



INSTITUTO TECNOLÓGICO DE TUXTLA GUTIÉRREZ

INFORME DEL PROYECTO DE RESIDENCIA PROFESIONAL

Sistema de monitoreo y control de proceso industrial basado en placas de desarrollo Arduino y Raspberry comunicados vía internet

Carrera: Ingeniería en Electrónica

Autor: José David Arroyo Pérez

No. de Control: 10270474

Correo: jose_da_24@hotmail.com

Asesor Interno: Mtro. Raul Moreno Rincón

TUXTLA GUTIERREZ, CHIAPAS, ABRIL 2015

1. Introducción	4
1.1 Antecedentes	4
1.2 Estado del Arte	4
1.3 Justificación	5
1.4 Objetivo	5
1.5 Metodología	6
2. Caracterización del área donde se participó	7
2.1 Historia del ITTG	7
2.2 Misión	7
2.3 Visión	7
2.4 Localización	7
2.5 Área específica relacionada directamente con el proyecto	8
3. Problemas a resolver	8
4. Fundamento teórico	9
4.1 Placas de desarrollo	9
4.2 Sensores	10
4.3 Comunicaciones y base de datos	13
4.3 Herramientas de software	18
4.4 Lenguajes de programación	20
5. Procedimiento y actividades realizadas (Desarrollo)	24
5.1 Funcionamiento	24
5.3 Etapa de Monitoreo	26
5.4 Etapa de Control	26
6. Resultados	31

7. Conclusiones	36
8. Referencias	36
9. Anexos	38
9.1 Imágenes del invernadero	38
9.2 Diagramas eléctricos	39
9.3 Códigos	41
9.4 Instalación de Raspbian en tarjeta de memoria SD	80
9.5 Raspberry como servidor web	83

El trabajo del ingeniero implica resolver problemas aplicando métodos y tecnologías que derivan en un proceso reflexivo profundo de una situación y del entorno. La diversidad del saber científico y tecnológico que se involucran en la resolución de problemas requiere establecer puntos de vista unificadores[1].

1.1 Antecedentes

La crisis económica, combinada con las crecientes expectativas de los ciudadanos, está aumentando la presión, para proporcionar mejores soluciones y eficientes servicios, por menos costo. Esta tendencia ha contribuido a la creciente popularidad del uso de hardware libre y software de código abierto como son Arduino y Raspberry Pi.

Arduino y Raspberry Pi, pueden lucir parecidas y es posible que se asuma que este par de plataformas de hardware compiten por resolver problemas similares. En realidad son diferentes; Raspberry Pi es una computadora completamente funcional, que puede correr múltiples programas e incluso programar con ella mientras que Arduino es un microcontrolador, el cual sólo corre un programa, el que tenga cargado en su memoria.

Aunque Arduino puede ser programado con aplicaciones basadas en C, este no puede ejecutar un sistema operativo y no puede ser el sustituto de una computadora. Raspberry Pi es una plataforma apta para el desarrollo de aplicaciones domóticas y de control. Sin embargo, Raspberry Pi tiene limitaciones, por ejemplo; el número de entradas/salidas disponibles o problemas con los tiempos de respuesta que se consiguen programando en Python.

1.2 Estado del Arte

La tarjeta de desarrolló Raspberry Pi es 40 veces más rápida que Arduino, esto por su velocidad de reloj. Además, Pi tiene 128,000 veces más memoria RAM. Raspberry Pi es una computadora independiente que puede ejecutar un sistema operativo real basado en Linux. Puede realizar varias tareas, soportar dos puertos USB y conectarse de forma inalámbrica a Internet. Pi es lo suficientemente potente como para funcionar como una computadora personal, pero no lo suficientemente para competir con una Mac u otra PC.

Puede sonar que Raspberry Pi es superior a Arduino, pero eso es sólo para cierto tipo de aplicaciones de software. La simplicidad de Arduino hace que éste sea una apuesta mejor para proyectos de hardware con tareas repetitivas de control. Arduino y Raspberry Pi no son dos productos sin relación, juntos pueden hacer una simbiosis interesante y permitir ampliar el rango de posibilidades de una placa Arduino sola.

Lo que en el presente proyecto se hace es combinar Raspberry Pi con Arduino asignando el control del hardware a Arduino y utilizando al Raspberry como controlador maestro. La comunicación entre ambos dispositivos se puede hacer de varias maneras. Las formas físicas de hacerlo son a través de: el módulo Ethernet, el cable USB, o Bluetooth.

Una forma sencilla de comunicación es mediante el puerto serie. Los módulos Bluetooth para Arduino funcionan como modem que se conecta al puerto serie del microcontrolador. Una vez elegido el método físico, se encuentran diversos métodos de comunicación: aplicaciones para realizar el control, mediante la consola de Arduino IDE o desarrollando una aplicación propia escrita en cualquier lenguaje de programación que admita la comunicación serie; Python, C, Java, Processing.

1.3 Justificación

El presente proyecto es importante porque a través del uso de hardware libre y software de código abierto se desarrolla un sistema de monitoreo y control que proporciona soluciones eficientes, es de fácil uso y está enfocado a satisfacer los requerimientos propios de los procesos industriales, mejorando la seguridad en ellos y la calidad de sus productos. También el empleo del sistema desarrollado favorece a los usuarios en el logro de servicios eficientes.

1.4 Objetivo

Desarrollar un sistema de monitoreo de cinco variables las cuales son temperatura T [°C], humedad relativa del aire H_a [%], humedad relativa del suelo H_s [%], luminosidad L [Lux], consumo energético I [A]. También que el sistema controle las variables temperatura, luminosidad, y humedad del suelo con un rango amplio de valor programable.

1.5 Metodología

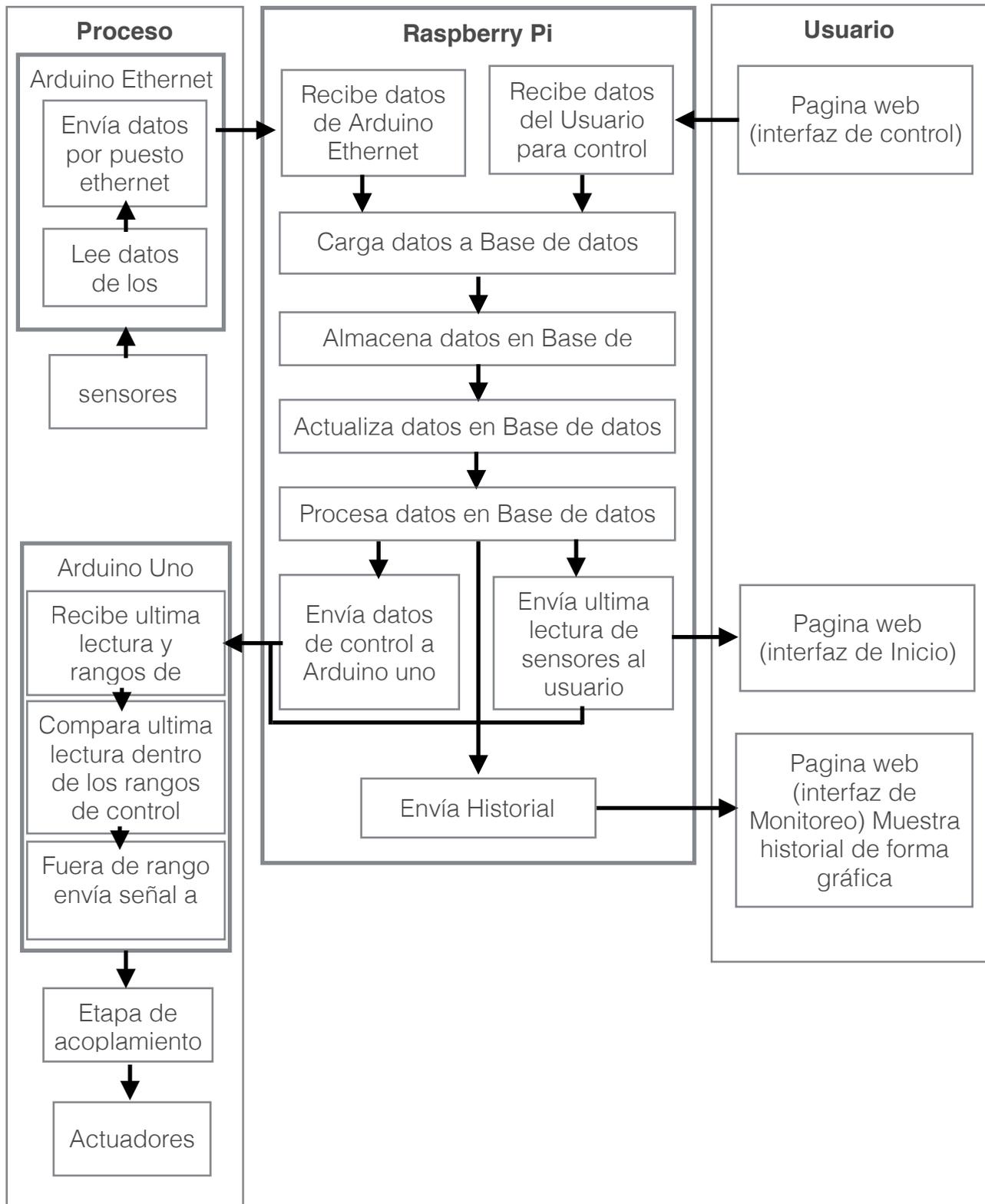


Fig. 1.1 Diagrama a bloques del sistema de monitoreo y control.

2. Caracterización del área donde se participó

2.1 Historia del ITTG

El Instituto Tecnológico de Tuxtla Gutiérrez es una institución pública dependiente de la Secretaría de Educación Pública. Imparte 8 licenciaturas y 2 programas de posgrado en las áreas de Ingeniería, Tecnología y Ciencias Administrativas.

El Instituto Tecnológico de Tuxtla Gutiérrez fue fundado el 24 de octubre de 1972, se encuentra ubicado en el centro del estado de Chiapas, que por su variedad de climas y suelos es propio para el cultivo de diversas especies vegetales nativas y adaptadas, con lo que se puede desarrollar la agroindustria, así como otros procesos industriales a partir del potencial que ofrece este estado, en el cual el Instituto debe constituirse en el sujeto de cambio, al presentar alternativas de desarrollo sustentable del estado en forma multidisciplinaria.

El Instituto Tecnológico de Tuxtla Gutiérrez cuenta con laboratorios de microbiología, biotecnología, química, mecánica, sistemas computacionales, ingeniería industrial, eléctrica y electrónica, plantas piloto y un Polo Tecnológico Nacional.

2.2 Misión

Formar de manera integral profesionistas de excelencia en el campo de la ciencia y la tecnología con actitud emprendedora, respeto al medio ambiente y apego a los valores éticos.

2.3 Visión

Ser una Institución de excelencia en la educación superior tecnológica del Sureste, comprometida con el desarrollo socioeconómico sustentable de la región.

2.4 Localización

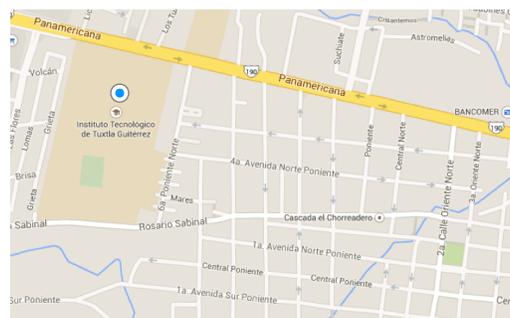


Fig.2.1 Ubicación del ITTG en Google Maps.

Ubicado en Carretera Panamericana Kilómetro 1080, Colonia Terán, C.P 29050 Tuxtla Gutiérrez, Chis.

2.5 Área específica relacionada directamente con el proyecto

El laboratorio de Ingeniería Electrónica cuenta con 8 áreas destinadas a la investigación, desarrollo e implementación de proyectos escolares, 5 de las cuales son educativas, en las que los docentes imparten clases de: electrónica digital, PLC, electrónica de potencia, instrumentación, mediciones eléctricas, programación en C y electrónicas analógicas.

Dos son para investigación, una pertenece al grupo de IEEE y la otra es de desarrollo de circuitos electrónicos. La última área está destinada para que los alumnos desarrollen sus actividades en extra clase. La carrera de Ingeniería Electrónica tiene como misión "formar profesionales de excelencia con competencias en el ámbito de la Ingeniería Electrónica, motivados para la promoción del desarrollo profesional y el conocimiento científico y tecnológico, con actitud emprendedora, respeto al medio ambiente y apego a los valores cívicos y éticos".

3. Problemas a resolver

El presente proyecto resuelve el problema de disponer de una herramienta genérica que permita desarrollar aplicaciones enfocadas al monitoreo, análisis y control de procesos. Esta herramienta permite solucionar una gama amplia de problemas a través de la implementación de aplicaciones específicas, por ejemplo; soluciona problemas de pequeñas industrias emergentes que busquen mejorar la calidad de los productos con un sistema que ayude a monitorear y controlar la línea de producción.

El desarrollo del proyecto se relaciona con las áreas de comunicaciones de datos, programación de lenguajes de mediano y alto nivel, control, electrónica analógica, digital y de potencia. Algunos de los problemas que se resolvieron son; la comunicación entre Arduino - Raspberry, el almacenamiento de datos recolectados de los sensores, el procesamiento de los datos, la visualización del historial del funcionamiento. Y como caso de estudio se desarrollo una aplicación de control para las tareas de los actuadores de un invernadero.

En la implementación del proyecto se solucionaron problemas derivados del uso de linux, y diversas aplicaciones de software y lenguajes de programación. Por ejemplo; se estudio y aprendió todo lo referente al desarrollo de una aplicación con phpmyadmin. El desarrollo se hizo desde cero para la configuración del servidor en Raspberry.

Se resolvió el problema de subir los datos de Arduino al servidor. Esta tarea fue complicada por el hecho de haber estudiado y comprendido los lenguajes php y SQL desde cero. El problema que aquí se solucionó fue el de mandar los cinco datos de los sensores a una misma tabla de la base de datos. A través del estudio de los lenguajes se diseñó un algoritmo para mandar los datos usando cadenas de caracteres. En Arduino se forma la cadena de caracteres y se transmite. En Raspberry pi se recibe la cadena de caracteres y se descodifica.

Teniendo la base de datos, mostrar su comportamiento y su historial fue otro problema que se resolvió. Para que este reto fuese superado se investigó y se encontró la herramienta de software libre llamada highcharts. Con programación en php se adaptó la aplicación para garantizar la posibilidad de tener acceso a los datos de la base de datos y mostrar todo el historial en forma de líneas.

Teniendo la parte del monitoreo solucionada, quedó el reto de mandar los datos de control a Arduino. Esto se hizo con controles deslizantes, programación php y una tabla de phpMyadmin. También se usó programación en eclipse y lenguaje java. Finalmente se desarrolló una aplicación para mandar los datos a Arduino uno a uno por medio del puerto serie.

4. Fundamento teórico

4.1 Placas de desarrollo

Arduino.- es una plataforma, de electrónica abierta para la creación de prototipos, basada en software y hardware flexibles y fáciles de usar. Se creó para estudiantes, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información de sensores del entorno a través de sus pines de entradas analógicas y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino y el entorno de desarrollo Arduino, basado en Processing. Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software (p.ej. Flash, Processing, MaxMSP)[2].

Raspberry Pi.- es un ordenador del tamaño de una tarjeta de crédito que se puede conectar a una TV y a un teclado. Es una PC en miniatura con procesador ARM que se puede utilizar para muchas de las cosas que una PC de escritorio puede hacer, como

hojas de cálculo, procesador de word y videojuegos. Ha sido desarrollado por la Fundación Raspberry Pi de Reino Unido con la intención de estimular la enseñanza de la informática básica en las escuelas [3].



Fig. 4.1 Raspberry pi modelos B.

4.2 Sensores

Sensor de temperatura y humedad relativa del aire DHT11.- es un sensor digital de temperatura y humedad integrado que contiene una salida para transmisión digital de la temperatura y la humedad. El sensor transmite un paquete de datos donde van codificados los valores de la temperatura y la humedad. El sensor incluye un microcontrolador de ocho bits para coordinar las tareas de medición y comunicaciones.

Aplicaciones del DTH11.- equipos de aire acondicionado, deshumidificador, pruebas y equipos de inspección, bienes de consumo, registradores de datos, estaciones meteorológicas, electrodomésticos, regulador de la humedad, la medición de la humedad médica y control.

Características del DTH11.- estabilidad a largo plazo, medición de la humedad relativa y la temperatura, respuesta fiable, capacidad anti-interferencia, aceptable distancia de transmisión de señales, salida de señal digital, y posibilidad de calibración. Detalles más específicos sobre los parámetros del sensor se encuentran en el la figura 4.2.

humedad relativa		temperatura	
Resolucion	16 bits	Resolucion	16 bits
Repetibilidad	$\pm 1\%$ RH	Repetibilidad	$\pm 0,2\text{ }^{\circ}\text{C}$
Precisión	A $25\text{ }^{\circ}\text{C} \pm 5\%$ de humedad relativa	Rango	A $25\text{ }^{\circ}\text{C} \pm 2\text{ }^{\circ}\text{C}$ Tiempo de respuesta: 1 / e (63%) 10S
Intercambiabilidad	Tiempo de respuesta totalmente intercambiables: 1 / e (63%) de $25\text{ }^{\circ}\text{C}$ 6s 1 m / s 6s aire	Fuente de alimentación: DC 3.5 ~ 5.5V	
		Características eléctricas	
Histéresis	$<\pm 0.3\%$ RH	Corriente de alimentación: medida 0.3mA espera 60μ Un período de muestreo: más de 2 segundos	
Estabilidad a largo plazo	$<\pm 0,5\%$ HR / año en		

Fig.4.2 Características del sensor de temperatura y humedad DTH11.

Sensor de luz; precision light sensor 1127.- es un sensor de luz de precisión que mide el nivel de luz perceptible humana en luxes. El rango de medición es desde 1 lux hasta 1000 lux. Mas detalles de los parámetros se encuentran en el la figura 4.3, 4.4 y 4.5

Propiedades del sensor	
Sensor de tipo	luz
Tipo de salida del sensor	No Ratiometric
Tiempo de respuesta máximo	20 ms
Error de medición	$\pm 5\%$
Pico de sensibilidad de longitud de onda	580 nm
Nivel de luz minimo	1 lx
Nivel maximo (3.3V)	660 lx
Nivel maximo (5V)	1 Klx

Fig. 4.3 Sensor de luminosidad; **Precision light sensor 1127.**

Propiedades eléctricas	
Consumo de corriente	2 mA
Impedancia	1 kW
Tensión de alimentación Min.	3.3 V DC
Tensión de alimentación Max.	5 V DC

Fig. 4.4 Sensor de luminosidad, propiedades eléctricas.

Propiedades físicas	
Temperatura de funcionamiento mínima	- 30° C
Temperatura de funcionamiento máxima	85 ° C

Fig. 4.5 Sensor de luminosidad, propiedades físicas.

Sensor de humedad del suelo.- es un sensor con inmersión de oro (ENIG), el cual protege al Nickel de la oxidación. La inmersión de oro origina ventajas sobre los sensores de electrodos con recubrimiento de soldadura (HASL). El sensor usa dos electrodos que pasan corriente a través del suelo. Luego es leída la caída de tensión para obtener el valor de la medición de la humedad. A mayor cantidad de agua provoca que la conductividad del sueño sea mayo, a menor resistencia menor caída de tensión y viceversa. Más detalles se encuentran en la fig. 4.6.

Especificaciones	
Voltaje de alimentacion	3.3 V - 5V
Voltaje de la señal de salida	0~4.2V
Corriente	35 mA

Fig. 4.6 Sensor de humedad del suelo

Sensor de corriente SCT-013-030.- La ventaja de este sensor es que no es invasivo, no precisa cortar el cable para efectuar la medición. El sensor usa un gancho que se abre para introducir y rodear al cable de la corriente que se desea monitorear. Se considera que el cable se comporta como un bobinado de espira única. El cable induce un voltaje en el transformador que incluye el sensor.

El principio de funcionamiento está basado en el transformador eléctrico. El sensor incorpora un transformador constituido por un núcleo y, en vez de dos devanados,

primario y secundario, tiene solo un enbobinado en uno de los extremos de su núcleo ferromagnético. Este enbobinado se desempeña como el secundario, la parte correspondiente al primario es el cable. En este caso es una espira única que constituirá el primario al momento de ser insertada en el interior del gancho del sensor.

Al hacer pasar por el “primario” del sensor un cable de la instalación, este capta el flujo magnético generado que es proporcional a la intensidad que circula en ese instante por el cable. Por efecto de la inducción electromagnética se obtiene en el secundario una intensidad proporcional a la del primario. Existe una resistencia de carga en la salida del transformador, de cuyos bornes se toma la señal a útil. Mas detalles de los parámetros se encuentran en la fig. 4.7.

Especificaciones	
Corriente de entrada	0-30 A
Tension de salida	0-1 V
No-linealidad	± 1%
Corriente de la resistencia de muestreo	62Ω
Relacion de vueltas	1800: 1
Grado de resistencia	Grado B
Temperatura de trabajo	-25 ° C ~ + 70 ° C
Rigides dielectrica (Entre la carcasa y la salida)	1500V AC / 1min 5 mA

Fig. 4.7 Sensor de corriente SCT-013-030.

4.3 Comunicaciones y base de datos

Servidor Web.- es un ordenador que está al “servicio” de otros ordenadores llamados clientes y que les suministra a estos la información almacenada en sus dispositivos de memoria. Por tanto un servidor en informática es un ordenador que suministra una información requerida por unos clientes, que pueden ser personas que accedan desde otros dispositivos como ordenadores o dispositivos móviles.

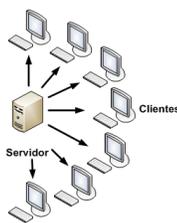


Fig. 4.8 Servidor web.

Base de datos.- es un “almacén” que permite guardar grandes cantidades de información de forma organizada para luego encontrarla y utilizarla fácilmente. El término de bases de datos fue empleado por primera vez en 1963, en un simposio celebrado en California, USA. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada de forma estructurada.

Una base de datos es un sistema formado por un conjunto de datos almacenados en discos y un conjunto de programas que manipulan ese conjunto de datos y permiten el acceso directo a ellos. Cada base de datos se compone de una o más tablas que guardan el conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que se guarda en la tabla, cada fila de la tabla conforma un registro.

Definición de base de datos.- Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio particular.

Características principales de los sistemas de base de datos.- Independencia lógica y física de los datos, redundancia mínima, acceso concurrente por parte de múltiples usuarios, integridad de los datos, consultas complejas optimizadas, seguridad de acceso y auditoría, respaldo y recuperación, acceso a través de lenguajes de programación estándar, Sistema de Gestión de Base de Datos (SGBD).

Los Sistemas de Gestión de Base de Datos - DataBase Management System.- son un tipo de software específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

Ventajas de las bases de datos - control sobre la redundancia.- los sistemas de archivos almacenan varias copias de los mismos datos en archivos distintos. Esto desperdicia espacio de almacenamiento. Además provoca inconsistencia de los datos. En los sistemas de bases de datos los archivos están integrados, por lo que se almacena solo una copia de los mismos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos.

Consistencia de datos.- controlando las redundancias de datos se reduce el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, entonces puede encargarse de garantizar que todas las copias se mantengan consistentes.

Compartir datos.- en los sistemas de archivos, los archivos pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados.

Mantenimiento de estándares.- Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

Mejora en la integridad de los datos.- la integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

Mejora en la seguridad.- La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin buenas medidas de seguridad, la integridad de los datos en los sistemas de bases de datos hace que éstos sean vulnerables a intrusos.

Mejora en la accesibilidad a los datos.- los SGBD proporcionan lenguajes de consulta o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

Mejora en la productividad.- el SGBD proporciona las funciones estándar que el programador necesita escribir en un sistema de archivos. A nivel básico, el SGBD proporciona todas las rutinas de manejo de archivos típicas de los programas de aplicación. El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel.

Mejora en el mantenimiento.- En los sistemas de archivos, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan. Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios en los programas cuyos datos se ven afectados. Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto se conoce como independencia de datos, y así se simplifica el mantenimiento de las aplicaciones.

Aumento de la concurrencia.- en algunos sistemas de archivos, si hay varios usuarios que pueden acceder simultáneamente a un mismo archivo, es posible que el acceso interfiera entre ellos de modo que se pierda información o se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

Mejora en la seguridad.- los sistemas de archivos dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido en caso de fallo.

Desventajas de las bases de datos – Complejidad.- los SGBD son conjuntos de programas que pueden tener gran funcionalidad, pero ser complejos. Es preciso comprender bien esta funcionalidad para realizar un buen uso de ellos. **Costo del equipamiento adicional.-** El SGBD y la base de datos pueden requerir más espacio de almacenamiento. Además, para alcanzar las prestaciones deseadas se recomienda adquirir una máquina más grande o una máquina dedicada al SGBD. Esto hace que la implantación de un sistema de bases de datos sea más cara.

Vulnerable a fallos.- El hecho de que el funcionamiento del SGBD esté centralizado hace que el sistema sea vulnerable ante fallos. Es por ello que deben tenerse copias de seguridad.

Tipos de Campos.- los Sistemas de Base de Datos poseen tipos de campos similares o diferentes. **Campos numéricos.-** pueden ser tipo entero o reales. **Booleanos.-** poseen dos estados; verdadero ó falso. **Memos.-** son campos alfanuméricos de longitud ilimitada. Presentan el inconveniente de no poder ser indexados. **Fechas.-** almacenan fechas facilitando su explotación. Almacenar fechas de esta forma posibilita ordenar los registros por fechas o calcular los días entre una fecha y otra.

Alfanuméricos.- contienen cifras y letras. Estos presentan una longitud limitada 255 caracteres. **Autoincrementables.-** son campos numéricos enteros que incrementan en una unidad su valor por cada registro incorporado. **Identificadores.-** son exclusivos de un registro.

Tipos de Bases de Datos.- MySql es una base de datos con licencia GPL basada en un servidor. Se caracteriza por su rapidez. No es recomendable para grandes volúmenes de datos. PostgreSQL y Oracle: Son sistemas de base de datos poderosos. Administran bien

grandes cantidades de datos, y suelen ser utilizadas en intranets y sistemas de gran calibre.

Access.- es una base de datos desarrollada por Microsoft. Esta base de datos, debe ser creada bajo el programa access, el cual crea un archivo .mdb. Microsoft SQL Server es una base de datos más potente que access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de información.

Modelo entidad-relación.- Los diagramas o modelos entidad-relación, denominados por su siglas, ERD "Diagram Entity relationship", son una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes de un sistema de información, sus inter-relaciones y propiedades.

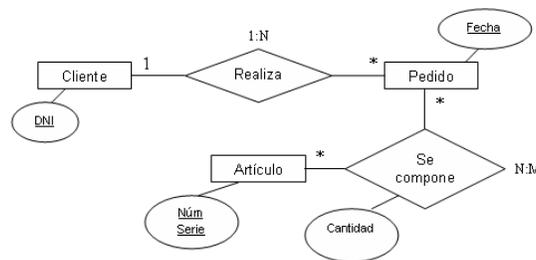


Fig. 4.9 Modelo de entidad-relación.

Cardinalidad de las Relaciones entre tablas - Relaciones de uno a uno.- una instancia de la entidad A se relaciona con una y solamente una de la entidad B. **Relaciones de uno a muchos.-** cada instancia de la entidad A se relaciona con varias instancias de la entidad B. **Relaciones de muchos a muchos.-** cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B.

Estructura de una Base de Datos.- una base de datos ordena la información de manera lógica y ese orden debe ser considerado para acceder a la información de manera coherente. Cada base de datos contiene tablas que cumplen la función de declarar los campos utilizados. En la figura 4.10 se muestra un ejemplo de una tabla de cuatro campos referentes a "comentarios". Los datos quedan organizados como se muestra en la Fig. 4.11.

Orden jerárquico.- una base de datos posee el siguiente orden; Tablas, Campos, Registros, Lenguaje SQL. El lenguaje SQL es el más universal en los sistemas de base de datos. Este lenguaje permite realizar consultas a bases de datos para mostrar, insertar, actualizar y borrar datos. **Ejemplo de Mostrar.-** para mostrar los registros se utiliza la instrucción Select junto con el nombre del registro: *Select * From comentarios.*

	Field	Type
<input type="checkbox"/>	id	int(11)
<input type="checkbox"/>	titulo	varchar(100)
<input type="checkbox"/>	texto	blob
<input type="checkbox"/>	fecha	varchar(10)

Fig. 4.10 Tabla de campos utilizados.

	id	titulo	texto	fecha
<input type="checkbox"/>	1	saludos	[BLOB - 0 B]	22-10-2007
<input type="checkbox"/>	2	como estas ???	[BLOB - 0 B]	23-10-2007

Fig. 4.11 Organización de datos.

Ejemplo de Insertar.- los registros se introducen a partir de sentencias que emplean la instrucción Insert: *Insert Into comentarios (titulo, texto, fecha) Values ('saludos', 'como estas???', '22-10-2007')*. **Ejemplo de Borrar.-** Para borrar un registro se utiliza la instrucción Delete. Despues se especifican los registros a borrar. Por ello es necesario establecer una selección mediante la cláusula Where. La instrucción completa especificando el registro queda: *Delete From comentarios Where id='1'*.

Ejemplo de Actualizar.- para actualizar los registros se utiliza la instrucción Update. Aquí también se requiere especificar con Where los registros de las modificaciones. Además, de especificar los nuevos valores de los campos a actualizar. La instrucción completa con el registro queda: *Update comentarios Set titulo='Mi Primer Comentario' Where id='1'*.

4.3 Herramientas de software

Highcharts.- es una biblioteca de gráficos escritos en JavaScript, que ofrece una manera fácil de agregar gráficos interactivos a una aplicación web. Highcharts soporta gráficos de tipo línea, spline, área, areaspline, column, barras, circulares, de dispersión, medidores angulares, arearange, areasplinerange, columnrange, burbuja, diagrama de caja, barras de error, embudo, cascada y tipos de gráficos polares[4].

Highcharts se basa en tecnologías de navegación nativas y funciona sin clientes plugins secundarios como Flash o Java. Además no es necesario instalar nada en el servidor, sin PHP o ASP.NET. Highcharts necesita sólo dos archivos JS a ejecutar: El núcleo highcharts.js y el jQuery, MooTools o marco Prototype.

El ajuste de las opciones de configuración Highcharts no requiere conocimientos de programación especiales. Las opciones se muestran en una estructura de notación de objetos JavaScript, que es básicamente un conjunto de claves y valores conectados por dos puntos, separados por comas y agrupados por llaves.

A través de una API completa, se puede añadir, eliminar, modificar series y puntos, modificar los ejes en cualquier momento después de la creación gráfica, o suministrar eventos de programación. En combinación con jQuery, MooTools y API Ajax de Prototype, abre soluciones con gráficos en vivo en constante actualización con los valores del servidor, y los datos suministrados por el usuario.

MYSQL.- es uno de los más importantes en diseño y programación de bases de datos de tipo relacional. Cuenta con múltiples aplicaciones y aparece en el mundo informático como una de las más utilizadas por usuarios del medio. El programa MySQL se usa como servidor a través del cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo.

MySQL inicia a principios de la década de 1980. Programadores de IBM lo desarrollaron para contar con un código de programación para generar bases de datos para empresas y organizaciones de diferente tipo. Desde esa época distintas versiones han surgido y han sido de gran importancia. Hoy en día MySQL es desarrollado por Sun Microsystems.

Una de las características de MySQL es que permite recurrir a bases de datos multiusuario a través de la web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos. También MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información, a diferencia de sistemas anteriores. Las plataformas que utiliza son LAMP, MAMP, SAMP, BAMP y WAMP aplicables a Mac, Windows, Linux, BSD, Open Solaris, Perl y Python.

PhpMyadmin.- es una herramienta de software libre escrito en PHP que permite administrar MySQL a través de Internet. PhpMyAdmin es compatible con operaciones en MySQL, MariaDB y Ilo Vizna. PhpMyAdmin es utilizado en operaciones de gestión de bases de datos, tablas, columnas, relaciones, índices, usuarios, permisos, y se puede realizar a través de la interfaz de usuario, y permite mantener la capacidad de ejecutar cualquier sentencia SQL. PhpMyAdmin es reconocido por SourceForge.net como "Mejor utilidad para los administradores de sistemas"[5].

Eclipse.- es una interfaz de usuario que pone a disposición un conjunto de complementos para desarrollo de código abierto basada en Java y ofrece un conjunto de servicios para el entorno de desarrollo de los componentes de entrada. Los usuarios de Eclipse ven a esta plataforma como una IDE de Java, pero Eclipse también incluye el Entorno de Desarrollo de Complementos (PDE), que es para extender Eclipse, dado que permite construir herramientas que se integran al entorno de Eclipse.

Eclipse se escribe en Java, pero no se limita a este. Los complementos se encuentran disponibles para incluir soporte para los lenguajes de programación C++ y COBOL. También Eclipse puede utilizarse como base para aplicaciones no relacionadas con el

desarrollo de software. Una aplicación de Eclipse es el entorno de trabajo de IBM® WebSphere® Studio, que forma la base de la familia de IBM de las herramientas de desarrollo de Java. WebSphere Studio Application Developer, agrega soporte para JSP, servlets, EJB, XML, servicios web y acceso a base de datos.

4.4 Lenguajes de programación

PHP.- es un lenguaje de código abierto adecuado para el desarrollo web, que puede ser incrustado en HTML, acrónimo recursivo de PHP: Hypertext Preprocessor. En lugar de usar muchos comandos para mostrar HTML, como en C o en Perl, las páginas de PHP contienen HTML con código incrustado que realiza las ordenes. El código de PHP está encerrado entre las etiquetas especiales de comienzo y final que permiten entrar y salir del "modo PHP"; `<?php, ?>`.

Lo que distingue a PHP como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no conocerá el código subyacente. El servidor web puede ser configurado para procesar los archivos HTML con PHP, por lo que no hay manera de que los usuarios conozcan el código ejecutado. Lo mejor de usar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores profesionales.

SQL.- es el lenguaje estándar ANSI/ISO de definición, manipulación y control de bases de datos relacionales. Es un lenguaje declarativo: sólo hay que indicar qué se quiere hacer. En cambio, en los lenguajes procedimentales es necesario especificar cómo hay que hacer la acción sobre la base de datos. SQL se parece al inglés, y es expresivo. Por estas razones, y como lenguaje estándar, el SQL es un lenguaje con el que se puede acceder a todos los sistemas relacionales comerciales.

El modelo relacional tiene como estructura de almacenamiento las relaciones de los datos. La intensión de una relación consiste en el nombre dado a la relación y un conjunto de atributos. La extensión de una relación es un conjunto de *tuplas*. Al trabajar con SQL, la nomenclatura cambia, lo cual esta ejemplificado en la figura 4.12. En lugar de *relaciones* se habla de tablas, en lugar de *atributos* se habla de columnas y en lugar de *tuplas* se habla de filas.

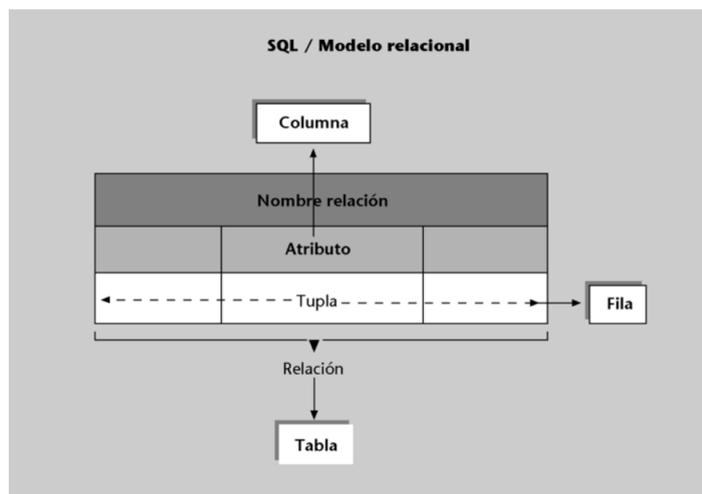


Fig. 4.12 SQL - Modelo relacional.

A pesar de que la nomenclatura utilizada los conceptos son los mismos. Con SQL se puede definir, manipular y controlar una base de datos relacional. A continuación se presenta el procedimiento para realizar acciones; **CREATE TABLE - crear tabla.**- es básico crear una tabla que contenga los datos de los productos de una empresa, ver fig. 4.13. **INSERT INTO - Insertar en.**- en una tabla creada con anterioridad se inserta un producto, ver fig. 4.14.

```
CREATE TABLE productos
(
  codigo_producto INTEGER,
  nombre_producto CHAR(20),
  tipo CHAR(20),
  descripcion CHAR(50),
  precio REAL,
  PRIMARY KEY (codigo_producto));
```

Nombre de la tabla
Nombre de las columnas y tipo
Clave primaria

Fig. 4.13 Insertar en la tabla producto valores

```
INSERT INTO productos
VALUES (1250, 'LENA', 'Mesa', 'Diseño Juan Pi. Año 1920.', 25000);
```

Valores de la fila

Fig. 4.14 Insertar en la tabla producto valores

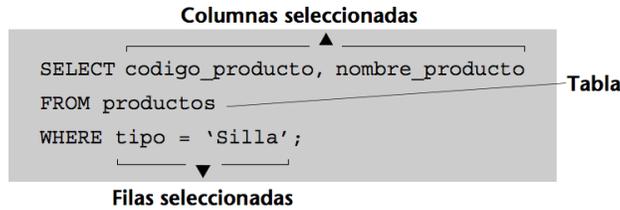


Fig. 4.15 Selecciona el producto silla de la tabla productos

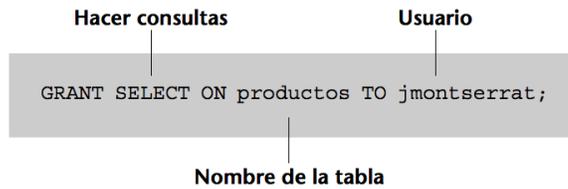


Fig. 4.16 Accede solo una persona a tabla productos.

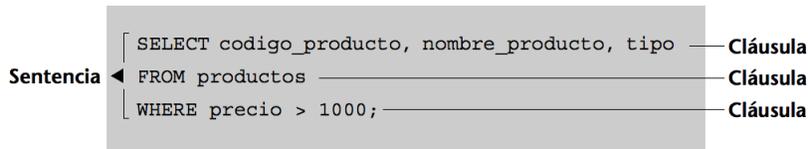


Fig. 4.17 Consulta de los productos que cuestan más de 1000.

SELECT - FROM - WHERE.- permite consultar los productos de una empresa que son sillas, ver fig. 4.15. **GRANT SELECT.**- permite dejar acceder a un vendedor a la información de la tabla productos, ver fig. 4.16. Las operaciones de SQL reciben el nombre de sentencias y están formadas por diferentes partes denominadas cláusulas. Un ejemplo de ello se muestra en la figura 4.17. En este ejemplo se hace una consulta para mostrar el código, el nombre y el tipo de los productos que cuestan más de 1000.

La idea de trabajar con un lenguaje de programación anfitrión que puede cobijar SQL como si fuese un huésped se conoce con el nombre de SQL hospedado. Para trabajar con SQL hospedado se requiere un compilador que separe las sentencias del lenguaje de programación de las del lenguaje de bases de datos. Una alternativa a esta forma de trabajar son las rutinas SQL/CLI*, SQL/Call-Level Interface, que resuelven el problema de acceder a SQL desde un lenguaje de programación y no necesitan compilador.

Lenguaje Java.- La principal característica de Java es la de ser un lenguaje compilado e interpretado. Todo programa en Java ha de compilarse y el código generado, bytecodes, es interpretado por una máquina virtual. De este modo se consigue la independencia de

la plataforma, el código compilado se ejecuta en máquinas virtuales que son dependientes de la plataforma.

Java es un lenguaje orientado a objetos de propósito general. Aunque Java comenzará a ser conocido como un lenguaje de programación de applets que se ejecutan en el entorno de un navegador web, se puede utilizar para construir cualquier tipo de proyecto. La sintaxis es parecida a la de C y C++. Pero Java no es una evolución de C ni un C++ mejorado. En el diseño de Java se prestó atención a la seguridad. Existen varios niveles de seguridad en Java, desde el ámbito del programador, hasta el ámbito de la ejecución en la máquina virtual.

Java realiza comprobación estricta de tipos durante la compilación, evitando problemas como el desbordamiento de la pila. Pero en la ejecución es donde se encuentra el método adecuado según el tipo de clase receptora del mensaje; aunque siempre es posible forzar un enlace estático declarando un método como final. Las instancias de una clase se crean con el operador new() y un recolector de basura se encarga de liberar la memoria ocupada por los objetos que ya no están referenciados. La máquina virtual de Java gestiona la memoria dinámicamente.

Una fuente común de errores en programación proviene del uso de punteros. En Java se han eliminado los punteros, el acceso a las instancias de clase se hace a través de referencias. Además, el programador está obligado a tratar las excepciones que se produzcan en tiempo de ejecución. Java define procedimientos para tratar estos errores. Java también posee mecanismos para garantizar la seguridad durante la ejecución comprobando, antes de ejecutar código, que este no viola ninguna restricción de seguridad del sistema donde se va a ejecutar.

También cuenta con un cargador de clases, de modo que todas las clases cargadas a través de la red tienen su propio espacio de nombres para no interferir con las clases locales. Otra característica de Java es que está preparado para la programación concurrente sin necesidad de utilizar biblioteca alguna. Finalmente, Java posee un gestor de seguridad con el que restringe el acceso a los recursos del sistema.

A menudo se argumenta que Java es un lenguaje lento porque debe interpretar los bytecodes a código nativo antes de ejecutar un método, pero gracias a la tecnología JIT, este proceso se lleva a cabo una única vez, después el código en código nativo se almacena de tal modo que está disponible para la siguiente vez que se llame.

5. Procedimiento y actividades realizadas (Desarrollo)

5.1 Funcionamiento

El sistema de monitoreo comienza a funcionar en el momento en que se enciende Raspberry pi. Arduino Ethernet manda los datos de las lecturas con el programa servidor.ino ubicado en anexos 9.3, el programa envía por puerto GET y puerto Ethernet cada 3 segundos mandando datos concatenados con los programas *iot.php* que se encuentra en el anexo 9.3, separa los datos de la cadena de caracteres que envía Ethernet y lo inserta en la tabla Valores de phpMyadmin, el programa permitirá llevar un historial del comportamiento de las variables analizadas, *iot1.php* que se encuentra en el anexo 9.3, el programa *iot1.php* recibe la cadena concatenada de temperatura, humedad del suelo y luminosidad, actualizando la ID 1 de la tabla de control1 en phpMyadmin. Esto sirve para comparar las lecturas con los datos de control. Figura 6.6 del capítulo de resultados dentro del servidor Raspberry pi.

El usuario observa en la página web de inicio la última lectura de los cinco sensores, la hora y dos links, figura 6.1 del capítulo de resultados. La programación de esta parte corresponde al código *index.php*. El primer enlace lleva a la página de control. En esta el usuario carga los valores a los que necesita que se mantenga la variable a controlar, figura 6.7 del capítulo de resultados. El segundo enlace permite abrir la ventana de visualización gráfica del historial. La programación de esta parte corresponde al código *grafica.php* el resultado se puede observar en la figura 6.5 del capítulo de resultados.

La página de control, figura 6.6 del capítulo de resultados, programada con el código *index.html*, sirve como interfaz para cargar dos valores a los que el usuario desea tener las variables a controlar, un límite superior y un límite inferior, la página utiliza los códigos *rangeslider.css*, *rangeslider.min.js*, *Responce.php* que se encuentran en el anexo 9.3. El código *index.html* muestra 6 controles deslizantes dentro de la página dándoles estilo con el código *rangeslider.css*, dentro de esta se encuentran seis funciones una por cada control deslizante, dentro de la página *index.html* se encuentran funciones *jquery* estas funcionan con el código *rangeslider.min.js*, dentro de esta función se utiliza el código *responce.php*, este código actualiza la tabla de control dentro de phpMyadmin.

Para enviar datos de control a Arduino uno se utiliza el software Eclipse, se carga el código *mian.java* que se encuentra en el anexo 9.3, el código envía cadenas de bits los cuales son los datos leídos de las bases de datos de control y control1, al correr el programa, figura 1.21 del capítulo de resultados, Arduino uno con el código *arduino_internet2.ino*, el código compara el bit de *addr* y *sync* si los dos números en binario son iguales

procede a leer los bits, cuando realiza la lectura el arduino compara que la lectura del sensor se encuentre entre los rangos de control cargados por el usuario siendo el caso en que la variable se encuentre fuera de dichos rangos se accionan los actuadores. El diagrama de flujo del código se muestra en la figura 5.1.

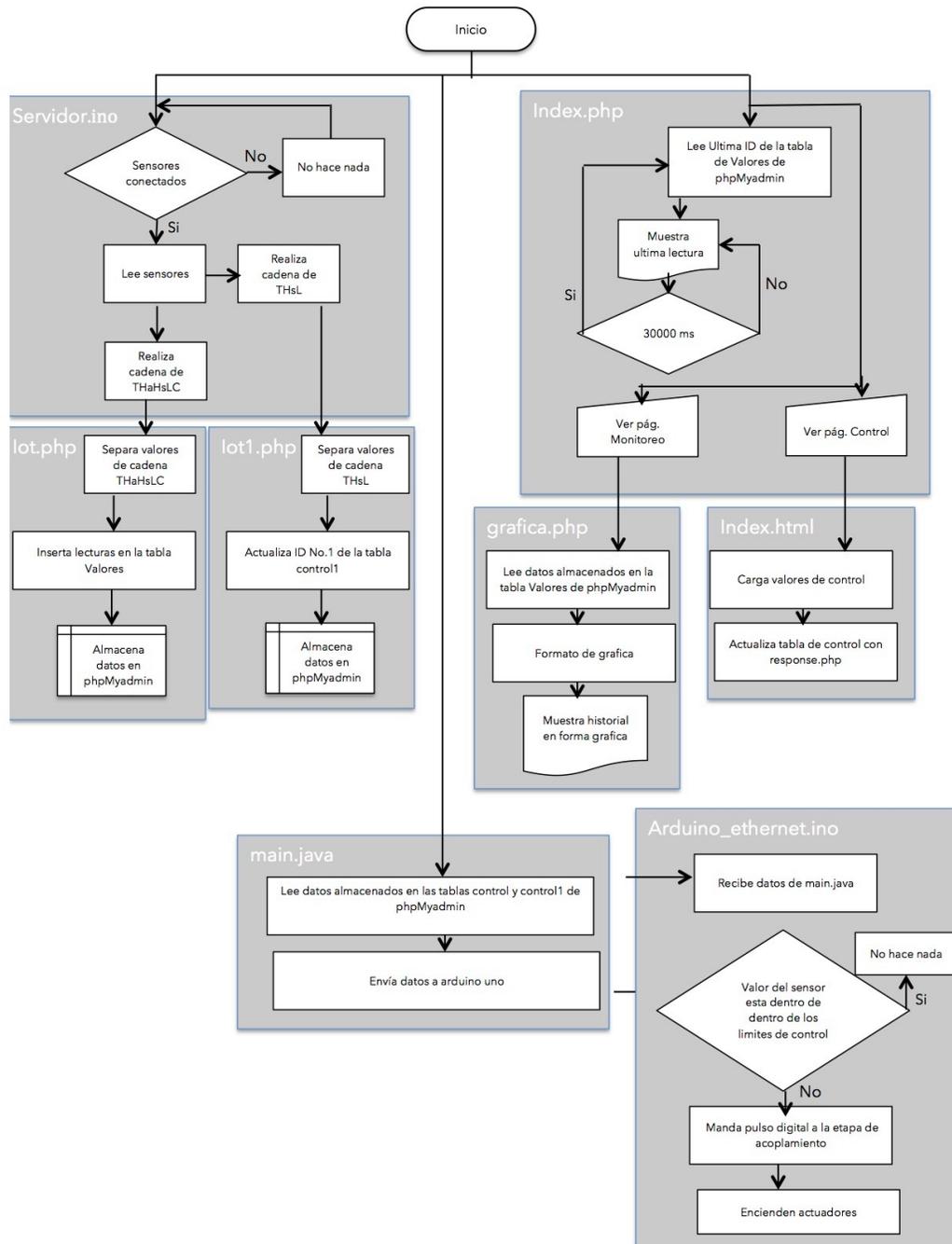


Fig. 5.1 Diagrama de flujo de códigos.

5.2 Caso de estudio; control de un invernadero a escala

Se diseñó un invernadero a escala con paredes de acrílico de 3 mm. Se seleccionó acrílico transparente para que la luz llegue a las plantas y puedan desarrollarse. Se colocaron sensores Moisture Sensor para medir la humedad del suelo. Se usó el sensor DTH11 para monitorear la temperatura y la humedad relativa que tiene el aire dentro del invernadero.

Se usó el sensor de luminosidad Precision light sensor 1127, el cual fue colocado en la pared frontal. La medición de la luminosidad se usó para encender a un conjunto de diodos LED variando la intensidad según los requerimientos de la planta. La idea es tener la luminosidad necesaria siempre que se requiera. Las imágenes del invernadero se encuentran en el Anexo 9.1.

5.3 Etapa de Monitoreo

Conexión de los sensores a Arduino Ethernet.- el sensor DTH11 se conecta al pin digital D2, el sensor de luminosidad al pin A0, el sensor de Humedad del suelo al pin A1 y el sensor de corriente al pin A2. En el diagrama electrónico, de la Figura 9.3.1 incluida en el Anexo 9.3, se muestran las conexiones de los sensores junto con las conexiones de las etapas de acoplamiento que se requieren. **Acoplamiento de sensores.**- la etapa de acoplamiento del sensor de corriente SCT-013-030 con Arduino se muestra en la figura 5.2.

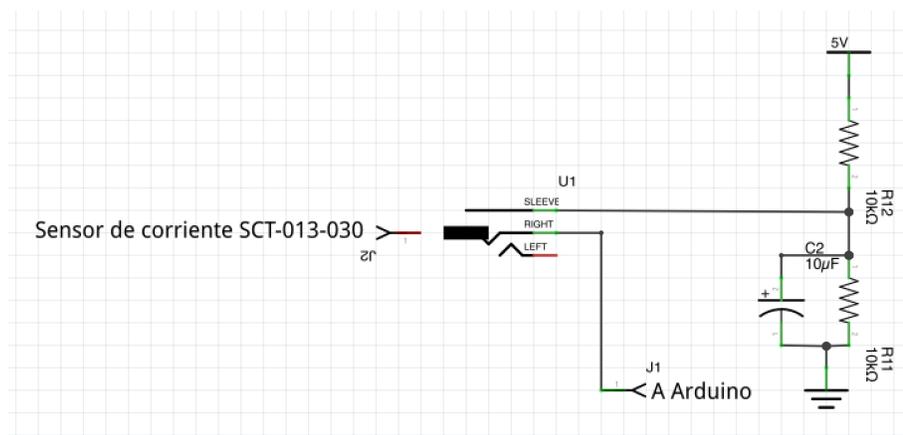


Fig. 5.2 Diagrama electrónico de acoplamiento del sensor de corriente SCT-013-030.

5.4 Etapa de Control

Para controlar la temperatura del aire se usó un ventilador inyector y un ventilador extractor de aire frío ó caliente según la necesidad. Una celda peltier se usó para enfriar ó

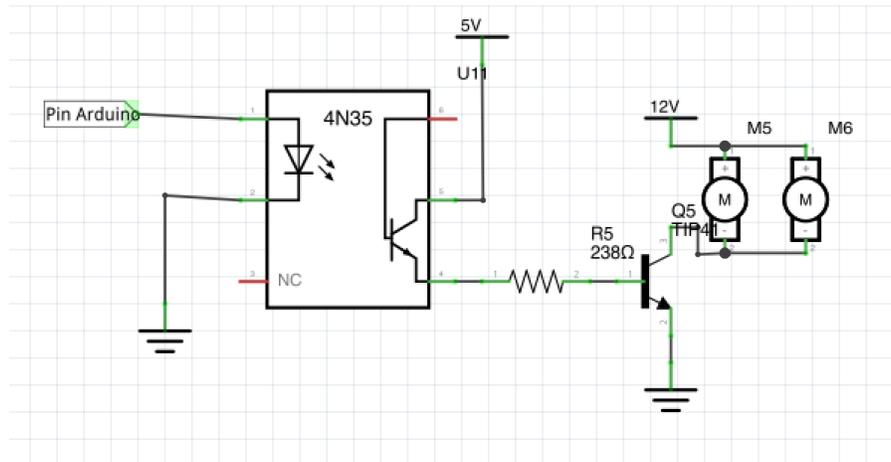


Fig. 5.5 Diagrama electrónico de acoplamiento para prendido de ventiladores inyectores/
extractores de aire

Para la resistencia de la base del TIP41 que activara los ventiladores de inyección y extracción los cálculos son los siguientes:

$$\begin{aligned} \beta &= 45 \\ I_c &= 800 \text{ mA} \\ I_B &= \frac{\beta}{I_c} \\ I_B &= \frac{800 \text{ mA}}{45} \\ I_B &= 17.77 \text{ mA} \\ R_B &= \frac{5 - 0.7}{17.77 \text{ mA}} \\ R_B &= 241.98 \Omega \approx 270 \Omega \end{aligned}$$

Fig.5.6 Resistencia de la base para la activación de los ventiladores.

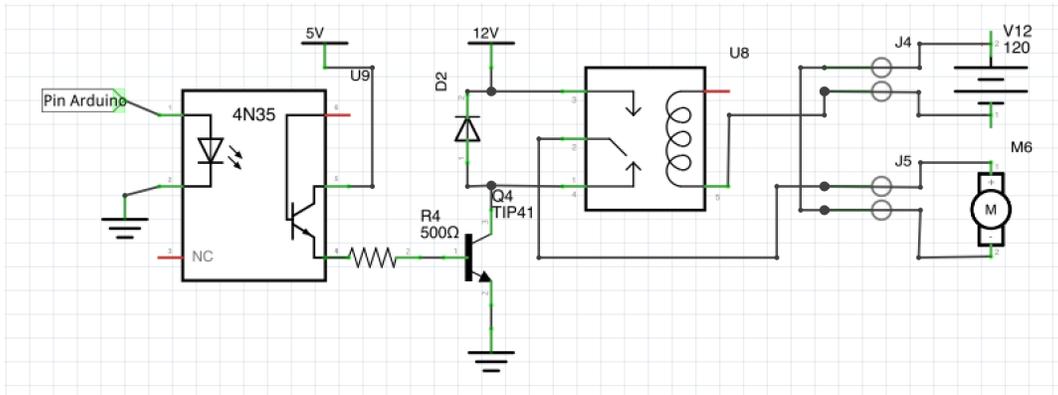


Fig.5.7 Diagrama electrónico de acoplamiento de la bomba de riego.

El sistema de control de la humedad del suelo utiliza una bomba de agua activada por un relevador. Cuando la humedad del suelo está debajo del límite inferior, el Arduino activa a la bomba por 10 segundos. Pasados los diez segundos desactiva a la bomba por cinco segundos y averigua si la humedad suministrada es suficiente. En caso contrario activa nuevamente realizando el mismo procedimiento. El diagrama eléctrico del acoplamiento de la bomba se presenta en la figura 5.7. Los cálculos de la resistencia de la base del TIP41 que activa al relevador se muestran en la figura 5.8.

$$\beta=93$$

$$I_c=800 \text{ mA}$$

$$I_B = \frac{\beta}{I_c}$$

$$I_B = \frac{800 \text{ mA}}{93}$$

$$I_B=8.60 \text{ mA}$$

$$R_B = \frac{5 - 0.7}{8.60 \text{ mA}}$$

$$R_B= 500\Omega$$

Fig.5.8 Resistencia de la base para la activación del relevador.

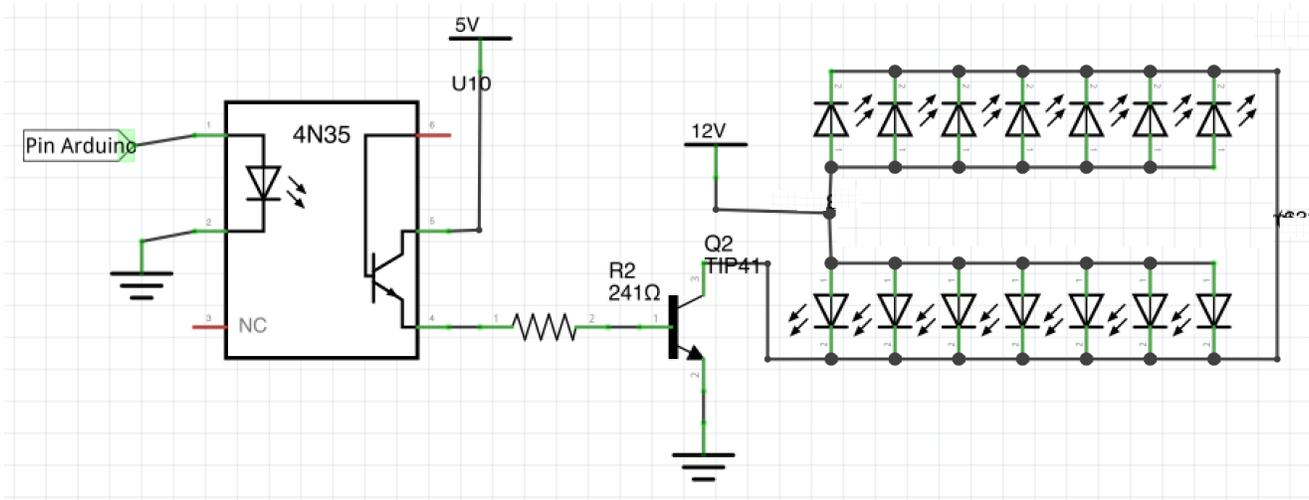


Fig.5.9 Diagrama electrónico del control de luminosidad con diodos LED.

$$\beta=45$$

$$I_c=800 \text{ mA}$$

$$I_B = \frac{\beta}{I_c}$$

$$I_B = \frac{800 \text{ mA}}{45}$$

$$I_B = 17.77 \text{ mA}$$

$$R_B = \frac{5 - 0.7}{17.77 \text{ mA}}$$

$$R_B = 241.98\Omega \approx 270\Omega$$

Fig.5.10 Resistencia de la base para la activación de los diodos LED

La variable luminosidad es controlada a través de tiras de diodos LED que prenden con la intensidad programada desde la página de control en el rango de 0 a 100%. El control de la intensidad se logra aprovechando una salida PWM del Arduino uno. El diagrama eléctrico de la etapa de acoplamiento para el control de las tiras de diodos se muestra en la figura 5.9. Los cálculos de la resistencia de la base del TIP41 que activa a la tira de diodos LED se muestran en la figura 5.10.

Las terminales 7,9,11,12 del Arduino sirven para mandar las señales de activación a los optoacopladores para las etapas de acoplamiento descritas en las figuras 5.3, 5.5, 5.7, 5.9. Arduino uno está conectado a Raspberry por medio del puerto serial. A través de ese puerto recibe los comandos de control desde la página Web.

6. Resultados

En la figura 6.1 se muestra la pagina de inicio del sistema de monitoreo, permitiendo ver los últimos valores leídos de los sensores, la hora de la lectura cada 30 segundos y dos links, uno para ir a la pagina de control y el siguiente para ver el historial de lecturas de los sensores.



Fig. 6.1 Pagina de inicio del sistema de monitoreo y control de procesos industriales basado en placas de desarrollo Arduino y Raspberri pi

En la figura 6.2 y 6.3 , se muestran los datos leídos de los sensores dentro del invernadero y almacenados en la tabla Valores de base de datos de phpMyadmin del servidor Pi.

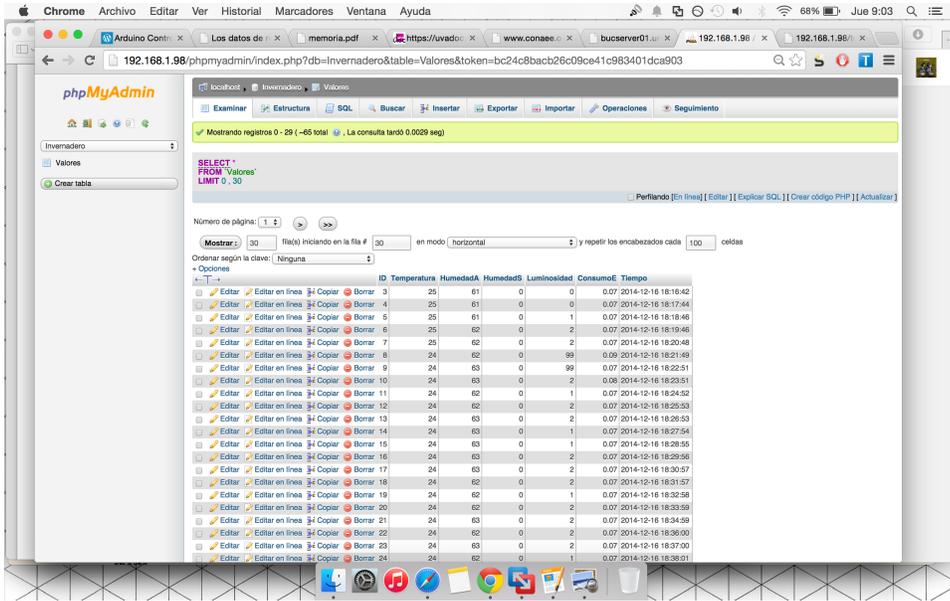


Fig. 6.2 Registro de phpMyadmin, inicio de la tabla Valores

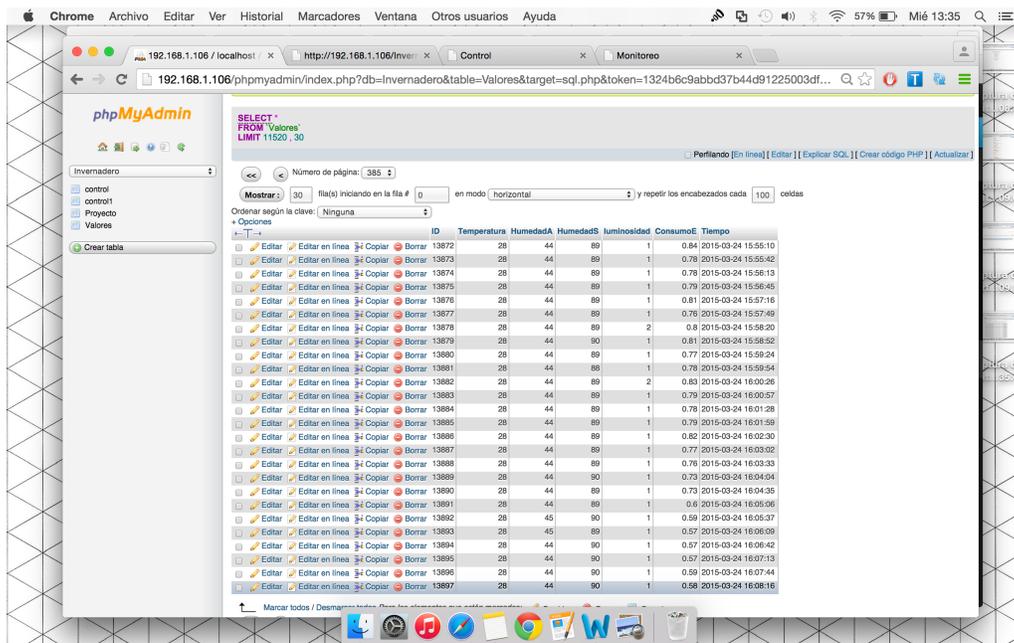


Fig. 6.3 Registro de phpMyadmin, final de la tabla de Valores

En la figura 6.4 muestra almacenamiento de la ultima lectura de los sensores en la tabla control1 actualizando la ID 1 cada medio segundo (30000 ms)

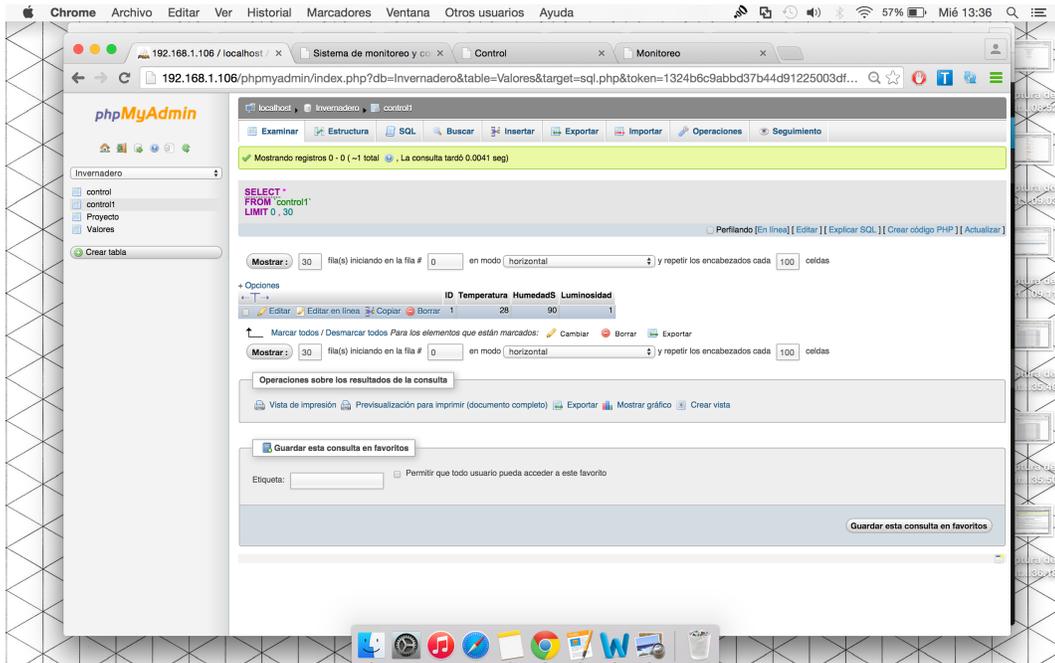


Fig. 6.4 Registro de phpMyadmin tabla de control1

En la figura 6.5 se observa el historial de los datos leídos por los sensores, aprovechando el registro de la tabla valores del servidor pi son mostrados en la pagina de monitoreo de forma gráfica.

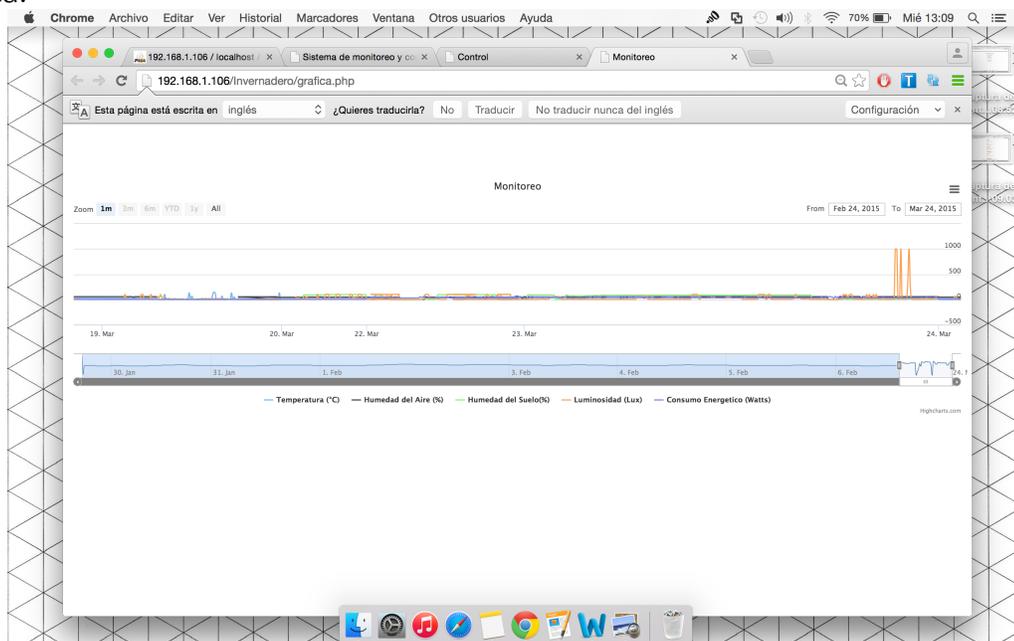


Fig. 6.5 Historial de lecturas

En la figura 6.6 observamos la pagina de control, lugar que nos permite programar los limites a los que desea el usuario se mantengan las variables temperatura, humedad del suelo y luminosidad, los datos se almacenan en la tabla control actualizando la ID1, figura 6.7, cuando se modifican los valores en la pagina de control.

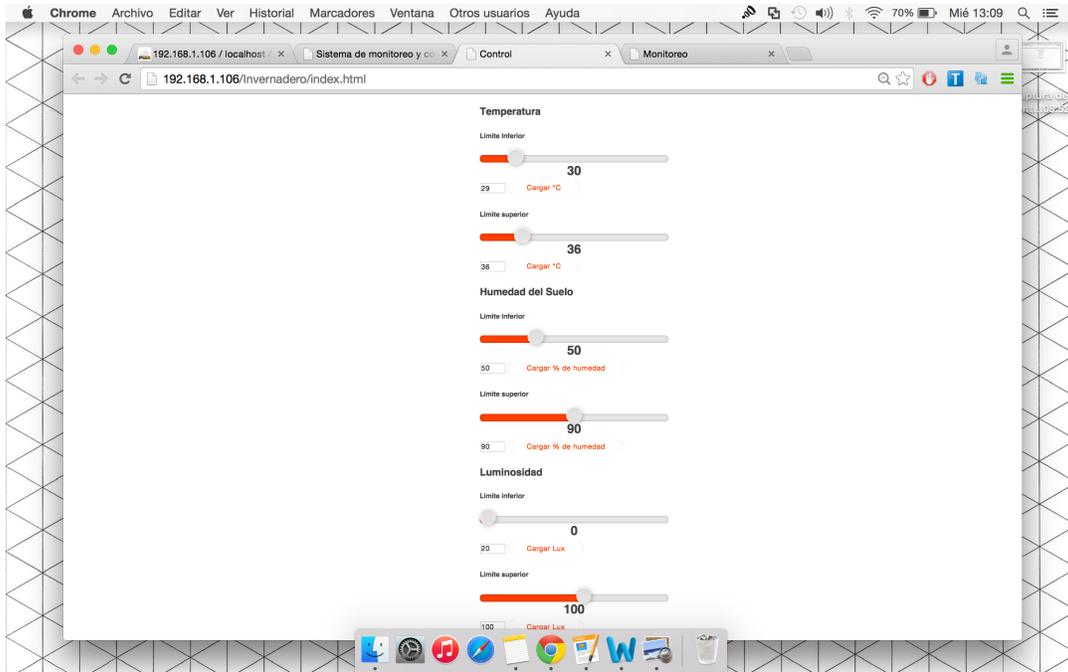


Figura 6.6 Pagina de control

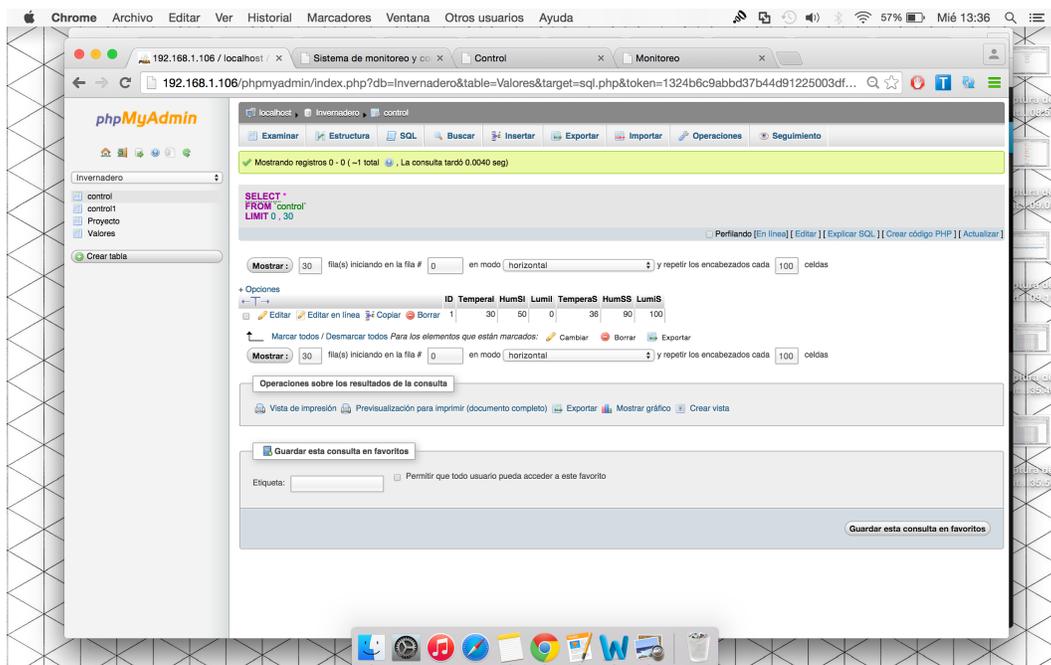


Figura 6.7 Tabla control de phpMyadmin

En la figura 6.8 se muestra el programa eclipse código main.java funcionando y mandando datos a arduino uno.

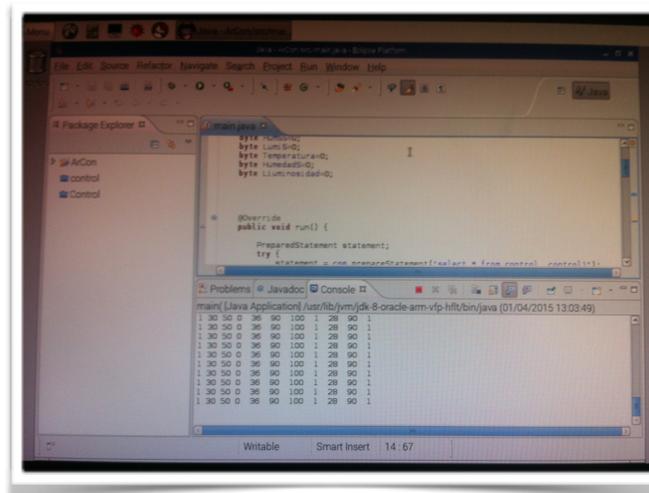


Figura 6.8 Envío de datos por puerto serial con software eclipse

En la figura 6.9 se muestra el invernadero funcionando, en el se observa los led's prendidos al 100 %, ya que el sensor estaba mandado 1 lux y en la pagina de control la variable luminosidad esta entre los rangos de 0 a 100 lo que nos dice que si menos luz recibe los led's prenderán a mayor intensidad, la cual será la señalada en el limite superior de la base de control.



Figura 6.9 Invernadero funcionando

7. Conclusiones

Existen sistemas de monitoreo y control de procesos industriales de altos costos. Derivado de ello en este proyecto se desarrollo un sistema genérico que para la demostración de su utilidad se aplico al monitoreo y control de las variables de control de un invernadero. De esta manera también se demostró como la tecnología con sus avances y desarrollo permite presentar propuestas nuevas e innovadoras de bajo costo.

El sistema presentado mejora el uso de los sistemas de control, teniendo una solución eficiente y a la medida que permite con el tiempo ser expandida. Esto significa que el sistema permite que un proceso industrial inicie pequeño y con el tiempo vaya creciendo y mejorando agregando sensores y actuadores. El sistema por ser genérico tiene la característica de adaptarse a cualquier proceso de control.

8. Referencias

- [1] Van Gigh , John P. Teoría General de Sistemas Aplicados. México. Trillas (1981)
- [2] <http://www.arduino.cc/>
- [3] <http://www.raspberrypi.org/>
- [4] <http://www.highcharts.com/products/highcharts>
- [5] http://www.phpmyadmin.net/home_page/index.php
- [6] <http://www.ibm.com/developerworks/ssa/library/os-ecov/>
- [7] <http://www.engineerathome.com/elektronika/log+arduino+data+to+a+web+server/>
34
- [8] <https://amtek.wordpress.com/2014/05/04/control-arduino-through-internet-on-windows/>
- [9] <http://geekytheory.com/php-mysql-highcharts-mostrar-grafica-dinamica-en-funcion-del-tiempo/>

- [10] <http://diymakers.es/raspberry-pi-como-servidor-web/>
- [11] <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- [12] <http://www.definicionabc.com/tecnologia/mysql.php>
- [13] <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>
- [14] http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02149.pdf
- [15] <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

9.1 Imágenes del invernadero

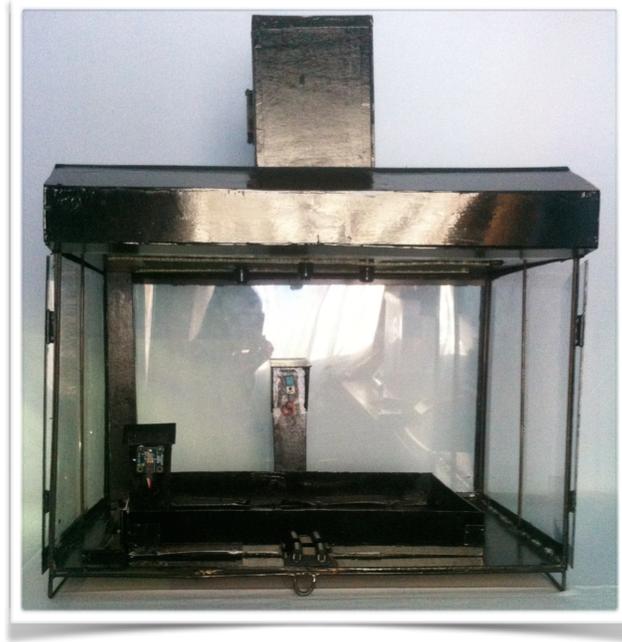
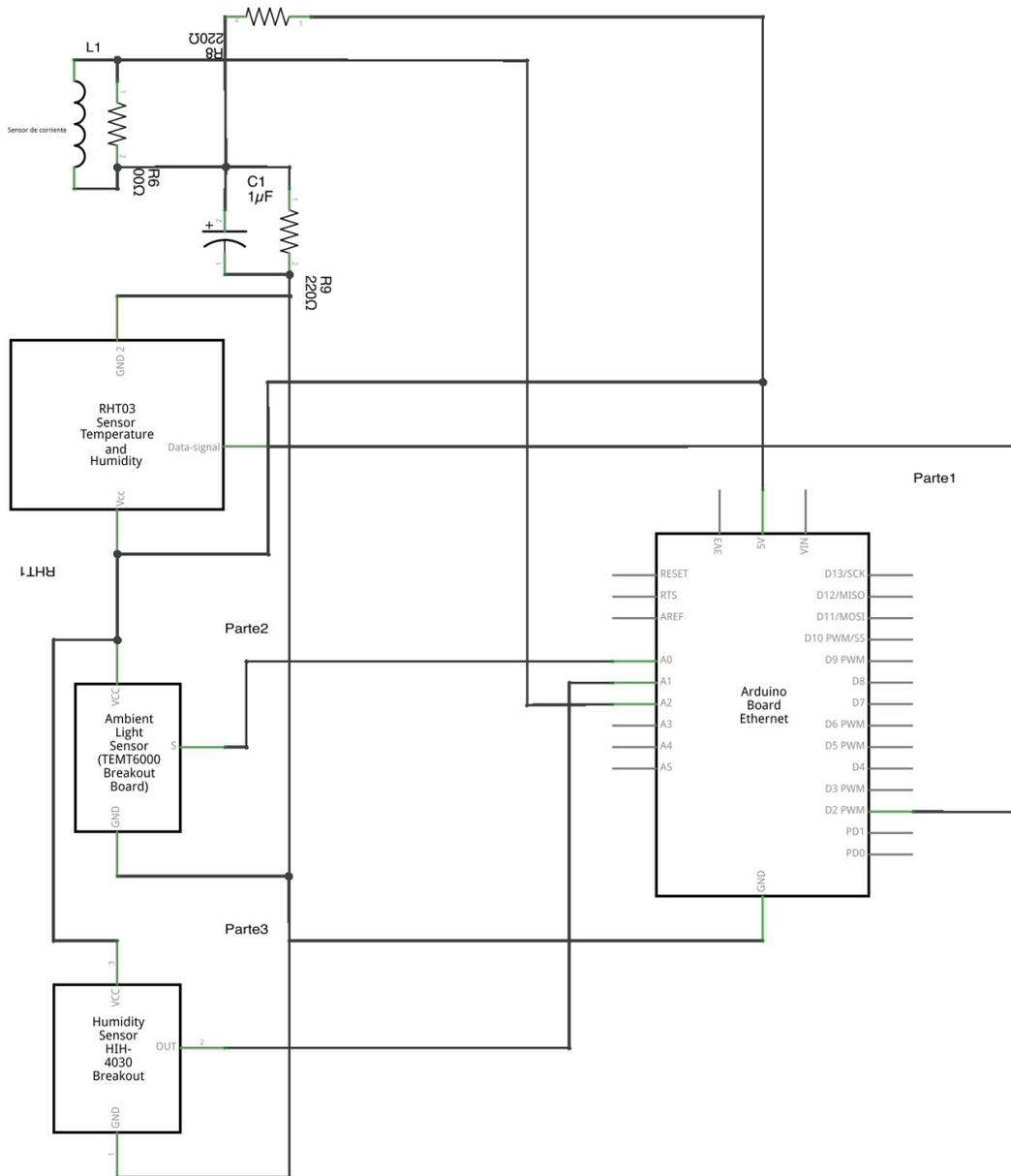


Fig.9.1.1 Invernadero vista frontal.



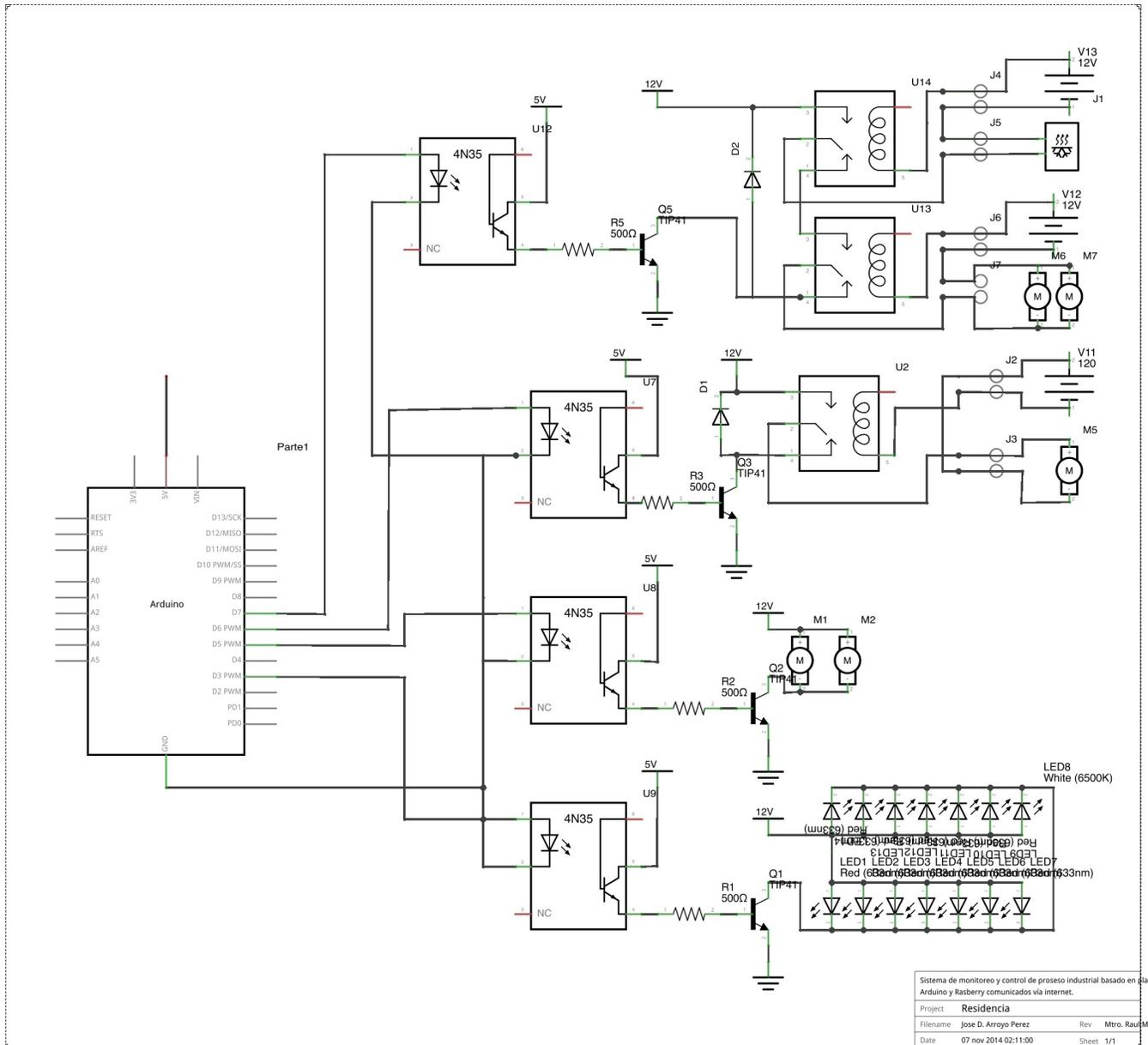
Fig.9.1.2 Invernadero vista posterior.

9.2 Diagramas eléctricos



fritzing

Fig.9.2.1 *Arduino Ethernet, acoplamiento de sensores.*



fritzing

Fig.9.2.2 Arduino uno, circuitos de acoplamiento y actuadores.

9.3 Códigos

- Arduino Ethernet

servidor.ino

```
#include <Ethernet.h>
#include <SPI.h>
#include "DHT.h"
#include "EmonLib.h"
#define DHTPIN 2 // what pin we're connected to
#define DHTTYPE DHT11 // DHT 11 (AM2302)
DHT dht(DHTPIN, DHTTYPE);
EnergyMonitor emon1;

// Configuracion del Ethernet Shield
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFF, 0xEE}; // Direccion MAC
byte ip[] = { 192,168,1,109 }; // Direccion IP del Arduino
byte server[] = { 192,168,1,103 }; // Direccion IP del servidor
EthernetClient client;
int lux;
int Valor;

void setup(void) {
  Ethernet.begin(mac, ip); // Inicializamos el Ethernet Shield
  delay(1000); // Esperamos 1 segundo de cortesia
  dht.begin();
  emon1.current(A2, 29);
}

void loop(void) {
  lux = analogRead(A0);
  lux = map (lux, 0, 1023, 0, 100);
  Valor = analogRead(A1);
  Valor = map (Valor, 0, 1023, 0, 100);
  double Irms = emon1.calcIrms(1480);
```

```

// Reading temperature or humidity takes about 250 milliseconds!
// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
float h = dht.readHumidity();
// Read temperature as Celsius
float t = dht.readTemperature();
// Read temperature as Fahrenheit
float f = dht.readTemperature(true);

//Display in Serial Monitor
Serial.println("Humedad Ambiental: ");
Serial.print(h);
Serial.print(" % ");
Serial.println("Temperatura Ambiental: ");
Serial.print(t);
Serial.print(" *C\t");
Serial.print(f);
Serial.println(" *F ");
Serial.println("Nivel de luminosidad ");
Serial.print(lux);
Serial.print(" % ");
Serial.println("Nivel de Humedad en el Suelo: ");
Serial.print(Valor);
Serial.print(" % ");
Serial.print("Corriente Suministrada: ");
Serial.print(Irms);
Serial.print("Amperios");
Serial.print("Consumo de Potencia: ");
Serial.print(Irms*120); // Apparent power
Serial.println("Watts ");

// Proceso de envio de muestras al servidor
Serial.println("Connecting...");
if (client.connect(server, 80)>0) { // Conexion con el servidor
  client.print("GET /Invernadero/iot.php?Temperatura="); // Enviamos los datos por GET

```

```

client.print(t);
client.print("&HumedadA="); // Enviamos los datos por GET
client.print(h);
client.print("&HumedadS="); // Enviamos los datos por GET
client.print(Valor);
client.print("&Luminosidad="); // Enviamos los datos por GET
client.print(lux);
client.print("&ConsumoE="); // Enviamos los datos por GET
client.print(Irms);
client.println(" HTTP/1.0");
client.println("User-Agent: Arduino 1.0");
client.println();
Serial.println("Conectado");
} else {
  Serial.println("Fallo en la conexion");
}
if (!client.connected()) {
  Serial.println("Disconnected!");
}
client.stop();
client.flush();
// Espero un minuto antes de tomar otra muestra

// Proceso de envio de muestras al servidor
Serial.println("Connecting...");
if (client.connect(server, 80)>0) { // Conexion con el servidor
  client.print("GET /Invernadero/iot1.php?Temperatura="); // Enviamos los datos por GET
  client.print(t);
  client.print("&HumedadS="); // Enviamos los datos por GET
  client.print(Valor);
  client.print("&Luminosidad="); // Enviamos los datos por GET
  client.print(lux);
  client.println(" HTTP/1.0");
  client.println("User-Agent: Arduino 1.0");
  client.println();
  Serial.println("Conectado");
}

```

```

} else {
  Serial.println("Fallo en la conexion");
}
if (!client.connected()){
  Serial.println("Disconnected!");
}
client.stop();
client.flush();
delay(30000); // Espero un minuto antes de tomar otra muestra
}

```

- Arduino uno

arduino_eternet2.ino

```

byte pre_sync = 0b01010101;
byte pre_addr = 0b01110111;
byte sync=0;
byte addr=0;
byte Temporal=0;
byte HumSI=0;
byte Lumil=0;
byte TemperaS=0;
byte HumSS=0;
byte LumiS=0;
byte Temperatura=0;
byte HumedadS=0;
byte Liuminosidad=0;

int lu_pin = 9;
int hum_pin = 7;
int temH_pin = 12;
int temF_pin = 11;

int ledState = LOW;      // ledState used to set the LED
long previousMillis = 0; // will store last time LED was updated

```

```

long intervalOn = 10000;    // 10 segundos ON
long intervalOff = 5000;   // 5 segundos OFF

void setup()
{
  pinMode(hum_pin, OUTPUT);
  pinMode(temF_pin, OUTPUT);
  pinMode(lu_pin, OUTPUT);
  pinMode(temH_pin, OUTPUT); // attaches the servo on pin 9 to the servo object
  Serial.begin(9600);
}
void loop()
{

  if (Serial.available() > 11) {
    // read the most recent byte (which will be from 0 to 255):
    sync = Serial.read();

    if(sync == pre_sync)
    {
      addr = Serial.read();
      if(addr == pre_addr)
      {
        Temperal = Serial.read();
        HumSI = Serial.read();
        Lumil = Serial.read();
        TemperaS = Serial.read();
        HumSS = Serial.read();
        LumiS = Serial.read();
        Temperatura = Serial.read();
        HumedadS = Serial.read();
        Liuminosidad = Serial.read();
      }
    }
  }

  //control de Temperatura//

```

```

int tmax=TemperaS;
int tmin=Temperal;

if(Temperatura >= tmax)
{
  digitalWrite(temF_pin,HIGH);
}
else
digitalWrite(temF_pin,LOW);

if(Temperatura <= tmin)
{
  digitalWrite(temH_pin,HIGH);
}
else
digitalWrite(temH_pin,LOW);

//-----
//control de luminosidad//
int lumi=Lumil;
int lumis=LumiS;
int LimS=10.24*LumiS;
int lux2=map(Liuminosidad,0,1000,lumi,lumis);
int nit=10.24*lux2;
int pwm=LimS-nit;

analogWrite(lu_pin,pwm/4);

//-----
//---Control de humedad---//

int HumSM=HumSI+20;

unsigned long currentMillis = millis();

```

```

if( HumedadS >= HumSI && HumedadS <= HumSM )
{

if (ledState == LOW)
{
if(currentMillis - previousMillis > intervalOff)
{
previousMillis = currentMillis;
ledState = HIGH;
}
}
if(ledState==HIGH)
{
if(currentMillis - previousMillis > intervalOn)
{
previousMillis = currentMillis;
ledState= LOW;
}
}
}

if(HumedadS >= HumSS)
{
ledState=LOW;
}
if(HumedadS <= HumSI)
{
ledState=HIGH;
}
digitalWrite(hum_pin, ledState);
}

}

```

- Raspberry

config.php

```
<?php
// config.php
// Credenciales
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "proyecto";
$dbname = "Invernadero";
// Conexión con la base de datos
$con = mysqli_connect($dbhost, $dbuser, $dbpass, $dbname);
?>
```

connect.php

```
<?php
$username = "root"; //mysql username
$password = "proyecto"; //mysql password
$hostname = "localhost"; //hostname
$databasename = "Invernadero"; //databasename

$con = mysqli_connect($hostname, $username, $password) or die('could not
connect to database');
mysql_select_db($databasename,$con);
?>
```

iot.php

```
<?php
// iot.php
// Importamos la configuración
require("config.php");

// Leemos los valores que nos llegan por GET
$Temperatura = mysqli_real_escape_string($con, $_GET['Temperatura']);
```

```

$HumedadA = mysqli_real_escape_string($con, $_GET['HumedadA']);
$HumedadS = mysqli_real_escape_string($con, $_GET['HumedadS']);
$Luminosidad = mysqli_real_escape_string($con, $_GET['Luminosidad']);
$ConsumoE = mysqli_real_escape_string($con, $_GET['ConsumoE']);

// Esta es la instrucción para insertar los valores
$query = "INSERT INTO
Valores(Temperatura,HumedadA,HumedadS,Luminosidad,ConsumoE) VALUES('".
$Temperatura."','".$HumedadA."','".$HumedadS."','".$Luminosidad."','".$ConsumoE."' )";

// Ejecutamos la instrucción
mysqli_query($con, $query);
mysqli_close($con);

?>

```

iot1.php

```

<?php
// iot.php
// Importamos la configuración
require("connect.php");

// Leemos los valores que nos llegan por GET
$Temperatura = filter_var($_GET["Temperatura"],FILTER_SANITIZE_STRING,
FILTER_FLAG_STRIP_HIGH);
$HumedadS = filter_var($_GET["HumedadS"],FILTER_SANITIZE_STRING,
FILTER_FLAG_STRIP_HIGH);
$Luminosidad = filter_var($_GET["Luminosidad"],FILTER_SANITIZE_STRING,
FILTER_FLAG_STRIP_HIGH);

// Esta es la instrucción para insertar los valore
// Ejecutamos la instrucción
mysql_query("UPDATE control1 SET Temperatura=$Temperatura, HumedadS=
$HumedadS, Luminosidad=$Luminosidad WHERE id='1'");
mysql_close($connecDB);

?>

```

index.htm

```
<!DOCTYPE html> <html> <head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Control</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="../../Invernadero/dist/rangeslider.css">
  <style>

    html {
      color: #404040;
      font-family: Helvetica, arial, sans-serif;
    }
    body {
      padding: 15px 5px;
      margin: 0 auto;
      max-width: 200px;
    }

    output {
      display: block;
      font-size: 20px;
      font-weight: bold;
      text-align: center;
      margin: 3px 0;
      width: 100%;
    }
  </style>
</head>
<body>

  <div class="wrapper">
    <div id="js-example-change-value">
      <h4>Temperatura</h4>
      <div>
```

```

<h6>Limite Inferior</h6>
<input type="range" min="0" max="180" value="50" data-rangeslider>
<output></output>
    <input type="number" value="10" min="0" max="180"><button class="flat
up">Cargar °C</button>
</div>
<div>
    <h6>Limite superior </h6>
    <input type="range" min="0" max="180" value="50" data-rangeslider2>
    <output></output>
    <input type="number" value="10" min="0" max="180"><button class="flat
up">Cargar °C</button>
</div>
</div>
<div id="js-example-change-value">
    <h4>Humedad del Suelo</h4>
    <div>
        <h6>Limite Inferior</h6>
        <input type="range" min="0" max="180" value="50" data-rangeslider1>
        <output></output>
        <input type="number" value="10" min="0" max="180"><button class="flat
up">Cargar % de humedad</button>
    </div>
    <div>
        <h6>Limite superior </h6>
        <input type="range" min="0" max="180" value="50" data-rangeslider3>
        <output></output>
        <input type="number" value="10" min="0" max="180"><button class="flat
up">Cargar % de humedad</button>
    </div>
</div>
<div id="js-example-change-value">
    <h4>Luminosidad</h4>
    <div>
        <h6>Limite inferior</h6>
        <input type="range" min="0" max="180" value="50" data-rangeslider0>
        <output></output>

```

```

        <input type="number" value="10" min="0" max="180"><button class="flat
up">Cargar Lux</button>
    </div>
    <div>
        <h6>Limite superior</h6>
        <input type="range" min="0" max="180" value="50" data-rangeslider4>
        <output></output>
        <input type="number" value="10" min="0" max="180"><button class="flat
up">Cargar Lux</button>
    </div>
</div>

```

```

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.0/jquery.min.js"></script>
<script src="../../Invernadero/dist/rangeslider.min.js"></script>
<script>

```

```

$(function () {

```

```

    var $document = $(document),
        selector = '[data-rangeslider]',
        $input = $(selector);

```

```

    $document.on('change', selector, function(e) {
        var value = e.target.value,
            output = e.target.parentNode.getElementsByTagName('output')[0];
        output.innerHTML = value;
    });

```

```

    var myData = 'column_name=' + 'Temperal' + '&' + 'value='+ value;

```

```

    jQuery.ajax({
        type: "POST", // HTTP method POST or GET
        url: "response.php", //Where to make Ajax calls
        dataType:"text", // Data type, HTML, json etc.
        data:myData, //post variables
        success:function(){

```

```
},  
error:function (xhr, ajaxOptions, thrownError){  
    alert(thrownError); //throw any errors  
}  
});
```

```
});
```

```
$input.rangeslider{
```

```
    // Deactivate the feature detection  
    polyfill: false,
```

```
    // Callback function  
    onInit: function() {},
```

```
    // Callback function  
    onSlide: function(position, value) {},
```

```
    // Callback function  
    onSlideEnd: function(position) {}
```

```
});
```

```
    // Example functionality to demonstrate programmatic value changes  
    $document.on('click', '#js-example-change-value button', function(e) {  
        var $inputRange = $('#input[type="range"]', e.target.parentNode),  
            value = $('#input[type="number"]', e.target.parentNode)[0].value;
```

```
        $inputRange.val(value).change();
```

```
    });
```

```
});
```

```

$(function(){

    var $document = $(document),
        selector = '[data-rangeslider0]',
        $input = $(selector);

    $document.on('change', selector, function(e){
        var value = e.target.value,
            output = e.target.parentNode.getElementsByTagName('output')[0];
        output.innerHTML = value;

        var myData = 'column_name=' + 'Lumil' + '&' + 'value='+ value;

        jQuery.ajax({
            type: "POST", // HTTP method POST or GET
            url: "response.php", //Where to make Ajax calls
            dataType:"text", // Data type, HTML, json etc.
            data:myData, //post variables
            success:function(){
            },
            error:function (xhr, ajaxOptions, thrownError){
                alert(thrownError); //throw any errors
            }
        });

    });

    $input.rangeslider({

        // Deactivate the feature detection
        polyfill: false,

        // Callback function
        onInit: function() {},
    });

```

```

// Callback function
onSlide: function(position, value) {},

// Callback function
onSlideEnd: function(position) {}
});

// Example functionality to demonstrate programmatic value changes
$(document.on('click', '#js-example-change-value button', function(e) {
    var $inputRange = $('input[type="range"]', e.target.parentNode),
        value = $('input[type="number"]', e.target.parentNode)[0].value;

    $inputRange.val(value).change();
}));

});

$(function() {

    var $document = $(document),
        selector = '[data-rangeslider1]',
        $input = $(selector);

    $document.on('change', selector, function(e) {
        var value = e.target.value,
            output = e.target.parentNode.getElementsByTagName('output')[0];
        output.innerHTML = value;

    var myData = 'column_name=' + 'HumSI' + '&' + 'value=' + value;

    jQuery.ajax({
        type: "POST", // HTTP method POST or GET
        url: "response.php", //Where to make Ajax calls
        dataType:"text", // Data type, HTML, json etc.
        data:myData, //post variables
    });
});

```

```

success:function(){
},
error:function (xhr, ajaxOptions, thrownError){
    alert(thrownError); //throw any errors
}
});

```

```
});
```

```
$input.rangeslider({
```

```

    // Deactivate the feature detection
    polyfill: false,

```

```

    // Callback function
    onInit: function() {},

```

```

    // Callback function
    onSlide: function(position, value) {},

```

```

    // Callback function
    onSlideEnd: function(position) {}

```

```
});
```

```

// Example functionality to demonstrate programmatic value changes
$document.on('click', '#js-example-change-value button', function(e) {
    var $inputRange = $('input[type="range"]', e.target.parentNode),
        value = $('input[type="number"]', e.target.parentNode)[0].value;

```

```

    $inputRange.val(value).change();

```

```
});
```

```
});
```

```
$(function () {
```

```

var $document = $(document),
    selector = '[data-rangeslider2]',
    $input = $(selector);

$document.on('change', selector, function(e) {
    var value = e.target.value,
        output = e.target.parentNode.getElementsByTagName('output')[0];
    output.innerHTML = value;

    var myData = 'column_name=' + 'TemperaS' + '&' + 'value='+ value;

    jQuery.ajax({
        type: "POST", // HTTP method POST or GET
        url: "response.php", //Where to make Ajax calls
        dataType:"text", // Data type, HTML, json etc.
        data:myData, //post variables
        success:function(){
        },
        error:function (xhr, ajaxOptions, thrownError){
            alert(thrownError); //throw any errors
        }
    });

});

$input.rangeslider({

    // Deactivate the feature detection
    polyfill: false,

    // Callback function
    onInit: function() {},

    // Callback function

```

```

    onSlide: function(position, value) {},

    // Callback function
    onSlideEnd: function(position) {}
});

// Example functionality to demonstrate programmatic value changes
$(document.on('click', '#js-example-change-value button', function(e) {
    var $inputRange = $('input[type="range"]', e.target.parentNode),
        value = $('input[type="number"]', e.target.parentNode)[0].value;

    $inputRange.val(value).change();
}));

});

$(function () {

    var $document = $(document),
        selector = '[data-rangeslider3]',
        $input = $(selector);

    $document.on('change', selector, function(e) {
        var value = e.target.value,
            output = e.target.parentNode.getElementsByTagName('output')[0];
        output.innerHTML = value;

        var myData = 'column_name=' + 'HumSS' + '&' + 'value='+ value;

        jQuery.ajax({
            type: "POST", // HTTP method POST or GET
            url: "response.php", //Where to make Ajax calls
            dataType:"text", // Data type, HTML, json etc.
            data:myData, //post variables
            success:function(){
            },
        },

```

```
error:function (xhr, ajaxOptions, thrownError){
    alert(thrownError); //throw any errors
}
});
```

```
});
```

```
$input.rangeslider{
```

```
    // Deactivate the feature detection
    polyfill: false,
```

```
    // Callback function
    onInit: function() {},
```

```
    // Callback function
    onSlide: function(position, value) {},
```

```
    // Callback function
    onSlideEnd: function(position) {}
```

```
});
```

```
    // Example functionality to demonstrate programmatic value changes
    $document.on('click', '#js-example-change-value button', function(e) {
        var $inputRange = $('input[type="range"]', e.target.parentNode),
            value = $('input[type="number"]', e.target.parentNode)[0].value;
```

```
        $inputRange.val(value).change();
```

```
    });
```

```
});
```

```
$(function () {
```

```
    var $document = $(document),
```

```

    selector = '[data-rangeslider4]',
    $input = $(selector);

    $document.on('change', selector, function(e) {
        var value = e.target.value,
            output = e.target.parentNode.getElementsByTagName('output')[0];
        output.innerHTML = value;

        var myData = 'column_name=' + 'LumiS' + '&' + 'value='+ value;

        jQuery.ajax({
            type: "POST", // HTTP method POST or GET
            url: "response.php", //Where to make Ajax calls
            dataType:"text", // Data type, HTML, json etc.
            data:myData, //post variables
            success:function(){
            },
            error:function (xhr, ajaxOptions, thrownError){
                alert(thrownError); //throw any errors
            }
        });

    });

    $input.rangeslider({

        // Deactivate the feature detection
        polyfill: false,

        // Callback function
        onInit: function() {},

        // Callback function
        onSlide: function(position, value) {},
    });

```

```

    // Callback function
    onSlideEnd: function(position) {}
  });

  // Example functionality to demonstrate programmatic value changes
  $document.on('click', '#js-example-change-value button', function(e) {
    var $inputRange = $('input[type="range"]', e.target.parentNode),
        value = $('input[type="number"]', e.target.parentNode)[0].value;

    $inputRange.val(value).change();
  });

});

</script>

</body>
</html>

```

rangeslider.css

```

.wrapper {
  width: 300px;
  margin: 000px auto 0;
}

.rangeslider,
.rangeslider__fill {
  background: #e6e6e6;
  border: none;
  display: block;
  height: 12px;
  width: 100%;
  -webkit-box-shadow: 0px 2px 2px rgba(255, 255, 255, 0.25), inset 0px 1px 3px rgba(0, 0, 0, 0.3);
}

```

```
-moz-box-shadow: 0px 2px 2px rgba(255, 255, 255, 0.25), inset 0px 1px 3px rgba(0, 0, 0, 0.3);
box-shadow: 0px 2px 2px rgba(255, 255, 255, 0.25), inset 0px 1px 3px rgba(0, 0, 0, 0.3);
-webkit-border-radius: 10px;
-moz-border-radius: 10px;
-ms-border-radius: 10px;
-o-border-radius: 10px;
border-radius: 10px;
}
```

```
.rangeslider {
  position: relative;
}
```

```
.rangeslider__fill {
  background: #FF4000;
  position: absolute;
  top: 0;
}
```

```
.rangeslider__handle {
  background: white;
  border: 0px solid #ccc;
  cursor: pointer;
  display: block;
  width: 25px;
  height: 25px;
  position: absolute;
  top: -10px;
  background-image: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%, rgba(255, 255, 255, 0)), color-stop(100%, rgba(0, 0, 0, 0.1)));
  background-image: -webkit-linear-gradient(rgba(255, 255, 255, 0), rgba(0, 0, 0, 0.1));
  background-image: -moz-linear-gradient(rgba(255, 255, 255, 0), rgba(0, 0, 0, 0.1));
  background-image: -o-linear-gradient(rgba(255, 255, 255, 0), rgba(0, 0, 0, 0.1));
  background-image: linear-gradient(rgba(255, 255, 255, 0), rgba(0, 0, 0, 0.1));
  -webkit-box-shadow: 1px 1px 8px rgba(0, 0, 0, 0.3);
}
```

```

-moz-box-shadow: 1px 1px 8px rgba(0, 0, 0, 0.3);
box-shadow: 1px 1px 8px rgba(0, 0, 0, 0.3);
-webkit-border-radius: 20px;
-moz-border-radius: 20px;
-ms-border-radius: 20px;
-o-border-radius: 20px;
border-radius: 20px;
}
.rangeslider__handle:after {
content: "";
display: block;
width: 23px;
height: 23px;
margin: auto;
position: absolute;
top: 0;
right: 0;
bottom: 0;
left: 0;
background-image: -webkit-gradient(linear, 50% 0%, 50% 100%, color-stop(0%, rgba(0, 0, 0, 0.13)), color-stop(100%, rgba(255, 255, 255, 0)));
background-image: -webkit-linear-gradient(rgba(0, 0, 0, 0.13), rgba(255, 255, 255, 0));
background-image: -moz-linear-gradient(rgba(0, 0, 0, 0.13), rgba(255, 255, 255, 0));
background-image: -o-linear-gradient(rgba(0, 0, 0, 0.13), rgba(255, 255, 255, 0));
background-image: linear-gradient(rgba(0, 0, 0, 0.13), rgba(255, 255, 255, 0));
-webkit-border-radius: 20px;
-moz-border-radius: 20px;
-ms-border-radius: 20px;
-o-border-radius: 20px;
border-radius: 20px;
}

button {
margin: 4px;
}

.classic{

```

```
padding: 10px 30px;
}
```

```
.flat{
background:transparent;

color:#FF4000;
border:none;
padding:3px 30px;
border-radius:20px;
box-shadow:0 0 0 1px #FF4000 inset;
outline:none;
font-weight:normal;
-webkit-transition:all 200ms ease-out;
}
```

```
.flat:hover{
box-shadow:0 0 0 3px #FF4000 inset;
-webkit-transition:none;
}
```

```
.flat:active{
color:#FF4000;
background: #FF4000;
}
```

rangeslider.min.js

```
/*! rangeslider.js - v0.2.7 | (c) 2014 @andreruffert | MIT license | https://github.com/
andreruffert/rangeslider.js */
"use strict";!function(a){"function"==typeof define&&define.amd?
define(["jquery"],a):a(jQuery)}(function(a){function b(){var
a=document.createElement("input");return a.setAttribute("type","range"),"text"!==a.type}
function c(){var a=!1,b=b||{};return("ontouchstart"in window||
window.DocumentTouch&&document instanceof b)&&(a=!0),a}function d(a,b){var
c=Array.prototype.slice.call(arguments,2);return setTimeout(function(){return
a.apply(null,c)},b)}function e(a,b){return b=b||100,function(){if(!a.debounce){var
```

```

c=Array.prototype.slice.apply(arguments);a.lastReturnVal=a.apply(window,c),a.debounceTime=0;return clearTimeout(a.debounceTimeout),a.debounceTimeout=setTimeout(function(){a.debouncing=!1},b),a.lastReturnVal}}function f(b,c){if(this.$window=a(window),this.$document=a(document),this.$element=a(b),this.options=a.extend({},k,c),this._defaults=k,this._name=g,this.polyfill=this.options.polyfill,this.onInit=this.options.onInit,this.onSlide=this.options.onSlide,this.onSlideEnd=this.options.onSlideEnd,this.polyfill&&j)return!1;this.identifier="js-"+g+"-"+new Date,this.value=parseFloat(this.$element[0].value)||0,this.min=parseFloat(this.$element[0].getAttribute("min"))||0,this.max=parseFloat(this.$element[0].getAttribute("max"))||100,this.step=parseFloat(this.$element[0].getAttribute("step"))||1,this.$fill=a('<div class="'+this.options.fillClass+' />'),this.$handle=a('<div class="'+this.options.handleClass+' />'),this.$range=a('<div class="'+this.options.rangeClass+' id="'+this.identifier+' />').insertBefore(this.$element).prepend(this.$fill,this.$handle),this.$element.css({position:"absolute",width:"1px",height:"1px",overflow:"hidden",visibility:"hidden"}),this.handleDown=a.proxy(this.handleDown,this),this.handleMove=a.proxy(this.handleMove,this),this.handleEnd=a.proxy(this.handleEnd,this),this.init();var f=this;this.$window.on("resize."+g,e(function(){d(function(){f.update()},300)},20)),this.$document.on(this.options.startEvent,"#" +this.identifier,this.handleDown),this.$element.on("change."+g,function(a,b){if(!b||b.origin!==g){var c=a.target.value,d=f.getPositionFromValue(c);f.setPosition(d)}})}var g="rangeslider",h=[],i=c(),j=b(),k={polyfill:!0,rangeClass:"rangeslider",fillClass:"rangeslider_fill",handleClass:"rangeslider_handle",startEvent:(i?"touchstart":"mousedown")+ "." +g,moveEvent:(i?"touchmove":"mousemove")+ "." +g,endEvent:(i?"touchend":"mouseup")+ "." +g};f.prototype.init=function(){this.update(),this.onInit&&"function"==typeof this.onInit&&this.onInit(),f.prototype.update=function(){this.handleWidth=this.$handle[0].offsetWidth,this.rangeWidth=this.$range[0].offsetWidth,this.maxHandleX=this.rangeWidth-this.handleWidth,this.grabX=this.handleWidth/2,this.position=this.getPositionFromValue(this.value),this.setPosition(this.position)},f.prototype.handleDown=function(a){if(a.preventDefault(),this.$document.on(this.options.moveEvent,this.handleMove),this.$document.on(this.options.endEvent,this.handleEnd),!((" "+a.target.className+" ").replace(/[\n\t]/g," ").indexOf(this.options.handleClass)>-1)){var b=this.getRelativePosition(this.$range[0],a),c=this.getPositionFromNode(this.$handle[0])-this.getPositionFromNode(this.$range[0]);this.setPosition(b-this.grabX),b>=c&&b<c+this.handleWidth&&(this.grabX=b-c)},f.prototype.handleMove=function(a){a.preventDefault();var b=this.getRelativePosition(this.$range[0],a);this.setPosition(b-this.grabX)},f.prototype.handleEnd=function(a){a.preventDefault(),this.

```

```

$document.off(this.options.moveEvent,this.handleMove),this.
$document.off(this.options.endEvent,this.handleEnd);var b=this.getRelativePosition(this.
$range[0],a);this.onSlideEnd&&"function"==typeof this.onSlideEnd&&this.onSlideEnd(b-
this.grabX,this.value)},f.prototype.cap=function(a,b,c){return b>a?b:a>c?
c:a},f.prototype.setPosition=function(a){var b=this.cap(a,
0,this.maxHandleX),c=this.getValueFromPosition(b);1!
==this.step&&(b=this.getPositionFromValue(c),c=this.getValueFromPosition(b)/
this.step*this.step),this.$fill[0].style.width=b+this.grabX+"px",this.$handle[0].style.left=b
+"px",this.setValue(c),this.position=b,this.value=c,this.onSlide&&"function"==typeof
this.onSlide&&this.onSlide(b,c)},f.prototype.getPositionFromNode=function(a){for(var
b=0;null!=a;)b+=a.offsetLeft,a=a.offsetParent;return
b},f.prototype.getRelativePosition=function(a,b){return(b.pageX||b.originalEvent.clientX||
b.originalEvent.touches[0].clientX||b.currentPoint.x)-
this.getPositionFromNode(a)},f.prototype.getPositionFromValue=function(a){var b,c;return
b=(a-this.min)/(this.max-this.min),c=b*this.maxHandleX},f.prototype.getValueFromPosition=function(a){var
b,c;return b=a/this.maxHandleX,c=this.step*Math.ceil((b*(this.max-this.min)+this.min)/
this.step),Number(c.toFixed(2))},f.prototype.setValue=function(a){this.
$element.val(a).trigger("change",{origin:g}),f.prototype.destroy=function(){this.
$document.off(this.options.startEvent,"#" + this.identifier,this.handleDown),this.
$element.off("." + g).removeAttr("style").removeData("plugin_" + g),this.$range&&this.
$range.length&&this.$range[0].parentNode.removeChild(this.
$range[0]),h.splice(h.indexOf(this.$element[0]),1),h.length||this.
$window.off("." + g),a.fn[g]=function(b){return this.each(function(){var
c=a(this),d=c.data("plugin_" + g);d||(c.data("plugin_" + g,d=new
f(this,b)),h.push(this)),"string"==typeof b&&d[b]()}});

```

response.php

```

<?php
//include database connection file
include_once("connect.php");

//sanitize post value, PHP filter FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH
$colname = filter_var($_POST["column_name"],FILTER_SANITIZE_STRING,
FILTER_FLAG_STRIP_HIGH);
$val = filter_var($_POST["value"],FILTER_SANITIZE_STRING, FILTER_FLAG_STRIP_HIGH);

// Insert sanitize string in record

```

```

if(mysql_query("UPDATE control SET $colname=$val WHERE id='1'"))
{
    mysql_close($conecDB);

}
else{
    //output error
    header('HTTP/1.1 500 Looks like mysql error, could not insert record!');
    exit();
}

```

?>

index.php

```

<Html>
<head>
<META HTTP-EQUIV = "REFRESH" CONTENT="30;URL= ../Invernadero/index.php">
</head>
<BODY>

```

<?php

```

function conectarBD(){
    $server = "localhost";
    $usuario = "root";
    $pass = "proyecto";
    $BD = "Invernadero";
    //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 conexion 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
    $conexion = conectarBD();
    //generamos la consulta
    if(!$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;
        $i++;
    }
    //Cerramos la base de datos
    desconectarBD($conexion);
    //devolvemos rawdata
    return $rawdata;
}

//Sentencia SQL

```

```
$sql = "SELECT Temperatura,HumedadA,HumedadS,Luminosidad,ConsumoE,Tiempo
from Valores ORDER BY id DESC Limit 1;";
//Array Multidimensional
$rowdata = getArraySQL($sql);
```

```
?>
```

```
<title>Sistema de monitoreo y control de proceso industrial basado en placas de desarrollo
arduino y raspberry comunicados via internet</title>
```

```
<div align=center>
```

```
<BR>
```

```
<BR>
```

```
<h2>Bienvenido al sistema de monitoreo y control de proceso industrial basado en placas
de desarrollo arduino y raspberry comunicados via internet</h2>
```

```
<br>
```

```
<h4>Valores de lecturas actualizados cada 30 segundos</h4>
```

```
<h6>Temperatura Actual: <?php for($i = 0 ;$i<count($rowdata);$i++){ ?> <?php echo
$rowdata[$i]["Temperatura"];?> <?php } ?> Â°C</h6>
```

```
<h6>Humerdad del Aire Actual: <?php for($i = 0 ;$i<count($rowdata);$i++){ ?> <?php
echo $rowdata[$i]["HumedadA"];?> <?php } ?> % Humedad relativa</h6>
```

```
<h6>Humerdad del Suelo Actual: <?php for($i = 0 ;$i<count($rowdata);$i++){ ?> <?php
echo $rowdata[$i]["HumedadS"];?> <?php } ?> % Humedad</h6>
```

```
<h6>Luminosidad Actual: <?php for($i = 0 ;$i<count($rowdata);$i++){ ?> <?php echo
$rowdata[$i]["Luminosidad"];?> <?php } ?> Lux</h6>
```

```
<h6>Corriente utilizada actualmente: <?php for($i = 0 ;$i<count($rowdata);$i++){ ?> <?
php echo $rowdata[$i]["ConsumoE"];?> <?php } ?> A</h6>
```

```
<h6>Hora de lectura: <?php for($i = 0 ;$i<count($rowdata);$i++){ ?> <?php echo
$rowdata[$i]["Tiempo"];?> <?php } ?> </h6>
```

```
<BR>
```

```
<BR>
```

```
<a href=" ../Invernadero/index.html" onclick="window.open(this.href);return false;">Ir a
cargar los valors para control</a>
```

```
<BR>
```

```
<BR>
```

```
<a href=" ../Invernadero/grafica.php" onclick="window.open(this.href);return false;">Ir a
ver el historial</a>
```

```
</div>
</html>
```

grafica.php

```
<?php
```

```
function conectarBD(){
    $server = "localhost";
    $usuario = "root";
    $pass = "proyecto";
    $BD = "Invernadero";
    //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
    $conexion = conectarBD();
    //generamos la consulta
    if(!$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;
        $i++;
    }
    //Cerramos la base de datos
    desconectarBD($conexion);
    //devolvemos rawdata
    return $rawdata;
}

//Sentencia SQL
$sql = "SELECT Temperatura,HumedadA,HumedadS,Luminosidad,ConsumoE,Tiempo
from Valores;";

//Array Multidimensional
$rawdata = getArraySQL($sql);

//Adaptar el tiempo
for($i=0;$i<count($rawdata);$i++){
    $time = $rawdata[$i]["Tiempo"];
    $date = new DateTime($time);
    $rawdata[$i]["Tiempo"]=$date->getTimestamp()*1000;
}

```

?>

<HTML>

<head>

<meta charset="utf-8">

<title>Monitoreo </title>

<!-- Latest compiled and minified JavaScript -->

<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 type='text/javascript'>

\$(function () {

\$(document).ready(function() {

Highcharts.setOptions({

global: {

useUTC: false

}

});

\$('#container').highcharts('StockChart', {

chart: {

type: 'spline',

animation: Highcharts.svg, // don't animate in old IE

marginRight: 10,

events: {

load: function(){

var series = this.series[0];

setInterval(function () {

var x = [<?php echo \$rawdata[\$i]["Tiempo"];?>];

var y = [<?php echo \$rawdata[\$i]["Temperatura"];?>];

series.addPoint([x, y], true, true);

}, 1000);

```

    }
  }
},
scrollbar: {
  barBackgroundColor: 'white',
  barBorderRadius: 7,
  barBorderWidth: 0,
  buttonBackgroundColor: 'gray',
  buttonBorderWhidth: 0,
  buttonArrowColor: 'white',
  buttonBorderRadius: 7,
  rifleColor: 'gray',
  trackBackgroundColor: 'gray',
  trackBorderWidth: 1,
  trackArrowColor: 'black',
  trackBorderRadius: 7
},
navigator: {
  maskInside: false,
  handles: {
    backgroundColor: 'gray',
    borderColor: 'white'
  }
},
plotOptions: {
  series: {
    allowPointSelect: true
  }
},
rangeSelector: {
  selected: 1
},
title: {
  text: 'Monitoreo'
},
xAxis: {

```

```

    type: 'datetime',
    tickPixelInterval: 150
  },
  legend: {
    enabled: true
  },
  exporting: {
    enabled: true
  },
  series: [{
    name: 'Temperatura (°C)',
    data: (function() {
      var data = [];
      <?php
        for($i = 0 ;$i<count($rawdata);$i++){
          ?>
          data.push([<?php echo $rawdata[$i]["Tiempo"];?>,<?php echo $rawdata[$i]
["Temperatura"];?>]);
          <?php } ?>
      return data;
    })()
  },{
    name: 'Humedad del Aire (%)',
    data: (function() {
      var data = [];
      <?php
        for($i = 0 ;$i<count($rawdata);$i++){
          ?>
          data.push([<?php echo $rawdata[$i]["Tiempo"];?>,<?php echo $rawdata[$i]
["HumedadA"];?>]);
          <?php } ?>
      return data;
    })()
  },{
    name: 'Humedad del Suelo(%)',
    data: (function() {

```

```

        var data = [];
    <?php
        for($i = 0;$i<count($rawdata);$i++){
    ?>
            data.push([<?php echo $rawdata[$i]["Tiempo"];?>,<?php echo $rawdata[$i]
["HumedadS"];?>]);
        <?php } ?>
    return data;
    })()
}
name: 'Luminosidad (Lux)',
    data: (function() {
        var data = [];
    <?php
        for($i = 0;$i<count($rawdata);$i++){
    ?>
            data.push([<?php echo $rawdata[$i]["Tiempo"];?>,<?php echo $rawdata[$i]
["Luminosidad"];?>]);
        <?php } ?>
    return data;
    })()
}
name: 'Consumo Energetico (Watts)',
    data: (function() {
        var data = [];
    <?php
        for($i = 0;$i<count($rawdata);$i++){
    ?>
            data.push([<?php echo $rawdata[$i]["Tiempo"];?>,<?php echo $rawdata[$i]
["ConsumoE"];?>]);
        <?php } ?>
    return data;
    })()
}
});
});

```

```
});
```

```
</script>
```

```
</head>
```

```
<BODY>
```

```
<BR>
```

```
<div id="container">
```

```
</div>
```

```
</html>
```

main.java

```
import java.util.Timer;
```

```
import java.util.TimerTask;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
import java.sql.PreparedStatement;
```

```
import java.sql.ResultSet;
```

```
import java.sql.SQLException;
```

```
import jssc.SerialPort;
```

```
import jssc.SerialPortException;
```

```
public class main extends TimerTask{
```

```
    static SerialPort serialPort = new SerialPort("/dev/ttyACM0");
```

```
    static Connection con = null;
```

```

byte sync = 0b01010101;
byte addr = 0b01110111;
byte Temporal=0;
byte HumSI=0;
byte Lumil=0;
byte TemperaS=0;
byte HumSS=0;
byte LumiS=0;
byte Temperatura=0;
byte HumedadS=0;
byte Luminosidad=0;
public void run() {
    PreparedStatement statement;
    try {
        statement = con.prepareStatement("select * from control, control1");
        ResultSet result = statement.executeQuery();
        while(result.next())
        {
            System.out.println(result.getString(1) + " " + result.getString(2) + " " +
result.getString(3) + " " + result.getString(4)+ " " + " " + result.getString(5)+ " " + " " +
result.getString(6)+ " " + " " + result.getString(7)+ " " + " " + result.getString(8)+ " " + " " +
result.getString(9)+ " " + " " + result.getString(10)+ " " + " " + result.getString(11));

            Temporal = (byte) Integer.parseInt(result.getString(2));
            HumSI = (byte) Integer.parseInt(result.getString(3));
            Lumil = (byte) Integer.parseInt(result.getString(4));
            TemperaS = (byte) Integer.parseInt(result.getString(5));

```

```

HumSS = (byte) Integer.parseInt(result.getString(6));
LumiS = (byte) Integer.parseInt(result.getString(7));
Temperatura = (byte) Integer.parseInt(result.getString(9));
HumedadS = (byte) Integer.parseInt(result.getString(10));
Luminosidad = (byte) Integer.parseInt(result.getString(11));
}
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
try {
    serialPort.writeByte(sync);
    serialPort.writeByte(addr);
    serialPort.writeByte(Temperal);
    serialPort.writeByte(HumSI);
    serialPort.writeByte(Lumil);
    serialPort.writeByte(TemperaS);
    serialPort.writeByte(HumSS);
    serialPort.writeByte(LumiS);
    serialPort.writeByte(Temperatura);
    serialPort.writeByte(HumedadS);
    serialPort.writeByte(Luminosidad);
}
catch (SerialPortException ex) {
    System.out.println(ex);
}

```

```

    }
}

public static void main(String[] args) throws Exception {
    // TODO Auto-generated method stub

    Class.forName("com.mysql.jdbc.Driver");

    con = DriverManager.getConnection("jdbc:mysql://localhost:3306/
Invernadero","root","proyecto");

    try {

        serialPort.openPort();//Open serial port

        serialPort.setParams(SerialPort.BAUDRATE_9600,
            SerialPort.DATABITS_8,
            SerialPort.STOPBITS_1,
            SerialPort.PARITY_NONE);//Set params. Also you can set params by this
string: serialPort.setParams(9600, 8, 1, 0);
    }

    catch (SerialPortException ex) {

        System.out.println(ex);
    }

    TimerTask timerTask = new main();

    Timer timer = new Timer(true);

    timer.scheduleAtFixedRate(timerTask, 0, 1000);

    try {

        Thread.sleep(278400000);
    } catch (InterruptedException e) {

        e.printStackTrace();
    }
}

```

```

timer.cancel();

serialPort.closePort();//Close serial port

con.close();

}

}

```

9.4 Instalación de Raspbian en tarjeta de memoria SD

A. Descargar la imagen del sistema Operativo desde <http://www.raspberrypi.org/downloads/>

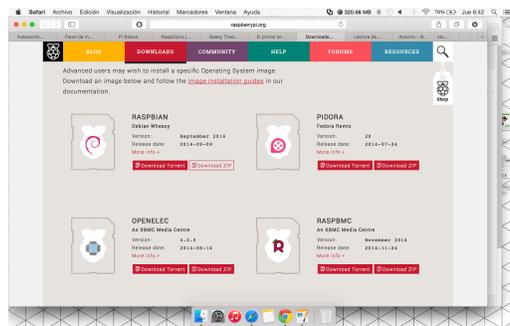


Fig. 9.4.1 Sección de descargas de raspberrypi.org

B. Grabamos la imagen en la sd, en mi caso con RPi-sd card builder v1.2.app

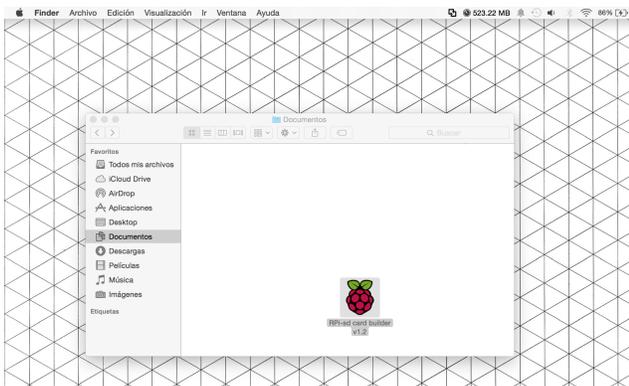


Fig. 9.4.2 Programa para cargar raspbian a tarjeta SD

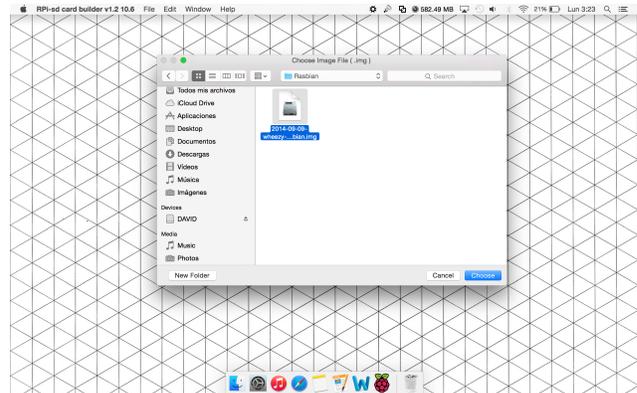


Fig. 9.4.3 Imagen del sistema operativo Raspbian

C. Insertar la tarjeta sd en la paca de Raspberry pi y conectamos

D. Configuramos la Raspberry pi

Nos aparecerá el siguiente menú. A continuación explicare en que orden hay que ejecutar y que acciones realizan las opciones mas importantes.

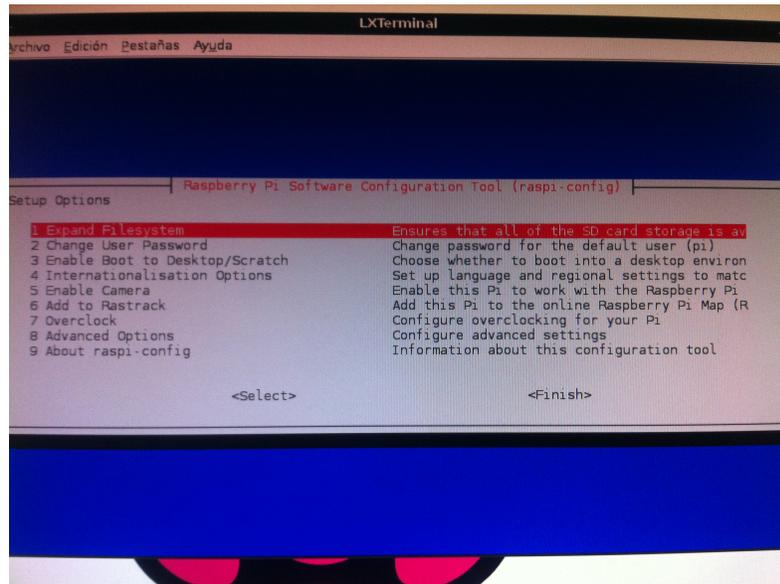


Fig. 9.4.4 Panel de configuración de Raspbian

I - Expandir el sistema: Esto se encarga de que el sistema de archivos de Raspbian ocupe todo el espacio de la tarjeta SD (si se instala con NOOBS creo que no es necesario hacerlo).

II - Cambio de contraseña: Por defecto el usuario es pi y la contraseña raspberry, recomendable cambiarla.

III - Activar inicio en modo escritorio: Con esta opción podemos elegir que nuestra Raspberry Pi inicie en modo de escritorio directamente.

IV - Opciones de internacionalización: Con esto podemos configurar el idioma de nuestra Raspberry Pi, la localización del teclado que usamos, etc.

V - Activar cámara: Esto es para activar la cámara de Raspberry Pi (que hay que adquirir a parte).

VI - Añadir a Rastrack: Esto es para tema de clusterización, (ya hablaremos en próximos artículos).

VII - Overclock: Opciones de Overclocking, para subir la velocidad del microprocesador (con el correspondiente aumento de temperatura y la disminución del tiempo de vida que supone).

VIII - Opciones Avanzadas: Aquí podemos cambiar el nombre de la máquina; elegir el uso de memoria para la GPU; y entre otras opciones más, lo que me parece más importante resaltar, activar el servidor SSH para poder acceder a la Raspberry remotamente via SSH desde cualquier terminal.

IX - Información sobre raspi-config

E. Finalizamos la configuración y entramos en modo consola.

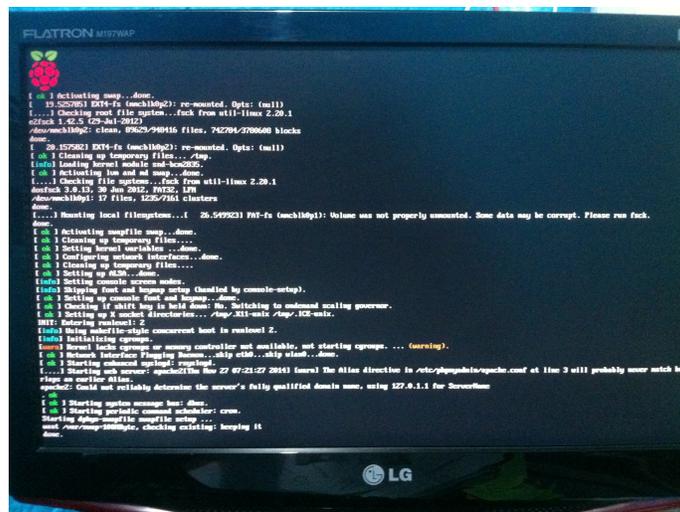


Fig. 9.4.5 Arranque de Raspberry con raspbian

F. Iniciamos poniendo el usuario pi y password franbuesa (si no se modifico es raspberry)



Fig. 9.4.6 Raspbian en modo consola

G. Entramos al modo gráfico con el comando start

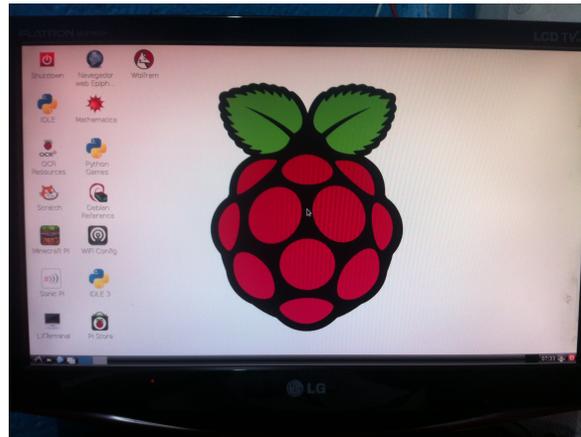


Fig. 9.4.7 Raspbian en modo gráfico

H. Empezamos a trabajar

9.5 Raspberry como servidor web

Antes de empezar con la instalación del servidor, se recomienda actualizar la Raspberry Pi:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Instalar Apache

Para instalar el servidor web Apache, primero tendremos que crear un grupo de usuarios para el servidor:

```
sudo groupadd www-data
```

```
sudo usermod-a -G www-data www-data
```

Instalamos el Apache:

```
sudo apt-get install apache2
```

Una vez finalizado, vemos que se ha creado un directorio en `/var/www` donde se ubicaran los archivos HTML y PHP del servidor. Por defecto, hay un archivo llamado `index.php`. Si ponemos en un navegador la dirección IP de la Raspberry Pi (en el tutorial Raspberry Pi: Guía inicial se explica cómo establecer una IP fija) veremos una página con la frase `It works!`. Esto quiere decir que el servidor se ha instalado correctamente. Ahora podemos probar de crear un archivo html:

`sudo nano /var/www/prueba.html` y escribimos el siguiente código:

```
<html>
<head>
<title>DIYMakers</title>
</head>
<body>
<h1>DIYMakers</h1>
<p>Prueba de archivo HTML</p>
</body>
</html>
```

Para guardar pulsamos CTRL+X, luego S e INTRO. Ahora en el navegador ponemos la dirección IP de la Raspberry más el nombre del archivo html para ver nuestra web:

`http://192.168.1.134/prueba.html` (poner vuestra dirección IP)

Instalar PHP

El lenguaje de programación PHP sirve para poder crear webs dinámicas. Procedemos a instalar el lenguaje PHP en la Raspberry Pi:

```
sudo apt-get install php5
```

Después instalaremos sus paquetes:

```
sudo apt-get install libapache2-mod-php5 libapache2-mod-perl2 php5 php5-cli php5-
common php5-curl php5-dev php5-gd php5-imagick php5-ldap php5-mhash php5-mysql
php5-odbc
```

Para comprobar que se haya instalado correctamente, creamos un archivo php en el servidor:

```
sudo nano /var/www/pruebaphp.php
```

escribimos el siguiente código php:

```
<?php
```

```
echo "Hello World";
```

```
?>
```

Para guardar pulsamos CTRL+X, luego S e INTRO. Ponemos en el navegador web lo siguiente:

```
http://192.168.1.134/pruebaphp.php (poner vuestra dirección IP)
```

Nos tiene que salir el texto "Hello World" en una página web.

Instalar MySQL

MySQL es una base de datos Open Source que sirve para almacenar datos. Para su instalación escribimos lo siguiente:

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

Nos pedirá que pongamos una contraseña que nosotros queramos. Al terminar la instalación, debemos iniciar el MySQL:

```
sudo service mysql start
```

Para comprobar si se ha instalado correctamente, tenemos que escribir el siguiente comando poniendo la contraseña que hemos establecido antes en la instalación de MySQL después de la letra p:

```
mysql -uroot -pCONTRASEÑA
```

Si aparece el comando mysql> quiere decir que funciona correctamente. Para salir pulsamos CTRL+C.

Instalar PHPmyAdmin

PHPmyAdmin es un software basado en PHP que sirve para administrar la base de datos MySQL a través de interfaz web. Para su instalación escribimos el siguiente comando:

```
sudo apt-get install libapache2-mod-auth-mysql php5-mysql phpmyadmin
```

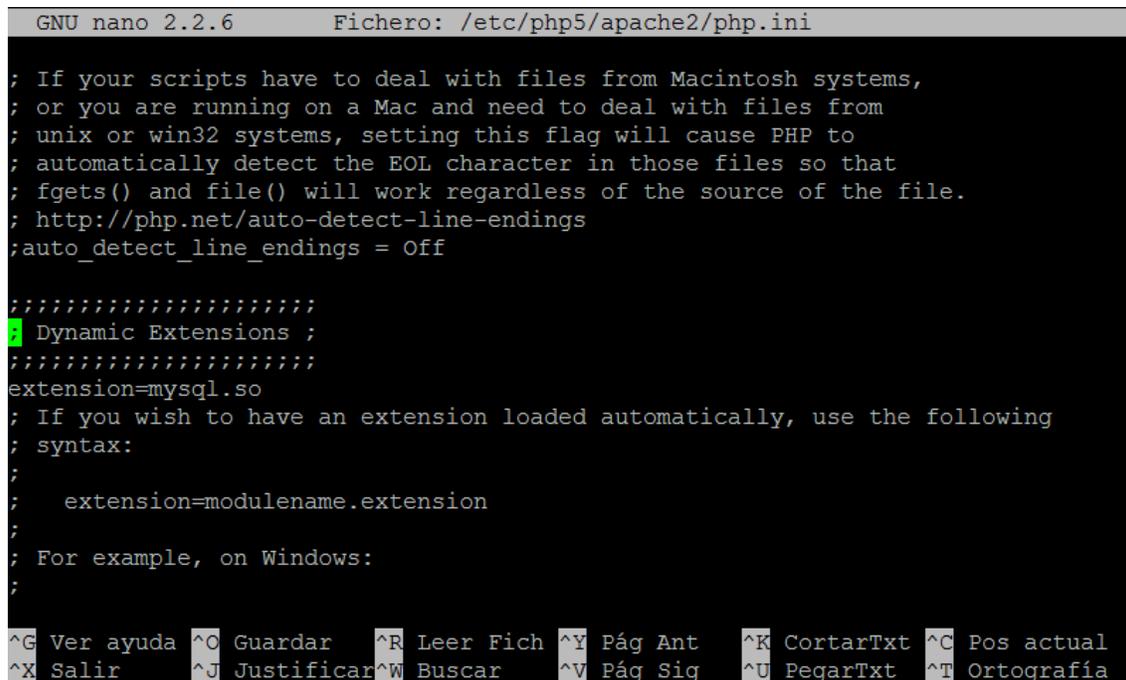
Nos preguntará que servidor web hemos instalado. Seleccionamos Apache con la tecla Espacio y pulsamos Enter. Después preguntará si queremos configurar la base de datos, decimos que Sí. Luego tendremos que poner la contraseña que establecimos previamente en el MySQL y finalmente nos pedirá que pongamos una nueva contraseña para PHPmyAdmin.

Al finalizar la instalación, tenemos que modificar el siguiente archivo:

```
sudo nano /etc/php5/apache2/php.ini
```

Tenemos que escribir el siguiente comando en una nueva línea en cualquier sitio, aunque si queremos tener el archivo ordenado lo pondremos en la sección "Dynamic Extensions":

```
extension=mysql.so
```



```
GNU nano 2.2.6 Fichero: /etc/php5/apache2/php.ini
; If your scripts have to deal with files from Macintosh systems,
; or you are running on a Mac and need to deal with files from
; unix or win32 systems, setting this flag will cause PHP to
; automatically detect the EOL character in those files so that
; fgets() and file() will work regardless of the source of the file.
; http://php.net/auto-detect-line-endings
;auto_detect_line_endings = Off

;;;;;;;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;;;;;;;
extension=mysql.so
; If you wish to have an extension loaded automatically, use the following
; syntax:
;
; extension=modulename.extension
;
; For example, on Windows:
;
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
```

Fig. 9.5.1 Ejemplo de donde escribir el caigo en *php.ini*

Para guardar pulsamos CTRL+X, luego S e INTRO. Esta modificación sirve para si usamos la base de datos MySQLi (distinta a MySQL).

Después, como hemos modificado un archivo tendremos que reiniciar la Raspberry Pi:

```
sudo reboot
```

Tras reiniciar, tenemos que ejecutar dos comandos más:

```
sudo ln -s /etc/phpmyadmin/apache.conf /etc/apache2/conf.d/phpmyadmin.conf
```

```
sudo /etc/init.d/apache2 reload
```

Finalmente, vamos al navegador y ponemos:

```
http://192.168.1.134/phpmyadmin/ (poner vuestra dirección IP)
```

Nos saldrá una interfaz como esta donde usuario es root y la contraseña la que hemos asignado en la instalación:

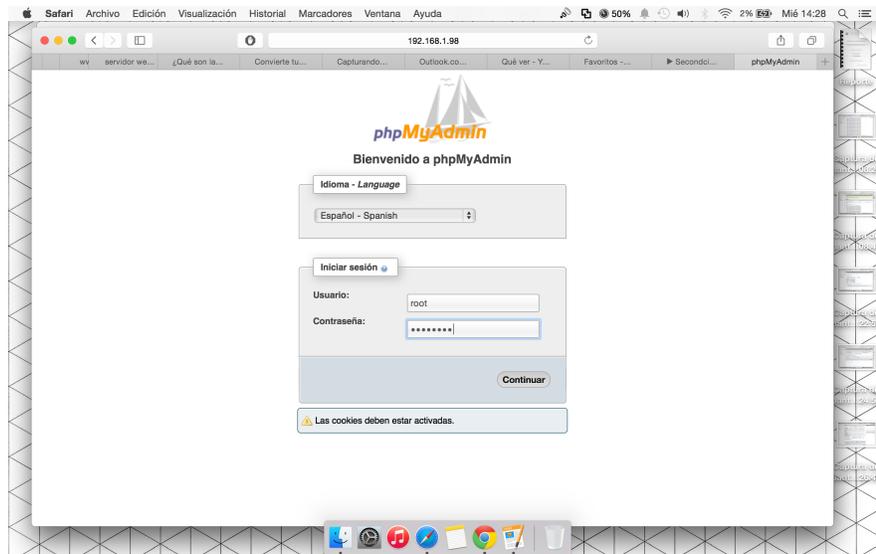


Fig. 9.5.2 Pagina de inicio de phpMyadmin