

Sistema de Bajo Coste para la Medida y Monitorización en Agricultura Inteligente

Rebeca Lopez-Chacón, Herminio Martínez-García
 Department of Electronics Engineering
 Eastern Barcelona School of Engineering (EEBE)
 Technical University of Catalonia (UPC). BarcelonaTech
 Diagonal-Besòs Campus. Eduard Maristany Ave., nº 10 – 14
 E-08019 - Barcelona. Spain
 rebecalch93@gmail.com, herminio.martinez@upc.edu

Abstract— El presente artículo presenta las principales características del diseño, desarrollo e implementación de dos prototipos llevados a cabo para la monitorización de parámetros climatológicos y ambientales, así como del suelo, para agricultura inteligente. Ambos prototipos han sido desarrollados utilizando tecnología de bajo coste. Además, los sistemas se han diseñado bajo la premisa de ser de bajo consumo y sostenibles, con alimentación a partir de un sistema solar fotovoltaico, y baterías recargables. En cuanto a la programación, el sistema se ha desarrollado para que sea modular y escalable, de forma que se adecue de la mejor forma posible a las necesidades particulares de cada cliente. Dicho cliente podrá obtener los datos requeridos mediante diferentes formas (SD card, SMS o envío por GSM), dependiendo del módulo escogido. La página web y la base de datos asociadas al sistema han sido desarrolladas en base a monitorizar los datos, poder representarlos en tiempo real, mostrar gráficos de históricos, así como poder realizar descargas de los mismos, con el fin de poder ser procesados *a posteriori* en un sistema informático complementario al sistema (PC, etc.).

Keywords— Microcontroller, sensors, monitoring, GSM, low-cost technology, Arduino, database, web page.

I. INTRODUCCIÓN

En la actualidad, la tecnología utilizada en los terrenos de cultivos agrícolas y explotaciones ganaderas está más presente que nunca. Conocer, monitorizar y registrar aquellos datos y eventos que nos rodean permiten analizar, prever y conocer las tendencias de éstos en un futuro. Además, la monitorización de datos recogidos a través de sensores ubicados de forma estratégica en los campos permite ayudar a los agricultores y ganaderos a obtener mejores resultados y beneficios.

Tradicionalmente los agricultores se basaban en su experiencia e intuición para realizar las distintas labores del campo siguiendo un calendario prefijado año tras año. En cambio, gracias al uso de las nuevas tecnologías de macrodatos podrán tomarse decisiones instantáneamente a través del sensado, el almacenamiento y el análisis de grandes volúmenes de datos en tiempo real procedentes de sensores distribuidos por el terreno. Hoy en día, los sectores agrícolas y ganaderos se encuentran ante una nueva revolución liderada por las tecnologías de la información y la comunicación; es la llamada *agricultura inteligente* o *agricultura de precisión*.

Las ventajas y objetivos principales que pretende cubrir la agricultura inteligente se resumen en la optimización de recursos, el incremento de la producción y la reducción de los costes asociados a ésta, la reducción del impacto medioambiental vinculado a la actividad agrícola y la mejora de la calidad del producto final.

En este artículo, se muestra el diseño y la implementación de dos prototipos basados en Arduino; en concreto, una estación meteorológica y una estación de suelo que permiten proporcionar información, obtenida a través de sensores, de parámetros meteorológicos y atmosféricos entre otros. Además, se presentan tres maneras diferentes para la obtención de estos datos; a saber, microSD, SMS y mediante página Web.

II. ARQUITECTURA *HARDWARE* DEL SISTEMA

El elemento principal de estos prototipos es Arduino[®], concretamente Arduino[®] UNO para las estaciones de suelo y Arduino[®] Mega ADK para la estación meteorológica. En el caso de la estación meteorológica, fue necesario utilizar otro modelo de Arduino[®] que tuviera mayor memoria SRAM debido a los requisitos técnicos de los programas desarrollados.

Para el almacenamiento de los datos en una tarjeta microSD se utiliza el módulo Arduino[®] Wireless SD, que además de disponer de un zócalo para tarjetas microSD permite añadir comunicación inalámbrica utilizando módulos XBee (Serie 1 y Serie 2) O Wifi 802.11 b/g con el módulo RN-XV WiFly.

A nivel de comunicaciones, se ha utilizado el módulo Arduino[®] GSM-GPRS V2 con antena integrada de la compañía Telefónica, ya que las dos grandes necesidades que se pretenden cubrir son la conexión inalámbrica y la necesidad de conectarse a Internet sin tener que depender de estar cerca de una casa o edificio que disponga de un *router*.

En cuanto a los sensores utilizados para la estación meteorológica, tal y como puede verse en la Fig. 1, encontramos:

- Sensor de temperatura y humedad DHT22.
- Sensor barométrico BMP180.
- Sensor de luminosidad TSL2561.

- Sensor de luz ultravioleta GUYA-S12SD.
- Anemómetro, veleta y pluviómetro de Argent Data System.

Por otro lado, para la estación de suelo se utilizó una sonda de temperatura y humedad SHT11, como puede verse en la Fig. 2.

Para ambas estaciones diseñadas e implementadas se ha utilizado el reloj RTC Chronodot V2.1 para poder vincular cada dato a la fecha y hora exacta de recogida.

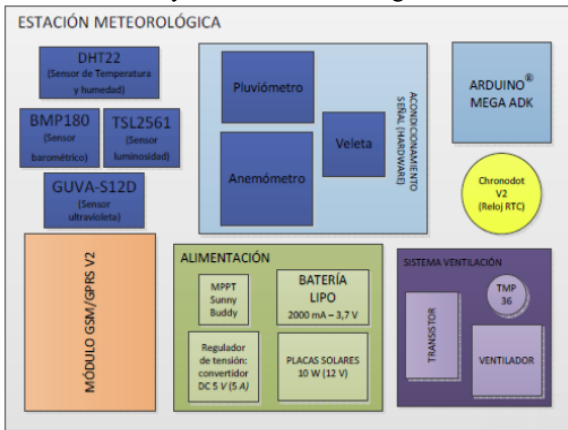


Fig. 1. Esquema de bloques: estación meteorológica.

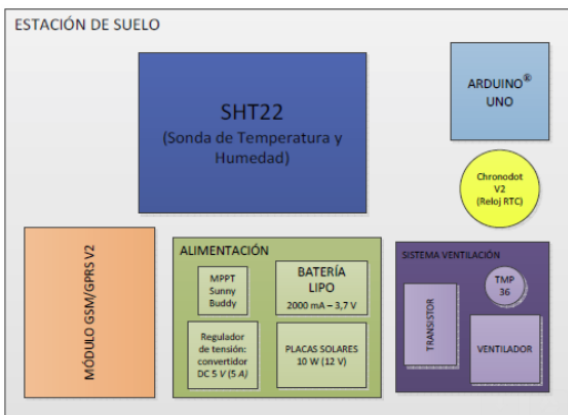


Fig. 2. Esquema de bloques: estación de suelo.

Finalmente, y debido a que la electrónica se encontrará ubicada en el interior de una caja, se ha diseñado un pequeño sistema de control para la ventilación que cuenta con un sensor de temperatura TMP36 (que se ubicará en el interior de la caja), un transistor *driver* y un ventilador.

III. ACONDICIONADO DE LA SEÑAL

Para la sensórica, ha sido necesario realizar el acondicionamiento de la señal de los sensores utilizados usando tanto *hardware* como *software*. Sólo, en el caso de algunos sensores específicos, no se ha realizado ningún tipo de acondicionamiento ya que se han utilizado librerías específicas que ya recogen estos acondicionamientos vía *software*, como es el caso del sensor de temperatura y humedad DHT22 entre otros.

Para el sensor de luz ultravioleta GUYA-S12SD ha sido

necesario acondicionar la señal vía *software*, teniendo en cuenta que existe una tabla de equivalencias entre el voltaje de salida y el índice UV, tal y como muestra la Fig. 3.

Para obtener el resultado en *mV*, simplemente será necesario multiplicar el resultado por un factor 10^3 . Para este sensor se ha realizado una función que se encarga de obtener el valor analógico del sensor GUYA-S12SD y, a través de los cálculos pertinentes, transforma la lectura del ADC en tensión (*mV*). Mediante estructuras de control se relaciona el índice UV con los *mV* obtenidos.

| | | | | | | |
|----------------------|-----|-----|-----|-----|------|-------|
| UV Index (Índice UV) | 0 | 1 | 2 | 3 | 4 | 5 |
| Vout(mV) | <50 | 227 | 318 | 408 | 503 | 606 |
| Valor analógico | <10 | 46 | 65 | 83 | 103 | 124 |
| UV Index (Índice UV) | 6 | 7 | 8 | 9 | 10 | 11 |
| Vout(mV) | 696 | 795 | 881 | 976 | 1079 | 1170+ |
| Valor analógico | 142 | 162 | 180 | 200 | 221 | 240 |

Fig. 3. Tabla de equivalencia entre el Índice UV, la tensión de salida y el valor analógico.

En el caso del anemómetro es necesario tener en cuenta que se trata de un anemómetro de cazoletas que mide la velocidad del viento en función de la cantidad de pulsos generados durante un tiempo determinado. Una velocidad de viento de $2,4 \text{ km/h}$ hace que el interruptor se cierre una vez por segundo. Por otro lado, el pluviómetro está basado en el principio de que cada $0,28 \text{ mm}$ se produce un cierre de contacto momentáneo, trabajando como si fuera un *switch*. Estos cierres momentáneos son leídos por el microcontrolador, y permiten así obtener en *mm* la cantidad de lluvia recogida en un período determinado.

Para la obtención de estos valores, ha sido necesario el uso de interrupciones externas y funciones IRQ. En el caso del anemómetro habrá que incrementar el número de vueltas que da cada vez que se detecta un pulso y en el caso del pluviómetro para contabilizar el número de veces que se vierte el depósito. En ambos casos, la única manera de obtener una correcta medición es interrumpiendo el programa para realizar el incremento del contador que acumula el número de pulsos, inicializando dichos contadores en cada intervalo de envío y guardando de datos.

En el caso de la velocidad del viento, se han desarrollado diferentes funciones que permiten obtener la media entre las vueltas que ha dado el anemómetro y el tiempo entre la medición actual (obtenida antes de realizar la media) y la medición anterior. También se guardará, tras haber realizado la media y realizar los cálculos pertinentes, el tiempo actual de la variable correspondiente al tiempo de la medición anterior.

Para la precipitación, en los programas de envío GSM (con o sin *buffer*) el valor total de la variable de la función IRQ se almacenará en otra variable y se reiniciará en cada iteración. De este modo, se enviará la precipitación acumulada cada 5 min (frecuencia entre envío y envío) y podrán visualizarse

correctamente las gráficas de precipitación acumulada en la página web.

Por otro lado, la veleta consta de ocho interruptores (Fig. 4), cada uno de ellos conectado a una resistencia diferente. El imán de la veleta puede cerrar dos interruptores a la vez, lo que permite un máximo de 16 posiciones diferentes.

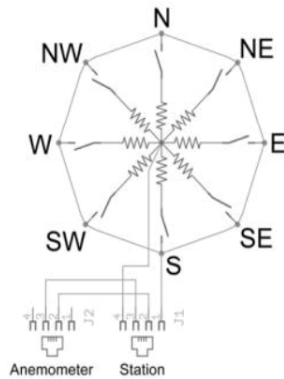


Fig. 4. Resistencias internas de la veleta.

Los distintos valores de voltaje son leídos por el ADC del Arduino® y posteriormente son tratados para obtener la conversión en grados.

El fabricante ofrece en el *datasheet* una tabla que relaciona cada grado asociado al valor resultante de la resistencia variable (Tabla I).

TABLA I
CÁLCULOS VELETA – LECTURA ADC TEÓRICO.

| Grados (°) | Rvar interna (Ω) | Rfija R1 (Ω) | Rfija RS (Ω) | Rfija RA (Ω) | Req (Rvar+R1+RA) | Voltaje salida (V) | Lectura ADC teórico |
|------------|------------------|--------------|--------------|--------------|------------------|--------------------|---------------------|
| 112,5 | 688 | 1000 | 4700 | 1000 | 2688 | 1,819166215 | 372,5652409 |
| 67,5 | 891 | 1000 | 4700 | 1000 | 2891 | 1,904228692 | 389,9860361 |
| 90 | 1000 | 1000 | 4700 | 1000 | 3000 | 1,948051948 | 398,961039 |
| 157,5 | 1410 | 1000 | 4700 | 1000 | 3410 | 2,102342787 | 430,5598027 |
| 135 | 2200 | 1000 | 4700 | 1000 | 4200 | 2,359550562 | 483,2359551 |
| 202,5 | 3140 | 1000 | 4700 | 1000 | 5140 | 2,611788618 | 534,8943089 |
| 180 | 3900 | 1000 | 4700 | 1000 | 5900 | 2,783018868 | 569,9622642 |
| 22,5 | 6570 | 1000 | 4700 | 1000 | 8570 | 3,229088169 | 661,317257 |
| 45 | 8200 | 1000 | 4700 | 1000 | 10200 | 3,422818792 | 700,9932886 |
| 247,5 | 14120 | 1000 | 4700 | 1000 | 16120 | 3,871277618 | 792,8376561 |
| 225 | 16000 | 1000 | 4700 | 1000 | 18000 | 3,964757709 | 811,9823789 |
| 337,5 | 21880 | 1000 | 4700 | 1000 | 23880 | 4,177746676 | 855,6025192 |
| 0 | 33000 | 1000 | 4700 | 1000 | 35000 | 4,408060453 | 902,7707809 |
| 292,5 | 42120 | 1000 | 4700 | 1000 | 44120 | 4,518639902 | 925,4174519 |
| 315 | 64900 | 1000 | 4700 | 1000 | 66900 | 4,671787709 | 956,7821229 |
| 270 | 120000 | 1000 | 4700 | 1000 | 122000 | 4,814522249 | 986,0142068 |

A nivel de programación, se ha realizado una función que se encarga de obtener la lectura analógica de la veleta y, mediante estructuras de control, asociar cada lectura del ADC al grado polar correspondiente.

IV. SISTEMA DE ALIMENTACIÓN

La sostenibilidad con el medio ambiente era uno de los objetivos principales del desarrollo de los prototipos, por ese motivo se ha utilizado la recarga de baterías mediante paneles solares.

En referencia al consumo, es importante destacar que éste se ve aumentado notablemente a causa de la utilización del módulo GSM/GPRS ya que su consumo fluctúa entre los 700 y los 1000 mA. Este hecho intervino considerablemente en la

selección de baterías, placas solares y *shields* necesarios para la correcta alimentación del prototipo. El resto de sensores y módulos utilizados no tienen un consumo significativo, por ejemplo en el caso del sensor de temperatura y humedad estamos hablando de 2,5 mA.

A continuación se presentan los elementos utilizados:

- MPPT Sunny Buddy: módulo diseñado por Sparkfun, basado en el cargador LT3652 (2 A) que permite optimizar la carga de las baterías LiPo de 3,7 V mediante paneles solares. La carga se conecta en paralelo directamente sobre la batería.
- Batería LiPo de 2000 mAh y 3,7 V.
- Regulador de tensión: convertidor DC 5 V (5 A) de la empresa Bricogeeek que permitirá obtener una salida de 5 V a partir de la tensión de entrada de 3,7 V dada por la batería LiPo utilizada.
- Placas Solares: *kit* de panel solar policristalino de 10 W (12 V) de la empresa Comercial Solar.

La electrónica está ubicada en el interior de una caja estanca y los paneles solares se encuentran instalados en la parte superior con una inclinación determinada, calculada mediante el *software* PVsyst®, teniendo en cuenta la localización donde se realizaran pruebas *outdoor* (en la población barcelonesa de Vilanova i la Geltrú). Teniendo en cuenta el diseño de las cajas, los paneles solares se encontrarán encastados en la parte superior de éstas por lo que tendrán un ángulo fijo que no podrá variar en función de la localidad ni de la época del año. El motivo principal por el que se decidió incrustar las placas en la caja es evitar el cableado exterior, evitando que los cables pudieran dañarse con facilidad, asegurando y protegiendo la alimentación del sistema. A partir de las diferentes simulaciones realizadas mediante el *software* PVsyst®, se decidió instalar las placas con un ángulo óptimo de 38° (ángulo fijo) y orientarlos hacia el sur para cubrir la demanda de todo el año.

V. ARQUITECTURA SOFTWARE

En el sistema global de medida y monitorización presentado, se han desarrollado tres módulos de programación enfocados a las funcionalidades deseadas (Fig. 5):

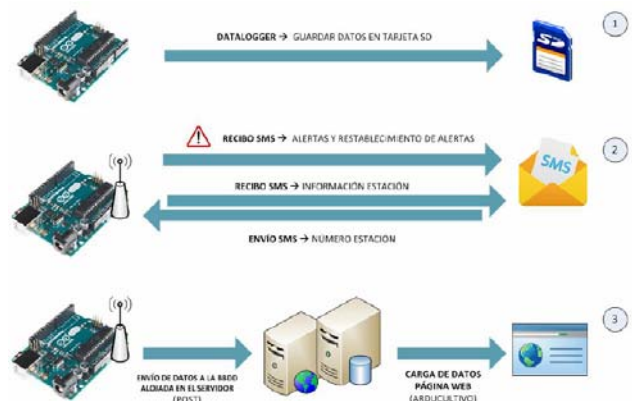


Fig. 5. Diagrama simplificado de la programación modular.

1. *Datalogger*: la misión principal de este bloque de programación consiste en registrar los valores obtenidos mediante las sondas y sensores ubicados en el terreno agrícola y almacenarlos en una tarjeta SD.
2. *SMS*: Este módulo ofrece, en primer lugar, poder recibir alertas y avisos vía SMS mediante el *shield* GSM/GPRS. También enviará un SMS cuando la alarma se haya restablecido. Esta gestión de alarmas se ha desarrollado mediante un sistema ON-OFF con histéresis diferencial. Además, ofrece la posibilidad de poder conocer los datos obtenidos en ese instante vía SMS, enviando previamente un mensaje de texto con el número de la estación utilizando un teléfono móvil. Ambas funcionalidades pueden verse en la Fig. 6.

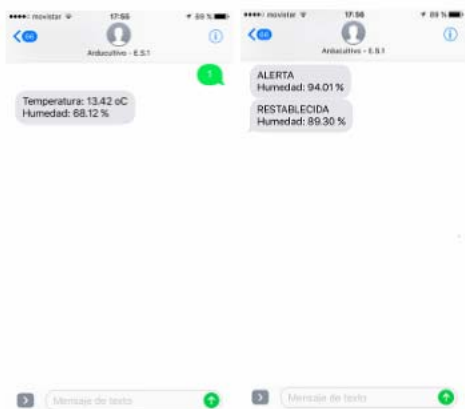


Fig. 6. SMS de parámetros de la estación y SMS de Alertas.

3. *BBDD y página web*: con el *shield* GSM/GPRS, este bloque permite enviar, a través de comandos POST, los datos, obtenidos a través de los sensores, a un servidor de base de datos y web. Posteriormente, estos datos podrán monitorizarse a través de gráficas, históricos y también podrán descargarse en formato *.csv. Para este modelo también se ha desarrollado una versión con buffer. Este buffer se ha realizado mediante programación y permite almacenar un total de 13 horas de datos. De este modo se garantiza que el prototipo pueda guardar todos los datos en el caso de no haber luz solar y por tanto las baterías no puedan ser recargadas. Los datos, almacenados en diferentes vectores y controlados mediante apuntadores, se enviarán una vez se detecte luz solar por un tiempo determinado mediante el sensor de luminosidad TSL2561.

Es importante destacar que la estación meteorológica y la estación de suelo disponen de programas distintos, adecuándose cada uno a las placas Arduino® utilizadas y a las variables específicas.

VI. ARQUITECTURA DE RED

Para el correcto funcionamiento de los programas de BBDD y página web (con y sin *buffer*) que se cargarán de insertar los datos en la base de datos se requiere tener en cuenta la arquitectura de red mostrada en la Fig. 7.

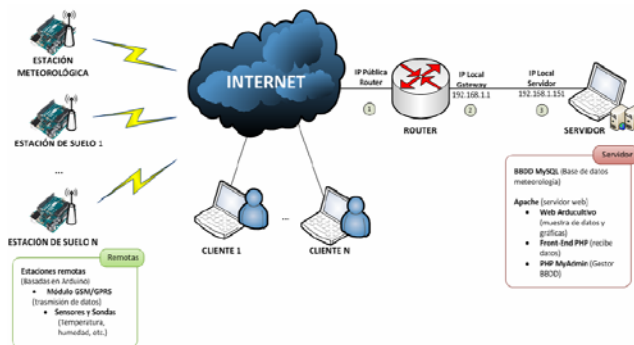


Fig. 7. Arquitectura de red: BBDD y página web.

El servidor, a nivel de *hardware*, realmente se trata de un antiguo portátil Sony VAIO con Sistema Operativo Windows 7 Ultimate x64. El ordenador tiene instalado el paquete WAMP (Windows-Apache-MySQL-PHP/Phyton/PERL). Además, el enrutador se compone de un *router* para uso doméstico de la compañía Movistar.

Por un lado, se realiza la configuración de la red local estática del servidor, asignándole una IP fija para poder desarrollar y gestionar la página web correctamente y haciendo que solo sea necesaria una única configuración del *port forwarding*. Esta configuración se realiza a través de la web gestionable del *router*, asignando la IP fija a la MAC del portátil.

Por otro lado, el concepto *port forwarding* hace referencia al redireccionamiento de puertos, que consiste en redirigir un puerto de red de un nodo de red a otro. De este modo, mediante el módulo GSM/GPRS de Arduino®, será posible el acceso a un puerto con una dirección IP privada asociada (dentro de la LAN) desde el exterior a través del *router* con la NAT activada. Así se podrá asegurar el correcto envío de los datos obtenidos mediante los sensores al servidor. Esta configuración también se realiza a través de la web gestionable del *router* de Movistar.

Para la configuración de la IP pública fija del *router* es necesario contratar directamente un servicio a través de la compañía telefónica contratada.

VII. BASE DE DATOS

En referencia a la base de datos, se ha diseñado una base de datos relacional (BDR) en MySQL para almacenar y procesar los datos enviados mediante GSM/GPRS por las placas basadas en Arduino® pertenecientes a la estación meteorológica y las estaciones de suelo. La base de datos consta de tres tablas diferentes: *Estaciones*, *Variables* y *Datos* tal y como muestra la Fig. 8.

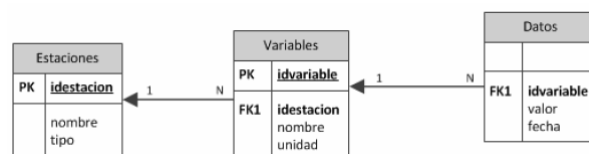


Fig. 8. Diagrama de la base de datos relacional. PK: clave primaria. FK: clave foránea.

Encontraremos registros fijos en la tabla *Estaciones* y en la tabla *Variables*, éstos hacen referencia a los números identificadores (*idestacion* o *idvariable*), al nombre (de la estación o de la variable en cuestión), al tipo (EM: Estación Meteorológica y ES: Estación de suelo) en el caso de la tabla estaciones, y unidad (p. ej., °C, %, lux, km/h, etc.) en el caso de la tabla *Variables*.

Por otro lado, los registros de la tabla datos se tratan de registros importados provenientes de los valores recogidos a través de los sensores distribuidos en el cultivo, por lo que en esta tabla encontraremos en cada registro el valor obtenido por el sensor, el número de la variable (*idvariable*) a la que corresponde el dato y la fecha de registro.

Es importante destacar que la fecha de registro se almacena en formato *UNIX Timestamp*, sistema que mide la cantidad de segundos transcurridos desde la medianoche (UTC) del 1 de enero de 1970.

A través de peticiones POST (mediante Arduino®) se realizará la importación de datos a la base de datos. En el servidor se encuentra un fichero PHP de inserción que permitirá realizar la carga de los datos para todas las variables a excepción de las variables referentes a la dirección del viento y a la velocidad del viento instantánea, las cuales tienen dos ficheros específicos de inserción de datos. Esto es así debido a que, a través de estas variables, se obtienen otras variables calculadas como son la dirección de la racha, la velocidad del viento, y la velocidad de la racha.

Mediante procesos ETL (*extract, transform and load*) se extraen los datos y se transforman para que sean capaces de ser analizados por el usuario. Por ejemplo, para interpretar la dirección del viento, la precipitación, la obtención de máximos, mínimos y promedios de las múltiples variables medidas, para insertarlos en gráficas, etc.

VIII. DESARROLLO WEB

La página web es la cara visible y funcional por parte del cliente ya que a través de ella podrá obtener la información necesaria del sistema en diferentes formatos, como, por ejemplo, gráficas e informes en formato .csv.

La página web se ha desarrollado mediante Wordpress (sistema de gestión de contenidos gratuito), MySQL (Base de datos donde se almacenan todos los registros referentes al sitio web) y un servidor con soporte PHP donde poder estar alojado. Los lenguajes utilizados para el desarrollo de la página web, para la exportación, tratamiento y carga de datos (procesos ETL) e inserción en gráficas, entre otros, son: SQL, PHP, HTML, JavaScript y JQuery. También se han utilizado formatos de texto JSON y CSV para la importación de datos a la página web y para la descarga de informes.

La web consta de tres perfiles diferentes de usuarios cuyos roles varían en función de las tareas que deben realizar, tal y como muestra la Fig 9.



Fig. 9. Perfiles de usuarios – Arducultivo.

La página web consta de cinco secciones bien diferenciadas: Datos a Tiempo Real, Gráficas en Tiempo Real, Históricas, Informes y Mapa. En la página web desarrollada hay dos páginas de “Datos a Tiempo Real”, una para la estación meteorológica y otra para las estaciones de suelo.

Para los Datos a Tiempo Real, existe un fichero PHP para cada variable que se encargará de extraer el último valor de la base de datos de la variable asociada al fichero y lo importará a la página web en formato JSON. Esto es así para todas las variables a excepción de la dirección del viento y la dirección de la racha, ya que han de calcularse de forma particular. Mediante el *plugin* JSON Content Importer se pueden importar los datos, almacenados en la BDD, en la página correspondiente.

Para las gráficas a “*tiempo real*”, se extraen los datos de la BBDD y posteriormente se realiza una importación en formato JSON de los últimos 50 valores de la variable correspondiente a la gráfica, ver Fig. 10.



Fig. 10. Gráficas a tiempo real (temperatura de suelo – ES1) – Arducultivo.

Además, se ha utilizado una función que permite cargar los datos desde una web en formato JSON a una gráfica creada mediante librerías *amCharts*. Además, se ha añadido un refresco cada 10 s para asegurar que el último dato guardado en la BBDD se mostrará en la gráfica que estamos visualizando en ese momento.

En el caso de los “*históricos*” el procedimiento es muy similar al de las gráficas a “*tiempo real*”. La diferencia principal se encuentra en los ficheros PHP de cada variable y en el código JavaScript de la gráfica.

De igual modo que en las gráficas a “*tiempo real*”, en el caso de las gráficas “*históricas*” también existe un fichero PHP asociado a cada variable. La diferencia principal de las gráficas “*históricas*” respecto de las de “*tiempo real*”, es la importación de *todos* los datos en la gráfica. Por ese motivo, se cuenta con la implementación de una función que carga un *slider* encima de la gráfica que permite al usuario navegar

entre un período de fechas comprendido entre los dos selectores de inicio y fin.

En el apartado “*Históricos*” también se pueden visualizar gráficas polares referentes a la dirección del viento y a la dirección de la racha (Fig. 11). Estas gráficas seleccionan los 288 últimos datos (valor y fecha) relacionando el valor numérico (grados) y su equivalente punto cardinal. A su vez, estos datos se agrupan y se determina el número de coincidencias de estos valores con los puntos cardinales.

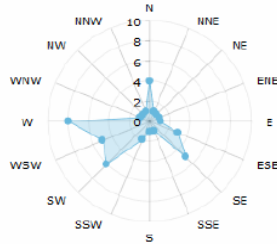


Fig. 11. Gráfica polar (dirección del viento) – Arducultivo.

Por otro lado, también se visualizan gráficas históricas de múltiples variables que tienen las mismas unidades, como es el caso de la temperatura y la humedad de la estación meteorológica y de las diversas estaciones de suelo que dispongamos, garantizando una rápida comparativa de forma visual, tal y como puede verse en la Fig. 12.



Fig. 12. Históricos (múltiples variables: Temperaturas) – Arducultivo.

En “*Históricos*” también encontraremos diagramas de barras y lineales que hacen referencia a los valores máximos, mínimos y promedios de algunas de las variables medidas (Fig. 13). Para ello, se han utilizado diferentes ficheros ETL encargados de realizar el tratado de datos y cálculos adecuados para la correcta visualización en las gráficas.



Fig. 13. Máximos, mínimos y promedios (temperatura de suelo ES1) – Arducultivo.

Finalmente, para la precipitación (Fig. 14) se ha optado por

la realización de un diagrama de barras que muestra la precipitación mensual acumulada. En este caso particular, el fichero ETL es el encargado de procesar los datos guardados en la base de datos de la variable precipitación para obtener la precipitación acumulada mensualmente.

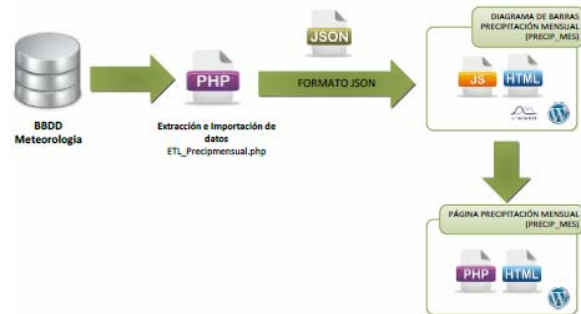


Fig. 14. Diagrama de gráfica de precipitación mensual – Desarrollo web.

En todas las gráficas podremos visualizar el valor exacto deslizando el ratón por encima de la barra tal y como puede verse en la Fig. 15, que en este caso muestra el valor de precipitación acumulada del mes de julio del 2016.

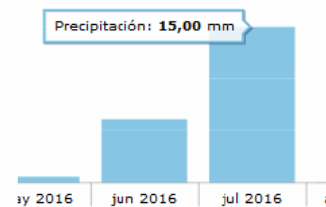


Fig. 15. Detalle *pop-up*: diagrama de barras de precipitación mensual acumulada – Arducultivo.

Para descargar los “*Informes*” en formato .csv, el cliente u operario previamente deberá seleccionar las variables que desea y la franja temporal elegida mediante un formulario que se enviará mediante método POST.

A través de un fichero PHP (ver Fig. 16) se realiza una consulta de selección de datos donde la fecha de inicio, la fecha de fin y las variables de las cuáles se quieren obtener los datos variarán en función de la selección realizada por parte del cliente en la página web. Finalmente, a través del fichero PHP, se creará un fichero llamado data.csv donde aparecerán los datos descargados.



Fig. 16. Diagrama de informes – Desarrollo web.

A modo de que el cliente tenga ubicadas sus estaciones, se ha realizado una página llamada “*Mapa*” donde pueden visualizarse las diferentes estaciones. Para ello se ha utilizado el *plugin* Huge IT Google Maps, donde, a través de éste, podremos modificar la ubicación, los iconos de posición y sus correspondientes etiquetas que aparecerán en el mapa.

Para concluir con el desarrollo web, es importante destacar que la página tiene un diseño *responsive* (ver Fig. 17), adaptándose, por tanto, a multitud de dispositivos móviles,

como tablets, *smartphones*, etc.



Figura 17. Diseño *responsive* – Arducultivo.

IX. CONCLUSIONES

El presente artículo ha presentado el proceso de diseño, desarrollo e implementación de dos prototipos llevados a cabo para la monitorización de parámetros climatológicos y ambientales, así como del suelo, para agricultura inteligente. Bajo la premisa de bajo coste, bajo consumo y sostenibilidad, ambos prototipos han sido diseñados para trabajar de forma autónoma mediante alimentación por paneles solares. Las plataformas llevadas a cabo se han diseñado para que se adecuen de la mejor forma posible a las necesidades particulares de cada cliente. Dicho cliente podrá obtener los datos requeridos mediante diferentes formas (SD card, SMS o envío por GSM), dependiendo del módulo escogido. El sistema permite una gran flexibilidad, para permitir monitorizar los datos, representarlos en tiempo real, mostrar gráficos de históricos, así como poder realizar descargas de los mismos, con el fin de poder ser procesados *a posteriori*.

REFERENCIAS

- [1] A. Barrera, “Nuevas realidades nuevos paradigmas: la nueva revolución agrícola”, Comunica online, España, pp. 10-20, 2011.
- [2] J. E. Gómez, S. Castaño, T. Mercado, A. Fernandez, J. García, “Sistema de Internet de las cosas (IoT) para el monitoreo de cultivos protegidos”, Ingeniería e Innovación, vol. 5 (1), pp. 27-36, 2017.
- [3] O. A. Orozco, G. Llano Ramírez. “Sistemas de información enfocados en tecnologías de agricultura de precisión y aplicables a la caña de azúcar, una revisión”, Revista Ingenierías Universidad de Medellín, vol. 15(28), pp. 103-124, 2016.
- [4] V. Rea-Sánchez, C. Maldonado-Cevallos, F. Villao-Santos, “Los Sistemas de Información para lograr un desarrollo competitivo en el sector agrícola”, Revista Ciencia UNEMI, vol. 8, no. 13, pp. 122-129, 2015.
- [5] A. J. Anaya-Isaza, D. H. Peluffo-Ordoñez, J. Ivan-Rios, J. A. Castro-Silva, D. A. Carvajal Ruiz, L. H. Espinosa Llanos, “Sistema de Riego Basado en la Internet de las cosas (IoT)”, ResearchGate, 2017.
Disponible en:
https://www.researchgate.net/publication/315793360_Sistema_de_Riego_Basado_En_La_Internet_De_Las_Cosas_IoT
- [6] J. A. Muñoz Arcentales, V. C. Calero Bravo, I. Marín-García, “Diseño e Implementación de un Sistema de Riego Inteligente basado en Sensores y Módulos de Radiofrecuencia para Transmisión y Sistema de Control”, ResearchGate, Guayaquil-Ecuador, 2014.
Disponible en:
https://www.researchgate.net/publication/258441493_Diseño_e_Implementación_de_un_Sistema_de_Riego_Inteligente_basado_en_Sensores_y_Módulos_de_Radiofrecuencia_para_Transmisión_y_Sistema_de_Control
- [7] M. Ríos Marín, L. Sánchez Padilla, J. Gómez Gómez, “Sistema de Monitoreo Agrícola mediante redes inalámbricas de sensores para el monitoreo de variables ambientales – SisMoAgro”, Ingeniería al Día, vol. 2, no. 2, 2016.