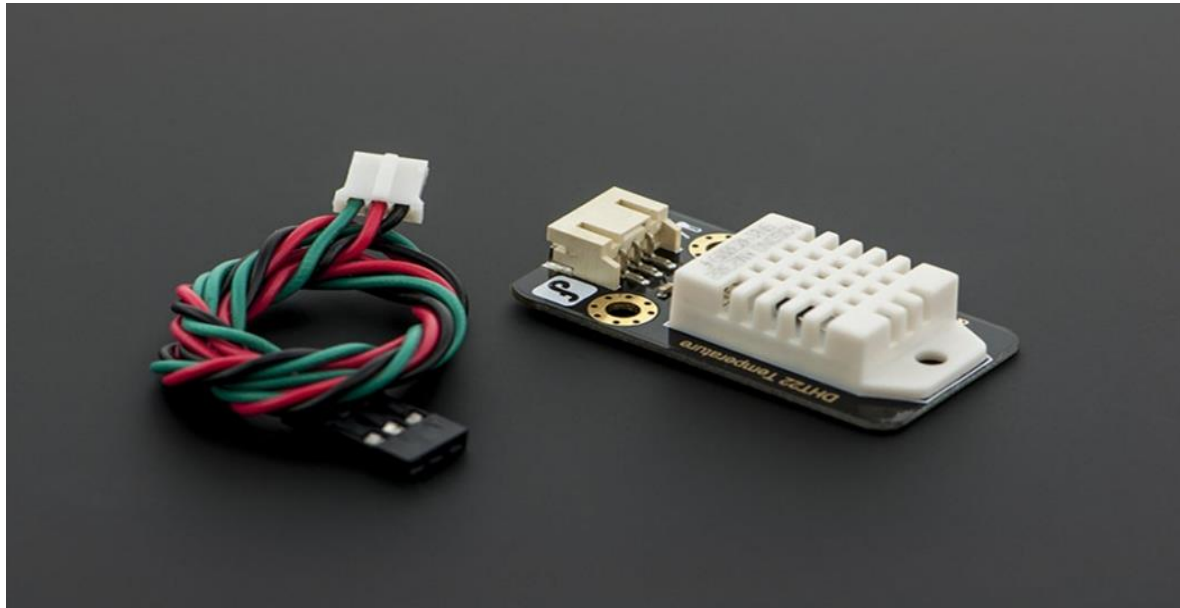


Figura 2. DTH22, Sensor de temperatura y Humedad

El DHT22 Sensor digital de temperatura y humedad. Utiliza un sensor capacitivo de humedad y un termistor para medir el aire circundante, y muestra los datos mediante una señal digital en el pin de datos (no hay pines de entrada analógica). Es bastante simple de usar, pero requiere sincronización cuidadosa para tomar datos. El único inconveniente de este sensor es que sólo se puede obtener nuevos datos una vez cada 2 segundos, así que las lecturas que se pueden realizar serán mínimo cada 2 segundos.

Sólo tienes que conectar el primer pin de la izquierda a la fuente de alimentación 3-5V, el segundo pin a tu pin de entrada de datos y el cuarto (último) pin a tierra. ¡A pesar de que utiliza un solo cable para enviar datos no es compatible con el protocolo de Dallas One-Wire! Si desea conectar varios sensores, cada uno debe tener su propio pin de datos, Hay una librería de ejemplo disponible para Arduino.

Características

- Alimentación: $3.3Vdc \leq Vcc \leq 6Vdc$
- Rango de medición de temperatura: $-40^{\circ}C$ a $80^{\circ}C$

- Precisión de medición de temperatura: $<\pm 0.5$ °C
- Resolución Temperatura: 0.1°C
- Rango de medición de humedad: De 0 a 100% RH
- Precisión de medición de humedad: 2% RH
- Resolución Humedad: 0.1%RH
- Tiempo de sensado: 2s

Sensor de humedad de suelo SKU (SEN0137)

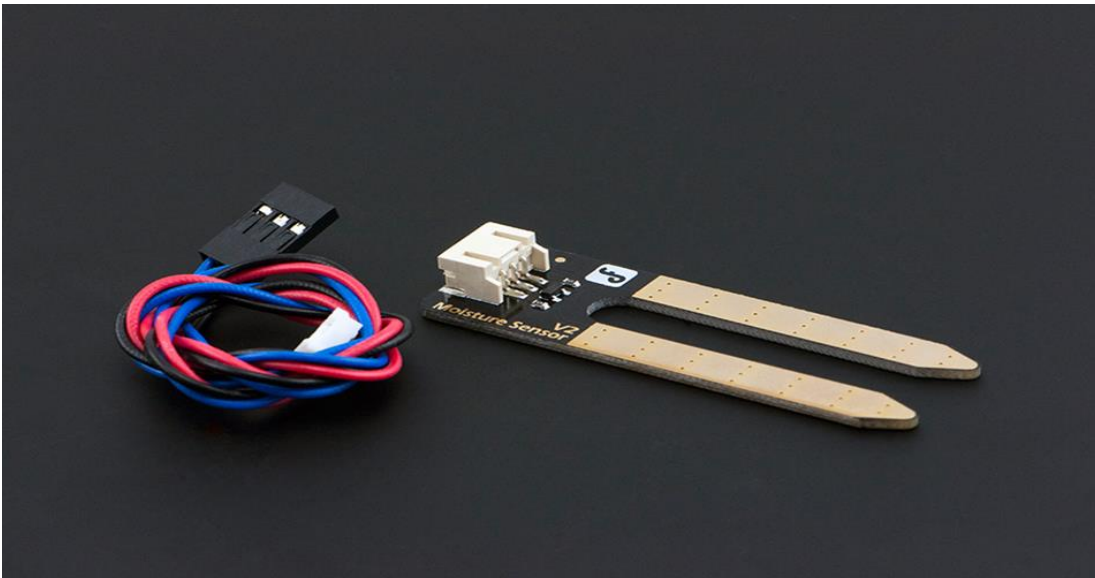


Figura 3. Higrómetro, módulo SKU (SEN0114).

Este sensor puede leer la cantidad de humedad presente en el suelo que lo rodea. Es un sensor de baja tecnología, pero es ideal para el seguimiento de un jardín urbano o un cultivo. Se trata de una herramienta indispensable para saber el estado de humedad del suelo. Cuenta con dos sondas para pasar corriente eléctrica a través del suelo circundante, y luego lee la resistencia para obtener el nivel de humedad. Más agua hace que el suelo conduzca la electricidad con mayor facilidad (menos resistencia), mientras que el suelo seco es un mal conductor de la electricidad (más resistencia). Será útil para recordarle que debe regar las plantas o para monitorear la humedad del suelo en su jardín o cultivo.

Especificaciones

- Fuente de alimentación: 3,3 V o 5 V
- señal de tensión de salida: 0 ~ 4.2v
- Corriente: 35 mA
- Definición de los pines:
 - Salida analógica (cable azul)
 - GND (alambre Negro)
 - Alimentación (cable rojo)
- Tamaño: 60x20x5mm (2.36x0.79x0.2 ")
- Acabado superficial: Oro de la inmersión

Sensor Luminosidad

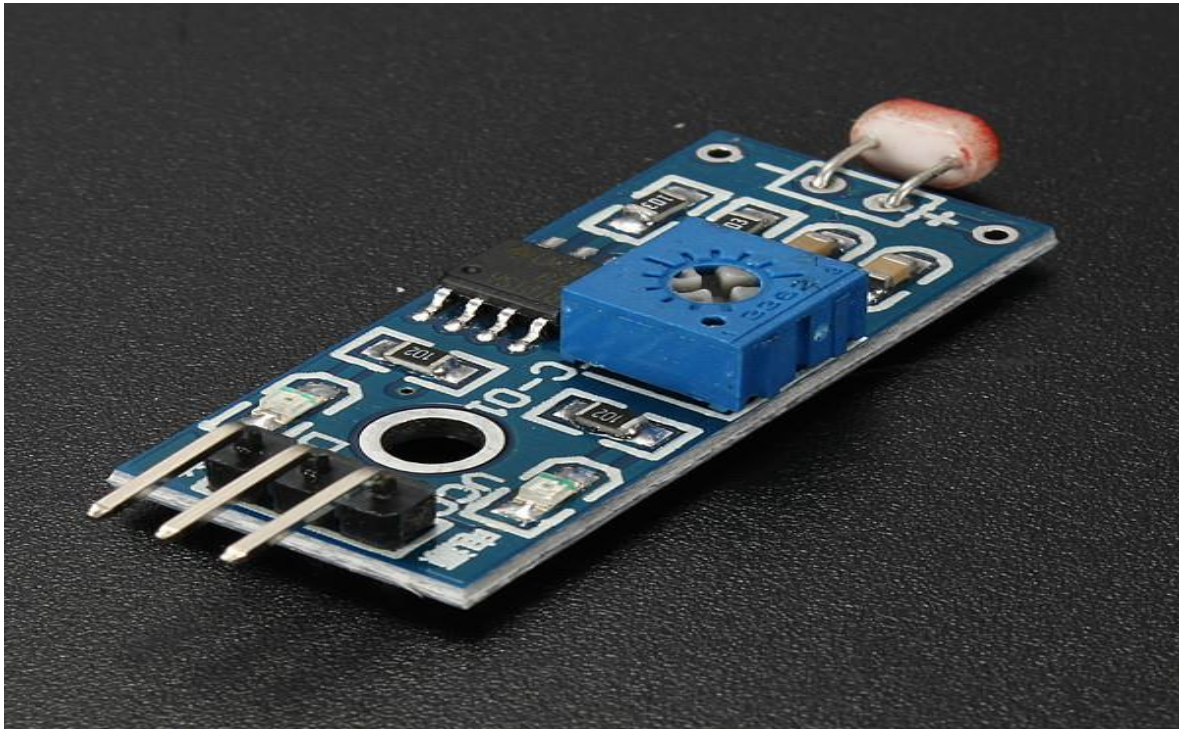


Figura 4. LDR (Sensor de luminosidad).

Un LDR es un dispositivo cuya resistencia varía de acuerdo con la cantidad de luz que reciba. Son muy útiles para proyectos de control de iluminación, seguidores solares, interruptores crepusculares entre otros.

La salida digital posee solo 2 estados: activo/apagado, el cambio de un estado a otro depende del umbral que se fije con el potenciómetro del módulo. La salida digital puede utilizarse para controlar un relay y así realizar una acción dependiente de la intensidad de luz.

Características:

- Capaz de detectar la luminosidad ambiente y la intensidad de la luz
- sensibilidad ajustable (mediante potenciómetro digital azul)
- Tensión de funcionamiento 3.3V-5V
- salidas de conmutación digitales (0 y 1) -D0
- Con el agujero de perno fijo para una fácil instalación
- Tamaño pequeño placa PCB: 3cm * 1.6cm
- Indicador de encendido (rojo) y el indicador de salida de conmutación digital (verde)
- Características de tensión amplia gama comparador LM393

5.4. Desarrollo metodológico

Para la implementación exitosa de este proyecto fue necesario llevar a cabo un conjunto de actividades las cuales se describen a continuación:

Etapas I.

Implementar el sistema de medidas

- Se seleccionaron los sensores
- Se escogió el dispositivo microcontrolado
- Se realizó una prueba registrando las variables medidas de manera local

Etapa II.

Implementar el sistema de control

- Se seleccionó el entorno de programación
- Se definieron las variables a controlar
- Se escogió el tipo de control
- Se realizaron pruebas para constatar que las variables controladas permanecen en el rango deseado

Etapa III.

Implementar la base de datos en internet

- Definir el gestor de bases de datos
- Crear las diferentes tablas
- Se realizó una prueba con el fin de verificar que la base de datos permite registrar de forma correcta

Etapa IV.

Conectar el sistema de medidas con la base de datos

- Se realizaron pruebas para corroborar que los valores sensados se registraban de forma correcta en la base de datos

Etapa V.

Diseño de una interfaz de consulta, gráficas y descargas.

- Definir una IDE o entorno de programación
- Se realizaron consultas a la base de datos seleccionando rangos de tiempo deseados y se hicieron descargas de los mismos como archivos de Microsoft office Excel
- Se graficaron los registros extraídos de la base de datos y se hicieron descargas en diferentes formatos de imagen (PNG, JPEG, PDF y vector de imagen SVG)

5.5. Circuito implementado

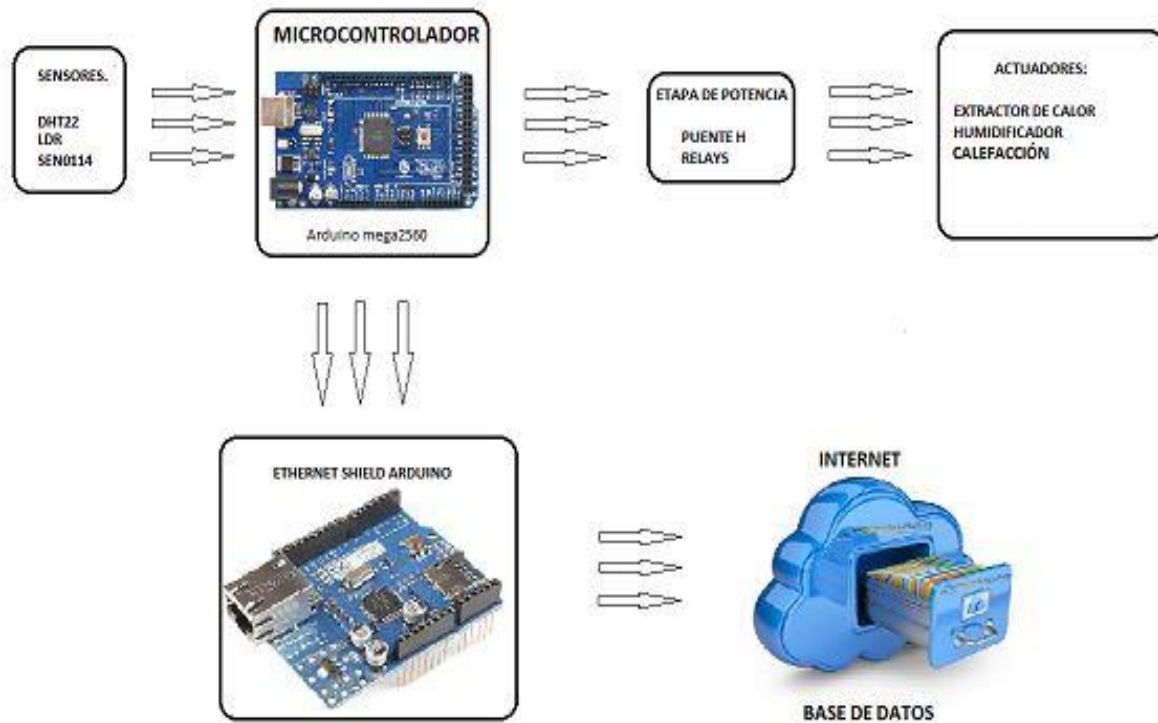


Figura 5. Circuito implementado

El circuito desarrollado está compuesto por un grupo de sensores que se encargan de medir las variables físicas al interior del invernadero, las cuales son leídas por un sistema de adquisición de datos microcontrolado, en éste caso el Arduino mega2560, el que a su vez se encarga del control de las variables temperatura y humedad ,para que permanezcan en un rango deseado, el microcontrolador también se encarga de enviar la señal de control para que la etapa de potencia suministre la corriente necesaria a los actuadores mediante modulación por ancho de pulsos o PWM. Las medidas adquiridas por el microcontrolador son enviados a un servidor web mediante el módulo de comunicación Ethernet Shield de Arduino para ser registrados en una base de datos, a la cual se puede acceder de forma fácil.

Como se puede observar en la Figura 5, el circuito implementado consta de seis etapas subtituladas las cuales se explican a continuación:

SENSORES

En esta etapa se observa la referencia de cada uno de los sensores utilizados con el fin de medir las variables físicas de interés para el proyecto. Dichos sensores son:

DHT22: Sensor de temperatura y humedad relativa

LDR: Sensor de luminosidad

SEN0114: sensor de humedad del suelo

MICROCONTROLADOR

Es la etapa encargada de leer los datos que entregan los sensores para ser enviados al servidor web (base de datos) y a su vez realizar un control del tipo on-off a la temperatura y la humedad relativa para mantenerlas dichas variables en rangos óptimos según la necesidad de cada cultivo.

ETAPA DE POTENCIA

Es la encargada de suministrar la potencia requerida para el funcionamiento de los actuadores, como se puede observar en la figura 5, se empleó un puente h para energizar el humidificador y el extractor de calor dicha acción es controlada mediante PWM desde el microcontrolador, también se utilizó un relé con el fin de activar el dispositivo de calefacción que en este proyecto es una luminaria de 7w a 110v.

ACTUADORES

Los actuadores son los dispositivos encargados de extraer calor y añadir humedad al invernadero según las necesidades del sistema.

ETHERNET SHIELD ARDUINO

Esta etapa es la encargada del enlace entre el microcontrolador y el servidor web permitiendo así poder registrar las variables sensadas en la base de datos.

INTERNET (BASE DE DATOS)

Hace referencia al servidor web en el cual se van a quedar almacenadas las medidas suministradas por los sensores, el servidor implementado en este proyecto e MySQL.

5.5.1. Medición de las variables físicas (Temperatura, humedad relativa, luminosidad y humedad del suelo) con Arduino

Para medir las variables físicas de interés, inicialmente se escogieron los sensores apropiados y el sistema microcontrolado, los cuales son descritos con anterioridad. El sistema embebido (Arduino) consta de dos partes fundamentales, la primera es el entorno de programación o IDE Arduino el cual se puede descargar de la página oficial de Arduino (www.arduino.cc/download) y sirve para compilar los códigos realizados y la segunda es el hardware en el cual se conectan los sensores y se envía la orden de activación para los actuadores entre otras.

Para realizar la Instalación del entorno de programación Arduino se recomienda seguir los pasos descritos en: [15] donde se explica de forma detallada.

5.5.2. Conexión Arduino a base de datos MySQL

Una vez sensadas las variables físicas de interés es importante establecer una conexión con el servidor para poder almacenar en este los dato capturados. Para conectar Arduino con la base de datos MySQL se realizaron los siguientes pasos:

1. Instalar el servidor independiente de plataforma XAMPP, el cual es un software libre, que consiste principalmente en la base de datos MySQL, el servidor web apache y los interpretes para lenguajes de script: PHP y Perl. Para la instalación de la plataforma XAMPP se recomienda seguir los pasos descritos en: [16], con el fin de llevar a cabo una buena configuración del programa.
2. Instalar el entorno de desarrollo NetBeans en donde se realizaron los códigos en PHP y HTML que permiten interactuar con la base de datos, la descarga e instalación se puede observar en: [16].

3. Utilizando NetBeans se implementaron los códigos que permiten la conexión Arduino servidor, registrar los datos, graficarlos y descargarlos. Los códigos se pueden observar en los anexos que van del [3] hasta el [14].
4. Por medio de Arduino y la Shield de Ethernet se realizó el registro de la información en la base de datos. Esto se logra primero asignando una IP fija al Arduino y al computador, con el fin de obtener una conexión única y estable entre ambos. Para lograr el registro de las variables sensadas se realizó un código en PHP que utiliza un método GET, para obtener el ID del sensor y su respectivo valor partir de la dirección web que es enviada desde el Arduino. El código realizado se observa en el anexo [15].

5.5.3. Interfaz gráfica



The screenshot shows a web browser window with the address bar displaying 'localhost/Conexion_arduino/graficas.php'. The main heading is 'Consulta de Sensores : Estadísticas'. Below the heading, there are several input fields and controls:

- SENSOR:** A dropdown menu with the text 'Escoja Sensor' and a downward arrow.
- FECHA INICIAL:** A date input field with the placeholder 'dd/mm/aaaa', followed by a time input field with the placeholder 'H/m/s' and the value '00:00:00' highlighted in yellow.
- FECHA FINAL:** A date input field with the placeholder 'dd/mm/aaaa', followed by a time input field with the placeholder 'H/m/s' and the value '23:59:59'.
- DESCARGAR REPORTE:** Radio buttons for 'SI' (selected) and 'NO'.
- Enviar:** A button to submit the form.

Figura 6. Interfaz gráfica.

Una vez implementados los códigos que permiten comunicación, leer, graficar y descargar los valores registrados en la base de datos se procedió a realizar una interfaz gráfica de usuario donde se permite escoger el tipo de sensor, la fecha inicial y final de registro, la opción de descarga como

archivo de Microsoft Excel y finalmente la opción de graficar en línea los datos extraídos del servidor.

En la figura 6, el usuario al hacer clic en la ventana (Escoja sensor) se despliega un submenú en cual puede escoger entre los diferentes sensores utilizados; al hacer clic sobre la ventana (dd/mm/aaaa) se despliega un submenú en forma de almanaque que permite poner de manera fácil la fecha inicial y final de consulta; al hacer clic sobre la ventana (H/m/s) el usuario puede asignar la hora inicial y final de la consulta a realizar; en la opción de (DESCARGAR REPORTE) se brinda la opción de hacerlo o no y al escoger la descarga se presenta una ventana en la cual se puede escoger el sitio de descarga en el pc en formato Microsoft Excel ; al hacer clic sobre (Enviar) nos aparece en pantalla la gráfica de las medidas del sensor seleccionado en el rango de tiempo escogido.El código implementado para tal fin se puede observar en el anexo [14]

5.6. Diagramas sistema de control y registro

5.6.1. Diagrama sistema de control

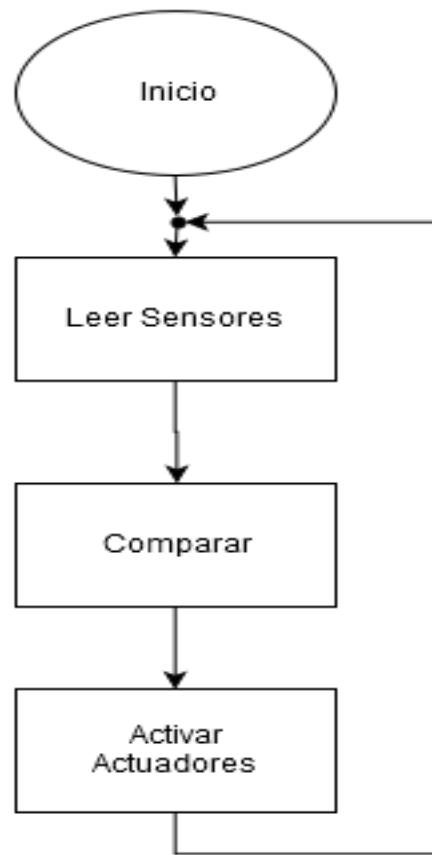


Figura 7. Diagrama sistema de control

En la figura 7, se puede observar un diagrama del control implementado en el proyecto el cual se realiza en cuatro partes así:

Inicio: es la etapa donde definimos archivos de cabecera y variables.

Leer Sensores: es la fase donde se capturan las medidas hechas por los sensores

Comparar: en esta etapa se toma la señal suministrada por el sensor de temperatura y se compara con una de referencia con el fin de poder tomar una decisión de control adecuada. Lo anterior se hace extensivo para la humedad relativa.

Activar Actuadores: en esta fase es donde se permite la activación o desactivación de los dispositivos encargados de la modificación del clima al interior del vivero.

5.6.2. Diagrama sistema de captura y registro

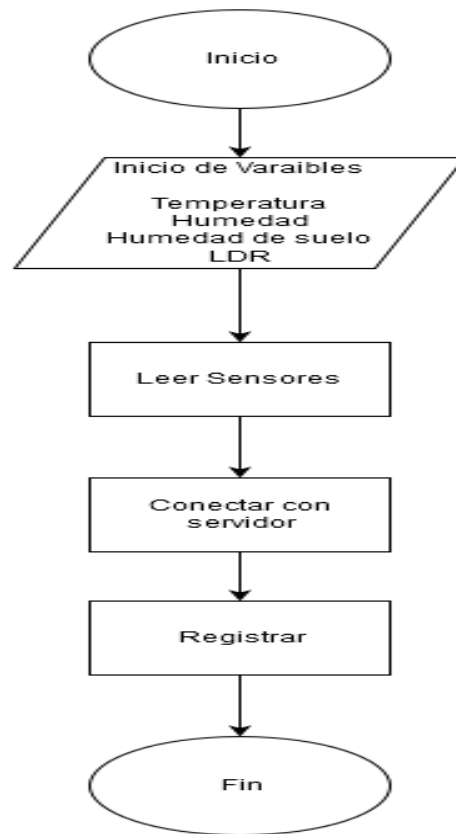


Figura 8. Diagrama sistema de captura y registro

En la figura 8, se puede observar el diagrama de captura u registros el cual se realiza en cinco partes descritas a continuación:

Inicio: es la etapa donde definimos archivos de cabecera y variables a utilizar

Inicio de variables: es la etapa donde definimos las variables a utilizar durante el control.

Leer Sensores: en esta etapa el microcontrolador captura las medidas realizadas por el conjunto de sensores utilizados

Conectar con el servidor: en esta fase es donde la Ethernet Shield permite enviar las medidas capturadas por el sistema microcontrolado y se encarga de enviarlas al servidor web.

Registrar: en esta etapa los datos enviados vía ethernet son almacenados en una base de datos previamente elaborada en MySQL.

5.7. Sistema desarrollado

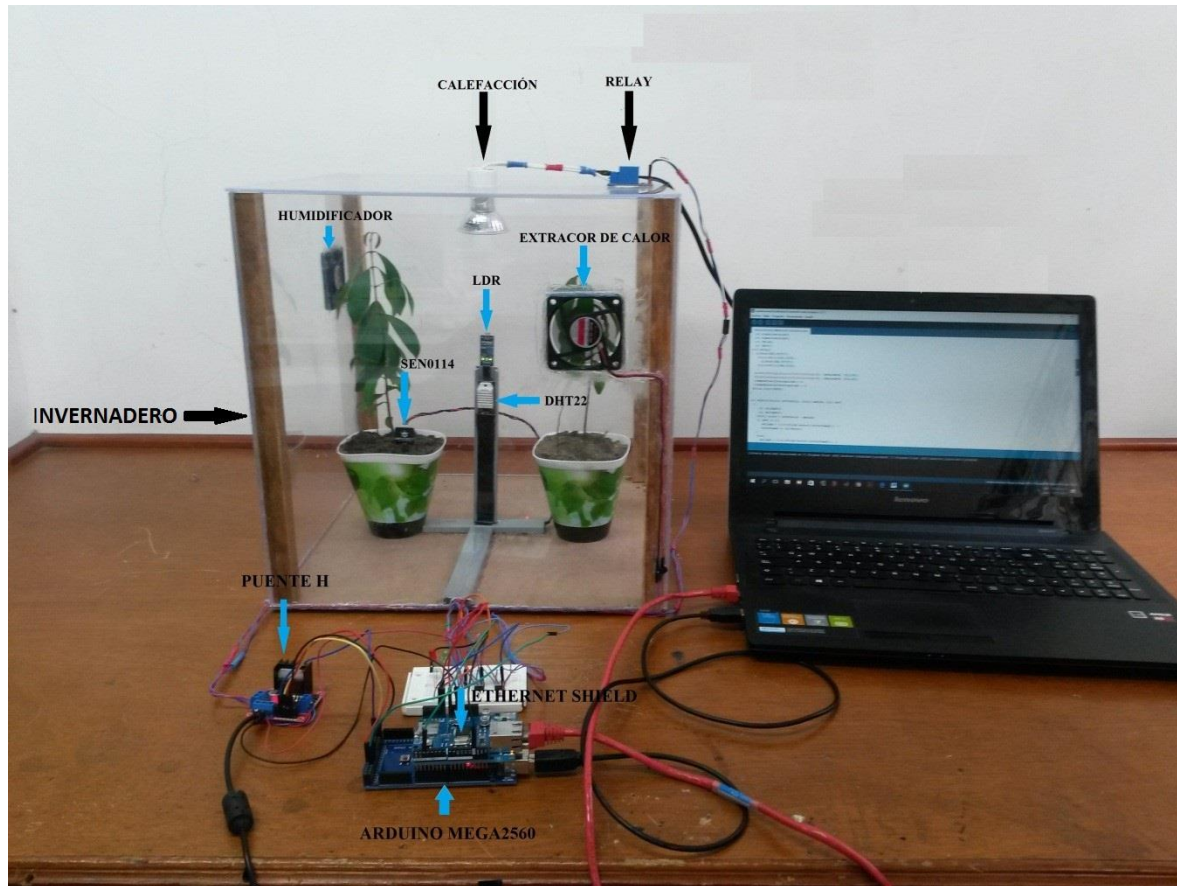


Figura 9. Sistema implementado

En la figura 9 se puede observar el sistema implementado, en este se aprecia el prototipo de un invernadero, el cual cuenta en su interior dos plantas de la familia de las mirtáceas más conocidas como *Eugenia uniflora* o pomelo australiano a las cuales se les está brindando unas condiciones climáticas controladas para su buen desarrollo, adicional se puede apreciar el sistema de adquisición y los sensores (DHT22, SEN0114 y LDR) ubicados de manera estratégica dentro del prototipo con el fin de obtener la correcta medida de las variables sensadas. También cabe destacar que el Arduino se encuentra conectado al computador por medio de un cable ethernet con el fin de registrar los datos captados por las diferentes fuentes sensoriales en una base de datos en internet.

También se puede observar los actuadores (Extractor de calor, humidificador y calefacción) con la finalidad cambiar se forma controlada las condiciones del ambiente al interior del invernadero.

6. RESULTADOS

Del desarrollo descrito en el capítulo anterior se obtuvieron los siguientes resultados:

Datos sensados en el microcontrolador Arduino mega 2560 obtenidos mediante el puerto serial.

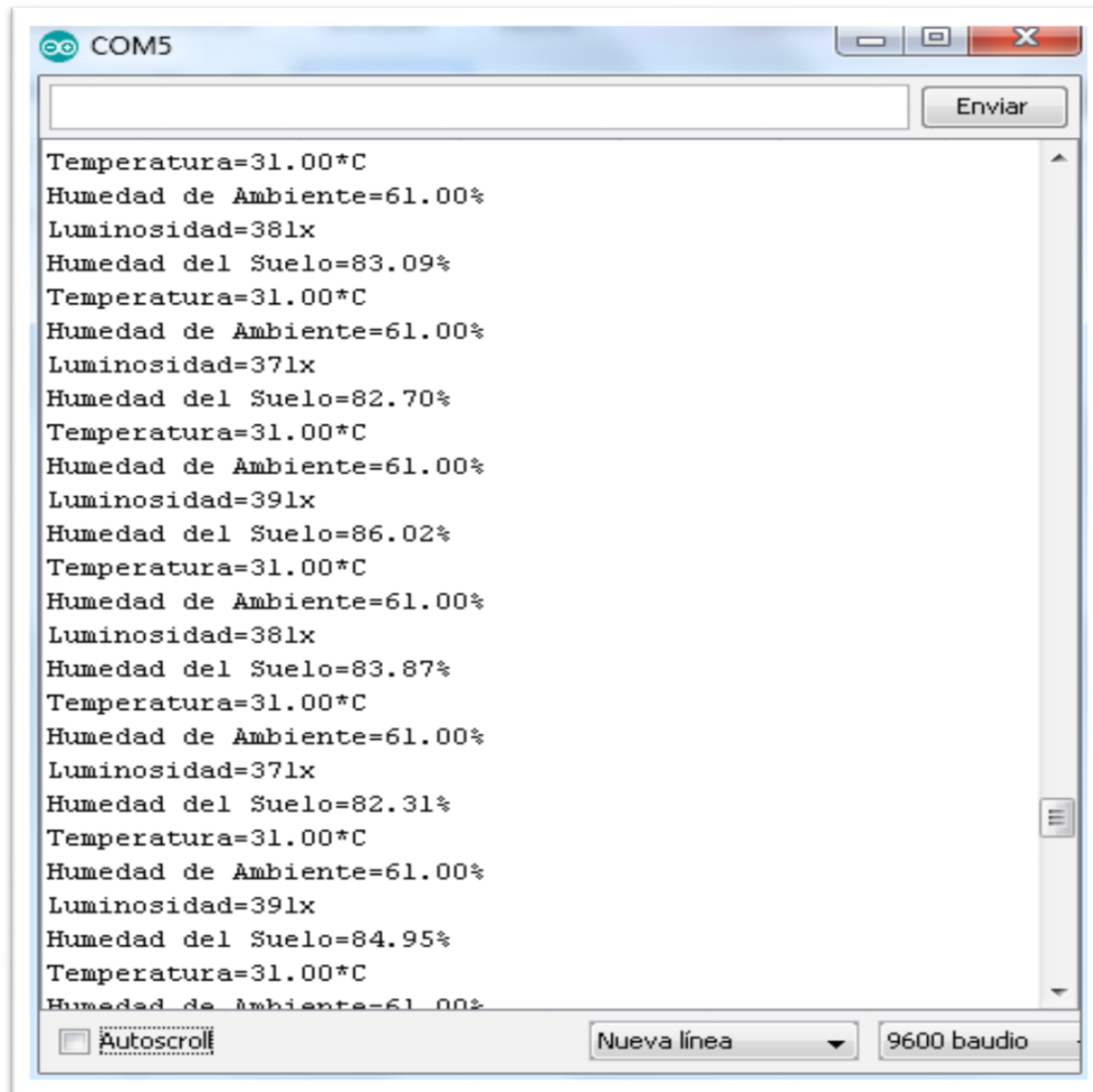
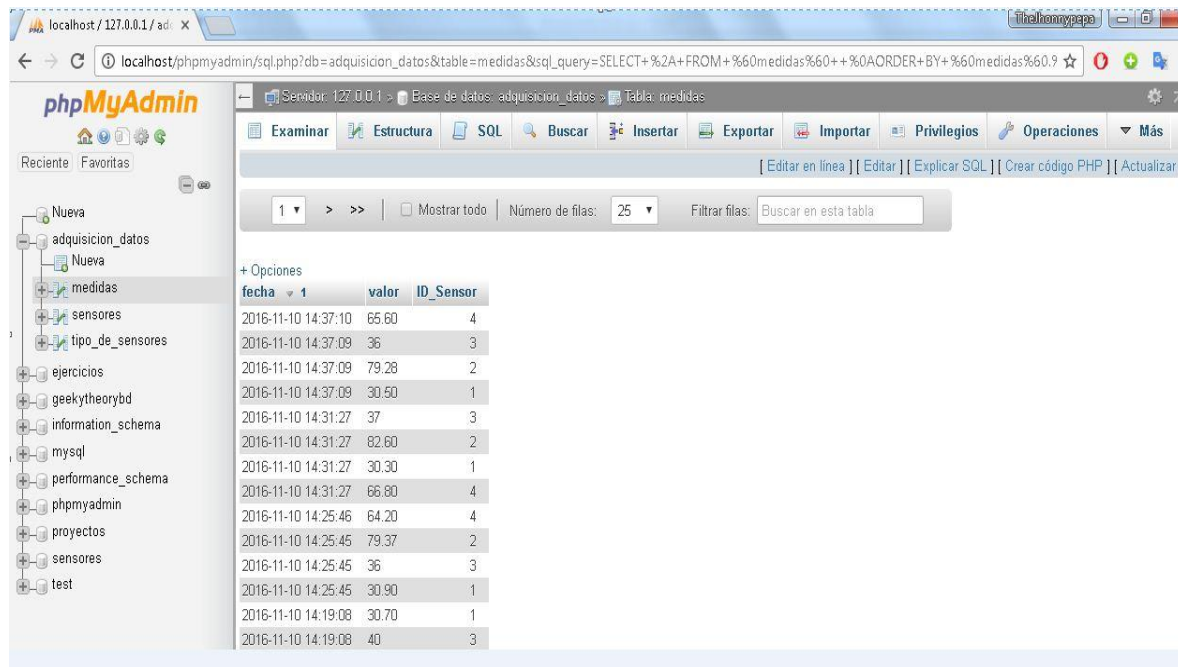


Figura 10. Visualización del COM Arduino.

En la figura 10 se puede observar los datos sensados de las diferentes variables mediante el puerto serial del microcontrolador Arduino

Base de datos creada en MySQL



The screenshot shows the phpMyAdmin interface for a MySQL database named 'adquisicion_datos'. The table 'medidas' is selected, and its data is displayed in a table format. The columns are 'fecha', 'valor', and 'ID_Sensor'. The data is sorted by 'fecha' in descending order, showing the most recent measurements first.

fecha	valor	ID_Sensor
2016-11-10 14:37:10	65.60	4
2016-11-10 14:37:09	36	3
2016-11-10 14:37:09	79.28	2
2016-11-10 14:37:09	30.50	1
2016-11-10 14:31:27	37	3
2016-11-10 14:31:27	82.60	2
2016-11-10 14:31:27	30.30	1
2016-11-10 14:31:27	66.80	4
2016-11-10 14:25:46	64.20	4
2016-11-10 14:25:45	79.37	2
2016-11-10 14:25:45	36	3
2016-11-10 14:25:45	30.90	1
2016-11-10 14:19:08	30.70	1
2016-11-10 14:19:08	40	3

Figura 11. Visualización de la base de datos en MySQL.

En la figura 11 Se puede observar que los datos sensados se registran de manera oportuna en la base de datos en internet, donde además se puede notar que el registro se realiza por orden cronológico (fecha y hora), el valor de la variable sensada y un id para identificar qué medida corresponde a cada sensor.

Interfaz grafica



← → ↻ ⓘ localhost/Conexion_arduino/graficas.php

Consulta de Sensores : Estadísticas

SENSOR:

FECHA INICIAL: H/m/s

FECHA FINAL: H/m/s

DESCARGAR REPORTE: SI: NO:

Figura 12. Visualización interfaz gráfica.

En la figura 12. Se puede observar la interfaz gráfica del proyecto, en la cual el usuario puede escoger el tipo de sensor, la fecha inicial y final de la medida almacenada en la base de datos, graficar los valores y la opción de descargar los reportes.

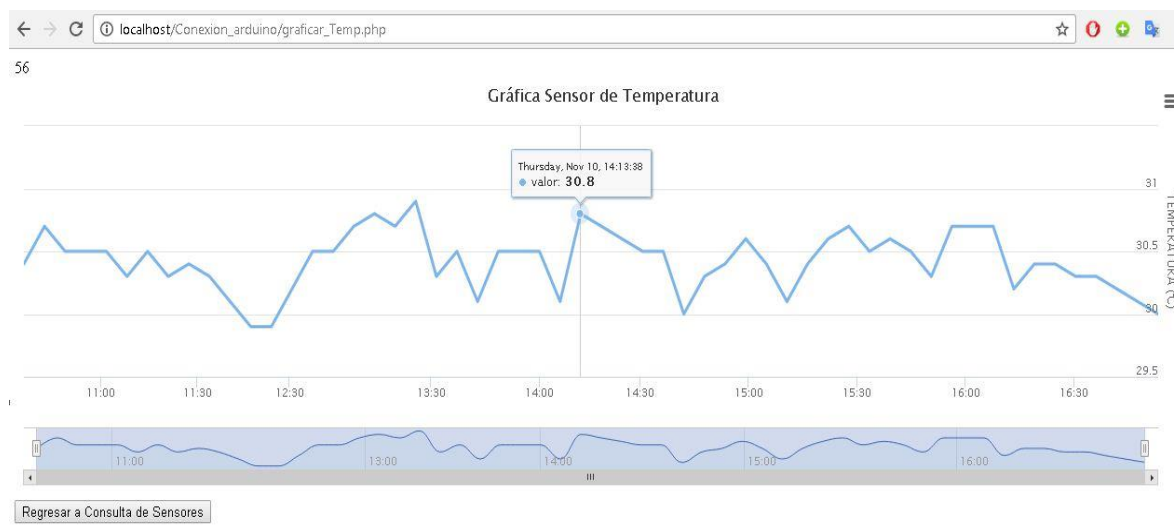


Figura 13. Visualización grafica Temperatura Ambiente.

En la figura 13. Se puede observar la gráfica de Temperatura de los valores extraídos por el usuario de la base de datos en internet, en rangos de tiempo predefinidos. Con el agregado que permite leer en la gráfica de forma clara la fecha, hora y valor de la medición además de que se puede realizar una descarga en diferentes formatos (PNG, JPEG, como archivo PDF, vector de imagen SVG).

Sumado a lo anterior se puede notar que el valor de temperatura permanece en el rango de control preestablecido

Lo anterior se repite para las figuras 14, 15 y 16 las cuales se muestran a continuación.

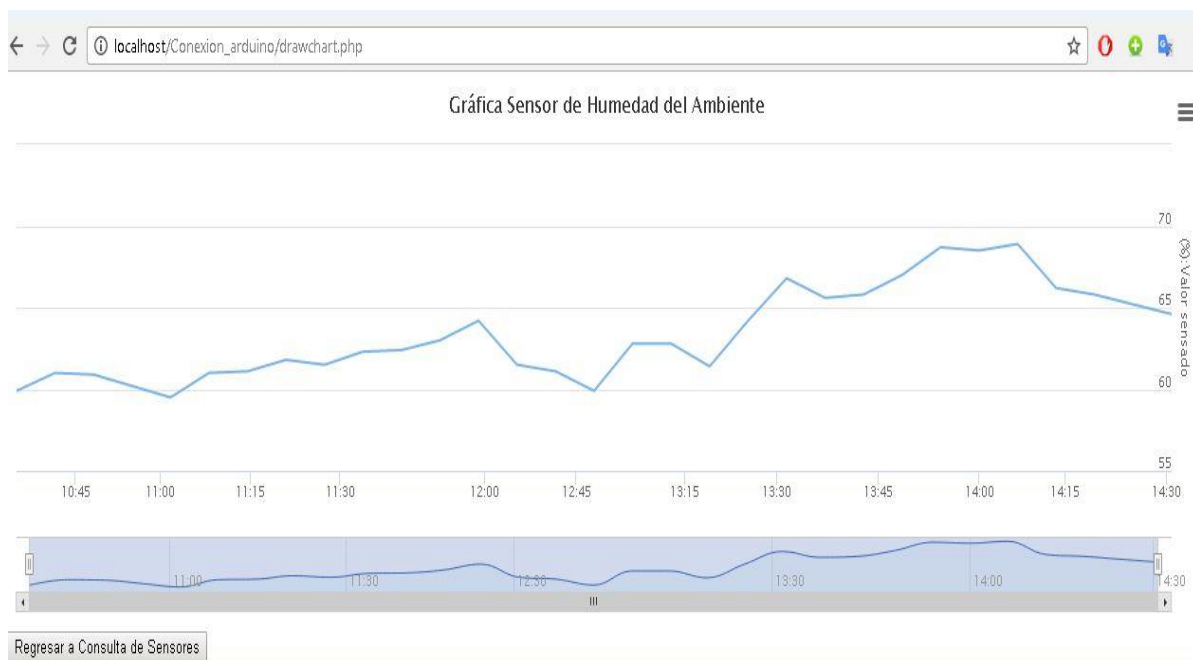


Figura 14. Visualización grafica Humedad del Ambiente.

En la figura 14. Se puede observar que la humedad del ambiente es controlada y permanece por encima del valor preestablecido, en nuestro caso del 60%.

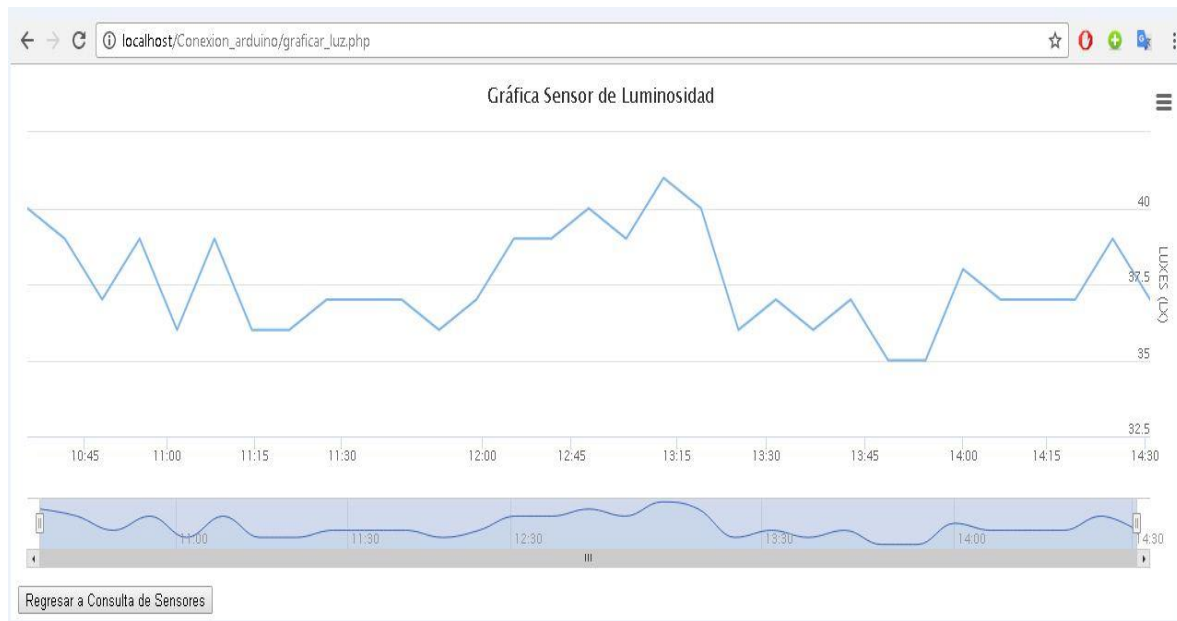


Figura 15. Visualización grafica Luminosidad.

En la figura 15. Se puede observar que la luminosidad permanece en valores normales durante el transcurso del día

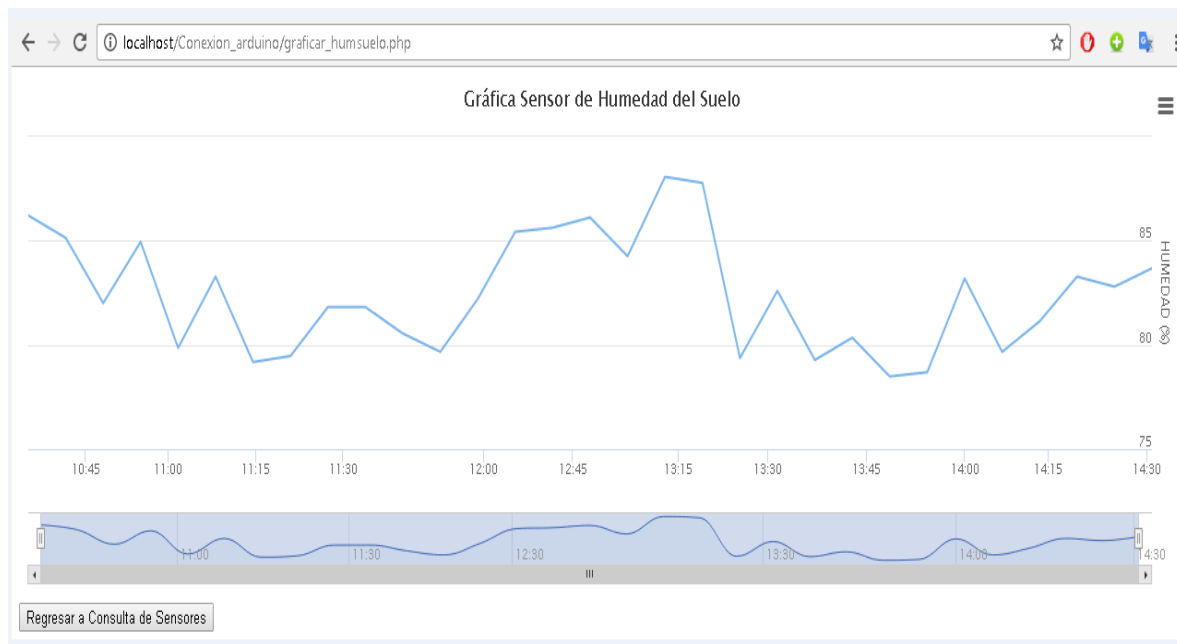


Figura 16. Visualización grafica Humedad del Suelo.

En la figura 16. Se puede observar que la humedad del suelo no sufre cambios significativos

Control de temperatura

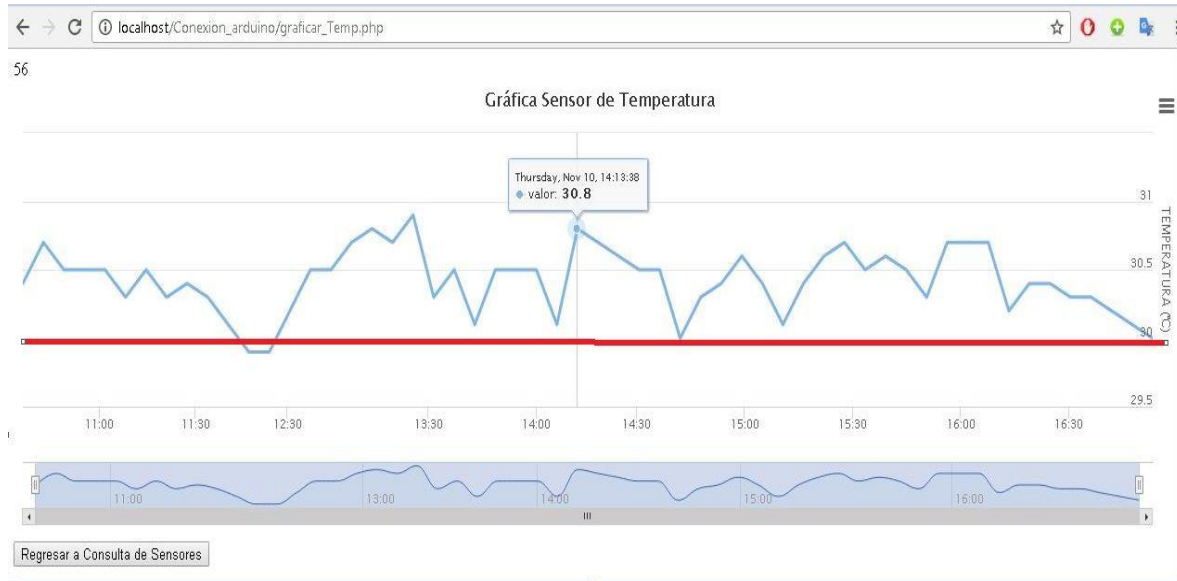


Figura 17. Visualización grafica control de temperatura

En la figura 17, se puede notar que la temperatura permanece en el margen preestablecido el cual fue de 30 (°C) con una histéresis de (0.1°C) aproximadamente

Control de humedad relativa del ambiente

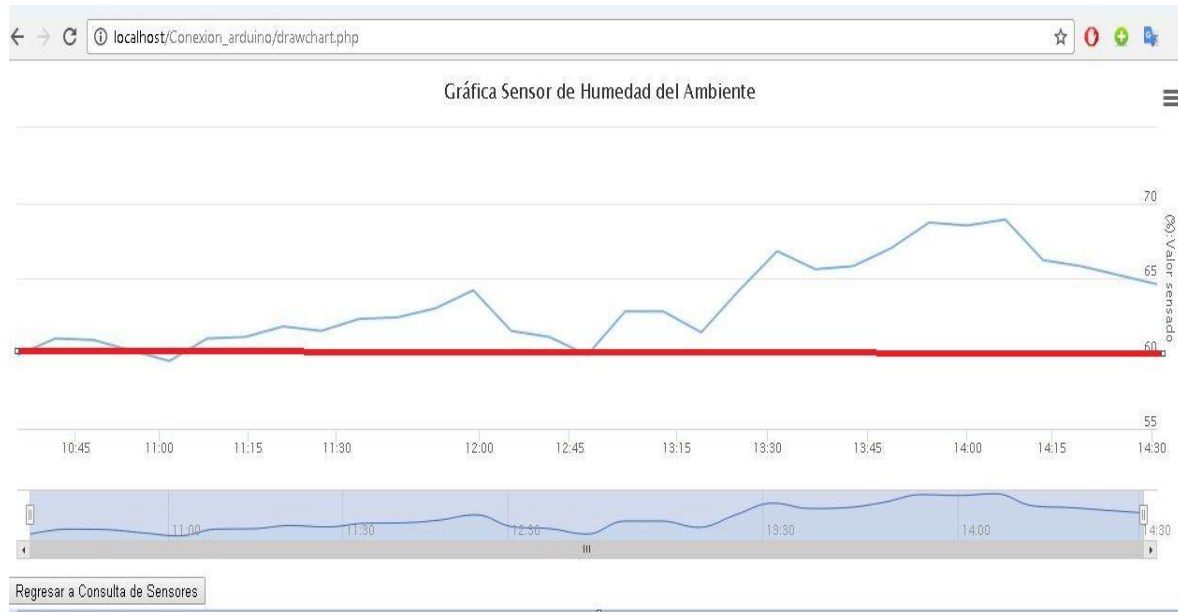


Figura 18. Visualización grafica control de humedad relativa del ambiente

En la figura 18, se puede notar que la humedad permanece en el rango preestablecido que en este trabajo fue de un valor por encima de 60% con un error de (+/-2%)

Verificación del rango de incertidumbre y el error del sensor DHT22

De los datos obtenido en el Anexo 1, se puede notar que el sensor empleado para medir temperatura se encuentra en los rangos estipulados en la figura 17 Anexo 2. De lo cual se concluye que no hay necesidad de calibración ya que el error de las medidas permanece en un rango aceptable.

7. CONCLUSIONES, APORTES Y RECOMENDACIONES

7.1. Conclusiones

- Se desarrolló un sistema de bajo costo y fácil implementación que permite el monitoreo, registro y control de variables físicas (Temperatura, humedad del ambiente, luminosidad y humedad del suelo) de un cultivo de invernadero.
- Del proyecto realizado se puede concluir que es una herramienta funcional y versátil para saber de forma rápida y precisa el comportamiento de las variables físicas de un invernadero
- El buen funcionamiento de la estrategia de control implementada permite mantener en niveles óptimos los valores de temperatura y humedad relativa dentro del invernadero.
- El buen funcionamiento de la estrategia de comunicación entre el sistema microcontrolado y el servidor permite ver el correcto registro de los valores sensados en la base de datos creada en MySQL
- Debido al modo en que se desarrolló el proyecto se puede agregar de manera fácil diferentes sensores y actuadores

7.2. Aportes

El aporte más significativo del proyecto es que permite conectar un dispositivo microcontrolador de bajo costo (Arduino Mega2560) con un servidor web para el almacenamiento de medidas en una base de datos en MySQL además de permitir la consulta y manipulación estadística de dichas variables. Además de ser un producto final que permitirá al grupo de investigación de ingeniería electrónica desarrollar trabajos futuros con el mismo.

7.3. Recomendaciones

- Reemplazar la fuente de suministro de energía por sistemas alternativos, tales como paneles solares o un conjunto de baterías.
- Reemplazar la conexión a internet por un sistema inalámbrico.
- Agregar más variables de medición (Ph, emisiones de CO₂, entre otros).
- Anexar un control para el suministro de agua con el fin de mejorar la humedad del suelo y poder realizar un suministro óptimo de nutrientes a las plantas.
- Agregar un sistema de captura de imagen para visualizar el estado de las plantas.

- Por medio del análisis de la información recopilada permitir que en trabajos futuros se puedan realizar detección de enfermedades, fumigación automática, entre otras.

8. BIBLIOGRAFÍA

[1] GLORIA ALEXANDRA HERNÁNDEZ GONZÁLEZ y Cesar Leonardo Romero Sánchez. “Análisis de la situación de los invernaderos de la Asociación de Tomateros en Garagoa (Boyacá)”. Tesis. Universidad de la Sabana. Bogotá D.C. Colombia. 2002.

[2] HERNÁN OCTAVIO DÍAZ SARMIENTO y Oscar Fabián Solano Rojas. Diseño y simulación del control climático para un invernadero y base de datos de registro. Tesis. Facultad de Ingeniería. Universidad de la Salle. Bogotá D.C. Colombia. 2006.

[3] JESÚS NÚÑEZ, Julio César Carvajal y Luis Armando Bautista. El TLC con Estados Unidos y su impacto en el sector agropecuario Colombiano: Entre esperanzas e incertidumbres. En: LEX UDES, 2004.

[4] PAULA RAMOS GIRALDO, Mauricio García Navarro, Juan Hoyos Suárez Carlos OliverosTascón y URIBE, Juan Sanz. Aplicación de una Metodología Estructurada para el Diseño de un Sistema de Cosecha Selectiva de Café. En: Scientia et Technica, tomo 20, no 1, 2015, págs. 10–19. ISSN 0122-1701.

[5] GOBERNACIÓN DE RISARALDA. Plan Frutícola Nacional, Desarrollo de la Fruticultura en Risaralda. Ministerio de Agricultura y Desarrollo Rural, 2006. 17

[6] ZHANG, Shanwen y ZHANG, Chuanlei. Orthogonal Locally Discriminant Projection for Classification of Plant Leaf Diseases. En: Computational Intelligence and Security (CIS), 2013 9th International Conference on, 2013, págs. 241–245.

[7] ZHOU, Rong, et al. Early Detection and Continuous Quantization of Plant Disease Using Template Matching and Support Vector Machine Algorithms. En: Computing and Networking (CANDAR), 2013 First International Symposium on, 2013, págs. 300–304.

[8] VIJAY, N. Application of sensor networks in agriculture. En: Sustainable Energy and Intelligent Systems (SEISCON 2012), IET Chennai 3rd International on, 2012, págs. 1–6.

[9] BHARGAVA, K.; KASHYAP, A. y GONSALVES, T.A. Wireless sensor network based advisory system for Apple Scab prevention. En: Communications (NCC), 2014 Twentieth National Conference on, 2014, págs. 1–6.

[10] SONE, Hironao. AI based agriculture support system with precisely deployed versatile sensors and sensor network. En: Advanced Intelligence and Awareness Internet (AIAI 2010), 2010 International Conference on, 2010, págs. 4–4.

[11] PÉREZ, Miguel A. Big Data para la agricultura, 2015.

[12] KLETTE, Reinhard. Concise Computer Vision: An Introduction into Theory and Algorithms. Springer Publishing Company, Incorporated, 2014. ISBN 1447163192, 9781447163190.

[13] Barroso García, Andrés. ” Control y monitorización de un invernadero a través de una aplicación móvil”. Máster en Ingeniería Electromecánica, Universidad Politécnica de Madrid, Madrid, España, Junio de 2015.

[14] Pinto Ríos. Wilson Daniel. “Monitoreo de Cultivos con Redes de Sensores XBEE, Arduino, Y Dispositivos de Medición de Suelos”. Tesis, Universidad Tecnológica de Pereira, Pereira, Risaralda, Colombia, 2015.

[15] Instalación del software Arduino, [Online]. Available: <http://www.cromalight.com/es/instalacion-del-software-arduino-en-el-ordenador->

[16] Manuel Dávila (2014), Instalación XAMPP y NetBeans, [Online]. Available: <https://www.youtube.com/watch?v=OU3nv6UI40E>

Anexos

Anexo 1.

	Datos Sensor DHT22 [°C]	Datos Termocupla (Sensor patron) [°C]	Error	Error RMS
	28,1	28,2	-0,1	0,1
	28,1	28,1	0	0
	28,1	27,9	0,2	0,2
	28,1	27,9	0,2	0,2
	28,1	28,3	-0,2	0,2
	28,1	28,4	-0,3	0,3
	28,2	28,3	-0,1	0,1
	28,2	28,2	0	0
	28,2	28,2	0	0
	28,2	28,5	-0,3	0,3
	28,2	28,6	-0,4	0,4
	28,2	28,4	-0,2	0,2
	28,2	28,4	-0,2	0,2
	28,2	28,4	-0,2	0,2
	28,2	28,2	0	0
	28,2	28,1	0,1	0,1
	28,2	28	0,2	0,2
	28,2	28	0,2	0,2
	28,2	28	0,2	0,2
	28,2	28	0,2	0,2
	28,2	28,1	0,1	0,1
	28,2	28,1	0,1	0,1
	28,2	28,1	0,1	0,1
	28,2	28,1	0,1	0,1

	28,2	28,3	-0,1	0,1
	28,2	28,3	-0,1	0,1
	28,2	28,3	-0,1	0,1
	28,2	28,3	-0,1	0,1
	28,2	28,2	0	0
	28,3	28,2	0,1	0,1
	28,2	28,4	-0,2	0,2
	28,2	28,4	-0,2	0,2
	28,2	28,1	0,1	0,1
	28,3	28,1	0,2	0,2
	28,2	28	0,2	0,2
	28,2	27,9	0,3	0,3
	28,2	27,9	0,3	0,3
	28,2	27,9	0,3	0,3
	28,2	28,1	0,1	0,1
	28,2	28,1	0,1	0,1
	28,2	28,2	0	0
	28,2	28,2	0	0
	28,2	28,3	-0,1	0,1
	28,2	28,3	-0,1	0,1
	28,2	28,4	-0,2	0,2
	28,2	28,4	-0,2	0,2
	28,2	28,3	-0,1	0,1
	28,2	28,2	0	0
	28,1	28	0,1	0,1
	28,1	28	0,1	0,1
	28,1	28	0,1	0,1
	28,1	27,8	0,3	0,3
	28,1	27,8	0,3	0,3

	28	27,9	0,1	0,1
	28	27,9	0,1	0,1
	28	28,2	-0,2	0,2
	28	28,2	-0,2	0,2
	27,9	28	-0,1	0,1
	27,9	28,1	-0,2	0,2
	27,9	28,1	-0,2	0,2
	27,9	28,2	-0,3	0,3
	27,9	28,2	-0,3	0,3
	27,9	28,1	-0,2	0,2
	27,8	28,1	-0,3	0,3
	27,8	28	-0,2	0,2
	27,8	28	-0,2	0,2
	27,8	27,9	-0,1	0,1
	27,8	28	-0,2	0,2
	27,7	28	-0,3	0,3
	27,7	29,9	-2,2	2,2
	27,7	27,9	-0,2	0,2
	27,7	28,1	-0,4	0,4
	27,7	28,1	-0,4	0,4
	27,7	28	-0,3	0,3
	27,7	27,9	-0,2	0,2
	27,7	27,7	0	0
	27,7	27,7	0	0
	27,6	27,8	-0,2	0,2
	27,6	27,8	-0,2	0,2
	27,6	27,9	-0,3	0,3
	27,6	27,7	-0,1	0,1
	27,6	27,5	0,1	0,1

	27,6	27,5	0,1	0,1
	27,5	27,5	0	0
	27,5	27,7	-0,2	0,2
	27,5	27,7	-0,2	0,2
	27,5	27,6	-0,1	0,1
	27,5	27,4	0,1	0,1
	27,5	27,4	0,1	0,1
	27,5	27,2	0,3	0,3
	27,5	27,2	0,3	0,3
	27,4	27	0,4	0,4
	27,4	27,1	0,3	0,3
	27,4	27,1	0,3	0,3
	27,4	27,2	0,2	0,2
	27,4	27	0,4	0,4
	27,4	26,9	0,5	0,5
	27,4	27,1	0,3	0,3
	27,4	27,1	0,3	0,3
	27,4	27	0,4	0,4
	27,4	27,1	0,3	0,3
	27,3	27	0,3	0,3
Media	28,1	28	0	0,2
Varianza	0,093353718	0,200337798	0,0934469	0,0523675
Desviacion estandar	0,305538406	0,447591106	0,30569086	0,22883947

Tabla 1. Verificación sensor DHT22

Anexo 2.

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

Pin sequence number: 1 2 3 4 (from left to right direction).

Pin	Function
1	VDD---power supply
2	DATA--signal
3	NULL
4	GND

Figura 16. Especificaciones técnicas sensor DHT22

Anexo 3.

Conexión con la base de datos

```
<?php
```

```
/*
```

```
* To change this license header, choose License Headers in Project Properties.
```

```
* To change this template file, choose Tools | Templates
```

```
* and open the template in the editor.
```

```
*/
```

```
class Conexion{
```

```

public function conectar() {
    $usuario='root';
    $password='';
    $host='127.0.0.1';
    $db='adquisicion_datos';
    return $conexion= new PDO("mysql:host=$host;
dbname=$db", $usuario, $password);
}

}

```

Anexo 4.

Leer de la base de datos

```
<?php
```

```

/*
Crear, Leer, Actualizar, Eliminar // crud
*/

```

```

class Crud{
    public $insertINTO;
    public $insertColumns;
    public $insertValues;
    public $mensaje;
    public $select;
    public $from;
    public $condition;
    public $rows;
    public $update;
    public $set;
    public $deleteFrom;

    public function Create() {
        $model=new Conexion();
        $conexion=$model->conectar();
        $insertInto = $this->insertINTO;
        $insertColumns = $this->insertColumns;
        $insertValues = $this->insertValues;
        $sql="INSERT INTO $insertInto($insertColumns)
VALUES($insertValues)";
        $consulta=$conexion->prepare($sql);
        if(!$consulta){
            $this->mensaje="Error al crear el registro";
        }
        else{
            $consulta->execute();
            $this->mensaje="Registro creado correctamente";
        }
    }
}

```

```

    }
}

public function Read() {
    $model=new Conexion();
    $conexion=$model->conectar();
    $select= $this->select;
    $from= $this->from;
    $condition= $this->condition;
    if ($condition != '' ){
        $condition= " WHERE " . $condition;
    }
    $sql= "SELECT $select FROM $from $condition";
    //echo $sql."<br>";
    $consulta=$conexion->prepare($sql);
    $consulta->execute();

    while($filas= $consulta->fetch())
    {
        $this->rows[]=$filas;
    }
}

public function Update() {
    $model=new Conexion();
    $conexion=$model->conectar();
    $update=$this->update;
    $set=$this->set;
    $condition= $this->condition;
    if ($condition != '' ){
        $condition= " WHERE " . $condition;
    }
    $sql= "UPDATE $update SET $set $condition";
    $consulta=$conexion->prepare($sql);
    if(!$consulta){
        $this->mensaje="Error al actualizar el
registro";
    }
    else{
        $consulta->execute();
        $this->mensaje="Registro guardado con exito";
    }
}

public function Delete() {
    $model=new Conexion();
    $conexion=$model->conectar();
    $deleteFrom=$this->deleteFrom;

```

```

        $condition= $this->condition;
        if ($condition != ' '){
            $condition= " WHERE " . $condition;
        }
        $sql= "DELETE FROM $deleteFrom $condition";
        $consulta=$conexion->prepare($sql);
        if(!$consulta){
            $this->mensaje="Error al eliminar el registro";
        }
        else{
            $consulta->execute();
            $this->mensaje="Registro eliminado correctamente";
        }
    }
}

```

Anexo 5.

Graficar temperatura

```

<?php
session_start();
require "crud/Conexion.php";
require "crud/Crud.php";
require "RandomClass.php";

$fechai=$_SESSION['fechai'];
$horai=$_SESSION['horai'];
$fechaf=$_SESSION['fechaf'];
$horaf=$_SESSION['horaf'];

$datei=$fechai." ".$horai;
$datef=$fechaf." ".$horaf;

$model = new Crud();
$model->select = '*';
$model->from = 'medidas';
$model->condition = "ID_Sensor=1 AND fecha>='$datei' AND
fecha<='$datef'";
$model->Read();
$filas = $model->rows;
$total = count($filas);
echo $total;
//nos creamos dos arrays para almacenar el tiempo y el valor numérico
$valoresy;
$valoresx;
//en un bucle for obtenemos en cada iteración el valor numérico y
//el TIMESTAMP del tiempo y lo almacenamos en los arrays
for($i = 0 ;$i<count($filas);$i++){

```

```

    $valoresy[$i]= $filas[$i][1];
    //OBTENEMOS EL TIMESTAMP
    $valoresx= $filas[$i][0];
    $date = new DateTime($valoresx);
    //ALMACENAMOS EL TIMESTAMP EN EL ARRAY
    $timeArray[$i] = $date->getTimestamp()*1000;
    //echo $timeArray[$i];
}

?>

<HTML>
<BODY>

<meta charset="utf-8">

<div id="container"></div>

<script src="https://code.jquery.com/jquery.js"></script>
    <!-- Importo el archivo Javascript de Highcharts directamente desde
su servidor -->
<script src="http://code.highcharts.com/stock/highstock.js"></script>
<script src="http://code.highcharts.com/modules/exporting.js"></script>
<script>

chartCPU = new Highcharts.StockChart({
    chart: {
        renderTo: 'container'
        //defaultSeriesType: 'spline'
    },
    rangeSelector : {
        enabled: false
    },
    title: {
        text: 'GrÃ¡fica Sensor de Temperatura'
    },
    xAxis: {
        type: 'datetime'
        //tickPixelInterval: 150,
        //maxZoom: 20 * 1000
    },
    yAxis: {
        minPadding: 0.2,
        maxPadding: 0.2,
        title: {
            text: 'TEMPERATURA (Â°C)',
            margin: 10
        }
    }
});

```

```

    }
  },
  series: [{
    name: 'valor',
    data: (function() {
      var data = [];
      <?php
        for($i = 0 ;$i<count($filas);$i++){
          ?>
          data.push([<?php echo $timeArray[$i];?>,<?php echo
$valoresy[$i];?>]);
          <?php } ?>
          return data;
        }) ()
    }],
    credits: {
      enabled: false
    }
  }
});

</script>
<form method="get"
action="http://localhost/Conexion_arduino/graficas.php"><button
type="submit">Regresar a Consulta de Sensores</button></form>

</BODY>

</html>

```

Anexo 6.

Grafica de la humedad relativa

```

<?php
session_start();
require "crud/Conexion.php";
require "crud/Crud.php";
require "RandomClass.php";

$fechai=$_SESSION['fechai'];
$horai=$_SESSION['horai'];
$fechaf=$_SESSION['fechaf'];
$horaf=$_SESSION['horaf'];

$datei=$fechai." ".$horai;
$datef=$fechaf." ".$horaf;

$model = new Crud();
$model->select = '*';

```

```

$model->from = 'medidas';
$model->condition = "ID_Sensor=4 AND fecha>='$datei' AND
fecha<='$datef'";
$model->Read();
$filas = $model->rows;
$total = count($filas);
echo $total;
//nos creamos dos arrays para almacenar el tiempo y el valor numérico
$valoresy;
$valoresx;
//en un bucle for obtenemos en cada iteración el valor numérico y
//el TIMESTAMP del tiempo y lo almacenamos en los arrays
for($i = 0 ;$i<count($filas);$i++){
    $valoresy[$i]= $filas[$i][1];
    //OBTENEMOS EL TIMESTAMP
    $valoresx= $filas[$i][0];
    $date = new DateTime($valoresx);
    //ALMACENAMOS EL TIMESTAMP EN EL ARRAY
    $timeArray[$i] = $date->getTimestamp()*1000;
    //echo $timeArray[$i];
}

?>

<HTML>
<BODY>

<meta charset="utf-8">

<div id="container"></div>

<script src="https://code.jquery.com/jquery.js"></script>
    <!-- Importo el archivo Javascript de Highcharts directamente desde
su servidor -->
<script src="http://code.highcharts.com/stock/highstock.js"></script>
<script src="http://code.highcharts.com/modules/exporting.js"></script>
<script>

chartCPU = new Highcharts.StockChart({
    chart: {
        renderTo: 'container'
        //defaultSeriesType: 'spline'

    },
    rangeSelector : {
        enabled: false
    },
    title: {

```



```

        text: 'Gráfica Sensor de Humedad del Ambiente'
    },
    xAxis: {
        type: 'datetime'
        //tickPixelInterval: 150,
        //maxZoom: 20 * 1000
    },
    yAxis: {
        minPadding: 0.2,
        maxPadding: 0.2,
        title: {
            text: '(%):Valor sensado',
            margin: 10
        }
    },
    series: [{
        name: 'valor',
        data: (function() {
            var data = [];
            <?php
                for($i = 0 ;$i<count($filas);$i++){
                    ?>
                    data.push([<?php echo $timeArray[$i];?>,<?php echo
$valoresy[$i];?>]);
                    <?php } ?>
                return data;
            })()
        }],
        credits: {
            enabled: false
        }
    }
});
</script>
<form method="get"
action="http://localhost/Conexion_arduino/graficas.php"><button
type="submit">Regresar a Consulta de Sensores</button></form>
</BODY>

</html>

```

Anexo 7.

Grafica de la luminosidad

```

<?php
session_start();
require "crud/Conexion.php";
require "crud/Crud.php";

```

```

require "RandomClass.php";

$fechai=$_SESSION['fechai'];
$horai=$_SESSION['horai'];
$fechaf=$_SESSION['fechaf'];
$horaf=$_SESSION['horaf'];

$datei=$fechai." ".$horai;
$datef=$fechaf." ".$horaf;

$model = new Crud();
$model->select = '*';
$model->from = 'medidas';
$model->condition = "ID_Sensor=3 AND fecha>='".$datei"' AND
fecha<='".$datef"'";
$model->Read();
$filas = $model->rows;
$total = count($filas);
echo $total;

//nos creamos dos arrays para almacenar el tiempo y el valor numérico
$valoresy;
$valoresx;
//en un bucle for obtenemos en cada iteración el valor numérico y
//el TIMESTAMP del tiempo y lo almacenamos en los arrays
for($i = 0 ;$i<count($filas);$i++){
    $valoresy[$i]= $filas[$i][1];
    //OBTENEMOS EL TIMESTAMP
    $valoresx= $filas[$i][0];
    $date = new DateTime($valoresx);
    //ALMACENAMOS EL TIMESTAMP EN EL ARRAY
    $timeArray[$i] = $date->getTimestamp()*1000;
    //echo $timeArray[$i];
}

?>

<HTML>
<BODY>

<meta charset="utf-8">

<div id="container"></div>

<script src="https://code.jquery.com/jquery.js"></script>
    <!-- Importo el archivo Javascript de Highcharts directamente desde
su servidor -->
<script src="http://code.highcharts.com/stock/highstock.js"></script>

```

```

<script src="http://code.highcharts.com/modules/exporting.js"></script>
<script>

chartCPU = new Highcharts.StockChart({
  chart: {
    renderTo: 'container'
    //defaultSeriesType: 'spline'

  },
  rangeSelector : {
    enabled: false
  },
  title: {
    text: 'Gráfica Sensor de Luminosidad'
  },
  xAxis: {
    type: 'datetime'
    //tickPixelInterval: 150,
    //maxZoom: 20 * 1000
  },
  yAxis: {
    minPadding: 0.2,
    maxPadding: 0.2,
    title: {
      text: 'LUXES (LX)',
      margin: 10
    }
  },
  series: [{
    name: 'valor',
    data: (function() {
      var data = [];
      <?php
        for($i = 0 ;$i<count($filas);$i++){
          ?>
          data.push([<?php echo $timeArray[$i];?>,<?php echo
$valoresy[$i];?>]);
          <?php } ?>
          return data;
        })()
    }],
  credits: {
    enabled: false
  }
});

</script>

```

```
<form method="get"
action="http://localhost/Conexion_arduino/graficas.php"><button
type="submit">Regresar a Consulta de Sensores</button></form>
```

```
</BODY>
```

```
</html>
```

Anexo 8.

Grafica de la humedad del suelo

```
<?php
session_start();
require "crud/Conexion.php";
require "crud/Crud.php";
require "RandomClass.php";

$fechai=$_SESSION['fechai'];
$horai=$_SESSION['horai'];
$fechaf=$_SESSION['fechaf'];
$horaf=$_SESSION['horaf'];

$datei=$fechai." ".$horai;
$datef=$fechaf." ".$horaf;
$model = new Crud();
$model->select = '*';
$model->from = 'medidas';
$model->condition = "ID_Sensor=2 AND fecha>='$datei' AND
fecha<='$datef'";
$model->Read();

$filas = $model->rows;
$total = count($filas);
echo $total;
//nos creamos dos arrays para almacenar el tiempo y el valor numérico
$valoresy;
$valoresx;
//en un bucle for obtenemos en cada iteración el valor numérico y
//el TIMESTAMP del tiempo y lo almacenamos en los arrays
for($i = 0 ;$i<count($filas);$i++){
    $valoresy[$i]= $filas[$i][1];
    //OBTENEMOS EL TIMESTAMP
    $valoresx= $filas[$i][0];
    $date = new DateTime($valoresx);
    //ALMACENAMOS EL TIMESTAMP EN EL ARRAY
    $timeArray[$i] = $date->getTimestamp()*1000;
    //echo $timeArray[$i];
}
```

?>

<HTML>

<BODY>

<meta charset="utf-8">

<div id="container"></div>

<script src="https://code.jquery.com/jquery.js"></script>

<!-- Importo el archivo Javascript de Highcharts directamente desde su servidor -->

<script src="http://code.highcharts.com/stock/highstock.js"></script>

<script src="http://code.highcharts.com/modules/exporting.js"></script>

<script>

```
chartCPU = new Highcharts.StockChart({
  chart: {
    renderTo: 'container'
    //defaultSeriesType: 'spline'

  },
  rangeSelector : {
    enabled: false
  },
  title: {
    text: 'GrÃ¡fica Sensor de Humedad del Suelo'
  },
  xAxis: {
    type: 'datetime'
    //tickPixelInterval: 150,
    //maxZoom: 20 * 1000
  },
  yAxis: {
    minPadding: 0.2,
    maxPadding: 0.2,
    title: {
      text: 'HUMEDAD (%)',
      margin: 10
    }
  },
  series: [{
    name: 'valor',
    data: (function() {
      var data = [];
      <?php
        for($i = 0 ;$i<count($filas);$i++){
```

```

        ?>
        data.push([<?php echo $timeArray[$i];?>,<?php echo
$valoresy[$i];?>]);
        <?php } ?>
        return data;
    }) ()
    }],
    credits: {
        enabled: false
    }
});

</script>
<form method="get"
action="http://localhost/Conexion_arduino/graficas.php"><button
type="submit">Regresar a Consulta de Sensores</button></form>

</BODY>

</html>

```

Anexo 9.

Descargar temperatura

```

<?php
require 'crud/Conexion.php';
require 'crud/Crud.php';
session_start();

$fechai=$_SESSION['fechai'];
$horai=$_SESSION['horai'];
$fechaf=$_SESSION['fechaf'];
$horaf=$_SESSION['horaf'];

$datei=$fechai." ".$horai;
$datef=$fechaf." ".$horaf;

$model = new Crud();
$model->select = '*';
$model->from = 'medidas';
$model->condition = "ID_Sensor=1 AND fecha>='$datei' AND
fecha<='$datef'";
$model->Read();
$filas = $model->rows;
$total = count($filas);

// Exporta a Excel
header("Content-type: application/vnd.ms-excel");

```

```

header("Content-Disposition: attachment;
filename=Reporte_Temperatura_Ambiente.xls");

?>
<!DOCTYPE html>
<html>
  <head>
    <!-- Latest compiled and minified CSS -->
    <!-- <link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.
css"> -->
    <link rel="stylesheet" href="bootstrap-
3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" href="estilo.css">
    <!-- jQuery library -->
    <!-- <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"
></script> -->
    <script src="bootstrap-3.3.7/jquery.min.js">
    <!-- Latest compiled JavaScript -->
    <!-- <script
src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
"></script> -->
    <script src="bootstrap-
3.3.7/js/bootstrap.min.js"></script>

    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
  </head>
  <body>
    <h1>Datos consultados </h1> <!-- Titulo-->
    <strong>El total de medidas son:<?php echo $total; ?> </strong>
    <table>
    <tr>
      <th>Fecha:</th>
      <th>Medida [Â°C]:</th>
    </tr>

    <?php
foreach($filas as $fila)
{
  echo "<tr>";
  echo "<td>".$fila['fecha']."</td>";
  echo "<td>".$fila['valor']."</td>";
  echo "</tr>";
}
}

```

```

    }
    ?>
    </table>
</body>
</html>

```

Anexo 10.

Descargar humedad relativa

```

<?php
require 'crud/Conexion.php';
require 'crud/Crud.php';
session_start();

$fechai=$_SESSION['fechai'];
$horai=$_SESSION['horai'];
$fechaf=$_SESSION['fechaf'];
$horaf=$_SESSION['fechaf'];

$datei=$fechai." ".$horai;
$datef=$fechaf." ".$horaf;

$model = new Crud();
$model->select = '*';
$model->from = 'medidas';
$model->condition = "ID_Sensor=2 AND fecha>='".$datei"' AND
fecha<='".$datef"'";
$model->Read();
$filas = $model->rows;
$total = count($filas);

// Exporta a Excel
header("Content-type: application/vnd.ms-excel");
header("Content-Disposition: attachment;
filename=Reporte_Humedad_Relativa.xls");

?>
<!DOCTYPE html>
<html>
    <head>
        <!-- Latest compiled and minified CSS -->
        <!-- <link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.
css"> -->
        <link rel="stylesheet" href="bootstrap-
3.3.7/css/bootstrap.min.css">

```



```

        <link rel="stylesheet" href="estilo.css">
        <!-- jQuery library -->
        <!-- <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"
></script> -->
        <script src="bootstrap-3.3.7/jquery.min.js">
        <!-- Latest compiled JavaScript -->
        <!-- <script
src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
"></script> -->
        <script src="bootstrap-
3.3.7/js/bootstrap.min.js"></script>

        <meta name="viewport" content="width=device-width,
initial-scale=1">
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
    </head>
    <body>
        <h1>Datos consultados </h1> <!-- Titulo-->
        <strong>El total de medidas son:<?php echo $total; ?> </strong>
        <table>
        <tr>
            <th>Fecha:</th>
            <th>Medida [%]:</th>
        </tr>

        <?php

        foreach($filas as $fila)
        {
            echo "<tr>";
            echo "<td>".$fila['fecha']."</td>";
            echo "<td>".$fila['valor']."</td>";
            echo "</tr>";
        }

        ?>
        </table>
    </body>
</html>

```

Anexo 11.

Descargar luminosidad

```

<?php
require 'crud/Conexion.php';

```

```

require 'crud/Crud.php';
session_start();

$fechai=$_SESSION['fechai'];
$horai=$_SESSION['horai'];
$fechaf=$_SESSION['fechaf'];
$horaf=$_SESSION['fechaf'];

$datei=$fechai." ".$horai;
$datef=$fechaf." ".$horaf;

$model = new Crud();
$model->select = '*';
$model->from = 'medidas';
$model->condition = "ID_Sensor=3 AND fecha>='$datei' AND
fecha<='$datef'";
$model->Read();
$filas = $model->rows;
$total = count($filas);

// Exporta a Excel
header("Content-type: application/vnd.ms-excel");
header("Content-Disposition: attachment;
filename=Reporte_Cantidad_Luz.xls");

?>
<!DOCTYPE html>
<html>
  <head>
    <!-- Latest compiled and minified CSS -->
    <!-- <link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.
css"> -->
    <link rel="stylesheet" href="bootstrap-
3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" href="estilo.css">
    <!-- jQuery library -->
    <!-- <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"
></script> -->
    <script src="bootstrap-3.3.7/jquery.min.js">
    <!-- Latest compiled JavaScript -->
    <!-- <script
src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
"></script> -->
    <script src="bootstrap-
3.3.7/js/bootstrap.min.js"></script>

```

```

                <meta name="viewport" content="width=device-width,
initial-scale=1">
                <meta http-equiv="Conect-Type" content="text/html; charset=UTF-
8">
        </head>
        <body>
            <h1>Datos consultados </h1> <!-- Titulo-->
            <strong>El total de medidas son:<?php echo $total; ?> </strong>
            <table>
            <tr>
                <th>Fecha:</th>
                <th>Medida [lx]:</th>
            </tr>

            <?php

            foreach($filas as $fila)
            {
                echo "<tr>";
                echo "<td>".$fila['fecha']."</td>";
                echo "<td>".$fila['valor']."</td>";
                echo "</tr>";
            }

            ?>
            </table>
        </body>
</html>

```

Anexo 12.

Descargar humedad del suelo

```

<?php
require 'crud/Conexion.php';
require 'crud/Crud.php';
session_start();

$fechai=$_SESSION['fechai'];
$horai=$_SESSION['horai'];
$fechaf=$_SESSION['fechaf'];
$horaf=$_SESSION['fechaf'];

$datei=$fechai." ".$horai;
$datef=$fechaf." ".$horaf;

$model = new Crud();
$model->select = '*';
$model->from = 'medidas';

```

```

$model->condition = "ID_Sensor=4 AND fecha>='$datei' AND
fecha<='$datef'";
$model->Read();
$filas = $model->rows;
$total = count($filas);

// Exporta a Excel
header("Content-type: application/vnd.ms-excel");
header("Content-Disposition: attachment;
filename=Reporte_Humedad_Suelo.xls");

?>
<!DOCTYPE html>
<html>
  <head>
    <!-- Latest compiled and minified CSS -->
    <!-- <link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.
css"> -->
    <link rel="stylesheet" href="bootstrap-
3.3.7/css/bootstrap.min.css">
    <link rel="stylesheet" href="estilo.css">
    <!-- jQuery library -->
    <!-- <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"
></script> -->
    <script src="bootstrap-3.3.7/jquery.min.js">
    <!-- Latest compiled JavaScript -->
    <!-- <script
src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js
"></script> -->
    <script src="bootstrap-
3.3.7/js/bootstrap.min.js"></script>

    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-
8">
  </head>
  <body>
    <h1>Datos consultados </h1> <!-- Titulo-->
    <strong>El total de medidas son:<?php echo $total; ?> </strong>
    <table>
    <tr>
      <th>Fecha:</th>
      <th>Medida [%]:</th>
    </tr>

```

```

<?php

foreach($filas as $fila)
{
    echo "<tr>";
    echo "<td>".$fila['fecha']."</td>";
    echo "<td>".$fila['valor']."</td>";
    echo "</tr>";
}

?>
</table>
</body>
</html>

```

Anexo 13.

Leer información de la base de datos

```

<?php
class RandomTable{

    public $IDr = 0 ;
    //Función que crea y devuelve un objeto de conexión a la base de
datos y chequea el estado de la misma.
    function conectarBD(){
        $server = "localhost";
        $usuario = "root";
        $pass = "";
        $BD = "adquisicion_datos";
        //variable que guarda la conexión de la base de datos
        $conexion = mysqli_connect($server, $usuario, $pass, $BD);
        //Comprobamos si la conexión ha tenido éxito
        if(!$conexion){
            echo 'Ha sucedido un error inesperado en la conexión de
la base de datos<br>';
        }
        //devolvemos el objeto de conexión para usarlo en las
consultas
        return $conexion;
    }
    /*Desconectar la conexión a la base de datos*/
    function desconectarBD($conexion){
        //Cierra la conexión y guarda el estado de la operación
en una variable
        $close = mysqli_close($conexion);
        //Comprobamos si se ha cerrado la conexión correctamente
        if(!$close){
            echo 'Ha sucedido un error inesperado en la desconexión
de la base de datos<br>';

```

```

    }
    //devuelve el estado del cierre de conexión
    return $close;
}

//Devuelve un array multidimensional con el resultado de la
consulta
function getArraySQL($sql){
    //Creamos la conexión
    $conexion = $this->conectarBD();
    //generamos la consulta
    if(!$result = mysqli_query($conexion, $sql)) die();

    $rawdata = array();
    //guardamos en un array multidimensional todos los datos de la
consulta
    $i=0;
    while($row = mysqli_fetch_array($result))
    {
        //guardamos en rawdata todos los vectores/filas que nos
devuelve la consulta
        $rawdata[$i] = $row;
        $i++;
    }
    //Cerramos la base de datos
    $this->desconectarBD($conexion);
    //devolvemos rawdata
    return $rawdata;
}

//inserta en la base de datos un nuevo registro en la tabla
usuarios
/*function insertRandom(){
    //Generamos un número entero aleatorio entre 0 y 100
    $ran = rand(0, 100);
    //creamos la conexión
    $conexion = $this->conectarBD();
    //Escribimos la sentencia sql necesaria respetando los tipos de
datos
    $sql = "insert into random (valor)
values (".$ran.)";
    //hacemos la consulta y la comprobamos
    $consulta = mysqli_query($conexion, $sql);
    if(!$consulta){
        echo "No se ha podido insertar en la base de
datos<br><br>".mysqli_error($conexion);
    }
    //Desconectamos la base de datos
    $this->desconectarBD($conexion);
    //devolvemos el resultado de la consulta (true o false)
}

```

```

        return $consulta;
    }*/
function getAllInfo(){
    //Creamos la consulta
    $sql = "SELECT * FROM random;";
    //obtenemos el array con toda la informaci3n
    return $this->getArraySQL($sql);
}
}
?>

```

Anexo 14.

Interfaz grfica

```

<?php
require "crud/Conexion.php";
require "crud/Crud.php";

$mensaje = null;
$model=new Crud();
$model->select= '*'; //que campos buscar
$model->from='sensores'; // nombre tabla
$model->Read();
$filas=$model->rows;

if (isset($_POST["create"]))
{

    $sensor=$_POST['sensor'];
    $fechai=$_POST['fecha'];
    $fechaf=$_POST['fechaF'];
    $horai=$_POST['hora'];
    $horaf=$_POST['horaF'];
    $op=$_POST['opcion'];

    session_start();
    $_SESSION['fechai']=$fechai;
    $_SESSION['fechaf']=$fechaf;
    $_SESSION['horai']=$horai;
    $_SESSION['horaf']=$horaf;

    if ($sensor == '')
    {
        $mensaje = "Error, llene el campo de Sensor";
    }
    else if($fechai == '')
    {

```

```

    $mensaje = 'Error, llene el campo de Fecha Inicial';
}
else if($fechaf == '')
{
    $mensaje = 'Error, llene el campo de Fecha Final';
}
else if($horai == '')
{
    $mensaje = 'Error, llene el campo de Hora Inicial';
}
else if($horaf == '')
{
    $mensaje = 'Error, llene el campo de Hora Final';
}
else
{
    if($sensor == '4' AND $op=='NO'){
        header("location:drawchart.php");
    }
    if($sensor == '1' AND $op=='NO'){
        header("location:graficar_Temp.php");
    }
    if($sensor == '3' AND $op=='NO'){
        header("location:graficar_luz.php");
    }
    if($sensor == '2'AND $op=='NO'){
        header("location:graficar_humsuelo.php");
    }

    if($sensor == '4' AND $op=='SI'){
        header("location:descargarHR.php");
    }
    if($sensor == '1' AND $op=='SI'){
        header("location:descargarTA.php");
    }
    if($sensor == '3' AND $op=='SI'){
        header("location:descargarLM.php");
    }
    if($sensor == '2' AND $op=='SI'){
        header("location:descargarHS.php");
    }
}
}
?>
<html>
<head>
<meta charset="UTF-8">

```



```

</head>
<body>
  <h1>Consulta de Sensores : Estadísticas</h1>
  <strong><?php echo $mensaje; ?>

  <form method="POST" action="<?php echo $_SERVER['PHP_SELF'];
?>">
    <TABLE>
      <tr>
        <td>SENSOR:</td>
        <td>
          <select name="sensor">
            <option value="" selected>Escoja
Sensor</option>
            <?php
              foreach($filas as $fila)
                {
                  <?php echo $fila['ID_Sensor']?><?php echo $fila['Nombre'].
". $fila['Tipo']?></option>
                  <?php }?>
                </select>
              </td>
            </tr>
            <tr>
              <td>FECHA INICIAL:</td>
              <td>
                <input id="fecha" name="fecha" class="element
text medium" type="date" value=""/>
                <?php echo "H/m/s";?>
                <input type="text" name="hora" value="<?php echo
date('H:i:s', time()); ?>" size="10" >
              </td>
            </tr>
            <tr>
              <td>FECHA FINAL:</td>
              <td>
                <input id="fechaF" name="fechaF" class="element
text medium" type="date" value=""/>
                <?php echo "H/m/s";?>
                <input type="text" name="horaF" value="<?php echo
date('H:i:s', time()); ?>" size="10" >
              </td>
            </tr>
            <tr>
            </tr>
            <tr>
            </tr>
            <tr>

```

```

                <td>DESCARGAR REPORTE:</td>
                <td>
                    SI: <input type='radio' name='opcion' checked
value='SI'>
                    NO: <input type='radio' name='opcion'
value='NO'>
                </td>
            </tr>
        </TABLE>
        <input type="hidden" name="create">
        <input type="submit" value="Enviar">
    </form>
</body>
</html>

```

Anexo 15.

Código en Arduino

```
#include <DHT.h>
```

```
/*
```

Web client

This sketch connects to a website (<http://www.google.com>)
using an Arduino Wiznet Ethernet shield.

Circuit:

Ethernet shield attached to pins 10, 11, 12, 13

```
*/
```

```
#include <SPI.h>
```

```
#include <Ethernet.h>
```

```
#include "DHT.h"
```

```
#define DHTPIN 2 // Pin donde está conectado el sensor DHT22(Temperatura/ Humedad)
```

```
#define DHTTYPE DHT22 // Sensor DHT22
```

```
#define M1 52//ESTRACTOR DE HUMEDAD
```

```
#define M2 53//HUMIDIFICADOR
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

```
IPAddress server(192,168,1,1); // numeric IP for Google (no DNS)
```

```
IPAddress ip(192, 168, 1, 2);
```

```
EthernetClient client;
```

```
int lightPin = 0; // Pin para la fotoresistencia
```

```
int humsueloPin = 1; //Pin para el sensor de humedad de suelo
```

```
int luz;
```

```
int humsuelo;
```

```
float hpor;
```

```
int relay = 28; //pin para activar el relé
```

```
int conta=0;
```

```
void setup() {
```

```
  pinMode(30, OUTPUT);
```

```
  pinMode(31, OUTPUT);
```

```
  pinMode(52, OUTPUT);
```

```
  pinMode(53, OUTPUT);
```

```
  pinMode(relay, OUTPUT);
```

```
  digitalWrite(relay,HIGH);
```

```
  digitalWrite(52, LOW);
```

```
  digitalWrite(53, LOW);
```

```
  Serial.println("Iniciando...DHT22 test!");
```

```
  dht.begin();
```

```
  Serial.begin(9600);
```

```
while (!Serial) {  
  ; // wait for serial port to connect. Needed for native USB port only  
  
}  
  
}  
  
void loop() {  
  conta++;  
  luz=analogRead(lightPin);//leemos luminosidad  
  
  humsuelo=analogRead(humsueloPin);//leemos humedad del suelo  
  hpor=(humsuelo/1023.0)*100.0;  
  
  delay(10);  
  float h = dht.readHumidity(); //Leemos la Humedad  
  float t = dht.readTemperature(); //Leemos la temperatura en grados Celsius  
  Serial.print("Temperatura=");  
  Serial.print(t);  
  Serial.println("°C");  
  
  Serial.print("Humedad de Ambiente=");  
  Serial.print(h);  
  Serial.println("%");  
  
  Serial.print("Luminosidad=");  
  Serial.print(luz);  
  Serial.println("lx");  
  
  Serial.print("Humedad del Suelo=");
```

```
Serial.print(hpor);  
Serial.println("%");
```

```
//CONTROL TEMPERATURA
```

```
if (t>31){  
    digitalWrite(52, LOW);  
    digitalWrite(53, HIGH);  
    digitalWrite(relay,HIGH);
```

```
}
```

```
else
```

```
{
```

```
    digitalWrite(52, LOW);  
    digitalWrite(53, LOW);
```

```
}
```

```
if (t<30){  
    digitalWrite(relay,LOW);
```

```
}
```

```
//CONTROL HUMEDAD
```

```
if (h<69){  
    digitalWrite(30, LOW);  
    digitalWrite(31, HIGH);
```

```
}
```

```
else
```

```

{
  digitalWrite(30, LOW);
  digitalWrite(31, LOW);

}

/* // if there are incoming bytes available
// from the server, read them and print them:
// start the Ethernet connection:*/
if (conta>1000)// CONEXION CON EL SERVIDOR VIA ETHERNET, BASE DE DATOS
{
  conta=0;
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
  }
  // try to configure using IP address instead of DHCP:
  Ethernet.begin(mac, ip);

  // give the Ethernet shield a second to initialize:
  Serial.println("connecting...");
  // if you get a connection, report back via serial:
  if (client.connect(server, 80)) {
    Serial.println("connected");
    // Make a HTTP request:
    client.print("Host:
http://localhost/Conexion_arduino/datossensados.php?ID_Sensor=1&valor=");
    client.print(t);
    client.print("&ID_Sensor1=2&valor1=");
    client.print(hpor);
    client.print("&ID_Sensor2=3&valor2=");

```

```
client.print(luz);
client.print("&ID_Sensor3=4&valor3=");
client.println(h);
client.print("Connection: close");
client.println();
}
else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}
delay(25);
if (client.available()) {
char c = client.read();
Serial.print(c);
}
// if the server's disconnected, stop the client:
if (!client.connected()) {
Serial.println();
Serial.println("disconnecting.");
client.stop();
// do nothing forevermore:
}
}
}
```