Escuela Politécnica Superior de Jaén

Grado en Ingeniería Geomática y Topográfica

UNIVERSIDAD DE JAÉN
Nombre del Centro

Trabajo Fin de Grado

DISEÑO E IMPLEMENTACIÓN DE SISTEMA SENSOR Y DE GESTIÓN DE RIEGO Y VISUALIZACIÓN EN TIEMPO REAL

Alumno: Luis Ros Labella

Tutor: Prof. D. Manuel A. Ureña Cámara
Prof. D. Francisco J. Ariza López
Dpto: Ingeniería Cartográfica, Geodésica y Fotogrametría

Noviembre, 2016

Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ingeniería Cartográfica, Geodésica y Fotogrametría.

Don Francisco Javier Ariza López y Don Manuel A. Ureña Cámara, tutores del Trabajo Fin de Grado titulado: Diseño e implementación de sistema sensor y de gestión de riego y visualización en tiempo real, que presenta Luis Ros Labella, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, noviembre de 2016

El alumno: Los tutores:

Luis Ros Labella

Francisco Javier Ariza López

Manuel A. Ureña Cámara

Índice

1. Introducción

1.1. Motivación

La creación del sistema sensor de adquisición de datos ambientales ha sido provocada por la necesidad de obtener temperaturas, humedades entre otras magnitudes sin tener que estar fisicamente en la toma de registros de datos. Actualmete gracias al desarrollo de la ténologia se encuentran en el mercado diferentes sistemas de de instrumentación portatil dedicadas a obtener obtenes datos ambientales y sistemas de comunicación que nos permiten acceder a esos datos.

Entre los diferentes sistemas de instrumentación portatil destacan aquellos que hace uso de sensores de temperatura y humedad, tanto del suelo como del aire. Estos sensores experimentan cambios al producirse una variación en alguna magnitud.

Un campo muy interesante de aplicación de instrumentación portátil, haciendo uso de sensores de humedad y temperatura, es la determinación del estrés hídrico y los parámetros ambientales de un cultivo. Esto se debe a que en la actualidad se ha popularizado el uso de estos sistemas de instrumentación portátil ya que te permiten tener controlado un cultivo en tiempo real y sin interrupción, y ademas son facilmete reproducibles.

La portabilidad el tamaño reducido y el coste son las primeras ventajas que aporta este sistema de instrumentación. Esto permite colocar diferentes sistemas de medidas en una misma superficie para tener un control preciso de el cultivo y evitar al usuario que tenga que desplazarse hasta el cultivo cada vez que este interesando en su estado. La información recogida por el sistema se envia a través de una red establecida a un terminal de control permitiendo la monitorización de los datos en tiempo real, de forma segura si unso de estructuras complrejas y de coste reducido.

1.2. Objetivos

El objetivo de este trabajo es desarrollar e implementar de un sistema sensor de riego para la gestión del estrés hídrico y captura de parámetros ambientales para su visualización en tiempo real.

Un sistema sensor portatil que sea capaz de detectar cambios en la humedad de la tierra, en la humedad relativa del aire y en la temperatura de un cultivo y monitorizar la información mediante un portal web. De esta forma tendremos controlado nuestro cultivo y gestionarlo con mucha mas precisión.

Para conseguirlo se usaran sensores simples y baratos que recogeran la información y se procesará usando un arduino, que será el encargado de enviar la informacion a la base de datos de donde posteriormente la visualizaremos. Este seria un sistema de gestión de riego.

A partir de esta idea general se determinan los objetivos particulares que pretende alcanzar el trabajo fin de grado:

- Diseñar e implementar uno o varios sistemas sensores de parámetros ambientales.
- Diseñar e implementar una base de datos para la gestión de la información en tiempo real de los parámetros capturados y visualización de la misma mediante un servidor.
- Comunicar ambos sistemas en tiempo real.

Con el fin de cumplir estos objetivos se ha realizado una metodologia a desarrollar para elaborar el trabajo fin de grado:

- Analisis de sistemas similares existentes en el mercado.
- Diseñar e implementar un sistema bajo Arduino para la obtención de información sobre parámetros ambientales.
- Diseñar e implementar la comunicación del sistema Arduino con un sistema gestor centralizado.
- Diseñar e implementar la estructura de la base de datos y el servidormpara la visualización de la información en tiempo real.

- Análisis de los resultados del proceso y sus pruebas.
- Conclusiones y mejoras futuras.

Para la realizacion del trabajo necesitamos tener unos conocimientos previos sobre la programación del Arduino, junto con sus diferentes conmponente como shields, sensores… y conocimientos sobre infraestructura de base de datos.

2. Antecedentes y Estado de la técnica

En este apartado se expone el entorno en el que esta ambientado el trabajo. Se pueden distinguir tres lineas claras:

- Los sistemas de adquisición de información, que esta orientada a la extracción de los datos.
- El almacenamiento de la información, orientada a la gestion y transporte de los datos.
- La representación de la informacion, orientada a la visualizacion y acceso de los datos.

2.1. Antecedentes

Las plantas dan vida y color a nuestras casas, pero debido a diferentes factores no podemos estar en ella para aplicarles los cuidados necesarios, como es el riego. Este problema lo podemos solucionar mediante la creación de un sistema de autorriego que sea capaz de regar la planta cada vez que lo necesite.

Un sistema de autorriego, como su propio nombre indica, permite que la planta no tenga deficiencia o abundancia de agua. Podemos encontrar diferentes modelos de sistemas de autorriego, y los podemos agrupar en dos grupos: tradicionales (no incluyen electrónica) y electrónicos.

Entre los sistemas de riego tradicionales encontramos:

- Conos de arcilla: Estos conos son fáciles de usar; económicos y dan muy buenos resultados. Se trata de pequeños depósitos fabricados generalmente de arcilla, que se llenan de agua y se introducen en la tierra de la planta; el otro extremo queda sumergido en un depósito de agua.
- Hidrojardineras: Son recipientes con un sistema de autorriego incorporado y además son muy decorativos. El sistema es realmente sencillo. Cuentan con un depósito generalmente dispuesto en la base, éste va proporcionando el agua poco a poco a las raíces.

- Retentor de agua: Se trata de un material gelatinoso capaz de acumular una cantidad de agua de hasta cuatro veces su peso para después liberarla progresivamente evitando que la tierra se seque en algún momento.

Los sistemas electrónicos de autorriego los podemos comprar, o como realizaremos en este trabajo, elaborarlo, con el fin de gestionar nosotros mismos la información referente a nuestro cultivo mediante una monitorización remota con sensores en tiempo real a través de un servidor web.

El sistema de autorriego se puede elaborar de varias maneras y utilizando diferentes tecnologías que tenemos a nuestro alcance. En nuestro caso utilizaremos Arduino como sistema de adquisición de datos para transportar la información de los sensores al sistema de almacenamiento, que será una base de datos.

En los siguientes apartados veremos el estado actual del entorno tecnológico que rodea a nuestro trabajo.

2.2. Sistemas de adquisición y procesamiento de datos

En este apartado vamos a hablar de cómo extraemos la información que necesitamos de nuestro cultivo. Dicha información la obtenemos de unos sensores que son controlados mediante tarjetas de desarrollo.

2.2.1. Tarjeta de desarrollo

Una Tarjeta de Desarrollo es una herramienta para diseño y prototipado rápido de sistemas digitales o analógicos, que se presenta como un elemento muy útil para el mejoramiento de los procesos de diseño. Estas tarjetas de desarrollo están gobernadas por un microcontrolador y que es su cerebro. Este elemento controla el comportamiento de la placa,y es el encargado de procesar toda la información.

El primer microcontrolador comercial fue el Intel 4004 de 4bits, aunque el primer microcontrolador que revolucionó el mercado fue el microcontrolador PIC debido a su precio, disponibilidad y proliferación de herramientas de software libre. Dado que este chip puede ser complicado de programar por su bajo nivel de lenguaje de programación, son más comunes los chips PICAXE (PIC estándar

reprogramado) o tarjetas BASIC Stamp que utilizan un lenguaje más fácil de entender.

De esta manera empezó a revolucionarse el mundo de los “makers” ya que tenían la posibilidad de controlar diferentes dispositivos electrónicos de una forma sencilla y barata. A continuación tenemos una lista de tarjetas de desarrollo que se usan en la actualidad.

BeagleBone

Beaglebone es una tarjeta de desarrollo de hardware libre y bajo consumo desarrollada por beaglebone.org. La placa fue desarrollada para enseñar las capacidades del software y hardware libre. La web donde acceder a la información es http://beagleboard.org/.

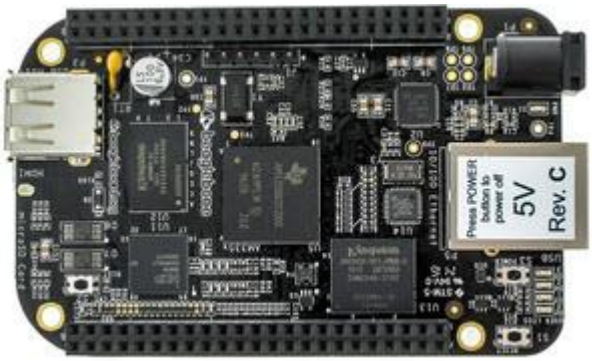

Ilustración 1. Tarjeta de desarrollo Beaglebone

Esta plataforma puede correr bajo un sistema operativo Linux (hay varias distribuciones de Linux para las plataformas Beaglebone). Este tiene varias entradas y salidas con diversas funciones: entradas y salidas digitales, entradas analógicas, salidas con PMW[1], un puerto Ethernet y otro USB 2.0 para la comunicación con los demás dispositivos. La alimentación energética se realiza mediante mini USB o por conexión tipo jack de 5V y 210mA o 460mA.

Hay dos tipos de prototipos que son:

[1] (pulse-width Modulation) señal de modulación por ancho de pulso.

- Beaglebone: Con un procesador Sitara ARM Cortex-A8 corriendo a 720 MHz, RAM de 256 MB, dos conectores de expansión de 46-pin, ranura microSD y un puerto de dispositivos multipropósito que incluye un serial de control a bajo.
- Beaglebone black: Incrementa la RAM a 512 MB, el reloj de procesador a 1 GHz, y añade HDMI y 2 GB de memoria flash eMMC. La BeagleBone Black también se entrega con kernel Linux 3.8.

Raspberry Pi

Es una tarjeta de desarrollo o en este caso más concreto un ordenador de placa única de bajo coste desarrollada por la fundación Raspberry Pi con el fin de instruir en la enseñanza de ciencias de la computación en las escuelas. La web donde acceder a la información es https://www.raspberrypi.org/.

Ilustración 2. Tarjeta de desarrollo Rapsberry-PI.

Raspberry corre, generalmente, bajo sistemas operativos basados en Linux. Para su uso es necesaria una tarjeta de memoria como sistema de almacenamiento y conectarlo a la red eléctrica. Generalmente todas las placas disponen de un puerto de salida de video (HDMI, RCA) y de audio (minijack). También tiene un puerto USB 2.0 y una conexión Ethernet para la comunicación. La alimentación energética se realiza mediante mini USB o por conexión tipo jack de 5V y 750mA.

Hay diferentes tipos de Rapsberry:

- Rapsberry PI A: es la tarjeta más básica, Utiliza un chip Broadcom BCM2835 con una CPU ARM1176JZF-S a 700 MHz, un procesador gráfico VideoCore IV a 250 Mhz, y 256 MB de memoria RAM.
- Rapsberry PI B: es igual que el hardware del modelo A excepto que tiene 512 MB de memoria RAM e incluye una conexión de red Ethernet10/100.
- Rapsberry PI A +: tiene el mismo hardware que las anteriores añadiendo más conectores GPIO[2] (hasta 17), soporta tarjetas microSD y es más pequeña.
- Rapsberry PI B+: es como la Raspberry Pi B añadiéndole dos puertos UBS2.0 adicionales, tarjeta microSD, mejora de audio y menor consumo.
- Rapsberry PI 2 B: esta es la tarjeta más completa, añade una nueva CPU ARM Cortex-A7 de cuatro núcleos a 900 MHz, así como 1 GB de memoria RAM a 450 Mhz, es 6 veces más potente que los modelos anteriores.

Waspmote

Es una plataforma de código abierto que permite la construcción de redes de sensores inalámbricas de bajo consumo creada y desarrollada por la empresa Libelium. Este producto a diferencia de los demás es que cuenta con muchas shields en las que van integradas más de 80 tipos de sensores diferentes. La web donde acceder a la información es http://www.libelium.com/products/waspmote/.

[2] (General Purpose Input/Output, Entrada/Salida de Propósito General) es un pin genérico en un chip.

Ilustración 3. Tarjeta de desarrollo Waspmote.

La placa waspmote usa el mismo IDE (compilador y librerías principales) que Arduino, ya que sus lenguajes (de alto niel) son similares. La placa incluye la memoria, la batería, un acelerómetro y diferentes sockets para añadir los módulos. La alimentación energética se realiza mediante mini USB o por conexión tipo jack de 5V a 12V y 100mA a 280mA

La plataforma Waspmote está especialmente diseñada para crear redes sensoriales inalámbricas que necesitan una larga durabilidad y ser instalados en un entorno real.

Arduino

Arduino surgió como proyecto para que los artistas tuvieran acceso a microprocesadores embebidos para el diseño interactivo. Es una plataforma de código abierto que permite la creación de proyectos sencillos y asequibles. Este producto también cuenta con una gran cantidad de shields en los que se incluyen sistemas de sensores o sistemas actuadores. La web donde acceder a la información es https://www.arduino.cc/

.

Ilustración 4. Tarjeta de desarrollo Arduino UNO.

Arduino se compone de 3 cosas:

- Una placa de hardware libre (cuando hablamos de placa hardware nos referimos a una placa de circuito impreso (PCB) que incorpora un microcontolador, que se puede reprogramar, y una serie de pines que utilizamos para conectar sensores o actuadores. Hay diferentes modelos de placa Arduino con diferentes caracteríscas, pero lo que no cambia en ninguno de ellos es la “familia tecnológica” a la que pertenecen los microcontroladores (son de tipo AVR, desarrollado por Atmel).
- Un software gratis, libre y multiplataforma que nos permite escribir, compilar y guardar instrucciones en la memoria del microcontrolador de la placa mediante cable USB.
- Un lenguaje de programación basado en el lenguaje de programación de alto nivel que es similar a C++.

Hay multitud de prototipos como: Arduino UNO, Leonardo, Due, Mega… Cada placa es diferente, ya sea por el tamaño, microcontrolador o número de pines, por ejemplo. Todas utilizan el mismo lenguaje de programación y utilizan el mismo software. En la siguiente tabla exponemos algunas caracteristicas de algunos modelos.

Tabla 1. Características principales de la familia de tarjetas de desarrollo Arduino (no se muestran todos los modelos existentes).

Caracteristicas	UNO	Mega	Leonardo	DUE
Microprodesaror	Atmega 328	Atmega 2560	Atmega 32U4	AT91SAM3X8E
Velocidad de reloj	16MHz	16MHz	16MHz	84MHz
Pines digitales E/S	14	54	20	54
Entradas analógicas	6	16	12	12
Salidas analógicas	0	0	0	2(DAC)
Memoria de programa	32Kb	256Kb	32Kb	512Kb
Memoria de datos	2Kb	8Kb	2.5Kb	96Kb
Memoria auxiliar	1Kb	4Kb	1Kb	0Kb

2.2.2. Shields

Las shields son placas de circuitos modulares que se conectan a la tarjeta de desarrollo permitiendo aumentar el hardware. Se comunican los pines digitales o analógicos. Existe una gran variedad de shields que se usan para la comunicación o interacción con el entorno. Las shields por lo general son especificas de arduino, aunque existen formas de adaptar estos a otras targetas de desarrollo .

Aqui se dejan algunos ejemplos:

- WiFi shield: proporciona acceso a internet vía WiFi usando la WiFi Library.

Ilustración 5. Shield WiFi modelo 2 para Arduino.

- Ethernet Shield: Proporciona acceso a internet vía Ethernet con un cable RJ45 usando la Ethernet Library.

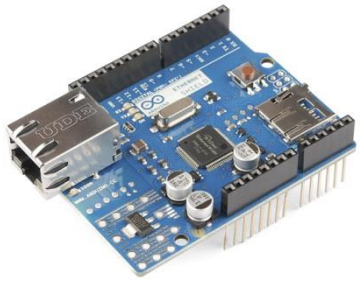

Ilustración 6. Ethernet Shield para Arduino.

-

- GSM/GRPS shield: Proporciona acceso a internet mediante GPRS usando una tarjeta SIM.

Ilustración 7. Shield para comunicaciones GSM/GPRS.

- Relay shield: esta proporciona uno o varios relés para controlar circuitos que no puedan controlarse directamente con el Arduino.

Ilustración 8. Shield de relés para Arduino.

2.2.3. Sensores de adquisición de datos

Un sensor es un dispositivo electronico capaz de detectar magnitudes físicas o químicas, llamadas variables de instrumentación, y transformarlas en variables eléctricas. Estos dispositivos se desarrollan mediante la utilización de componentes pasivos y activos. Los sensores hacen legibles las magnitudes físicas convirtiéndolas en señales eléctricas.

Los sensores tienen diferentes características como:

- Rango de medida: dominio en la magnitud medida en el que puede aplicarse el sensor.
- Offset o desviación de cero: valor de la variable de salida cuando la variable de entrada es nula.
- Sensibilidad: el valor mínimo de activación, la mínima cantidad de incremento del parámetro físico para que el sensor pueda dar una respuesta diferente.
- Repetitividad: error esperado al repetir varias veces la misma medida.
- Resolución: mínima variación de la magnitud de entrada que puede detectarse a la salida.
- Precisión: es el error de medida máximo esperado.

Las mas importantes son la resolucion y la precision del dispositivo.

Según la magnitud física que midan los sensores nos encontramos sensores de temperatura, humedad, velocidad, sonido... A continuación veremos unos apartados describiendo los diferentes tipos de sensores:

Sensores de contacto

Se emplean para detectar el final del recorrido o la posición límite de componentes mecánicos. Los principales son los llamados fines de carrera.

Sensores de humedad del suelo

Los sensores de humedad del suelo transforman los cambios de humedad en señales eléctricas. Son sensores capacitivos que permiten la medición de la constante dieléctrica para calcular el contenido de humedad del suelo. La fracción volumétrica del suelo ocupada por agua tiene gran influencia en la permisividad dieléctrica del suelo ya que su valor dieléctrico es mucho más alto que los valores asociados a otros constituyentes del suelo.

Sensores de temperatura

Los sensores de temperatura transforman los cambios de temperatura en señales eléctricas. El sensor está formado por diferentes elementos: el sensor, la vaina o envoltura y el cable de conexión al equipo.

Hay 3 tipos de sensores:

- Termistor: se basa en el comportamiento de la resistencia de los semiconductores, según la temperatura.
- RTD[3]: se basa en el comportamiento de la resistencia de un conductor según la temperatura.
- Termopares: se basa en el efecto termoeléctrico (un material termoeléctrico permite transformar el calor en electricidad).

Sensores ópticos

Detectan la presencia de una persona o de un objeto que interrumpen el haz de luz que le llega al sensor. Los principales sensores ópticos son las fotorresistencias, las LDR.

Sensores de humedad relativa

Los sensores de humedad relativa transforman los cambios en la humedad del aire en una señal eléctrica. Los sensores de humedad relativa pueden ser analógicos y digitales:

[3] (resistance temperature detector, detector de temperatura resistivo).

Analógicos: el sistema está basado en un condensador. El sensor está hecho de una película de vidrio. El material aislante absorbe y libera el agua basándose en la humedad relativa de la zona dada. Esto cambia el nivel de carga en el condensador del circuito en el cuadro eléctrico.

Digital: se basa en dos microsensores que se calibran según la humedad relativa de cada zona tomando su valor analógico. Luego mediante un chip se realiza una conversión de la señal analógica a la digital.

2.3. Almacenamiento de la información

Los datos obtenidos por los sensores deben de ser almacenados para su posterior procesamiento y visualización de los mismos.

Para almacenar los datos se tiene que hacer uso de un soporte de almacenamiento de datos o medio de almacenamiento de datos que es el dispositivo físico en el que se almacenaran los mismos. Los diferentes tipos de dispositivos físicos que encontramos son discos duros (soporte magnético), tarjetas de memoria (soporte de estado sólido), discos compactos (CD, soporte óptico)…

Para almacenar la información en el soporte de almacenamiento de forma ordenada para que luego podamos encontrar y utilizar la información deseada se pueden emplear tanto una base de datos o el almacenamiento en la nube. El almacenamiento en la nube está basado en redes de ordenadores donde los datos están alojados en espacios de almacenamiento virtuales, generalmente estos espacios los da un tercero.

2.3.1. Bases de datos

Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada. Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Cada base de datos se compone de tablas que guardan los datos, estas tablas se dividen en filas (donde está guardado cada registro) y columnas (donde se guarda la información sobre los elementos de la tabla).

La mayor ventaja de las bases de datos es que conseguimos que la información almacenada en ellas se pueda compartir y que terceros autorizados puedan acceder a los datos. Entre otras ventajas encontramos:

- Independencia de los datos programas y procesos.
- Menor redundancia de datos.
- Integridad de los datos.
- Mayor seguridad en los datos.
- Acceso simultaneo a los datos.
- Reducción del espacio de almacenamiento.
- Acceso a los datos más eficiente.
- Aumenta la productividad de los programadores.

La desventaja de estos almacenes de información es que son conjuntos de programas que pueden llegar a ser muy complejos aunque luego se obtiene una gran funcionalidad de ellos.

Para la gestión de una base de datos se necesita un sistema de gestión de base de datos que son un conjunto de programas, software que sirve de interfaz entre la base de datos, las aplicaciones y el usuario. Estos sistemas de gestión se componen de 3 lenguajes basados en el lenguaje de programación SQL.

SQL es un lenguaje de descripción de datos que utiliza diferentes acciones que modifican el esquema de la base de datos, creando, modificando o borrando tablas y objetos. Este lenguaje es el más extendido entre los gestores de bases de datos.

A continuacion se describen los diferentes lenguajes de gestión de datos:

- De definición de datos (DDL): Este lenguaje de gestión de datos permite al usuario la creación de las estructuras donde se almacenaran los datos y los procedimientos para consultarlos. Las sentencias que se incluyen en este

lenguaje son CREATE (para crear), ALTER (para modificar) y DROP (para eliminar)

- De manipulación de datos (DML): Este lenguaje de gestión de datos permite al usuario la consulta o modificación de los datos contenidos dentro de las estructuras. Las sentencias que se incluyen en este lenguaje son SELECT (para seleccionar datos), INSERT (para insertar datos), DELETE (para borrar datos) y Update (para modificar datos)
- De control de datos: Este lenguaje de gestión de datos permite al administrador controlar el acceso a los registros dentro de la base de datos. Los comandos que incluye son GRANT (para dar permisos), REVOKE (para eliminar permisos).

La gran mayoría de los sistemas gestores de base de datos siguen un modelo entidad-relación. Este modelo representa abstracciones y conocimientos en un sistema de información formado por diferentes objetos llamados entidades y relaciones. Este modelo también incorpora una representación visual denominada diagrama entidad-relación. Ya en la actualidad encontramos bases de datos orientadas a objetos y bases de datos orientadas a objetos.

Como conclusión podemos decir que las bases de datos son un conjunto de datos que se encuentran en algún soporte físico permanente y un conjunto de programas que nos ayudan a manipularlos.

Hay gran variedad de programas para manipular una base de datos, aquí se recopila una breve lista:

MYSQL (https://www.mysql.com): Este sistema de base de datos relacional de software libre desarrollado por MySQL AB que ofrece una licencia dual: ofrece la GNU GL y también se puede comprar una licencia de uso.

ORACLE (https://www.oracle.com/e/): Este sistema de base de datos relacional está desarrollado por Oracle Corporation. Es uno de los sistemas más completos. El problema de este sistema es su precio que es bastante elevado, según la versión y las licencias.

Access(https://products.office.com/es-es/access): Este sistema de base de datos relacional se creó para su uso en pequeñas organizaciones. Es parte del paquete suite Microsoft Office.

PostgreSQL(www.postgresql.org.es): Este sistema de base de datos relacional está orientado a objetos y es libre, bajo licencia BSD. Este es dirigido por una comunidad (PostgreSQL Global Development Group) de desarrolladores que se apoyan en empresas comerciales.

SQLite (https://sqlite.org): Este sistema de gestión de bases relacional es compatible con ACID. La diferencia con los demás gestores de bases de datos cliente-servidor es que el motor SQLite es un proceso más dentro del programa principal, lo que hace que se puede integrar en cualquier software porque no requiere de un servidor activo.

2.4 Aspectos esenciales en las aplicaciones desarrolladas en la web.

En este apartado se describe el entorno web que rodea al trabajo.

2.4.1. Internet

Internet es un conjunto de redes individuales de ordenadores interconectados físicamente, de acceso público y global, es una red de redes. Cuando la red tiene un ambito reducido, es una red local.

Toda esta lógica esta soportada por una red física. Esta red física está divida en subredes que pertenecen a diferentes empresas que son responsables del mantenimiento de la misma y su interconexión con sus adyacentes.

Para conectar el ordenador a la red hacemos uso de un router que nos permita el acceso a la red global, una tarjeta de red (WiFi) o cable LAN que nos permita conectar el ordenador con el router. La conexión con la red principal la realizamos mediante una red local (LAN), una red de área extensa (WAN) o un proveedor de servicio de internet (ISP). Las redes principales son las columnas vertebrales de internet de un ancho de banda muy elevado.

Lo que asegura la comunicación entre los ordenadores conectados a la red son los protocolos de comunicaciones, que permiten que ordenadores distantes que pertenecen a unas subredes diferentes e incluso sistemas operativos o aplicaciones diferentes puedan comunicarse bien. Estos protocolos son una serie de normas formales que tienen que cumplir ambos ordenadores para comunicarse.

2.4.2. Comunicaciones, protocolos e identificación de recursos

Las comunicaciones entre dos aplicaciones residentes en distintos sistemy esas se realizan según un modelo estructurado en capas. Este modelo se denomina OSI[4] y es un modelo de referencia para los protocolos de la red de arquitectura en capas. En la imagen siguiente se muestran las diferentes capas de las que esta formado este modelo.

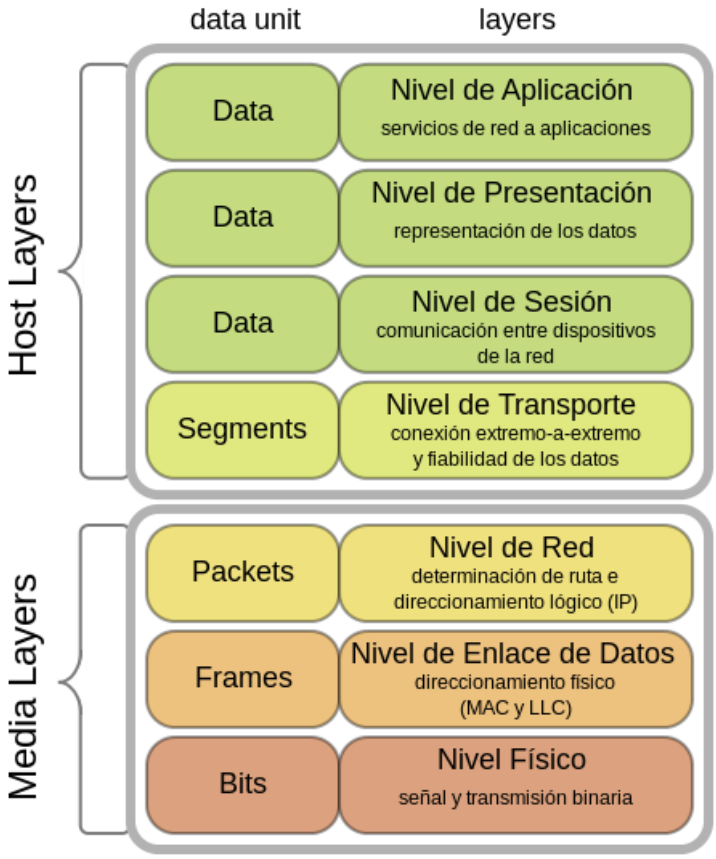

Ilustración 9. Esquema del modelo OSI. (https://es.wikipedia.org/wiki/Modelo_OSI)

En una conexión entre equipos existe comunicación entre las capas que se encuentran al mismo nivel en cada uno de los equipos, yendo la información desde

[4] Open System Interconnection. Modelo de interconexión abierto. ISO/IEC 7498-1.

las capas superiores hasta las que tiene por debajo, de forma que la comunicación directa entre aplicaciones tiene lugar entre las capas inferiores. Una vez que la capa inferior de la aplicación destino de la comunicación recibe los datos, esta capa ofrece servicios a las capas superiores.

Protocolo TCP/IP

TCP/IP es una familia de protocolos de comunicaciones que definen la comunicación entre dos aplicaciones según cuatro capas o niveles, TCP/IP encapsula de forma algo diferente los niveles del modelo OSI. Estas cuatro capas son:

- Capa de aplicación: En esta capa se definen los protocolos que permiten gestionar los servicios más importantes de internet. Incluye los niveles de sesión, presentación y aplicación del modelo OSI.
- Capa de transporte: Provee comunicación, de extremo a extremo entre las aplicaciones.
- Capa de red: Controla la comunicación entre un equipo y otro, decide que rutas deben seguir los paquetes de información para alcanzar su destino, y en el caso de la aplicación que envía la información.
- Capa física: Lanza al medio físico los paquetes de datos (datagramas) emitidos por los otros niveles. Incluye los niveles de enlace de datos y físico del modelo OSI.

Direcciones IP

Son la base de la identificación de dispositivos. En la actualidad se utiliza el estándar IPv4 que será superado por el IPv6.

La gestión de las redes se consigue mediante la denominada mascara, que permite distinguir los bits que idientifican la red y los que identifican el host de una dirección IP.

A la hora de crear una subred se pueden utilizar 4 tipos de IP:

- IP pública: es una IP asignada que es visible en la red de manera directa, y es administrada por el proveedor de servicios de internet.
- IP privada: es una IP asignada a un dispositivo conectada a una red local y esta es asignada por el ruter de la misma.
- IP dinámica: esta es aquella que es asignada mediante un servidor DHCP. La IP que se obtiene tiene una duración máxima determinada.
- IP fija: esta es aquella que es asignada por el proveedor de servicios de internet o el router de una red local. Esta m cambia en el tiempo.

DNS

Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. El uso que más se leda es el de la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico.

URI

Un URI es un identificador único global que se expresa como una cadena de caracteres que identifica inequívocamente un recurso fuente. Las URI están formadas por dos identificadores:

- Localizador uniforme de recursos (URL): Son unas cadenas de texto que se usan para nombrar rutas de acceso en internet para su localización.
- Nombre uniforme de recursos (URN): Son unas cadenas de texto que se usan para nombrar un recurso en internet para su localización.

Protocolo y transacción http

El protocolo HTTP es el que da soporte al WWW. (Sistema hipermedia para la distribución de ficheros digitales). Es un protocolo orientado a sistemas distribuidos para la colaboración e hipermedia. Es un protocolo genérico, sin estado orientado a objetos y que puede ser utilizado por muchas aplicaciones y sistemas de gestión de

objetos distribuidos a través de las extensiones de los métodos de petición. Con este protocolo se permite que los sistemas no dependan del tipo de datos utilizado.

La transacción se inicia con una petición del cliente al servidor. La petición es una URI en la que se alberga una solicitud y el servidor la contestara. La URL consta de 3 partes:

- El tipo de protocolo de transferencia a través del que se pretender acceder a un recurso.
- La dirección IP del servidor donde está el recurso deseado y el puerto de comunicaciones por el que la aplicación encargada de servir el recurso escucha.
- La ruta de acceso para llegar al recurso deseado en la estructura de directorios virtuales.

Los métodos HTTP más importantes invocados por el cliente en el servidor son:

- El método GET: Este método se utiliza para recoger cualquier tipo de información del servidor.
- El método POST: Este método se utiliza para enviar información al servidor.
- El método HEAD: Este método se utiliza para solicitar información sobre un objeto (fichero).

Arquitectura cliente/servidor

Es la arquitectura de computación en la que se basa el funcionamiento de internet, y en especial en el funcionamiento del web. Es una arquitectura distribuida y cooperativa basada en dos capas: El cliente, Capa desde la cual se emite una solicitud y se muestra el resultado y el servidor, Capa que recibe la solicitud del cliente, la procesa y genera una respuesta que es enviada de la vuelta al cliente.

Cualquier aplicación que funcione a través de a través de internet ha de estar basado en el empleo de esta arquitectura. Tanto hardware como software pueden actuar de clientes como de servidores. El cliente y el servidor son entidades abstractas que pueden estar en la misma o diferentes máquinas.

Cliente

El cliente inicia el dialogo al enviar la petición al servidor, esperando hasta que recibe la respuesta del mismo. Los clientes pueden ser:

- Clientes ligeros: que no requieren de ninguna instalación (en todo caso pueden requerir tener instalados algunos componentes desarrollados por terceros que actúan en calidad de plataforma de ejecución, como Google Chrome o el plugin Adobe Flash).
- Clientes pesados: Estos si precisan de una instalación explicita de la aplicación en el ordenador o dispositivo del usuario.

Servidor

El servidor es el componente encargado de procesar las solicitudes de los clientes. Tras la recepción de una solicitud, lo proceso y luego envía la respuesta al cliente. Generalmente acepta conexiones de múltiples clientes.

Modelo de 3 capas

La programación por capas es el origen de la arquitectura basada en capas. El objeto de esta forma de proceder es separar en capas distintas a aquellos elementos que son diferenciables desde un punto de vista lógico. Este tipo de arquitectura respeta los siguientes principios:

- Cada capa reside en un sistema físico independiente.
- Una capa solo intercambia información con las capas situadas que lindan con ella.
- El contenido de cada capa puede ser intercambiable mientras se respeta la definición concisa de la interfaz de comunicación con las capas adyacentes.

Cuando realizamos una aplicación ajustada al modelo cliente-servidor, se pueden distinguir algunas funciones que son comunes en las aplicaciones generales y de la información. Estas funciones se pueden agrupar en tres grupos o logias:

- Lógica de presentación: Esta serie de funciones se encarga de controlar los aspectos relacionados con la interacción entre el usuario y la aplicación.
- Lógica de negocio: las funciones que se incluyen en este grupo son las encargadas del cumplimiento de las reglas de negocio propias de la aplicación.
- Lógica de datos: En este conjunto entran los procesos encargados de la gestión de los datos, es decir, los procesos encargados del mantenimiento de los datos.

Como su propio nombre indica el modelo de 3 capas consta de 3 capas:

- Capa del cliente: En un navegador web o cualquier otra aplicación que mande peticiones al servidor.
- Capa del servidor web: Es un servidor web que se encarga de gestión del tráfico de los datos.
- Capa del servidor de aplicaciones: Es un servidor de datos se encarga del almacenamiento y definición de los datos.

Lenguajes de programación y etiquetado

Lenguajes de programación

- JavaScript (JS): es un lenguaje de programación interpretado. En su definición se describe como orientados a objetos, basado en prototipos, imperativo y dinámico. JavaScript sigue un estándar definido de los Scripts, el ECMAScript (ISO/IEC 16262). Se utiliza, generalmente, en el lado del cliente, donde se implementa como parte del navegador con el fin de mejorar el interfaz entre el usuario y el servidor (portal web) dinámico aunque también se utiliza en el lado del servidor. Este lenguaje se utiliza en las páginas web HTML para enviar y recibir información del servidor y para realizar operaciones en el marco del cliente.
- PHP (Pre hypertext–procesor): es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta de lenguaje es

permitir rápidamente a los desarrolladores la generación dinámica de páginas web.

Definición de web

- eXtensible Markup Languaje (XML): Es un lenguaje que se usa para definir lenguajes de marcas con el fin de almacenar los datos de forma legible.Este lenguaje es un estandar ISO en el que estan basados HTML o CSS. XML da soporte a bases de datos , siendo útil cuando diferentes aplicaciones se comunican o se integran.
- HiperText Markup Language (HTML): es un lenguaje de marcado utilizado para la elaboración de páginas web. Es un estándar que se utiliza como referencia del software que conecta con la creación y edición de páginas web. Define una estructura básica y un cogido (HTML) para la definición del contenido de la página. Es considerado como el lenguaje web más importante dado a que propició la aparición, desarrollo y expansión de la World Wide Web y es el estándar en la visualización de páginas web que han adoptado todos los navegadores.
- HTML5: Es la ultima version de HTML. Esta última versión contiene nuevos elementos, atributos y comportamientos que permiten a los sitios Web y a las aplicaciones ser mas diversas y tener mayor alcance.
- Cascading Style Sheets (CSS): es el lenguaje que se encarga de definir y crear la presentación de un documento HTML o XML... Mediante este separamos la estructura del documento de su presentación.
- CSS3: Es la última versión de CSS que incluye características avanzadas para aplicar un mejor aspecto a los elementos para realizar una maquetación mas precisa.

Librerías

- Bootstrap: es un framework creado por Twitter, con el que se pueden crear interfaces web con CSS y JavaScript que se adaptan automaticamente al formato de la pantalla del dispositivo. Esta técnica de diseño es “responsive design”.

- Highcharts: es una biblioteca de gráficos en JavaScript. Es gratuito para uso no comercial y personal, las licencias solo se necesitan en aplicaciones comerciales.

Lenguaje de consulta

- SQL: es un lenguaje de consulta que nos permite el acceso a bases de datos relacionales. Se caracteriza por su manejo del álgebra y del calculo relacional que pertmite hacer consultas para recuperar una información de forma sencilla.

3. Diseño del sistema

3.1. Requerimientos del sistema

A la hora de elegir los diferentes elementos de los que está hecho el trabajo hay que considerar una serie de requerimientos que se deben establecer:

- El requisito principal del trabajo propuesto es que es “low cost”, por lo que a la hora de decidir sobre un componente u otro lo que más pesara en la decisión será el precio del elemento.
- Otro de los principales es que al ser un trabajo “low cost” en la elección del software este tiene que ser de código abierto y gratuito (Free Open Source Software).
- En lo referente a la transferencia de datos, al ser muy baja, no necesitamos de un procesado con mucha potencia.
- Otra es el tipo de red que utilizamos, lo que buscamos es que sea lo más reducida posible, aunque este requerimiento no es de los más relevantes.
- En una de los objetivos es que el trabajo sea de fácil reproducción por lo que los componentes tienen que ser fácilmente accesibles.

3.2. Elección de componentes

3.2.1. Sistema de adquisición y procesamiento de datos

El sistema de adquisición de datos lo componen dos sensores: el sensor de humedad YL-69 y el sensor de temperatura y humedad relativa DHT 11. Hemos elegido estos sensores, básicamente por su precio y su accesibilidad. La elección no es muy complicada que en la gama de sensores de este tipo es reducida y son todos muy parecidos.

Para el procesado de datos hemos elegido los siguientes componentes: Arduino UNO y Arduino Ethernet Shield. Estos dos elementos también han sido básicamente elegidos por su precio ya que son los más económicos dentro de la gama de productos de este tipo. En las evoluciones de este trabajo se planteara introducir una WiFi Shield para mejorar su portabilidad.

Para este tipo de trabajos, la tarjeta de desarrollo Waspmote es más funcional, pero su elevado precio hace que se inclumpla el requisito de “low cost”.

3.2.2. Almacenamiento de la información

Dentro de la gama de sistemas de base de datos que se ha expuesto se ha elegido MySQL ya que es un software libre y este se incluye en el paquete XAMPP, que lo hace mucho más fácil de usar, ya que incluye un gestor de base de datos muy intuitivo que es phpMyadmin.

3.2.3. Tecnologías utilizadas

Dentro del ambiente del desarrollo web tenemos una gran variedad de lenguajes de programación o de etiquetado junto con sus librerías. En este caso se van utilizar los siguientes:

- HTML5
- CSS
- JavaScript
- PHP
- Librería JavaScript de Highcharts, que se descarga directamente de su página oficial.

- Librería Boostrap de Twitter, que se descarga directamente de su página oficial

La librería Boostrap se ha elegido para permitir que el portar web tenga una buena visualización en los dispositivos móviles, donde el formato de la pantalla es más reducido y esta librería permite un escalado y ajuste automático a cada tamaño de pantalla. El resto de tecnologías han sido elegidas por accesibilidad ya que son las más conocidas.

3.3. Esquema de sistema sensor

El trabajo se compone de dos partes: un software, ya explicado y un hardware.

En el siguiente esquema se puede observar que hay una placa Arduino que será el nodo donde irán conectados los sensores. También se puede ver el servidor local, instalado en el PC y la base de datos también instalada en el PC.

También está incluida la infraestructura de red, donde el router es el que la soporta con punto de acceso inalámbrico (WiFi), el cableado Ethernet y la conexión a internet.

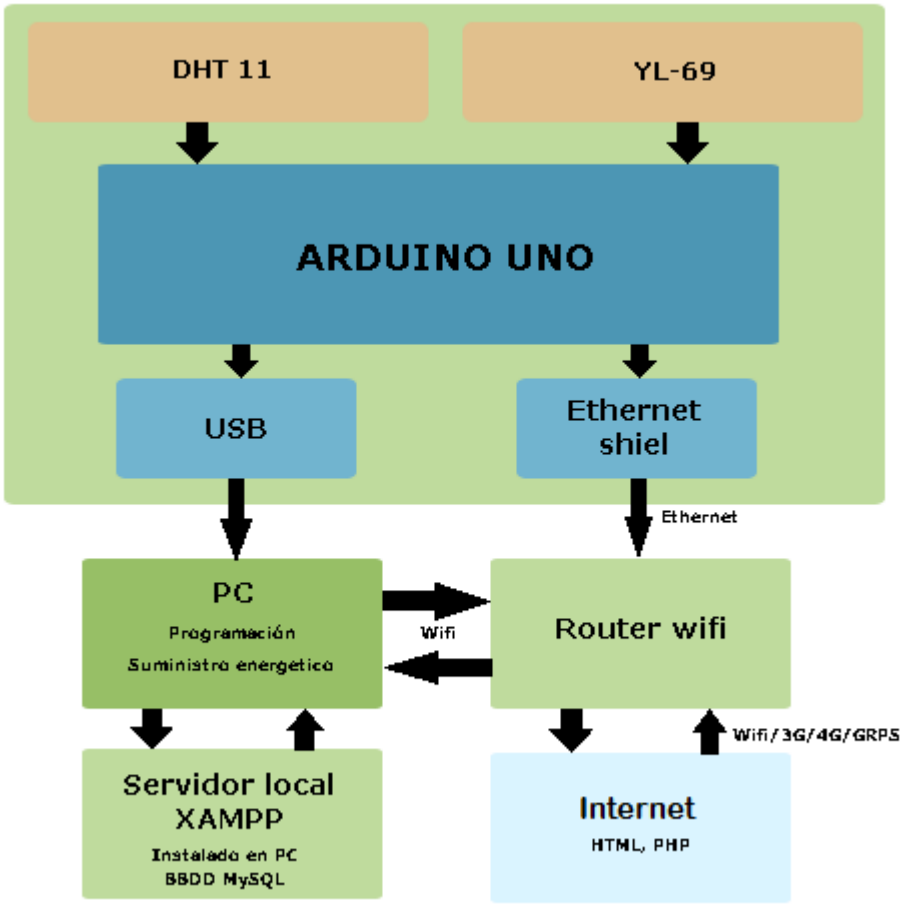

Ilustración 10. Esquema de comunicaciones entre los diferentes componentes del trabajo.

El sistema sensor está formado por la placa Arduino UNO junto con el Ethernet shield al que se le ha incorporado un sensor de temperatura y humedad relativa DHT 11 y un sensor de humedad de la tierra YL-69. Esta está conectada a internet vía Ethernet y permite que los registros lleguen a la base de datos MySQL. Aquí se

muestra un esquema del circuito del Arduino, donde Ethernet shield está ya incorporada en el Arduino UNO.

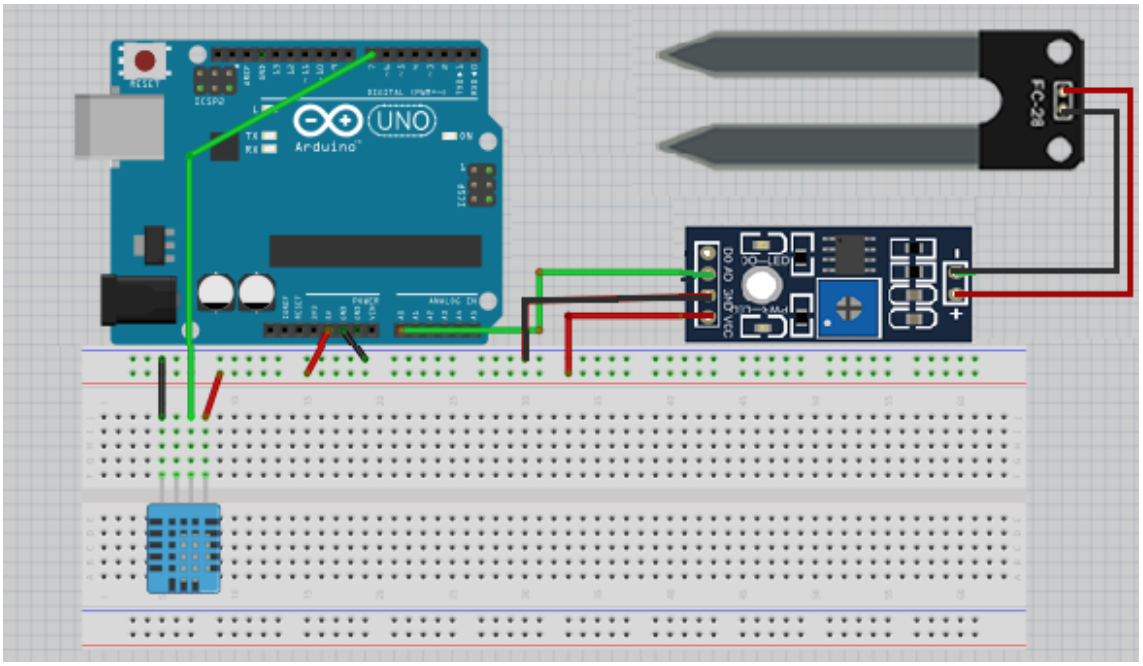

Ilustración 11. Esquema de conexiones de los sensores con la tarjeta de desarrollo Arduino UNO.

3.4. Hardware

Para comenzar a explicar cómo se ha desarrollado el trabajo se analizarán todos los componentes que conforman el hardware utilizado:

- Arduino UNO.
- Arduino Ethernet shield.
- Sensor de humedad: módulo YL-69.
- Sensor de temperatura y humedad relativa: módulo DHT11.
- Router inalámbrico Comtrend AR-5387un.
- Cable USB.
- Cable LAN.

La placa Arduino UNO junto con Arduino Ethernet shield serán los componentes principales que se encargarán de la transmisión de los datos. Para la recepción de los datos utilizaremos los sensores.

Al no tener una fuente de alimentación externa (ya que es un prototipo en proceso de desarrollo) utilizaremos un cable USB para la alimentación de nuestro sistema mientras realizamos las pruebas.

El router será el encargado de conectar nuestro sistema sensor a nuestra base de datos mediante una red local (telemática) conectándolo al mismo sistema con ayuda del cable LAN. También será el que nos permita el acceso a la World Wide Web (www) si disponemos de conexión a internet.

3.4.1. Arduino UNO

Este modelo de placa Arduino es el modelo estándar y más utilizado, desde que apareció en 2010 ha sufrido 3 revisiones por lo que el modelo actual es Arduino UNO R3.

Hay dos tipos de placas Arduino UNO ambas con el mismo modelo con la diferencia de que el microcontrolador una lo lleva montado en formato DIP[5] y la otra en formato SMD[6]. En este trabajo vamos a utilizar una con el formato DIP, dado que el microcontrolador se puede extraer fácilmente y llevarlo a otra placa.

El microcontrolador de la placa es un modelo ATmega328 de la marca Atmel de arquitectura AVR. Estos microcontroladores tienen diferentes memorias que se alojan en el:

- Memoria Flash: memoria persistente donde se almacena permanentemente el programa que se ejecuta, en este caso son 32 KB de almacenamiento. Los procesadores que viene ya incluidos en la placa tiene un poco menos de memoria debido al gestor de arranque que nos facilita la utilización de la placa de una forma sencilla y cómoda.
- Memoria SRAM: es una memoria volátil donde se almacenan los datos que en ese momento el programa necesita crear o manipular para su funcionamiento, en este caso son 2 KB de almacenamiento.
- Memoria EEPROM: Es la memoria que no se borra donde se almacenan los datos que queremos que permanezcan grabados después de apagar el microcontrolador, en este caso tenemos 1 KB de almacenamiento.

[5] (Dual in-line package,paquete el línea dual)
[6] (Surface Mounted Device, conponente de superficie de montaje).

La alimentación de la placa es otra característica de relevancia ya que si no la conectamos a una fuente de alimentación esta no funcionará. El voltaje de la placa es de 5V y podemos obtenerlos de diferentes maneras:

- Conectando la placa Arduino a una fuente externa, como una batería que se recomienda que ofrezca un voltaje de 7 a 12 voltios con el fin de obtener una estabilidad y seguridad eléctrica.
- Conectando la placa a nuestro ordenador mediante USB, para ello la placa lleva integrado un conector USB hembra.

La conexión USB de la placa Arduino ya hemos visto que sirve como alimentación pero también sirve para transmitir datos entre nuestro ordenador y la placa, y viceversa. Esta transmisión de datos se da gracias al protocolo USB que al ser demasiado complejo para nuestro microcontrolador ATmega328 necesita de otro chip que realiza la función de “traductor” del protocolo USB a uno serie más sencillo. Este es el chip ATmega16U2.

Otra de las características más importantes de la placa Arduino UNO son sus diferentes pines hembra, que son:

- Entradas y salidas digitales: Están situadas en la parte de arriba de la placa, van del 0 hasta el 13, este último pin lleva una resistencia interna incluida. La señal digital puede estar o encendida o apagada (LOW o HIGH). Los pines cero (RX) y uno (TX) se pueden utilizar para cargar el programa en la placa. Por ejemplo, se utilizan para parpadear un LED o; como entrada, un pulsador.
- Salidas analógicas: Son los pines 11, 10, 9, 6, 5 y 3, si os fijáis tienen una raya curva al lado, se denominan salidas PWM (Pulse Width Modulation) que realmente son salidas digitales que imitan salidas analógicas, modificando la separación entre los diferentes pulsos de la señal. La señal PWM puede dar diversos valores hasta 255, se utilizan, por ejemplo para variar la intensidad de un LED o hacer funcionar un servo. Hay que decir que estos pines funcionan como salidas o entradas digitales o como salidas analógicas.
- Entradas analógicas: Son los pines A0, A1, A2, A3, A4 y A5 (analog in). Se utilizan para que entre una señal de un sensor analógico, tipo un

potenciómetro o un sensor de temperatura, que dan un valor variable. También se pueden utilizar como pines digitales.

- Pines de alimentación:

  - GND: Son los pines a tierra de la placa, el negativo.
  - 5v: Por este pin suministra 5v
  - 3,3v: Por este pin suministra 3,3v
  - Vin: Voltaje de entrada, por este pin también se puede alimentar la placa.

- RESET: Por este pin se puede reiniciar la placa

IOREF: Sirve para que la placa reconozca el tipo de alimentación que requieren los shields

Pin sin utilizar: justo el pin a continuación del IOREF, el cual está sin etiquetar, actualmente no se utiliza para nada, pero se reserva para un posible uso futuro.

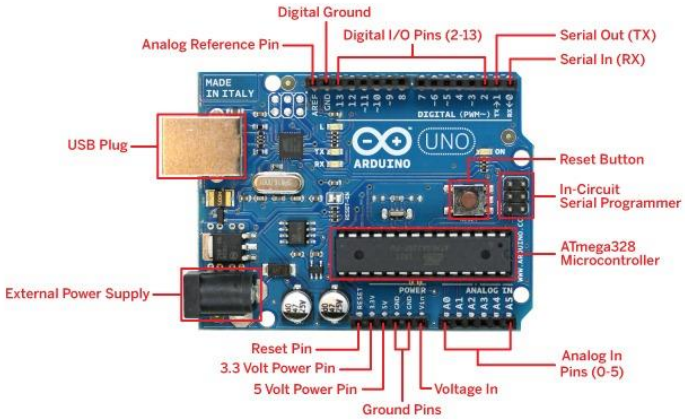

Ilustración 12. Esquema de conexiones y puertos del arduino UNO(http://manteniment-industrial.cat/descargas-manuales/) .

3.4.2. Arduino Ethernet shield

Este shield está pensado para los que le quieren añadir a la placa Arduino UNO la capacidad de conectarse a una red cableada TCP/IP. El chip controlador que incluye esta placa es: W5100 y se configura con la librería de programación: “Ethernet”, la cual ya viene por defecto en el lenguaje Arduino.

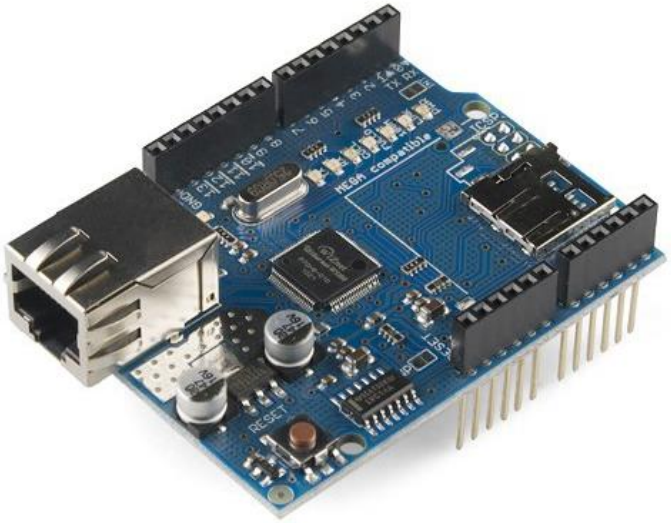

Ilustración 13. Imagen del Ethernet Shield.

Una vez conectado este shield sobre la placa UNO utilizaremos los pines de entrada y salida que ofrece el shield Ethernet. Estas entradas y salidas tienen exactamente la misma disposición y funcionalidad que las de la placa UNO.

El procedimiento de cargar nuestros programas en el microcontrolador de la placa UNO acoplada al shield Ethernet no varía respecto al realizado normalmente con una placa UNO independiente: primero debemos conectar la placa UNO a nuestro computador mediante el cable USB, y una vez cargado el programa, como siempre, podremos seguir alimentando la placa vía USB o bien desconectarla del computador y enchufarla a una fuente de alimentación externa. A partir de entonces, al conector RJ-45 del shield le podremos conectar un cable de red de tipo “estándar” si deseamos comunicar el shield a un switch o un router.

Este shield requiere 5 V para funcionar. Este voltaje lo aporta la placa UNO mediante el encaje del pin de alimentación correspondiente entre placa y shield.

La comunicación entre el chip W5100 y la placa UNO se establece mediante los pines 10, 11, 12 y 13 (vía SPI) por lo que estos pines no se pueden utilizar para otro propósito. Esto implica que realmente al usar este shield tenemos 4 entradas/salidas digitales menos.

Este shield también tiene una entrada para colocar una tarjeta microSD, la cual se podrá utilizar mediante la librería de programación “SD”, que viene por

defecto en el lenguaje Arduino. Al igual que ocurre con el chip W5100, para poder comunicarse con la tarjeta microSD la placa Arduino UNO utiliza el protocolo SPI, pero para ello esta vez emplea como pin el número 4 en vez del 10 (reservado al W5100). Es decir, emplea los pines 4, 11, 12 y 13.

El hecho de compartir el canal SPI entre el chip W5100 y la tarjeta microSD implica que no se pueden utilizar estos dos dispositivos a la vez. Si en nuestros programas queremos utilizar ambos elementos, deberemos tener en cuenta esto para programar nuestro código de forma correcta. Si, en cambio, no queremos utilizar uno de los dos dispositivos, hay que tener la precaución de desactivarlo explícitamente en nuestro código.

Este shield también dispone de su propio botón de “reset”, el cual reinicia tanto el chip W5100 como la propia placa Arduino

La placa incluye una serie de LEDs informativos:

- “PWR”: Indica que la placa y el shield reciben alimentación eléctrica.
- “LINK”: Indica la presencia de una conexión de red, y parpadea cuando el shield recibe o transmite datos.
- “FULLD”: Indica que la conexión de red es “full duplex”.
- “100M”: Indica la presencia de una conexión de red de 100 megabits/s –en vez de una de 10megabits/s–).
- “RX”: Parpadea cuando el shield recibe datos.
- ”TX”: Parpadea cuando el shield envía datos.
- “COLL”: Parpadea cuando se detectan colisiones de paquetes en la red.

3.4.3 Sensor de humedad: módulo YL-69

El módulo YL-69 es un sensor de humedad del suelo que usa la conductividad entre dos terminales para determinar parámetros relacionados con la humedad y conductividad.

EL modulo consta de dos partes: el circuito de control, YL-38 (a la izquierda de la imagen) y la sonda receptora, YL-69, que se introduce en la tierra para recibir la información (a la derecha de la imagen)

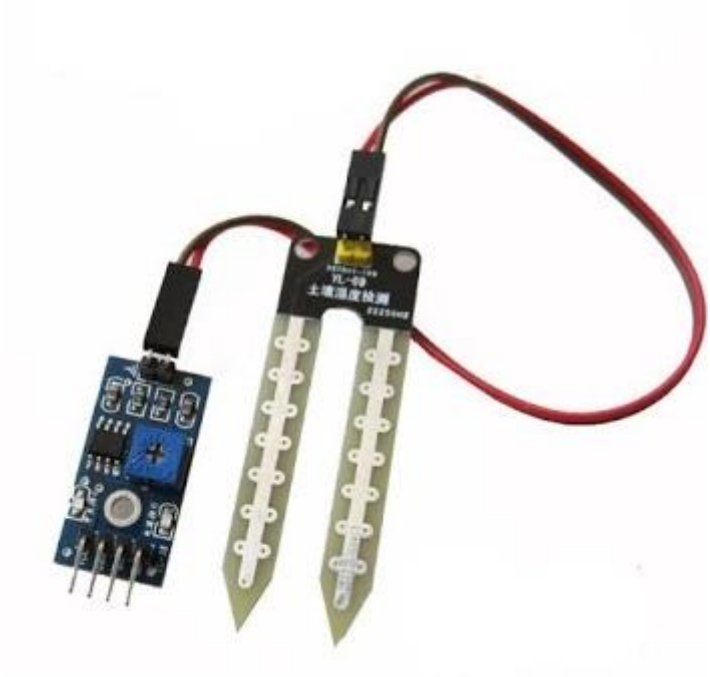

Ilustración 14. Imagen del sensor de humedad YL-38.

- YL-38: es el que posee las resistencias limitadoras de corriente y es el encargado de alimentar el módulo. Posee un amplificador operacional, específicamente el circuito integrado LM392. Este es el encargado de amplificar el pequeño diferencial de voltaje que se general cuando hay agua entre pistas del módulo. Aquí es donde se genera la señal de salida que puede ser del tipo analógica o digital. La señal digital oscilará entre los valores HIGH y LOW dependiendo de si hay agua o no sobre las pistas de la placa YL-69 y del grado de sensibilidad que calibremos. La señal analógica oscilará entre los valores 0 y 1023 dependiendo de la humedad del suelo. Este módulo tiene 6 pines: dos para la conexión al módulo YL-69, otros dos para la alimentación y otros dos para comunicarse con el Arduino.
- YL-69: Consiste en dos placas separadas entre sí por una distancia determinada. Ambas placas están recubiertas de una capa de material conductor. Si existe humedad en el suelo se creará un puente entre una punta y otra, el amplificador operacional transformará la conductividad registrada a un valor analógico que podrá ser leído por Arduino.

3.4.4. Sensor de temperatura y humedad relativa: módulo DHT11

EL módulo DHT 11 es un sensor de temperatura y humedad digital de bajo coste. Utiliza un sensor capacitivo de humedad y un termisor para medir el aire que

le rodea, mostrando los datos mediante una señal digital. Esto supone una gran ventaja frente a los sensores analógicos ya que en estos las fluctuaciones en el voltaje alteran la lectura de los datos. La toma de datos la debemos hacer con cautela ya que este sensor solo puede obtener nuevos datos cada 2 segundos.

El modelo que se utiliza en el trabajo es el sensor soldado en una placa con tres pines disponibles una resistencia pull-up y un condensador de filtrado soldados

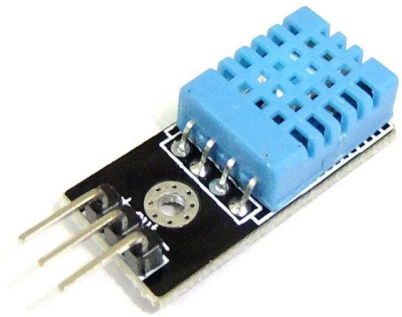

Ilustración 15. Imagen del sensor DHT11.

Tabla 2. Características técnicas del sensor DHT11.

Sensor	Rango de medida	Precisión	Resolución
Temperatura	De 20 a 90%RH	±5%RH	1
Humedad	De 0 a 50ºC	±2ºC	1

Por último vamos a ver los pasos que sigue el proceso de comunicación módulo DHT11 con la placa Arduino:

1. La placa Arduino envía una señal de arranque.
2. El módulo DHT11 se activa, dejando atrás el modo de bajo consumo, y le envía una señal al Arduino comunicándole que ya está activo (el tiempo invertido en este paso es de al menos un segundo).
3. Una vez se haya completado, el módulo DHT11 envía una señal de 40 bits que incluye la información de temperatura y humedad relativa.
4. Una vez recopilados los datos por el Arduino el sensor vuelve al modo de bajo consumo, esperando otra señal de arranque.

3.4.5. Router Comtrend AR-5387un

El Comtrend AR-5387un es un router inalámbrico ADSL2+ 802.11n. Dispone de cuatro puertos 10/100 Base-T Ethernet.

El punto de acceso WiFi soporta WEP, cifrado inalámbrico WPA / WPA2 y autenticación 802.1x. Tiene la opción de habilitar WPS en la página de configuración del router, que permite una fácil configuración de la seguridad inalámbrica.

Ilustración 16. Imagen del router inalámbrico Comtren AR-5387un.

3.5. Software

Para utilizar el hardware necesitamos un software que lo controle por lo que realizaremos un análisis de los diferentes programas que vamos a utilizar en este trabajo:

- Arduino IDE
- XAMPP
- Notepad ++

Arduino IDE es el programa que nos va a permitir interactuar con la placa Arduino UNO, para subir el código a la placa y para comunicarnos con la placa mediante el puerto serie.

XAMPP es el programa con el que vamos a establecer el servidor (Apache), la base de datos (MariaDB). También nos va a permitir la comunicación entre el portal web, la base de datos y el sistema sensor.

Notepad++ nos proporciona las herramientas necesarias para crear y editar las diferentes páginas web que forman el portal web.

3.5.1. Arduino IDE

Arduino tiene su entorno de desarrollo (IDE) basado en java, es decir, nos permite editar, compilar y subir el código a nuestra plataforma de prototipos. Además también permite la visualización del puerto serie estableciendo previamente la velocidad de conexión. El puerto serie nos permite comunicarnos con el Arduino tanto para mandar como para recibir información.

El entorno de desarrollo se puede extender gracias al uso de librerías que proporcionan funciones adicionales.

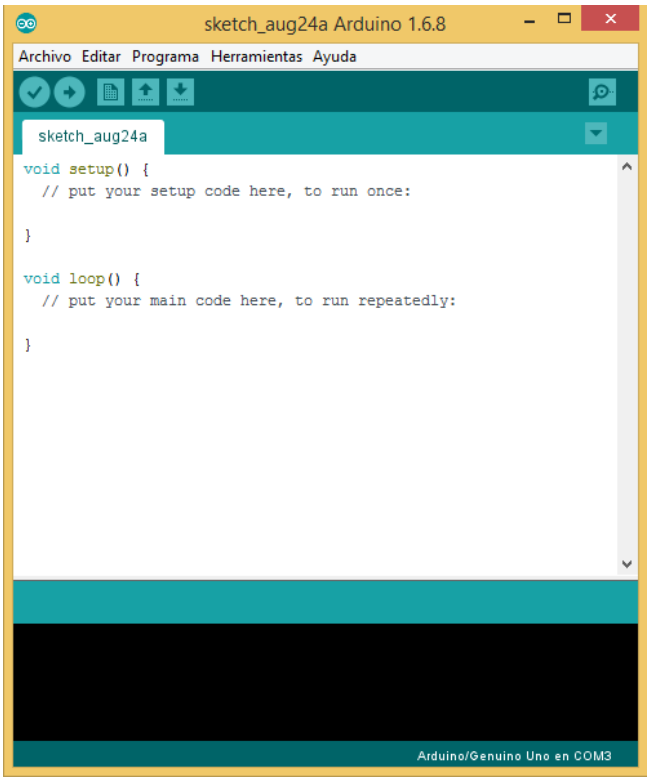

Ilustración 17. Imagen del entorno de desarrollo usado para Arduino.

La versión que se usa para la realización del trabajo es: Arduino-1.6.8-windows.

3.5.2. XAMPP

XAMPP es un servidor independiente de plataforma, software libre, que consiste principalmente en el sistema de gestión de bases de datos MariaDB (MySQL) con su respectivo gestor phpMyadmin, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl (en este trabajo solo utilizaremos el lenguaje PHP). Incluye también un servidor FTP como ProFTPD o FileZilla FTP Serve además de muchas más utilidades y herramientas.

XAMPP nos permite la creación y configuración de un host virtual (un portal web) que nos muestra el contenido de las diferentes páginas web que hemos editado. Uno de los pocos problemas que podemos encontrar es el relacionado con la seguridad de los datos, por lo que no es aconsejable utilizarlo en ambientes grandes o de producción. En el caso del prototipo del trabajo este software nos va facilitar el trabajo y al utilizarlo en una red casera el problema de la protección de

datos no es tan grave, aunque si se exporta a un sistema productivo este será uno de los requisitos mas importantes.

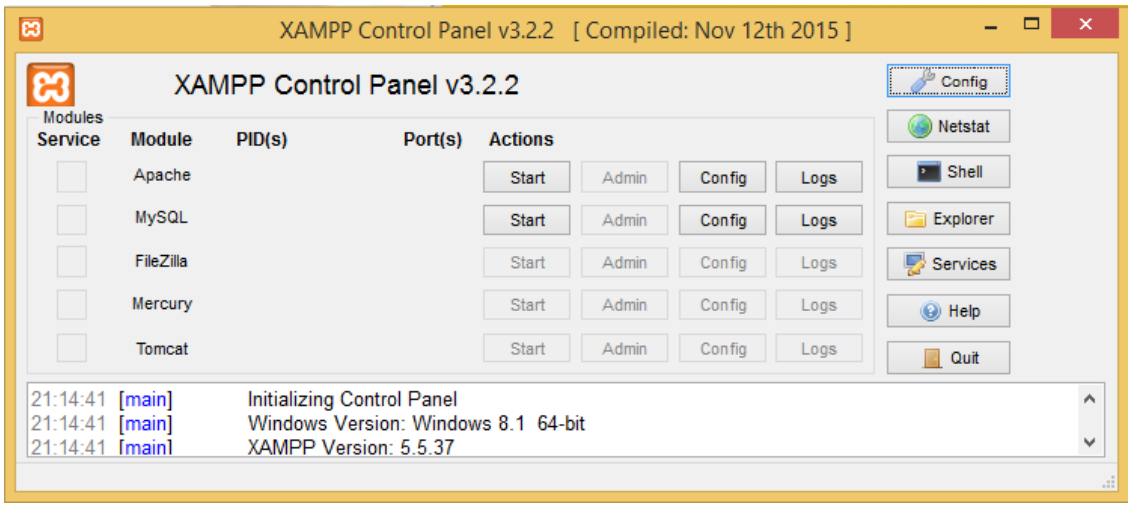

Ilustración 18. Imagen del panel de control de XAmpp.

La versión que se usa para la realización del trabajo es: xampp-win32-5.5.37.

Apache HTTP Server

El proyecto Apache HTTP Server, cuyo nombre proviene de la frase inglesa "a patchy server", es un software libre en continuo desarrollo y su objetivo es crear un sistema robusto, comercial y que la implementación del código fuente del servidor HTTP (web) este a libre disposición.

Este proyecto esta administrado por un grupo de desarrolladores (Apache Software Foundation) distribuidos por todo el mundo que utilizan internet y la Web para comunicarse planificar y desarrollar el servidor. Además de este grupo de administradores, los usuarios de Apache contribuyen con ideas para mejorar el proyecto. Desde el año 1996, es el servidor web más popular del mundo, debido a su estabilidad y seguridad.

MariaDB (MySQL)

MariaDB es un es un sistema de gestión de bases de datos derivado de MySQL que va a su compás, aunque MariaDB suele tener un retraso al liberar la versión estable, equivalente en nomenclatura a la de MySQL, para poder

implementar sus mejoras y realizar las pruebas pertinentes. Está desarrollado por Michael (Monty) Widenius (fundador de MySQL) y la comunidad de desarrolladores de software libre.

Para gestionar la base de datos podemos acceder al monitor serie y ejecutar diferentes comandos o utilizar un gestor de base de datos como phpMyadmin. Este gestor de base de datos está escrito el lenguaje PHP y tiene la función de administrar la base de datos a través de páginas web utilizando internet.

PHP

El código es interpretado por un servidor web (Apache HTTP Server, en este caso) con un módulo de procesado de PHP que genera la página web resultante.

Este lenguaje funciona en la mayoría de servidores web, sistemas operativos y plataformas sin coste. La implementación principal de PHP es producida por The PHP Group y sirve como estándar.

3.5.3. Notepad++

Es un editor de texto y código fuente libre que soporta gran variedad de lenguajes de programación. Tiene un formato simple e intuitivo.

4. Desarrollo de la aplicación

4.1. Instalación de los programas

El primer programa que se va a instalar es Arduino IDE. Para la instalación del mismo se accede a la página web de Arduino: https://www.arduino.cc. una vez dentro cliqueamos sobre la pestaña de descargas y se descarga el programa para el sistema operativo del ordenador en el que se trabaje.

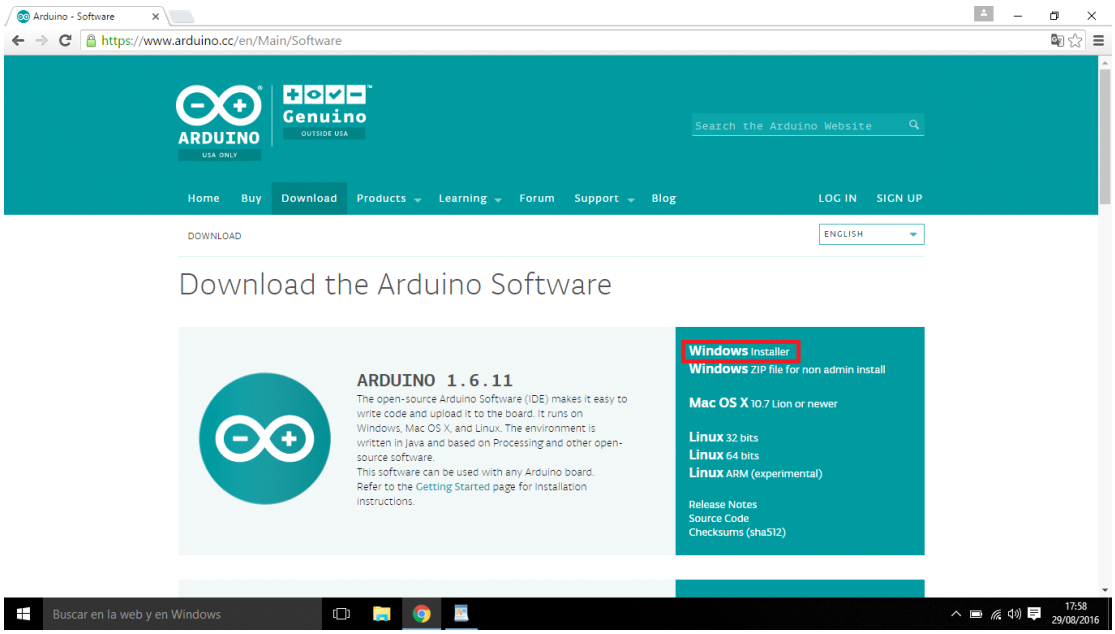

Ilustración 19. Web principal del proyecto Arduino. Detalle del lugar de descarga del instalador de Windows.

En este caso se descarga el instalador para el sistema operativo Windows ya que es el que estamos utilizando.

Una vez descargado el ejecutable lo abrimos, seguimos las instrucciones, instalamos el programa y ya se puede trabajar con él.

El segundo programa que se va a instalar es XAMPP. Para la instalación del mismo realizamos lo mismo que con Arduino IDE, se accede a la página web oficial de XAMPP: https://www.apachefriends.org/es/index.html, y pinchamos en la pestaña descargas y elegimos la versión más adecuada para nuestro sistema operativo.

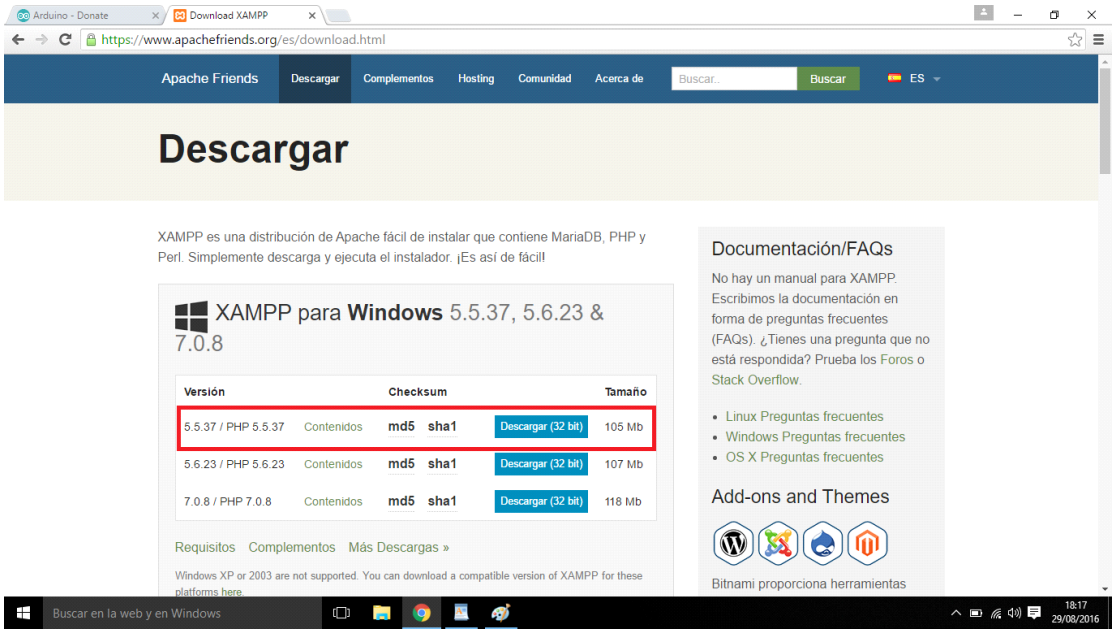

Ilustración 20. Web del proyecto XAmpp. Detalle del lugar de descarga del instalador de Windows.

En este caso se descarga la versión 5.5.37 / PHP 5.5.37 (Incluye: Apache 2.4.17, MariaDB 10.1.13, PHP 5.5.37, phpMyadmin 4.5.1, Open SSL 1.0.2, XAMPP Control Panel 3.2.2, Webalizer 2.23-04, Mercury Mail Transport System 4.63, FileZilla FTP Server 0.9.41, Tomcat 7.0.56 (with mod_proxy_ajp as connector), Strawberry Perl 7.0.56 Portable) ya que es la versión actualizada más antigua por lo que será más estable que las siguientes versiones.

Descargado el ejecutable se inicia y se siguen las instrucciones que indica el instalador, teniendo en cuenta los recursos que vamos a necesitar de los muchos que incluye el programa XAMPP y la carpeta raíz en la que se instale el programa:

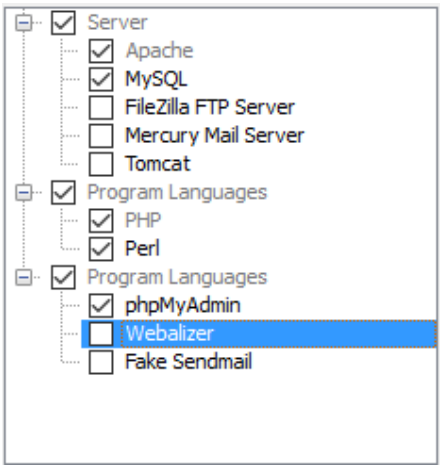

Ilustración 21. Detalle del software a instalar en el servidor del proyecto XAmpp.

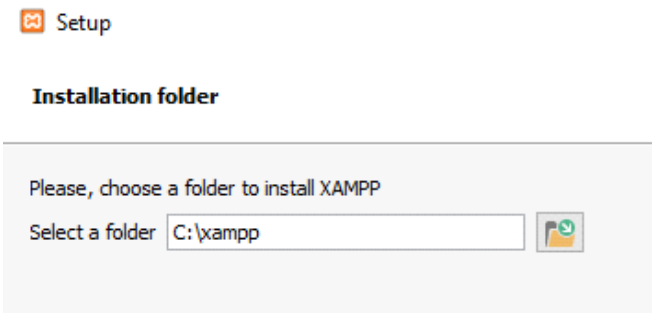

Ilustración 22. Detalle de la elección del directorio de instalación de XAmpp.

El último software que se va a instalar es Notepad++ que es el que permite la creación y edición de los diferentes archivos que necesitamos para establecer la web. Para la instalación del mismo simplemente se abre el archivo ejecutable, proporcionado por el curso.

4.2. Montaje del sistema sensor

Como se ha mencionado anteriormente el sistema sensor en este trabajo debe de ser portátil por lo que la mayoría de los componentes deben estar encapsulados.

Para el montaje del sistema se necesita un recipiente o capsula que proteja nuestro prototipo. En este caso haremos uso de una cajita de madera en la que encaja el Arduino a la perfección. Hay un pequeño inconveniente, que no tenemos salida para el cableado, para solucionarlo le haremos un pequeño orificio con una sierra. En el caso de que el trabajo se convierta en un sistema de producción se desarrollará una caja adecuada cumpliendo las normativas de protecciones de exteriores como las series IPH de la Comisión Electrotécnica Internacional.

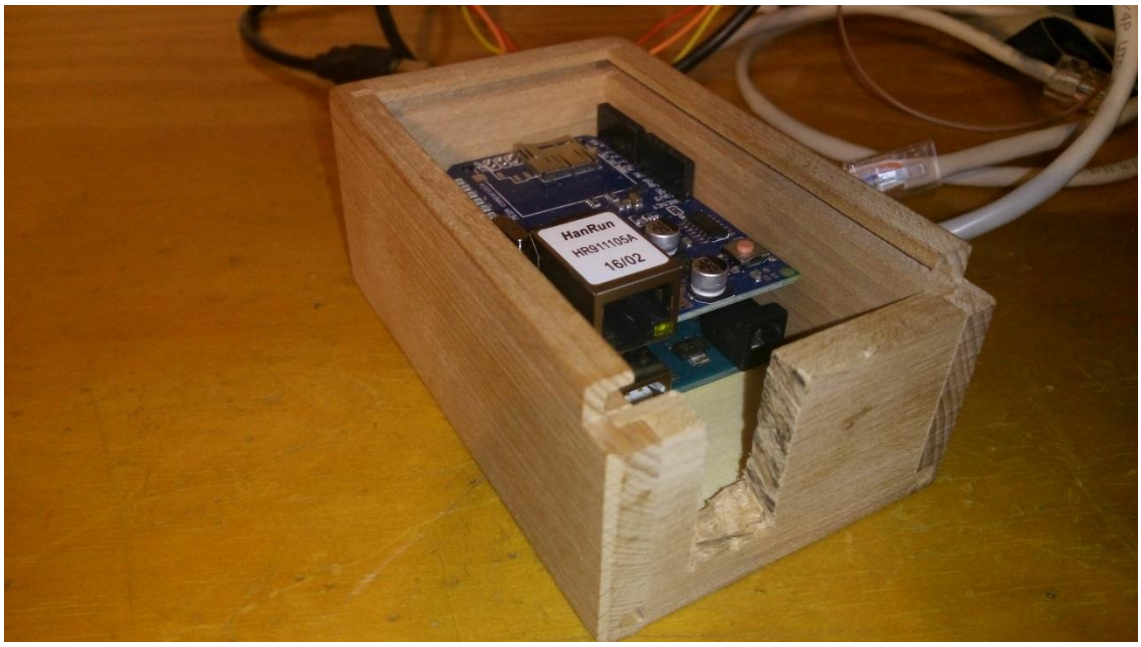

Ilustración 23. Vista de la caja y el Arduino junto con todo el material desarrollado para este trabajo. Nótese los cortes laterales para facilitar la conexión de los cables.

Una vez introducido el Arduino en su interior pasamos al cableado. Al tener dos sensores del mismo voltaje (5V) y solo un pin de salida de ese voltaje tendremos que realizar un empalme de cables para darles corriente a ambos sensores. Para transmitir la información utilizaremos los pines A0 (analógico) para módulo YL-69 y 7 (digital) para el módulo DHT 11.

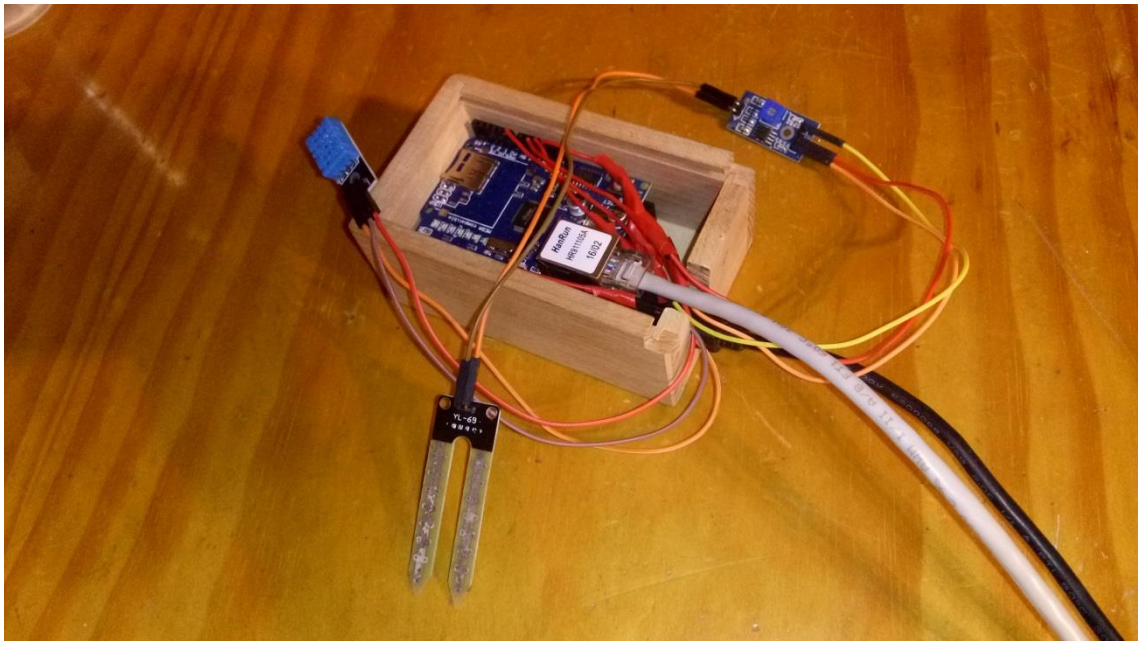

Ilustración 24. Vista del sistema de sensores montado con Arduino. Detalle que incluye los cables conexión y los sensores.

Los sensores deben de ir fuera de la caja ya que se necesita que recopilen información sobre nuestro cultivo, por eso irán lo más cerca posible de él. El módulo DHT 11 que se colocara cerca del entorno del cultivo, anclado a la caja de madera, el módulo YL-69 que se introduce, parcialmente, dentro de la tierra y el módulo YL-38 lo anclaremos a la caja de madera junto al DHT 11.

La caja se sitúa cerca del cultivo, ya que el cableado del módulo YL-69 es muy corto y para la conexión del shield con el router se puede utilizar un cable LAN más largo.

Para cerrar el apartado del montaje se va a mostrar una tabla con la que se confirmara que el precio del sistema sensor es reducido. Estos precios son variables, según el proveedor. En la siguiente tabla se exponen los costes del prototipo según los precios de compra a principios de 2016.

Tabla 3. Presupuesto del proyecto

Componente	Precio
Arduino UNO	14.95 €
Arduino Ethernet Shield	12.95 €
Módulo YL-69 (sensor de humedad)	1.31 €
Módulo DHT 11(sensor de temperatura y humedad relativa)	2.56€
Comtrend AR-5387un	15.00€
Cajita de madera	3.50€
TOTAL	50.27 €

4.3. Programación del Arduino

Para que el Arduino cumpla con su función se le tienen que dar una serie de instrucciones que son el código de programación.

El código para este trabajo utiliza diferentes librerías (conjunto de instrucciones codificadas que ofrecen mayor facilidad a la hora de programar). Algunas de ellas vienen incluidas en el programa Arduino IDE, otras las se tienen que descargar de internet e incluirlas dentro del programa, como la librería DHT 11.

El primer código que se implementa es el relacionado con la recogida de datos de los sensores y su impresión en pantalla mediante el puerto serie (como comprobación de que funcionan):

```
//declaramos las librerías y declaramos los pines de entrada
void setup()
//iniciamos el puerto serie y el sensor dht
void loop()
//declaramos todas las variables que vamos a usar asignándole los
//valores obtenidos por los sensores
//mostramos los valores de las variables en el puerto serie
//esperamos 5 segundos hasta la siguiente toma de datos
```

Para comprobar si el código es válido se compila, si lo es, se sube el código a la placa y se abre el puerto serie para verificar la entrada de los datos.

4.4. Comunicación

4.4.1. Identificación de dispositivos

Para establecer la comunicación entre los dispositivos estos deben de reconocer a quien tienen que mandarle la información. Para esto se utilizan los protocolos de identificación. En este caso se usa el protocolo IP.

En la red local todos los dispositivos están identificados con una IP, que según el DHCP puede ser fija o dinámica. Esta IP es privada, pertenece solo a la red local. En este caso la IP privada es 192.168.1.244.

Fuera de la red local no podemos acceder al portal utilizando un IP privada, por lo que haremos uso de la IP publica para acceder. En este caso la IP es 17.216.102.55. Para que esto funcione necesitamos redireccionar los puertos, esto nos permite que un usuario externo a la red local pueda tener acceso a un puerto en la dirección privada.

Para realizar esto tenemos que introducirnos en el router (mediante su ip privada, normalmente es 192.168.1.1) y buscar la opción de reenviar puertos. En este caso el puertos de comunicaciones con el servidor son el 80 y el 447 por lo que rediccionaremos estos dos puertos.

Otra forma de reconocimiento es mediante el DNS. Que en este caso es agricusion.hopto.org.

Para el establecimiento de el DNS primero se debe acceder al router y en la configuración avanzada buscamos la pestaña de DNS dinamico y entramos. Esta página permite utilizar a los usuarios de Internet un nombre en lugar de una dirección IP para acceder a sus servidores virtuales. La página nos pedirá el proveedor de DNS dinámico y información referente a nuesto perfil en el servidor del proveedor. Activando esta opcion y registrandonos en un proveedor de DNS dinámico como NO-IP.com ya se podra acceder al portal mediante el DNS.

4.4.2. Flujo de datos

En este apartado se explicara el recorrido que llevan los datos desde la toma de datos con los sensores hasta la visualización de los mismos en el portal web.

Arduino-Base de datos

En esta fase intervienen tres partes: el cliente, el Arduino, el servidor programado en PHP y el servidor de datos MySQL.

Para subir datos desde el Arduino a la base de datos se necesita un espacio libre dentro de la misma base de datos. Para crear un espacio dentro de la base de datos exclusiva para el Arduino se utiliza phpMyadmin, herramienta que gestiona la base de datos MariaDB (MySQL) a través de páginas web.

Podemos acceder a esta herramienta de diferentes formas, en este trabajo accederemos a ella con la ayuda de un navegador web. Para esto primero debemos iniciar el servidor Apache y MySQL.

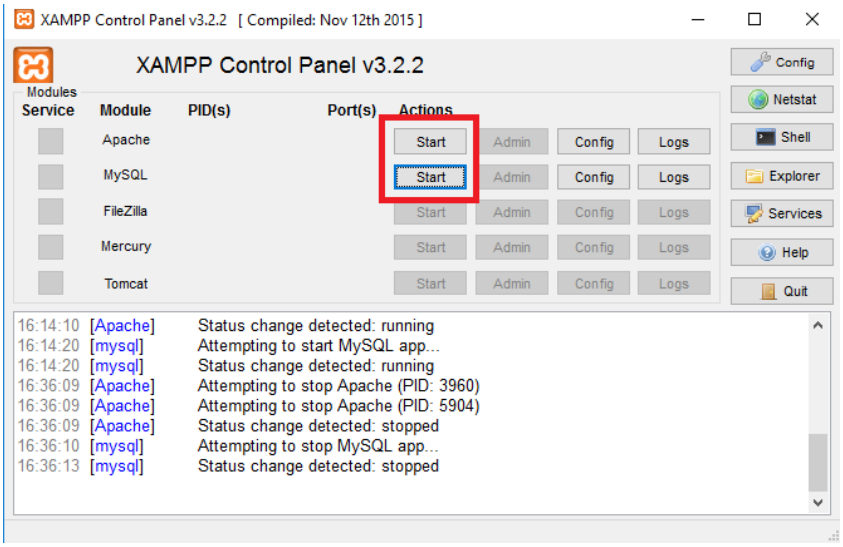

Ilustración 25. Detalle del panel de control de XAmpp. Se marcan los servicios a mantener activados.

Una vez iniciado el servidor abrimos el navegador web y en la barra url y se esribe: http://localhost/phpmyadmin/. Con esta dirección se accede al servidor local localhost (con una IP privada 192.168.1.11), y dentro de este se entra en phpmyadmin.

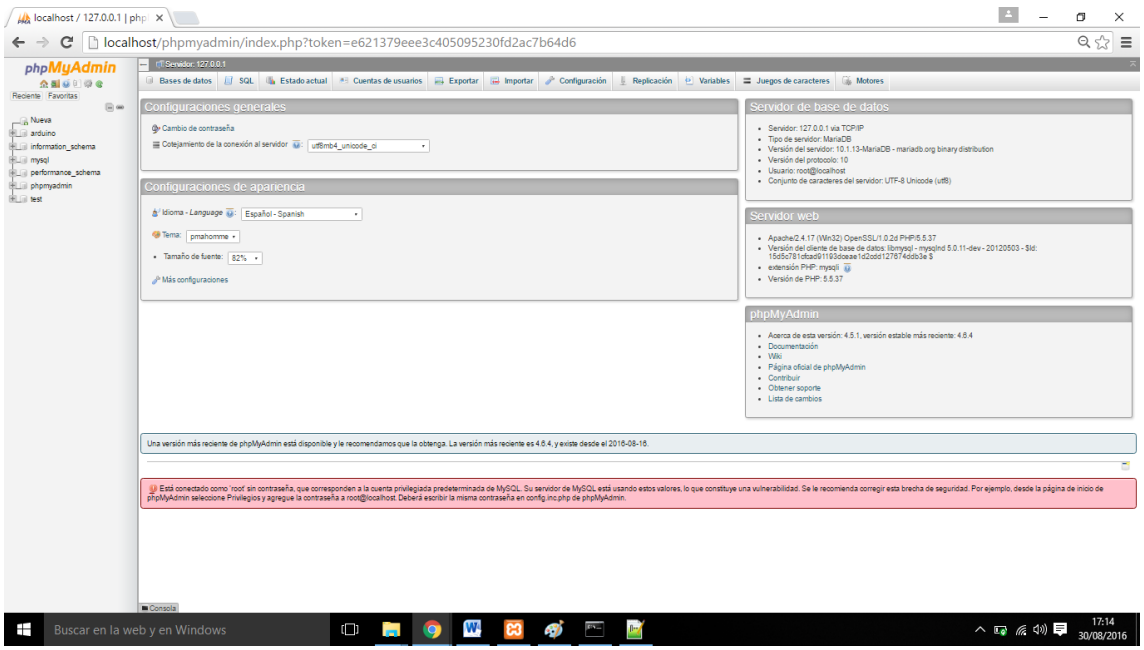

Ilustración 26. Ejemplo de la ventana de control de Php de XAmpp.

Aquí se ve que se abre la ventana principal y que esta pertenece ya a un usuario, que se llama root. Para poder acceder al usuario que queramos tenemos que cambiar el codigo PHP dentro del archivo de configuración, config.inc.php. (lo encontramos en la siguiente ubucación: C:\xampp\phpMyAdmin) se cambia el modo

de identificación cambiando el comando “config” por “cookie” donde el usuario deberá intriducir el nombre y la contraseña para acceder a la aplicación.

Provechando que se abre la aplicación con el ususario root, el administrador que tiene todos los permisos, se va a crear un usuario nuevo llamado Arduino, al que se le van a dar todos los permisos de administración como el usuario root. Para esto se clica en la pestana de cuentas de usuarios y entramos en el administrador de usuarios y pinchamos en la opcion de agregar cuenta de usuario. Se le asgina el nombre, la contraseña y le damos todos los privilegios.

Ya creado el nuevo usuario salimos de la sesion del usuario principal e introduciomos nuestro nuevo usuario con su contraseña. Accdemos a la página principal y clicamos en la pestaña de base de datos y creamos una base de datos que sera el espacio reservado para almacenar los datos que aporta el arduino. A la base de datos le vamos a poner le nombre de arduino, como al usuario.

Una vez creada la base de datos vamos a crear una tabla que almacene la informacion de los sensores de forma ordenada. Para esto entramos dentro de la base de datos y pinchamos en la opción crear tabla que la vamos a llamar datos. La tabla tendra 4 columnas:

- ID: identificaion de la medida (INT,clave principal)
- Temperatura: valor del sensor (INT)
- Humedad relativa del aire: valor del sensor (INT)
- Humedad: valor del sensor (INT)
- Fecha: fecha y hora en la que se ha tomado la muestra (TIMESTAMP, el valor predeterminado de este atributo es CURRENT_TIMESTAMP, cada vez que insertemos una muestra de datos se grabara la hora a la que se ha producido)

De esta manera hemos creado un usuario y una base de datos solo para nuestro sistema sensor.

Como ya tenemos un espacio en el servidor solo para nuestros datos ahora procedemos a la programación del mismo para que se pueda comunicar el arduino. Para esto necesitamos dos archivos “.php”:

- El primer archivo (identificacion.php) que vamos a crear es el que nos permite establecer la conexión con la base de datos, dandole la informacion necesaria para que entre dentro del servidor con un usuario y contraseña determinadas, y eligiendo la base de datos donde queremos almacenar la información.

```
<?php
   //nombre o IP del servidor
   //nombre del usuario
   //nombre de la contraseña
   //nombre de la base de datos
   // Conexión con la base de datos
 ?>
```

- El segundo archivo (comunicacion.php) que creamos es el que se encarga de subir los datos recibidos del Arduino a la base de datos en al que hemos accedido con el archivo anterior. Los datos los vamos a pasar mediante una sentencia GET.

```
<?php
// Importamos la configuración de identificacion
// Leemos los valores que nos llegan por GET y los guardamos en variables
// Esta es la instrucción para insertar los valores en la tabla sensores donde cada
//atributo toma su valor
// Ejecutamos la instrucción
//cerramos la conexion
?>
```

Estos dos archivos los tenemos que ubicar en el directorio php de nuestra pagina web (C:\xampp\htdocs\php) para que cuando el arduino mande la información los archivos .php le indiquen donde tiene que almacenarla.

Con el servidor ya programado vamos a pasar al código del Arduino conectandolo a un servidor remoto que tenemos en nuestro ordenador (servidor). Para ello utilizaremos el código escrito en la fase anterior y lo modificaremos.

```
//Incluimos las librerias necesarias para que funcionen nuestros sensores y la shield
//declaramos los pines de entrada como en la fase anterior
//en las siguientes lineas de código declaramos la dirección
```

```
//mac de nuestro arduino, que en este caso es aleatoria junto
//con la direccion ip de nuestro arduino (le damos tambien una
//aleatoria y la ip del servidor (el router)
void setup()
//en las siguientes lineas iniciamos el puerto serie la shield
//y el sensor DHT 11
void loop()
//declaramos las variables que contienen los datos de los sensor
//la siguiente linea de codigo hace que el arduino se conecte al router
//por el puerto 80 (que lo tenemos abierto)
// GET para enviar
//los datos a la base de datos mediante los archivos .php
//cuando la transferencia de datos haya terminado se
//se desconecta del servidor
//esperamos 5 segundos para la siguiente toma de datos
```

Cuando hallamos modificado el código, lo subimos a la placa Arduino y comprobamos mediante el puerto serie que la comunicación entre la base de datos y la placa Arduino funciona a la perfección. Podemos comprobar que tenemos registros de datos cada 5 segundos, que es lo que le hemos progamado a nuestro sistema sensor.

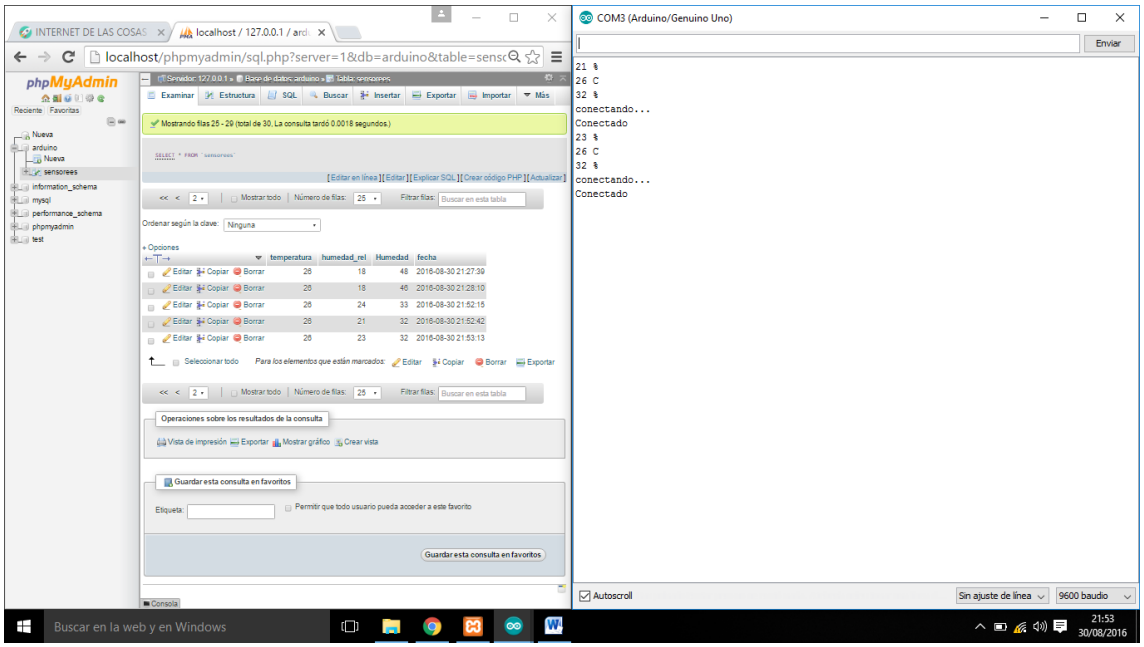

Ilustración 27. Detalle de la ventana de control de Php, de la información en MySQL y de la ventana de retorno de lectura del puerto COM del Arduino.

Base de datos – cliente ligero

En esta fase intervienen dos partes: el cliente, usuario del portal, el servidor programado en PHP y el servidor de datos MySQL.

Este tramo de la comunicación enlaza los registros guardados en la base de datos con el usuario que demanda esos registros. El usuario manda una petición que el servidor se encarga de responder adecuadamente, según la consulta.

La comunicación entre estas dos capas se basa en el lenguaje PHP que trabaja en el lado del servidor. Para la comunicación entre ambos vamos a utilizar el método POST del protocolo HTTP que lo invocamos en los archivos HTML. Con este el usuario envía al servidor unos parámetros que referencian a los datos que pide el mismo.

Realizado el envío de datos mediante el método POST el servidor busca en la base de datos, mediante consultas SQL, la información y la almacenamos en una matriz que luego usaremos para: la visualización de una tabla para mostrar los datos, apoyándonos en la librera Boostrap y a visualización de diferentes graficas mostrando la variación de las magnitudes en el tiempo, apoyándonos en la librería Highcharts.

Este servicio de visualización es el objetivo principal del trabajo junto a otras funciones que ofrece el portal web. Otra de las funciones que también comunican el cliente ligero del usuario con la base de datos es el gestor e usuarios que se ha creado.

Esta parte de la comunicación es posible gracias al método POST y a la variable superglobal $_SESSION que incluye PHP. Esto facilita mucho el diseño web ya que para un mismo archivo hay infinidad de representaciones según la base de datos y la consulta seleccionada.

Las sesiones son una forma sencilla de almacenar o consultar datos para usuarios de manera individual usando un ID de sesión único. Esto se puede usar para hacer persistente la información de estado entre peticiones de páginas. Los ID de sesiones normalmente son enviados al navegador mediante cookies de sesión.

Cuando una sesión se inicia, PHP recuperará una sesión existente usando el ID pasado (normalmente desde una cookie de sesión).

También hay otra función con la que nos comunicamos mandado correos electrónicos, mail (). El formulario de contacto que se incluye en el portal está basado en esta función. En este caso manda un correo electrónico al correo oficial del portal.

Para establecer la comunicación entre estas dos capas necesitamos iniciar el servidor apache y MySQL.

A continuación se describe el comportamiento que tienen los diferentes archivos que se encargan del desarrollo de la web.

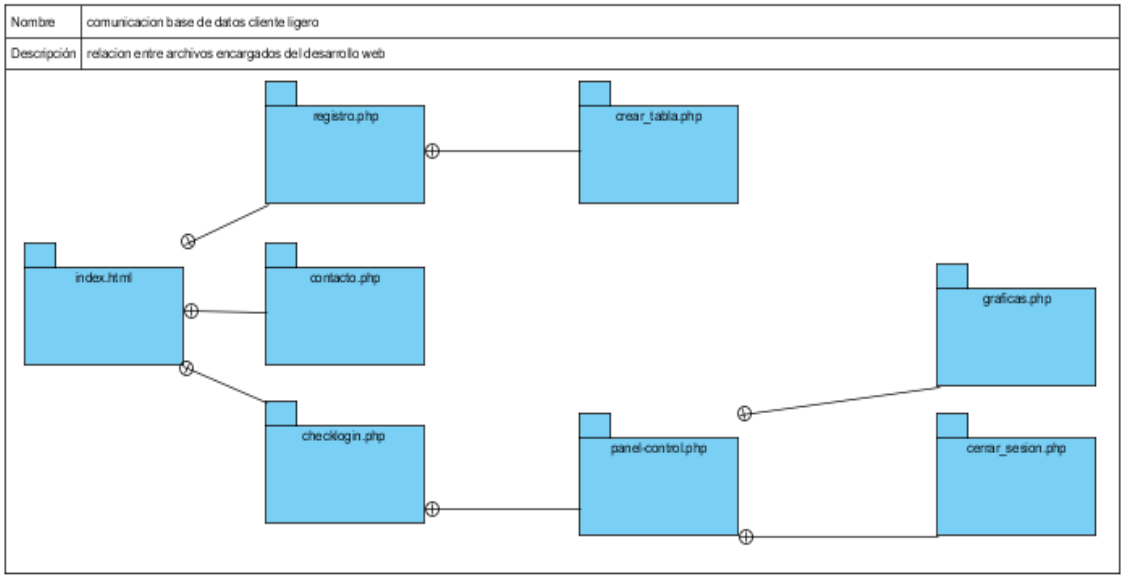

Ilustración 28. Comunicación de archivos encargados del desarrollo web

- index.html(incluye las librerias boostrap). Cuando el ususario entra en el cliente ligero y accede al portal mediante la direccion IP o DNS el navegador abre este archivo. Este archivo es el encargado de contener la pagina principal del portal. En la pagina principal se encuentran diferentes opciones como contacto, registro o acceso. Estas tres opciones estan referenciadas a dierentes archivos PHP, ya que son formularios que se implementan para enviarle la informacion a la base de datos mediante el metodo POST, excepto contacto.php que no interactua con el servidor.

- Contacto.php. este archivo recibe el formulacio de datos de la pagina principal y se encarga de mandar un correo electronico a la dirección de correo electronico del portal.
- registro.php. Este archivo se conecta con la base de datos y recibe el formulario de datos de la pagina princial y se encarga de registrar el usuario en una tabla (usuarios) dentro de la base de datos y crear otra trabla (la tabla tiene el nombre del usuario) para contener los registros del sistema sensor.
- checklogin.php. Este archivo se conecta con la base de datos y recibe el formulario de datos de la pagina principal y comprueba que los datos introducidos se encuentran dentro de la tabla usuarios. Una vez comprobado inicia sesion con el nombre de usuario que le hemos introducido e incluye en su codigo el archivo panel-control.php. si los datos introducidos no estan en la base de datos rediccionara al archivo index.html.
- panel-contro.php(incluye las librerias boostrap). Este archivo se incluye dentro de checklogin.php. Este se conecta a la base de datos y es el encargado de contener la página de perfil del usuario. Este archivo incluye un formulario de datos en el que introducimos dos fechas, intervalo de tiempo del que queremos los datos, estas dos fechas son buscadas en la base de datos y si encuentra registros que este contenidos entre estas dos fechas los guarda en una matriz de datos que utilizaremos posteriormente. Esta matriz de datos la representaremos de dos formas mediante una tabla y mediante 3 gráficas. Las tres graficas las llamaremos incluyendeo el código de graficas.php.
- Graficas.php(incluye las librerias Hihgcharts). Este archivo se encarga de se la representación de las gráficas mediante 3 funciones JavaScript, una para cada gráfica.
- Cerrar_sesion.php. Este archivo es el encargado de cerrar la sesion del usuario y redireccionar a la pagina principal index.html.

Para profundizar en el código de los archivos anteirores se deja una copia de todos estos en el CD, junto al documento.

4.5. Plataforma web de control

Para poder iniciar la comunicación con la base de datos, el usuario debe de enviarle una petición con el fin de obtener o enviar unos datos, iniciar una sesión, crear un usuario… Esta petición la enviamos mediante un formulario que invoca el método GET o POST.

Para controlar el envío de peticiones se van a utilizar formularios dentro de archivos HTML que recogerán la información que introduzca el usuario y la enviran posteriormente.

La plataforma web esta formada por una serie de archivos que interactuan entre si. Los archivos HTML son los que definen el contenido de la pagina web, CSS define los estilos de los elementos, PHP define la comunicación con el servidor, JavaScript define el dinamismo de la página. La imagen siguiente representa los archivos que se encuentran en el directorio raíz de nuéstra página web.

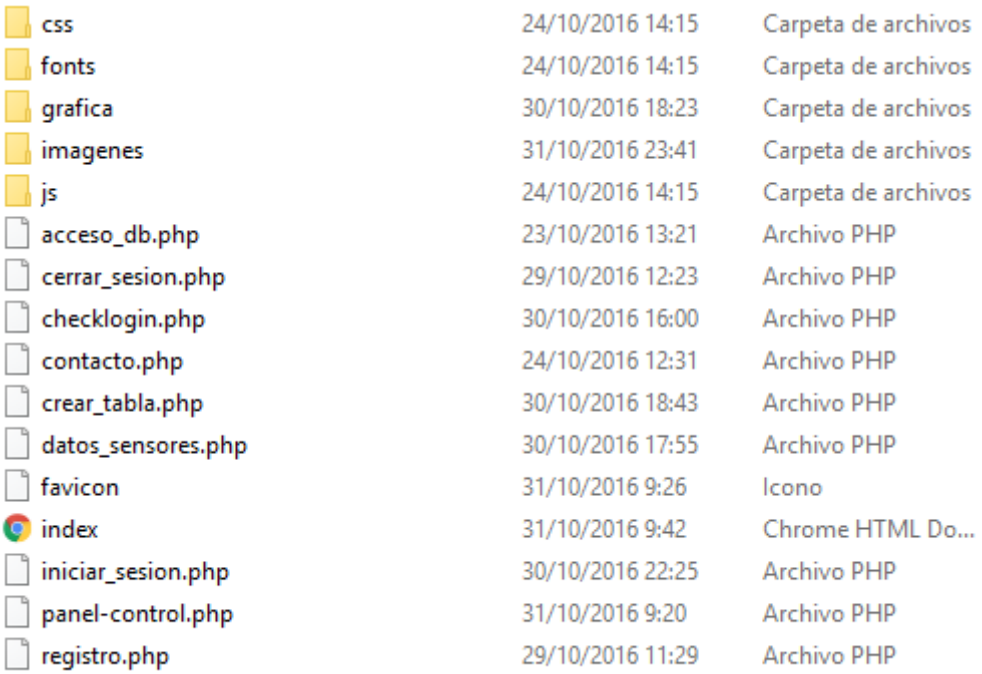

Ilustración 29. Estructura de archivos de la página web que incluye todos los módulos php.

En el caso de la base de datos y el sistema sensor la comunicación se inicia cuando el Arduino realiza un registro de información. Es entonces cuando el Arduino envía la petición con el método GET o POST en la que se incluyen los datos del registro.

5. Ejemplo de uso

En este ejemplo se van a mostrar los diferentes casos de uso que puede realizar el portal web.

Para acceder al portal, lo podemos hacer de diferentes maneras:

- Red local: Si estás conectado a la red local se necesita la ip privada del dispositivo que almacene el servidor asignada por el servidor DCHP.
- Red global: Para acceder desde cualquier conexión externa a nuestra red local se necesita la ip publica del dispositivo que almacene el servidor o el identificador DNS.

Una vez se accede vemos en la página principal en la que se ofrece una información sobre el portal y las diferentes opciones que tienen los usuarios sin tener ninguna restricción.

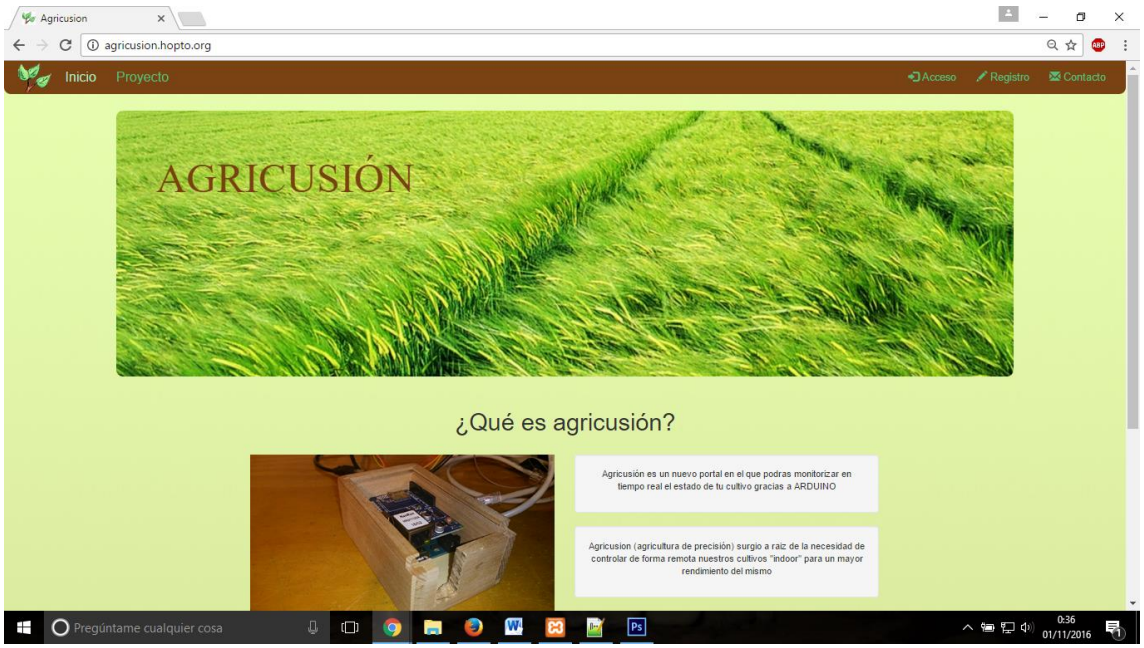

Ilustración 30. Página web principal del trabajo.

Exploramos la página y encontramos las opciones que nos ofrece:

- Contacto: esta pestaña despliega una ventana con un formulario para poder enviar un correo electrónico para contactar con el autor del portal

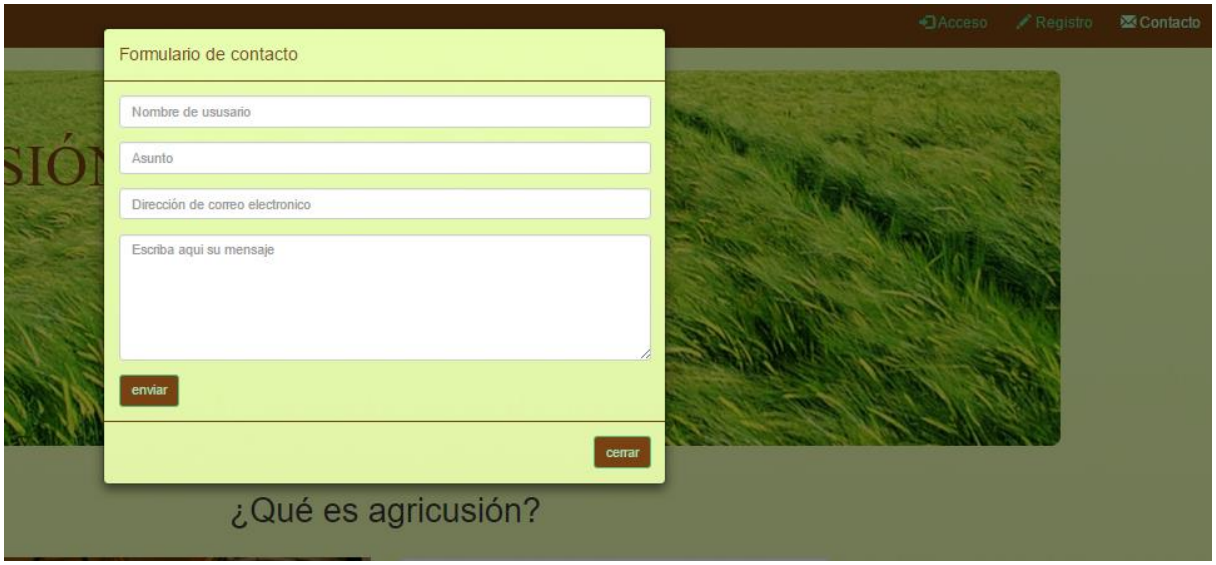

Ilustración 31. Detalle de la ventana emergente para la creación de un nuevo contacto.

- Registro: Esta pestaña despliega una ventana con un formulario de registro que crea un fila, con el nombre del usuario y sus datos, en la tabla de usuarios y a su vez crea una tabla donde se almacenaran los datos de los sensores.

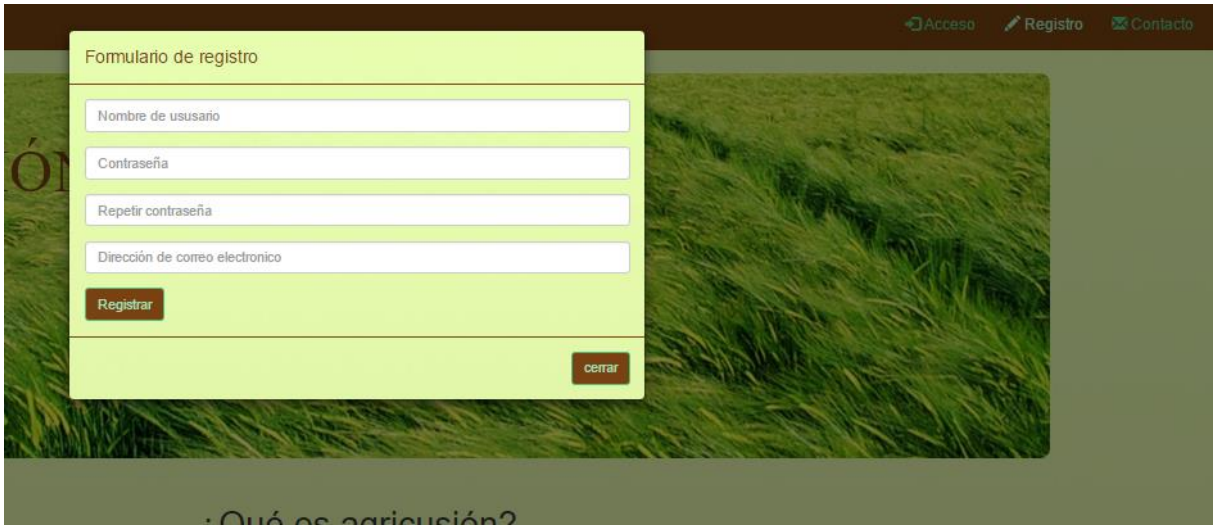

Ilustración 32. Detalle del formulario de registro de un nuevo usuario en la página web.

- Acceso: Esta pestaña despliega un pequeño formulario de acceso que inicia la sesión del usuario que introduce el usuario, siempre y cuando este registrado.

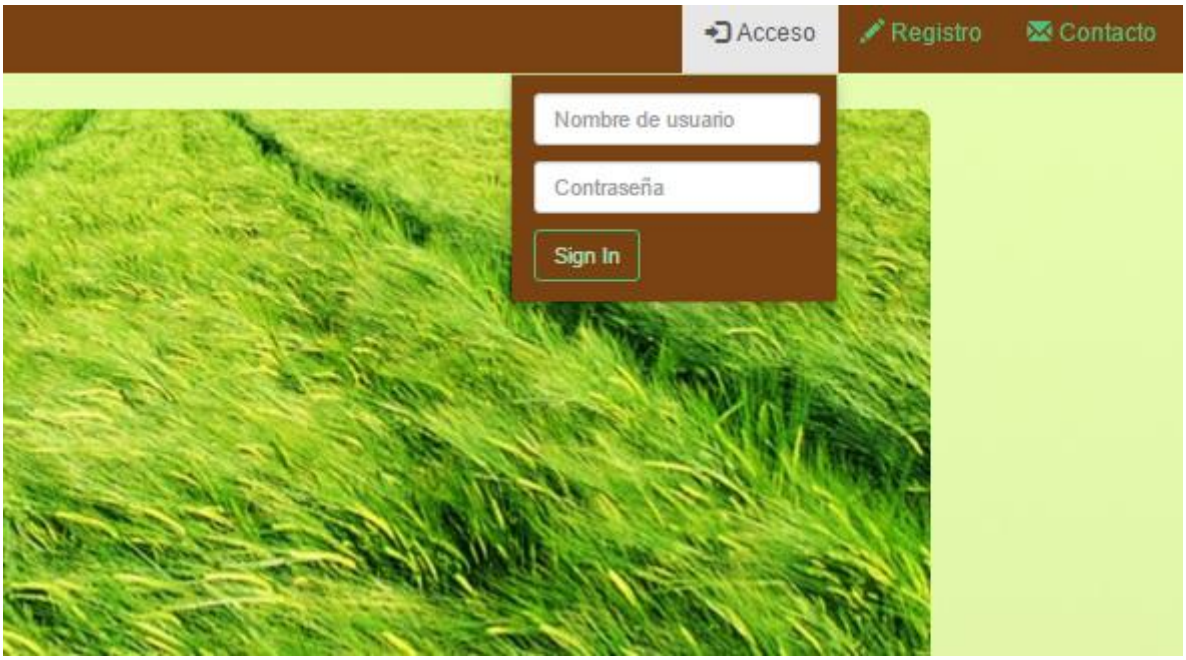

Ilustración 33. Detalle del método de acceso de los usuarios a la página web.

Accedemos con un nombre de usuario y contraseña validos y entramos en el panel de control del usuario. Tiene un diseño sencillo y original. Aquí se puede observar que hay un formulario en el que nos piden que introduzcamos la fecha y hora del inicio de la toma de datos y la fecha y hora del final de la toma de datos.

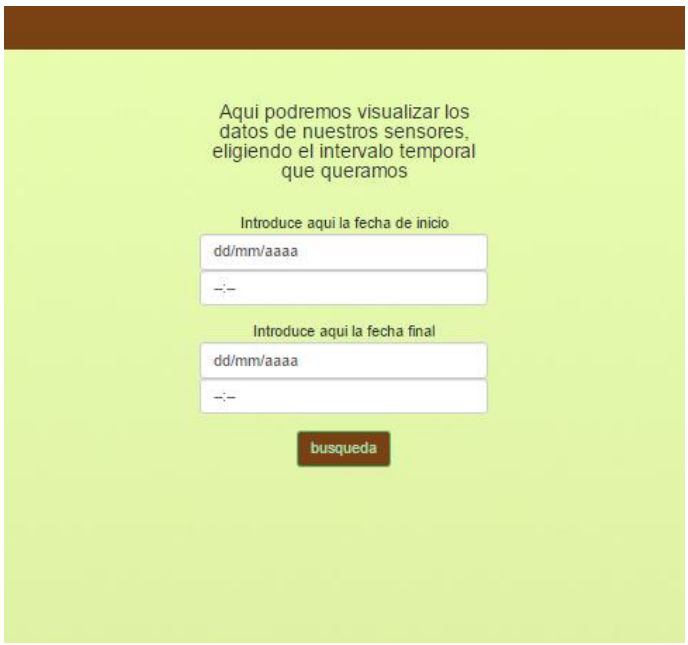

Ilustración 34. Detalle de elección del rango de intervalo de mostrado de información sobre el sensor.

Cuando introducimos los datos nos aparece la información seleccionada visualizada en una tabla y en 3 graficas, una para cada magnitud observada.

ID	Humedad relativa del aire	humedad	temperatura	fecha
2508	22	80	18	2016-10-31 16:27:47
2509	22	80	18	2016-10-31 16:27:53
2510	22	79	18	2016-10-31 16:27:59
2511	22	79	18	2016-10-31 16:28:04
2512	22	79	18	2016-10-31 16:28:10
2513	22	78	18	2016-10-31 16:28:23
2514	22	78	18	2016-10-31 16:28:29
2515	22	78	18	2016-10-31 16:28:35

Ilustración 35. Visualización de los datos de MySQL en una tabla con estilo Boostrap

Ilustración 36. . Visualizacionde los datos, de humedad relativa, en una grafica con Highcharts

Ilustración 37. . Visualizacionde los datos, de humedad del suelo, en una grafica con Highcharts

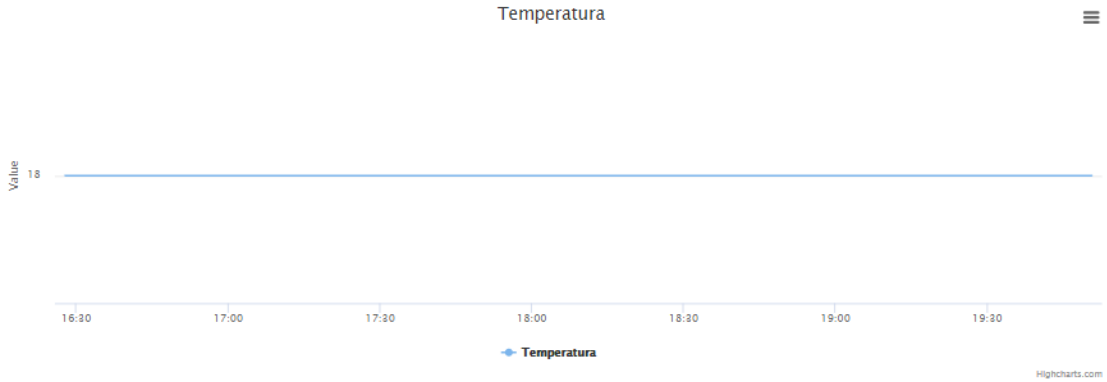

Ilustración 38. Visualizacionde los datos, de temperatura, en una grafica con Highcharts

6. Conclusiones

En este trabajo se ha conseguido implementar un sistema senor portatil que, mediante el uso de un Arduino como centro de procesamiento de la información y unos sensores de humedad, temperatura y humedad del aire como sistema de adquisicion, es capaz de gestionar y visualizar un cultivo en tiempo real.

El sistema se caracteriza por

- Es “low cost”. El uso de una targeta de desarrollo de bajo coste y componentes simples permite reducir los costes del producto.
- Sofware libre. Los programas que se han utilizado son totalmente gratuitos ya que otroga al usuario todas las libertades de manera adecuada. Esta caracteristica también influye en el precio, lo reduce.
- Es portatil. Al ser portatil este sistema se puede colocar en cualquier lugar del cultivo, siempre que este conectado al router, y conocer los datos ambientales de esa zona del cultivo.
- Sistema simple. No es necesario que el usuario del portal tenga conocimientos de como funciona el sistema. Se ofrece un interfaz simple donde el usuario puede acceder a los datos que deseé.
- Facil reproducción. El sistema al ser barato y no muy complicado de desarrollar se pueden fabricar varios sistemas facilmente.
- Monitorizacion remota en tiempo real. La comunicación entre los diversos componenetes del sistema junto con el portal web, hacen posible que, simplemete con una conexion a internet, se pueda visualizar los parametros ambientales del cultivo en el presente para realizar una gestion precisa.
- Almacen de datos. Una de las ventajas del almacenamiento de los registros en la base de datos es posterior manipulación. Con la informacion obtenida se pueden crear modelos de comportamiento del cultivo y ajustarlos para un mayor aprovechamiento del mismo.

Este trabajo deja abierta una via para su futuro desarrollo. A este sisteme se le pueden incluir y cambiar componenetes para una mejor toma de datos o añadir nuevas aplicaciones a nuestro sistema, no solo la de gestion y visualizacion de los datos recogidos.

La mejora mas inmediata sería la creación de un sistema actuador para el riego del cultivo. Para llevarla a cabo se necesitaria una electroválvula, una fuente de alimentación externa y un relé. De esta manera conectando la electroválvula a la fuente de alimentacion externa y a una toma de agua, se puede controlar la misma con el arduino mediante el rele.

Otra de las mejoras de mas relevancia es el encapsulamiento de los sensores segun el grado de protección IP[7] que se necesite.

[7] (Degrees of Protection, grados de protección).

7. Anejo I

En este apartado tenemos la parte del código mas relevante de los archivos principales que desarrollan la plataforma web.

7.1. Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Agricusion</title>
  <link rel="shortcut icon" type="image/x-icon" href="/favicon.ico">
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"></s
cript>
  <script src="js/bootstrap.min.js"></script>

  <style>

  .btn-default {
    color: #A9DFBF;
    background-color: #784212;
    border-color: #52BE80;
}

.navbar-default {
  background-color: #784212;
  border-color: #784212;
  border-bottom-left-radius: 10px;
  border-bottom-right-radius: 10px;

}

.navbar-brand {
    float: left;
    height: 50px;
    padding: 3px 6px;
    font-size: 18px;
    line-height: 20px;
}

.navbar-default .navbar-brand {
  color: #52BE80;

}
.navbar-default .navbar-brand:hover,
.navbar-default .navbar-brand:focus {
  color: #ABEBC6;
  background-color: transparent;
}
.navbar-default .navbar-text {
  color: #2ECC71;
}
.navbar-default .navbar-nav > li > a {
```

```
  color: #52BE80;
  font-size: 20px;
}
.navbar-default .navbar-right > li > a {
  color: #52BE80;
  font-size: 16px;
}
.navbar-default .navbar-nav > li > a:hover,
.navbar-default .navbar-nav > li > a:focus {
  color: #ABEBC6;
  background-color: transparent;
}
.navbar-default .navbar-nav > .active > a,
.navbar-default .navbar-nav > .active > a:hover,
.navbar-default .navbar-nav > .active > a:focus {
  color: #ABEBC6;
  background-color: transparent;
}
.navbar-nav > li > .dropdown-menu {
    background-color: #784212;
    margin-top: 0;
      padding-right: 10px;
    border-top-left-radius: 0;
    border-top-right-radius: 0;
}

.modal-content{
background-image: url(imagenes/degradado.jpg);
}
.modal-header {
      color:#784212;
    padding: 15px;
    border-bottom: 1px solid #784212;
}

  .modal-footer {
    padding: 15px;
    text-align: right;
    border-top: 1px solid #784212;
      }

      .jumbotron{
      border-radius:10px;
      margin-top:25px;
      margin-left:10%;
      background-image: url(imagenes/jumbo.jpg);
      height: 400px;
      width:80%;
      }

      .jumbotron h1 {
      padding-left:60px;
    font-size: 63px;
    font-family: -webkit-pictograph;
    color: #784212;
}

body{
background-image: url(imagenes/degradado.jpg);
}
```

```
.navbar-form .form-group {
   margin-bottom: 10px;
}

</style>


</head>
<body>
<!--navegador-->
      <nav class="navbar-default">
        <div class="container-fluid">
            <div class="navbar-header">
              <button type="button" class="navbar-toggle"
                   data-toggle="collapse" data-target="#myNavbar">
                  <span class="icon-bar" ></span>
                  <span class="icon-bar"></span>
                  <span class="icon-bar"></span>
              </button>
              <a class="navbar-brand"
                   href="#"><img src="imagenes\icono.png"></a>
            </div>
            <div class="collapse navbar-collapse" id="myNavbar">
              <ul class="nav navbar-nav">
                  <li class="active"><a href="index.html">Inicio</a></li>
                  <li><a href="checklogin.php">Proyecto</a></li>

              </ul>
              <ul class="nav navbar-nav navbar-right">
                  <li class="dropdown">
                        <a class="dropdown-toggle" data-toggle="dropdown"
                         href="#"><span   class="glyphicon   glyphicon-log-
                        in"></span> Acceso</a>
                        <ul class="dropdown-menu">
                              <!--formulario de login-->
                              <form class="navbar-form navbar-right"
                               method="post" action="checklogin.php">
                                    <div class="form-group">
                                    <input type="text" class="form-control"
                                     name="usuario_nombre"
                                     placeholder="Nombre de usuario">
                                    </div>
                                    <div class="form-group">
                                    <input type="text" class="form-control"
                                    name="usuario_clave"
                                    placeholder="Contraseña">
                                    </div>
                                    <button type="submit"
                                    class="btn btn-default">Sign In</button>
                              </form>
                              <!--formulario de login-->
                        </ul>
                  </li>
                  <li><a data-toggle="modal" data-target="#myModal"
                   href="#">
                        <span   class="glyphicon   glyphicon-pencil"></span>
                        Registro</a>

                        <!--formulario de registro-->
                        <div class="modal fade" id="myModal" role="dialog">
                              <div class="modal-dialog">
```

```
                    <!-- Contenido del modal-->
                              <div class="modal-content">
                                    <div class="modal-header">
                                    <h4 class="modal-title">
                                    Formulario de registro</h4>
                                    </div>
                                    <div class="modal-body">
                                    <form name="registro"
                                     method="post"
                                     action="registro.php">
                                    <div class="form-group">
                                    <input   type="text"   class="form-
                                    control"     name="usuario_nombre"
                                    placeholder="Nombre de ususario">
                                    </div>
                                    <div class="form-group">
                                    <input   type="text"   class="form-
                                    control" name="usuario_clave"
                                    placeholder="Contraseña">
                                    </div>
                                    <div class="form-group">
                                    <input   type="text"   class="form-
                                    control" name="usuario_clave_conf"
                                    placeholder="Repetir contraseña">
                                    </div>
                                    <div class="form-group">
                                    <input type="text" class="form-
                                    control"      name="usuario_email"
                                    placeholder="Dirección  de  correo
                                    electronico">
                                    </div>

                                    <button  type="submit"  class="btn
                                    btn-default">Registrar</button>
                                    </form>

                                    </div>
                                    <div class="modal-footer">
                                    <button  type="button"  class="btn
                                    btn-default"                data-
                                    dismiss="modal">cerrar</button>
                                    </div>
                              </div>
                        </div>
                  </div>
                  <!--formulario de registro-->
            </li>
                  <li><a  data-toggle="modal"  data-target="#myModal1"
                  href="#"><span      class="glyphicon      glyphicon-
                  envelope"></span> Contacto</a>
                  <!--formulario de contacto-->
                  <div      class="modal      fade"      id="myModal1"
                  role="dialog">
                        <div class="modal-dialog">

                    <!-- Contenido del modal-->
                              <div class="modal-content">
                                    <div class="modal-header">
                                    <h4  class="modaltitle">Formulario
                                    de contacto</h4>
```

```
                                    </div>
                                    <div class="modal-body">
                                    <form              name="contacto"
                                    method="post"
                                    action="contacto.php">

                                    <div class="form-group">
                                    <input   type="text"   class="form-
                                    control"     name="usuario_nombre"
                                    placeholder="Nombre de ususario">
                                    </div>
                                    <div class="form-group">
                                    <input   type="text"   class="form-
                                    control"             name="asunto"
                                    placeholder="Asunto">
                                    </div>
                                    <div class="form-group">
                                    <input   type="text"   class="form-
                                    control"      name="usuario_email"
                                    placeholder="Dirección  de  correo
                                    electronico">
                                    </div>
                                    <div class="form-group">
                                    <textarea     class="form-control"
                                    rows="6"              name="texto"
                                    placeholder="Escriba    aqui    su
                                    mensaje"></textarea>
                                    </div>
                                    <button  type="submit"  class="btn
                                    btn-default">enviar</button>
                                          </form>

                                    </div>
                                    <div class="modal-footer">
                                          <button        type="button"
                                          class="btn      btn-default"
                                          data-
                                          dismiss="modal">cerrar</butt
                                          on>
                                    </div>
                                 </div>
                              </div>
                           </div>
                           <!--formulario de contacto-->
                        </li>
                     </ul>
                  </div>
               </div>
            </nav>
<!--navegador-->

<div class="jumbotron">
<h1>AGRICUSIÓN</h1>
</div>

<div class="container text-center">
  <h1>¿Qué es agricusión?</h1><br>
  <div class="row">

    <div class="col-sm-5 col-sm-offset-1">
      <img src="imagenes/arduino.png" style="width:100%" alt="Image">
```

```
    </div>

    <div class="col-sm-5">
      <div class="well">
       <p>Agricusión es un nuevo portal en el que podras monitorizar en
tiempo real el estado de tu cultivo gracias a ARDUINO</p>
      </div>
      <div class="well">
       <p>Agricusion (agricultura de precisión) surgio a raiz de la
necesidad de controlar de forma remota nuestros cultivos "indoor" para un
mayor rendimiento del mismo </p>
      </div>
    </div>

  </div>
</div><br>

<div class="container text-center">
  <br>
  <div class="row">

    <div class="col-sm-6 col-sm-offset-3">
      <img src="imagenes/arduino2.png"style="width:100%" alt="Image">

    </div>



  </div>
</div><br>

</body>
</html>
```

7.2. Contacto.php

```
<?php
      $email_to="agricusion@gmail.com";
      $usuario_nombre = ($_POST['usuario_nombre']);
      $asunto = ($_POST['asunto']);
      $usuario_email = ($_POST['usuario_email']);
      $texto = ($usuario_nombre.": \n  "
                        .$_POST['texto']);

      $headers = "MIME-Version: 1.0\r\n";
      $headers .= "Content-type: text/html; charset=iso-8859-1\r\n";
      //dirección del remitente
      $headers .= "From: agricusion@gmail.com\r\n".
                        'Reply-To: '.$usuario_email."\r\n".
                        'X-Mailer: PHP/' . phpversion();
    if(mail($email_to, $asunto, $texto, $headers)){
            echo "<div class='alert alert-success alert-dismissable'>
                        <button    type='button'    class='close'    data-
dismiss='alert'>&times;</button>
                        El mensaje se ha enviado correctamente.
                        </div>";
                        include("/index.html");
      }
?>
```

7.3. Registro.php

```
<?php
    // incluimos el archivo de conexión a la Base de Datos
      include('acceso_db.php');

      // creamos una función que nos parmita validar el email
    function valida_email($correo) {
        if          (preg_match('/^[A-Za-z0-9-_.+%]+@[A-Za-z0-9-.]+\.[A-Za-
z]{2,4}$/', $correo)) return true;
        else return false;
    }

      // Procedemos a comprobar que los campos del formulario no estén
vacíos
    $sin_espacios = count_chars($_POST['usuario_nombre'], 1);
    if(!empty($sin_espacios[32]))   {   //   comprobamos   que   el   campo
usuario_nombre no tenga espacios en blanco
        echo "<div class='alert alert-danger alert-dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        El nombre usuario elegido no puede tener espacios
en blanco.
                        </div>";
                        include("/index.html");
    }elseif(empty($_POST['usuario_nombre'])) { // comprobamos que el campo
usuario_nombre no esté vacío
        echo "<div class='alert alert-danger alert-dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        El nombre usuario esta vacio, por favor completelo.
                        </div>";
                        include("/index.html");
    }elseif(empty($_POST['usuario_clave'])) { // comprobamos que el campo
usuario_clave no esté vacío
        echo "<div class='alert alert-danger alert-dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        La contraseña esta vacia, por favor completela.
                        </div>";
                        include("/index.html");
    }elseif($_POST['usuario_clave']  !=  $_POST['usuario_clave_conf'])  {  //
comprobamos que las contraseñas ingresadas coincidan
        echo "<div class='alert alert-danger alert-dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        Las contraseñas no coinciden.
                        </div>";
                        include("/index.html");
    }elseif(!valida_email($_POST['usuario_email'])) { // validamos que el
email ingresado sea correcto
        echo "<div class='alert alert-danger alert-dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        La direccion de correo electronico no es valida.
                        </div>";
                        include("/index.html");
    }else {

            // "limpiamos" los campos del formulario de posibles códigos
maliciosos
```

```
        $usuario_nombre                                                    =
mysql_real_escape_string($_POST['usuario_nombre']);
        $usuario_clave = mysql_real_escape_string($_POST['usuario_clave']);
        $usuario_email = mysql_real_escape_string($_POST['usuario_email']);

            // comprobamos que el usuario ingresado no haya sido registrado
antes
            $sql = mysql_query("SELECT usuario_nombre FROM usuarios WHERE
usuario_nombre='".$usuario_nombre."'");
            if(mysql_num_rows($sql) > 0) {
                echo "<div class='alert alert-danger alert-dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        El nombre usuario elegido ya ha sido registrado
anteriormente.
                        </div>";
                        include("/index.html");
                  }else {
                        // encriptamos la contraseña ingresada con md5
                //$usuario_clave = md5($usuario_clave); // encriptamos la
contraseña ingresada con md5
                // ingresamos los datos a la BD
                $reg = mysql_query("INSERT INTO usuarios (usuario_nombre,
usuario_clave, usuario_email, usuario_freg) VALUES ('".$usuario_nombre."',
'".$usuario_clave."', '".$usuario_email."', NOW())");
                if($reg) {
                    echo    "<div    class='alert    alert-success    alert-
dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        El usuario ha sido registrado correctamente.
                        </div>";
                        include("/index.html");
                }else {
                    echo    "<div    class='alert    alert-danger    alert-
dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        El usuario no se ha registrado correctamente.
                        </div>";
                        include("/index.html");
                }
            }
        }

      mysql_close(mysql_connect($host_db, $usuario_db, $clave_db));
      include 'crear_tabla.php';


?>
```

7.4. cheklogin.php

```
<?php

session_start();

if(!empty($_SESSION['username'])){
 include("panel-control.php");
}else{
```

```
$host_db = "localhost";
$user_db = "root";
$pass_db = "";
$db_name = "registro";


$conexion = new mysqli($host_db, $user_db, $pass_db, $db_name);

if ($conexion->connect_error) {
 die("La conexion falló: " . $conexion->connect_error);
}

$username = $_POST['usuario_nombre'];
$password = $_POST['usuario_clave'];



$sql = "SELECT * FROM usuarios WHERE usuario_nombre = '$username'";

$result = $conexion->query($sql);


if ($result->num_rows > 0) {
 }
 $row = $result->fetch_array(MYSQLI_ASSOC);

 if ($password == $row['usuario_clave'] ) {

    $_SESSION['loggedin'] = true;
    $_SESSION['username'] = $username;
      $_SESSION['password'] = $password;
    $_SESSION['start'] = time();
    $_SESSION['expire'] = $_SESSION['start'] + (5 * 60);



   include("panel-control.php");

 } else {
   echo "<div class='alert alert-danger alert-dismissable'>
                        <button     type='button'     class='close'     data-
dismiss='alert'>&times;</button>
                        El   usuario   o   contraseñas   introducidas   no   son
correctas.
                        </div>";
                        include("index.html");
}
 mysqli_close($conexion);
}
 ?>
```

- panel-control.php (aquí solo mostramos parte del el codigo ya que la mitad se repite en el archivo de index.html)

```
<?php
session_start();
if(!empty($_SESSION['username'])){
 include("panel-control.php");
```

```
}else{
$host_db = "localhost";
$user_db = "root";
$pass_db = "";
$db_name = "registro";
$conexion = new mysqli($host_db, $user_db, $pass_db, $db_name);

if ($conexion->connect_error) {
 die("La conexion falló: " . $conexion->connect_error);
}
$username = $_POST['usuario_nombre'];
$password = $_POST['usuario_clave'];
$sql = "SELECT * FROM usuarios WHERE usuario_nombre = '$username'";
$result = $conexion->query($sql);
if ($result->num_rows > 0) {
 }
 $row = $result->fetch_array(MYSQLI_ASSOC);
if ($password == $row['usuario_clave'] ) {
    $_SESSION['loggedin'] = true;
    $_SESSION['username'] = $username;
      $_SESSION['password'] = $password;
    $_SESSION['start'] = time();
    $_SESSION['expire'] = $_SESSION['start'] + (5 * 60);



   include("panel-control.php");
 } else {
   echo "<div class='alert alert-danger alert-dismissable'>
                        <button    type='button'    class='close'    data-
dismiss='alert'>&times;</button>
                        El  usuario  o  contraseñas  introducidas  no  son
correctas.
                        </div>";
                        include("index.html");
}
 mysqli_close($conexion);
}
?>
<html>
<?php include 'iniciar_sesion.php'?>



<div class="formulario">
<h4>Aqui podremos visualizar los datos de nuestros sensores, eligiendo el
intervalo temporal que queramos</h4>
</br>
<form method="post"  name="busqueda" action="checklogin.php">

                <div class="form-group" >
                Introduce aqui la fecha de inicio
                <input type="date" class="form-control" name="f_inicial">
                <input type="time" class="form-control" name="h_inicial">
                </div>



                <div class="form-group">
                Introduce aqui la fecha final
```

```
                <input type="date" class="form-control" name="f_final" >
                <input type="time" class="form-control" name="h_final" >
                </div>
                button        type="submit"        class="btn        btn-
                default">busqueda</button>

</form>
</div>
<?php
$f_inicial = '0';
$h_inicial = '0';
$f_final = '0';
$h_final = '0';
if (!empty ($_POST['f_inicial']) && !empty ($_POST['h_inicial']) && !empty
($_POST['f_final']) && !empty ($_POST['h_final'])){
 $f_inicial = $_POST['f_inicial'];
 $h_inicial = $_POST['h_inicial'];
 $f_final = $_POST['f_final'];
 $h_final = $_POST['h_final'];
}
//Sentencia SQL
      $sql = "SELECT * from ".$_SESSION['username']." where time BETWEEN
'".$f_inicial." ".$h_inicial."' AND  '".$f_final." ".$h_final."';";
      //Array Multidimensional
      $rawdata = getArraySQL($sql);
if(!empty ($rawdata)){
?>
<div class="divtabla">
<table class="table table-hover">
    <thead>
    <tr>
        <th data-field="name">ID</th>
        <th data-field="stargazers_count">Humedad relativa del aire</th>
        <th data-field="forks_count">humedad</th>
            <th data-field="forks_count">temperatura</th>
        <th data-field="description">fecha</th>
    </tr>
    </thead>
      <?php for($i=0;$i<count($rawdata);$i++){
      echo'<tr>';
      echo'<td>'.$rawdata[$i]["id"].'</td>';
      echo'<td>'.$rawdata[$i]["humedad_rel"].'</td>';
      echo'<td>'.$rawdata[$i]["humedad"].'</td> ';
      echo'<td>'.$rawdata[$i]["temperatura"].'</td>';
      echo'<td>'.$rawdata[$i]["time"].'</td>';
      echo'</tr> ';
      }
      ?>
</table>
</div>
<div class="col-sm-8 col-sm-offset-2">
    <?php include 'grafica/graficas.php' ?>
</div>
<?php
}
?>
</body>
</html>
```

7.5. crear_tabla-php

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "sensores";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

$sql= "CREATE TABLE ".$usuario_nombre." (
            id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
            humedad_rel INT(30) NOT NULL,
            humedad INT(30) NOT NULL,
            temperatura INT(50) NOT NULL,
            time TIMESTAMP
            )";

if ($conn->query($sql) === TRUE) {
    echo "Tabla creada correctamente";
} else {
    echo "Error al crear la tabla" . $conn->error;
}
$conn->close();
?>
```

7.6. graficas.php

```
<?php
      for($i=0;$i<count($rawdata);$i++){
    $time = $rawdata[$i]["time"];
    $date = new DateTime($time);
    $rawdata[$i]["time"]=$date->getTimestamp()*1000;
}
?>
<HTML>
<head>
<meta charset="utf-8">

<!-- 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
            }
        });

        var chart;
        $('#container').highcharts({
            chart: {
```

```
                type: 'spline',
                animation: Highcharts.svg, // don't animate in old IE
                marginRight: 10,
                events: {
                    load: function() {

                    }
                }
            },
            title: {
                text: 'humedad relativa del aire'
            },
            xAxis: {
                type: 'datetime',
                tickPixelInterval: 150
            },
            yAxis: {
                title: {
                    text: 'tanto por ciento'
                },
                plotLines: [{
                    value: 0,
                    width: 1,
                    color: '#808080'
                }]
            },
            tooltip: {
                formatter: function() {
                        return '<b>'+ this.series.name +'</b><br/>'+
                                Highcharts.dateFormat('%Y-%m-%d
%H:%M:%S', this.x) +'<br/>'+
                        Highcharts.numberFormat(this.y, 2);
                }
            },
            legend: {
                enabled: true
            },
            exporting: {
                enabled: true
            },
            series: [{
                name: 'humedad relativa',
                     data: (function() {
                        var data = [];
                    <?php
                        for($i = 0 ;$i<count($rawdata);$i++){
                    ?>
            data.push([<?php     echo     $rawdata[$i]["time"];?>,<?phpecho
            $rawdata[$i]["humedad_rel"];?>]);
                    <?php } ?>
                return data;
                     })()
            }]
        });
    });

});
<div id="container">
</div>

</br>
```

```
</br>
</html>
```

7.7. cerrar_sesion.php

```
<?php
session_start();
session_destroy();
header("Location: index.html");
?>
```

Bibliografía

- Alonsojpd. Uso de desiones en PHP, variables de sesión, cerrar sesión, iniciar sesión. http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=486 (15/10/2016).
- D-Robotics (2010). DHT 11 Humidity & Temperature sensor.
- García Moreno Flor, Juan Bedoya Fierro, Germán Arturo López Martínez (2013). Modelo a escala de un sistema de riego automatizado, alimentado con energía solar fotovoltaica: nueva perspectiva para el desarrollo agroindustrial colombiano. Universidad distrital Francisco José de Caldas.
- Lozano Ivan (2013).Cuatro alternativas a arduino. http://blogthinkbig.com/4-alternativas-arduino-beaglebone-raspberrypi-nanode-waspmote/ (30/10/2016).
- Manu (2014). configurando nuestro sitio con Boostrap 3. http://www.tutosytips.com/tutorial-configurando-nuestro-sitio-con-bootstrap-3/ (16/10/2016).
- Silberschatz Abraham, Henry F.Korth, S.Sudarshan (2002). Fundamentos de base de datos. Editorial Mc Graw Hill.
- Torrente Artero Oscar (2013). ARDUINO Curso práctico de formación. Editorial Alfaomega.
- Pérez Esteso Mario (2014). PHP + MYSQL + HIGHCHART: mostrar varias gráficas dinamicamente. https://geekytheory.com/php-mysql-highchart-mostrar-varias-graficas-dinamicamente/ (15/10/2016).
- Apuntes de la asignatura infraestructura de datos espaciales.