

# Desarrollo de una Estación de Medición Agroclimática con Hardware Libre

Emiliano López<sup>1</sup>, Carlos Vionnet<sup>1,2</sup>, Guillermo Contini<sup>1</sup>, Mariana Perez Elena<sup>1</sup>  
y Renato Manzo<sup>1</sup>

<sup>1</sup> Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral. RN  
168 - KM 472.4. (3000)

<sup>2</sup> Consejo Nacional de Investigaciones Científicas y Técnicas, Universidad Nacional  
del Litoral, Ciudad Universitaria, Paraje El Pozo, (3000) Santa Fe, Argentina

**Resumen** La medición de variables agroclimáticas es indispensable para comprender un fenómeno o proceso particular. Mantener registros periódicos con una buena resolución espacial demanda una considerable inversión en equipamiento comercial, generalmente de acotada flexibilidad para adaptarse a diferentes escenarios. El rápido avance de tecnologías emergentes han permitido el acceso a numerosos componentes electrónicos compatibles con plataformas de hardware libre como Arduino, cuyas especificaciones y software para su programación son de libre disponibilidad para su uso y modificación. Sensores variados y la plataforma de desarrollo Arduino fueron usados para diseñar una estación de medición automática con capacidades de almacenamiento y transmisión para su uso en proyectos relacionados con la agricultura y los recursos naturales. En el estudio actual se presentan resultados preliminares obtenidos en pruebas de laboratorio y campo donde se evaluaron variados sensores y componentes que dieron cuenta de su factibilidad en el desarrollo de una estación de monitoreo agroclimática.

**Keywords:** Hardware Libre, Arduino, Microcontroladores, Datalogger

## 1. Introducción

El registro periódico de variables agroclimáticas demanda una considerable inversión en equipamiento especializado. Difícilmente un único equipo mida la totalidad de las variables de interés, por lo que suele ser necesario contar con más de un dispositivo para realizar esta tarea. Esto no sólo incrementa los costos sino que dificulta el mantenimiento e integración de los datos ya que cada fabricante dispone de su propio software y formatos de salida con escasa flexibilidad para adaptarlo a necesidades particulares. La concepción cerrada que le imprime cada fabricante a su equipo dificulta el manejo flexible de la información, lo que en ocasiones repercute en la adecuada cobertura espacial y temporal de los parámetros de interés.

En los últimos años, el rápido avance de las tecnologías electrónicas posibilitó el acceso a una nueva variedad de componentes de bajo costo para el monitoreo y

control. Estas tecnologías emergentes permiten el acceso a numerosos elementos como microcontroladores programables, sensores, relojes de tiempo real (RTC por sus siglas en inglés), y módulos de memoria y comunicación, entre muchos otros que pueden ser fácilmente incorporados a un circuito simplificando de esta manera el diseño de un dispositivo de medición.

La elección de la plataforma de hardware impondrá el lenguaje de programación necesario para su uso, generalmente una variante de lenguaje de bajo nivel, esto es, ensamblador. Distintos fabricantes utilizarán lenguajes completamente incompatibles entre sí, por lo que un cambio de plataforma implicará la reescritura total del código. Un breve ejemplo de dos programas que realizan una lectura de datos de un mismo sensor se observa en el Listado 1.1 y 1.2. Si bien estos códigos realizan idéntica acción, esto es, leer de un sensor distanciómetro su valor, se observa una gran diferencia entre ambos programas. El Listado 1.2 muestra el lenguaje de un microcontrolador BasicX-24p cuyo conjunto de comandos es sustancialmente diferente al del Listado 1.1 para la familia de microcontroladores Atmel AVR.

```
Dim Dist As Word , Strdist As
    String * 8 , Serdist As Byte
Open "comd.7:9600,8,n,1,INVERTED"
    " For Input As #1
Config Adc = Single , Prescaler =
    Auto , Reference =
        Internal
Start Adc
Do
Dist = Getadc(0)
Shift Dist , Right , 2
Input #1 , Strdist Strdist =
    Right(strdist , 3)
Serdist = Val(strdist)
Print "Analog Distance = " ;
    Dist ;
Print "Serial Distance = " ;
    Serdist ;
Loop
```

**Listado 1.1.** Hardware AVR

```
Const RX As Byte = 10
Const AN As Byte = 13
Dim AnalogOutPut As Byte
Dim SerialOutPut As Byte
Dim PWOutPut As Byte
Sub Main()
Do
    AnalogOutPut = RangeA
    Debug.Print CStr(AnalogOutPut)
    Call SLEEP(512)
Loop
End Sub
Function RangeA() As Byte
Dim AValue As Integer
Call PutPin (RX, 0)
Call PutPin (RX, 1)
Call Sleep(40)
AValue = GetADC(AN)
RangeA = CByte(AValue \ 2)
End Function
```

**Listado 1.2.** Hardware BX24p

Una alternativa como solución a esta limitante es el uso del hardware libre, cuyo concepto se asemeja al del software libre pero aplicándose a los diseños, esquemáticos, especificaciones técnicas y software para programar el hardware [20]. De esto surge una serie de ventajas que van mas allá de la reducción de costos. Las licencias libres (Creative Commons<sup>3</sup> y GNU Public License<sup>4</sup>, por ejemplo) permiten el acceso y la modificación de los códigos fuente del software y diseños del hardware, lo que ha dado como resultado una comunidad activa de desarrolladores que han publicado herramientas de software que transparentan el proceso de programación de diferentes plataformas de hardware utilizando lenguajes de alto nivel (por ej, C/C++) facilitando la independencia del hard-

<sup>3</sup> creativecommons.org/licenses

<sup>4</sup> gnu.org/licenses/gpl.html

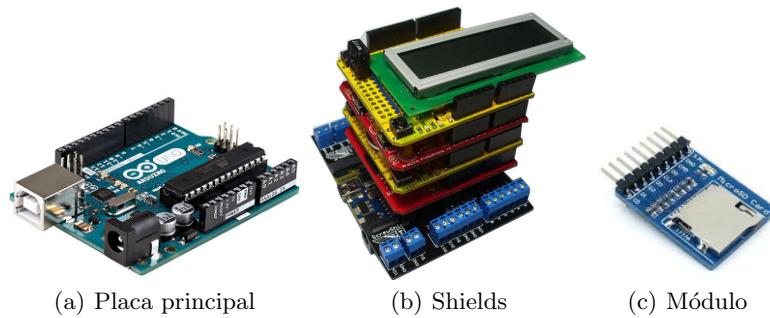
ware. De esta manera, diferentes placas, con variados microcontroladores, se programan bajo un mismo lenguaje<sup>5</sup>.

El uso de lenguajes de alto nivel facilitan el diseño del software y la reutilización de módulos a través de bibliotecas, cuya incorporación extiende las funcionalidades de un programa. En el Listado 1.3 se observa el uso de una biblioteca externa -bajo el paradigma de orientación a objetos- para la lectura del mismo sensor de los casos previos (Listados 1.1 y 1.2). Tal como se observa, el código se asemeja al de un lenguaje de programación de propósito general, con todas las ventajas sobre el lenguaje ensamblador que ello implica.

```
#include "Maxbotix.h" // uso de biblioteca externa
Maxbotix Sensor(6, Maxbotix::TX, Maxbotix::HRLV);
void setup() {
    Serial.begin(9600); // velocidad del puerto serie
}
void loop(){
    float dist = Sensor.getRange(); // lectura del sensor
    Serial.println(dist); // salida por puerto serie
    delay(2000); // espera de 2 segundos
}
```

**Listado 1.3.** Código de alto nivel

Una de las plataformas de hardware libre más populares es Arduino [4], compuesta por una placa principal, shields, módulos y un entorno integrado de desarrollo para su programación. La placa principal consiste en el microcontrolador programable y un conjunto de puertos de entradas/salidas analógicas y digitales. Los shields son componentes que agregan una determinada funcionalidad (almacenamiento y red por ej.), cuya disposición de pines coincide con los de la placa principal, por lo que puede conectarse fácilmente en forma de pila. Los módulos sólo se diferencian de los shields en que no mantienen la disposición de conexión estándar a la placa principal. Todo estos componentes pueden interactuar con el mundo físico a través de sensores analógicos o digitales y, en este último caso diferentes protocolos de comunicación (RS232, TTL, SPI, I2C, etc.). En la Fig. 1 se observan estos componentes.



**Figura 1.** Componentes de la plataforma Arduino

<sup>5</sup> Referencia del lenguaje Arduino: [www.arduino.cc/en/Reference](http://www.arduino.cc/en/Reference)

En los últimos años se desarrollaron una gran variedad de instrumentos de medición y control basados en la plataforma de hardware libre Arduino, sistemas que van desde el riego automático [8], evaporación diaria [9], hasta dispositivos para determinar la concentración de contaminantes en la atmósfera [6]. Otros trabajos como [10] y [11] desarrollaron dispositivos de almacenamiento con módulos de transmisión inalámbrica en una gran cantidad de puntos de medición en una cuenca hidrológica<sup>6</sup>. Publicaciones más recientes muestran que el uso de estas tecnologías continúa expandiéndose, principalmente en lo que refiere a dispositivos para el desarrollo de redes de monitoreo hidrológico [13]. Estos antecedentes dan cuenta de la factibilidad de la plataforma Arduino para el desarrollo de dispositivos de medición de variables agroclimáticas.

Si bien este trabajo aún se encuentra en plena ejecución, la presente comunicación muestra los avances significativos realizados a la fecha, puesto que se analizaron una variada gama de placas, sensores, módulos y shields a la par que se diseñaron componentes propios que permitieron cubrir una gran cuota de las actividades desarrolladas por los diversos grupos que desempeñan sus actividades de investigación dentro del Centro de Estudio Hidroambientales (CENEHA).

## 2. Materiales y Métodos

### 2.1. Arquitectura del Sistema

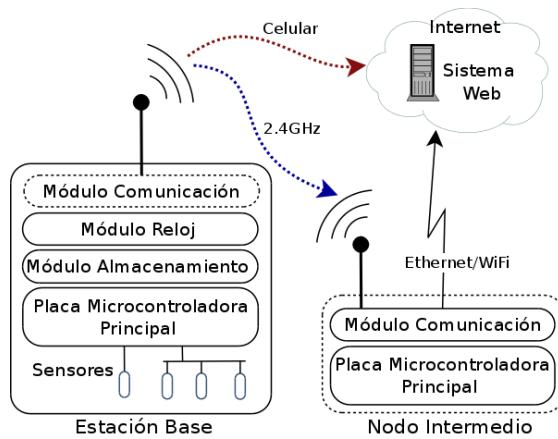
El sistema consiste en dos subsistemas, por un lado el dispositivo electrónico de medición instalado en campo y, por otro lado, el sistema web que recibe y almacena los datos para su visualización y gestión remota. El dispositivo fue diseñado de manera modular, teniendo en cuenta diferentes escenarios: una estación aislada que sólo almacena las mediciones, sin transmisión de datos, una que los envía utilizando tecnología de comunicación celular y otra donde los datos se envían a un nodo intermedio que se encuentra conectado a Internet y desde aquí se suben hacia el servidor. El esquema fue diseñado teniendo en cuenta sitios geográficos sin ninguna posibilidad de conexión a Internet y otros donde es posible utilizar una conexión a la red en las cercanías evitando afrontar el costo de la transmisión vía tecnología celular. En la Fig. 2 se muestra la estructura general del sistema teniendo en cuenta ambos escenarios.

### 2.2. Estación Base

La estación base consta de las funcionalidades indispensables para un equipo de medición en campo, esto es, que sea capaz de medir una amplia gama de variables de interés y almacenarlas en una tarjeta de memoria SD fácilmente extraíble y reemplazable. Más precisamente, la estación base está conformada por un microcontrolador principal y un módulo para almacenamiento, tal como se detalla a continuación.

---

<sup>6</sup> Critical Zone Observatory: <http://criticalzone.org/christina>



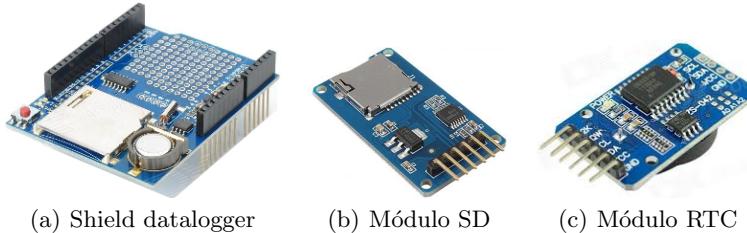
**Figura 2.** Esquema general del sistema. Dispositivo de medición (izq.) con alternativa de envío hacia Internet por telefonía celular o a través de un Nodo intermedio (der.)

**Placa Principal.** El Arduino UNO (Fig. 1a) fue la placa utilizada en gran parte del desarrollo inicial del proyecto. Cuenta con una serie de ventajas y un buen equilibrio entre los recursos disponibles (procesamiento y memoria) y consumo energético. Al ser la placa más popular, la mayoría de los shields presentes en el mercado son compatibles. Su flexibilidad en la alimentación (USB / CC 5v a 20v ), y la facilidad de conexión a una PC para transferir el firmware la hicieron la opción por defecto en el comienzo del proyecto. No obstante sus ventajas, durante la etapa de pruebas y análisis se presentaron dificultades para reducir el elevado consumo del módulo USB, aún programando los modos de bajo consumo soportados por el microcontrolador. Otro componente ineficiente es el regulador de voltaje interno debido a que no es una placa orientada al trabajo en campo, tal como se lo necesita para una estación de medición. Por tanto, su aparente flexibilidad va en detrimento de su supuesta autonomía.

Por otra parte, la placa Arduino Pro Mini cuenta con el mismo microcontrolador (ATmega328P), por lo que es compatible con cualquier componente que funcione con Arduino UNO. No obstante, carece del módulo USB por lo que no cuenta con una gran flexibilidad, pero es capaz de lograr una mayor autonomía que la anterior. La dificultad que impone está relacionada con la disposición de sus pines, lo que impide conectar directamente un shield, por lo que a la hora de programarlo es necesario utilizar un módulo conversor USB/TTL para transferir el programa desde la PC.

**Almacenamiento.** Se utilizó y analizó el shield datalogger XD-05 capaz de brindar la fecha y hora de una medición y a su vez almacenarlo en una tarjeta de memoria SD. En los prototipos iniciales fue una opción acertada ya que posibilitó enfocarse en el funcionamiento global del sistema. El inconveniente

de este componente estuvo relacionado al consumo energético, por lo que en etapas posteriores fue reemplazado por módulos independientes de reloj (RTC DS3231) y almacenamiento (Catalex microSD). Ambos fueron soldados en una placa de prototipado manteniendo el conexionado estándar, de esta manera se implementó un shield datalogger propio de consumo reducido. Los componentes se observan en la Fig. 3.



**Figura 3.** Alternativas para el almacenamiento

### 2.3. Variables Registradas

**Humedad y Temperatura Ambiente.** Se utilizó el sensor digital DHT22. Mide valores de humedad entre 0 y 100 % con una precisión de 2 % y un rango de valores de temperatura entre -40°C y 80°C con +/- 0.5°C de precisión. El intervalo mínimo entre muestras es de 2 segundos [2].

**Humedad y Temperatura de Suelo.** Para la medición de temperatura (en suelo/agua) se analizó el sensor analógico LM35 y el digital DS18B20. Este último, tiene un rango de medición va de entre -55C y 125°C (+/-0.5°C). Utiliza el protocolo One-Wire por lo que cada sensor es identificado con un número serie único de 64 bits lo que permite una enorme cantidad de sensores sobre un mismo canal de datos [15].

La medición de la humedad del suelo presenta una serie de dificultades asociadas al tipo de suelo. Dependiendo de la tecnología del sensor, un mismo valor puede significar estados de saturación muy diferentes en distintos suelos. Por este motivo, generalmente se requiere una correcta caracterización del terreno donde se utilizará el sensor. En el presente trabajo se utilizaron tres sensores.

Se programó el dispositivo Arduino como interfaz para el sensor HydraProbeII, debido a que es ampliamente utilizado a nivel mundial. Se probó además el sensor analógico FC28 que registra la resistencia eléctrica del suelo, a través de dos electrodos con diferente potencial eléctrico para pasar corriente a través del suelo y el sensor digital SHT10 permite medir temperatura y humedad de suelo con una precisión de 0.5°C para la temperatura y 4.5 % para humedad. Su rango de operación es de -40°C a 120°C y 0 a 100 % rh (Relative Humidity).

El sensor se encuentra protegido por una cápsula porosa que evita filtración de agua pero permite el paso del aire para que pueda medir la humedad [3].

**Nivel freático, Evaporación y Altura Hidrométrica.** Para la medición de niveles se utilizaron distancímetros acústicos e infrarrojos. En los primeros, el principio de funcionamiento consiste en la emisión de un pulso ultrasónico detectando el tiempo de demora en recibir la onda reflejada en el obstáculo. De la gran variedad de sensores existentes, se diferencian principalmente en el rango de medición (centímetros o metros), resolución (centímetros o milímetros) y tipo de comunicación (analógico, TTL, PWM, etc) entre otras características. Se analizaron una amplia gama de sensores, entre ellos SRF04, E00416, HCSR04, MB7092-XL y MB1013-HRLV.

Los distancímetros infrarrojos están equipados con dos lentes, uno que emite un haz de luz y otro -que lo recibe- provisto de un fotodetector sensible a la posición (PSD, de sus siglas en inglés), por lo que es posible determinar la distancia de un objeto en función de la posición del sensor que se capta el haz. Para este caso se analizó el sensor analógico Sharp 2Y0A710, con un rango de medición de entre 100 cm a 550 cm.

**Humedad de Hoja/Lluvia** El sensor de humedad en superficie, que emula la humedad de hoja de las plantas, es útil para prever posibles infecciones por hongos. En este caso se utilizó un sensor analógico que mide el cambio de resistencia en función de la superficie mojada.

**Presión atmosférica** Se utilizó el sensor digital BMP183<sup>7</sup> cuyo rango de medición es de 300 a 1100 hPa, contiene además un sensor de temperatura, utiliza el protocolo SPI para su comunicación y tiene un bajo consumo por lo que es ideal para integrar la estación de medición.

**Velocidad del Viento.** Se utilizó un anemómetro analógico, cuyo voltaje de salida se corresponde linealmente con la velocidad del viento. Su rango de operación es de 0 a 50 m/s y una resolución de 0.1 m/s [1].

#### 2.4. Transmisión

La estación base permite incorporar alternativas de transmisión hacia Internet contemplando diferentes escenarios: i) para aquellos lugares donde la única posibilidad de acceso sea a través de la red de telefonía celular o, ii) para aquellos sitios con una conexión cercana a Internet (1km aprox.).

Para la transmisión celular se programaron los shields SM5100b [19] y SIM900 [14]. En ambos casos, el envío de datos es a través de solicitudes HTTP mediante

---

<sup>7</sup> BMP183 datasheet: <http://media.digikey.com/pdf/Data%20Sheets/Bosch/BMP183.pdf>



**Figura 4.** Sensores utilizados para el registro de las variables

comandos AT, por lo que para el análisis inicial se utilizaron servicios gratuitos de IoT como Xively<sup>8</sup> y finalmente el sistema web desarrollado.

El otro escenario planteado envía los datos hacia Internet usando un nodo intermedio. Esta alternativa suele ser frecuente en la instalación de estaciones, donde algún establecimiento rural en las cercanías facilita la conexión, evitando afrontar los costos mensuales de la transmisión celular y protegiendo la estación del vandalismo. El nodo intermedio (Fig. 2) funciona como una estación de retransmisión, por un lado recibe los datos a través de un enlace inalámbrico y por otro lado se conecta a Internet utilizando Ethernet o WiFi 802.11, dependiendo de la conexión del lugar.

El enlace de radiofrecuencia entre la estación base y el nodo intermedio se realizó utilizando el módulo NRF24L01 [16] cuyas características destacables son la corrección de errores y protocolo de reenvío, comunicación dúplex, banda de libre uso a nivel mundial (2.4GHz), velocidad configurable de 250 kbps, 1 o 2 Mbps, bajo consumo, alcance hasta 1.1km. Si el nodo intermedio cuenta con una conexión cableada al alcance es posible incorporar el shield Ethernet y programarlo usando la biblioteca que permite establecer solicitudes HTTP para el envío de los datos hacia Internet. Por otra parte, si se dispone de una conexión inalámbrica 802.11 se utiliza la placa ESP8266[7] que permite una comunicación serial por software para generar una solicitud HTTP. Al igual que el módulo de transmisión celular, las solicitudes se realizan mediante comandos AT.

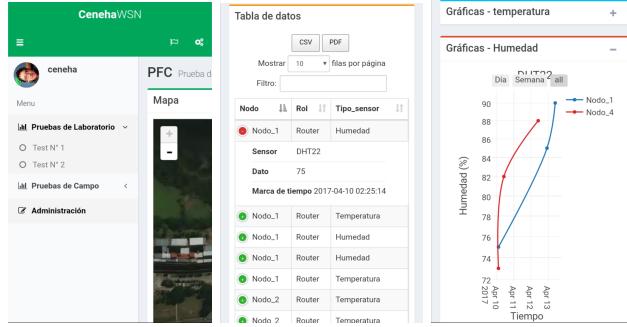
## 2.5. Aplicación Web

Las estaciones de medición con transmisión envían sus datos a un sistema web alojado en un servidor en Internet. El desarrollo del sistema fue incremental basado en el modelo de prototipos [18]. La aplicación, que actualmente corre

---

<sup>8</sup> IoT platform for connected devices, [www.xively.com](http://www.xively.com)

sobre la plataforma Heroku<sup>9</sup>, fue desarrollada en lenguaje Python [17] utilizando el framework Django [5].



**Figura 5.** visualización del sistema web desde smartphone

La aplicación tiene una doble funcionalidad, es una interfaz gráfica para visualizar y descargar los datos en formato estándar (CSV, del inglés Comma Separated Value) y, provee una API (del inglés Application Programming Interface) para que las estaciones se puedan comunicar y enviar sus registros.

## 2.6. Autonomía de la estación

Una de las cualidades buscadas en el diseño de una estación de medición es su autonomía. En el presente trabajo se avanzó sobre la reducción del consumo de la estación base, sin los módulos de transmisión. El microcontrolador que utiliza tanto la placa Arduino UNO como la ProMini (ATmega328P) soporta un modo de trabajo de bajo consumo por software. Esto fue programado utilizando la biblioteca *LowPower*<sup>10</sup> lo que permite un estado de suspensión de entre seis posibles, donde cada uno desactiva distintos componentes de la placa. Una vez que el Arduino entra en estado de suspensión existen dos alternativas para reactivarlo: mediante una interrupción externa o después de un tiempo determinado. Esta última es la que se utilizó permitiendo ser modificada por el usuario de manera remota.

Desde el punto de vista electrónico se deshabilitaron los leds de la placa Pro Mini, y se implementó un circuito de Darlington [12] que desconecta los módulos de almacenamiento y reloj (SD y RTC) cuando no son utilizados.

<sup>9</sup> Cloud Application Platform: [www.heroku.com](http://www.heroku.com)

<sup>10</sup> Biblioteca LowPower: <https://github.com/rocketscream/Low-Power>

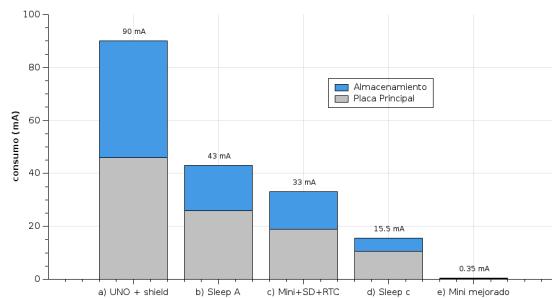
### 3. Resultados y Discusión

El sistema fue analizado a partir de pruebas en laboratorio -bajo condiciones controladas y exhaustivas- y a la intemperie en el predio de la Ciudad Universitaria de la FICH, UNL (Fig. 6). El diseño de las pruebas se orientó a evaluar tres elementos: *i*) la estación base -estabilidad y consumo eléctrico-; *ii*) la transmisión -distancia máxima y confiabilidad; *iii*) el desempeño de los sensores.



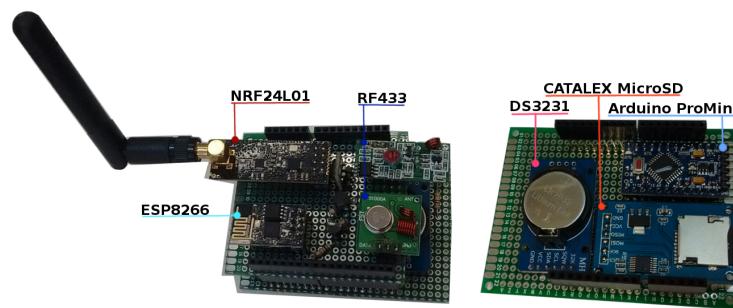
**Figura 6.** Pruebas en laboratorio y campo

La estación base fue modificada hasta alcanzar un nivel de autonomía aceptable para su instalación en campo. El consumo instantáneo se redujo de 90 mA a 0.35 mA, alcanzando una autonomía de aproximadamente 2 años con una batería NP7. Las sucesivas mejoras realizadas consistieron en cinco alternativas cuyo consumo se observa en la Fig. 7. La primera (a), conformada por la placa Arduino UNO y el shield de almacenamiento, en el caso (b) se programó el modo sleep, donde se observa que el consumo aún es elevado, en (c) se utilizó la placa Pro Mini con un módulo RTC y otro de tarjetas de memoria micro SD, la modificación (d) es el modo sleep de la alternativa previa y la última mejora consistió en la desconexión de los módulos de almacenamiento (SD y RTC) a partir de un circuito de Darlington para anularlos cuando no son utilizados.



**Figura 7.** Consumo de la estación base

El análisis de la transmisión hacia Internet a través de un nodo intermedio se realizó replicando lo enviado en la estación de medición, en el nodo intermedio y en el servidor final con una frecuencia de un minuto. La prueba se realizó durante ocho días sin pérdida de datos entre la estación y el nodo intermedio. Hubo fallas de menos de un 4 % entre el nodo intermedio y el servidor web, atribuibles a micro cortes en la conexión a Internet. Para el análisis de los módulos inalámbricos de transmisión se implementó un shield prototipo compatible con el diseño de la estación base (Fig. 8).



**Figura 8.** Diseños de shield de transmisión (izq.) y estación base (der.)

El comportamiento de los sensores fue contrastado con registros manuales (por ej, nivel freático) y con estaciones comerciales (por ej, humedad y temperatura) Pegasus<sup>11</sup>, Eddy Covariance<sup>12</sup>, Stevens<sup>13</sup> y Watchdog<sup>14</sup>. Si bien esta etapa se encuentra en pleno desarrollo, los resultados preliminares permitieron descartar sensores de funcionamiento anómalo y enfatizar en aquellos de comportamiento razonable.

En la Fig. 9 se observan resultados preliminares obtenidos de algunos de los sensores. La humedad ambiente medida por el DHT22 contrastada con la Eddy Covariance (a) muestra un desplazamiento constante por lo que con una calibración se obtuvieron buenos resultados. La velocidad del viento fue contrastada con la estación Pegasus (b) observándose un buen nivel de coincidencia entre ambos. El sensor de temperatura LM35 (c), al igual que otros sensores analógicos, presenta una gran susceptibilidad al ruido eléctrico. El sensor DS18B20 dio excelentes resultados en contraste con el sensor HydraProbeII de Stevens para registrar la temperatura del suelo (d). El registro del nivel freático (e) se realizó con el sensor MB7092, que dio buenos resultados tanto en laboratorio como en campo. Si bien el sensor Sharp 2Y0A710 tuvo un buen comportamiento en la-

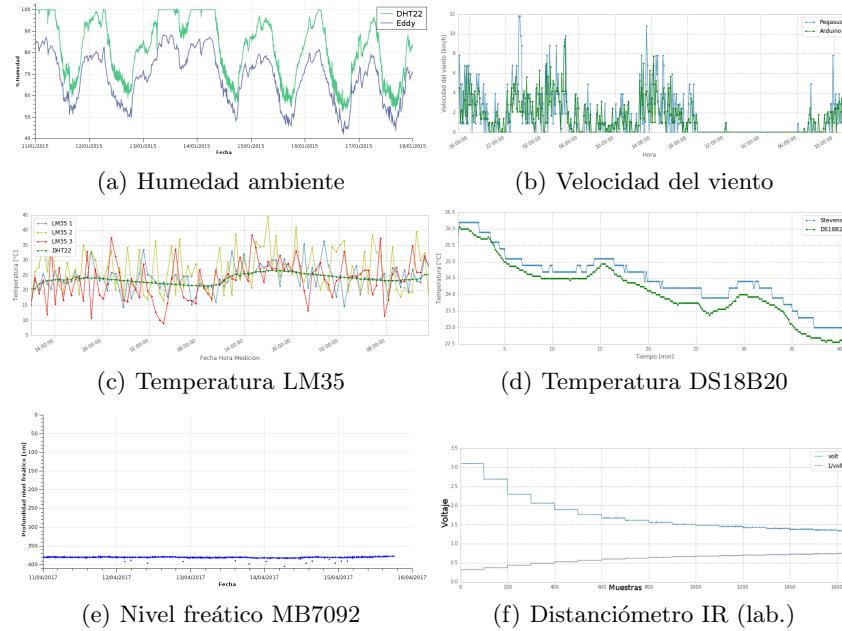
<sup>11</sup> [www.tecmes.com](http://www.tecmes.com)

<sup>12</sup> [www.campbellsci.com](http://www.campbellsci.com)

<sup>13</sup> [www.stevenswater.com](http://www.stevenswater.com)

<sup>14</sup> [www.specmeters.com](http://www.specmeters.com)

boratorio (f) aun no fue probado en campo debido a que debe acondicionarse la señal por software y hardware para filtrar el ruido presente en las lecturas.



**Figura 9.** Contraste con estaciones de medición comerciales

De acuerdo a los resultados obtenidos en pruebas de laboratorio y campo es posible considerar a Arduino como una plataforma válida en el desarrollo de instrumental de medición, esta afirmación se fortalece con la disponibilidad creciente de módulos y sensores fácilmente integrables. El sistema de hardware y software diseñado y desarrollado para la estación base dio pruebas de su estabilidad y autonomía, su bajo consumo es un avance relevante sumado a la capacidad de almacenamiento de varios GBs en contraste con las ofrecidas por equipos comerciales de unos pocos MBs.

Respecto a la vulnerabilidad al ruido, los resultados sobre los sensores analógicos han mostrado que requieren de un proceso de filtrado -por hardware o software- para evitar que los afecte. Vale la pena poner énfasis en que estos aspectos cuestionables pueden ser mejorados o modificados ya que todas sus especificaciones técnicas se encuentran disponibles bajo licencias libres que lo permiten estudiar, analizar y por lo tanto conocer al detalle su funcionamiento y por ende adaptarlo para mejorarlo.

#### 4. Conclusiones

Avances en tecnologías electrónica han dado lugar al surgimiento de microcontroladores, sensores y variados componentes de bajo costo para la medición, registro y transmisión de variables agroclimáticas. Las plataformas de hardware libre distribuyen sus diseños, esquemáticos y software bajo licencias libres, permitiendo su uso y modificación, lo que ha facilitado enormemente los esfuerzos necesarios para incorporar funcionalidades a un dispositivo de medición.

Basado en la plataforma Arduino se desarrolló un sistema de medición modular y de bajo costo capaz de transmitir datos hacia Internet adaptable a diferentes escenarios junto a un sitio web para su visualización. La estación desarrollada tiene la flexibilidad necesaria para modificar sus funciones, incorporar sensores y módulos con diferentes tecnologías de transmisión, reduciendo los costos en un orden de magnitud en contraposición a alternativas comerciales.

#### 5. Trabajos Futuros

Actualmente se encuentra en proceso de desarrollo el diseño del prototipo de la estructura de montaje y su impresión en diferentes materiales (PLA, ABS y nylon). Existen además dos líneas de investigación abiertas, por un lado la implementación de una red de sensores inalámbricos y por otro lado, la validación y pruebas de un pluviómetro desarrollado por el Centro de Informaciones Meteorológicas<sup>15</sup> bajo los estándares de la Organización Meteorológica Mundial, para ser incorporado a la estación aquí presentada.

#### Referencias

1. Adafruit: Anemometer wind speed sensor w/analog voltage output. <http://www.adafruit.com/product/385> (2014), online; último acceso 2014-11-28
2. Adafruit: DHT22 Temperature - Humidity sensor + extras. <http://www.adafruit.com/product/385> (2014), online; último acceso 2014-11-28
3. Adafruit: Soil temperature/moisture sen. <https://www.adafruit.com/product/1298> (2015), online; último acceso 2015-06-18
4. Arduino: Open-source electronics platform. <http://arduino.cc> (2014), <http://arduino.cc>, online; último Acceso 2014-06-02
5. Community, D.: Django: The web framework for perfectionists with deadlines. <https://www.djangoproject.com> (2015), online; último acceso 2016-07-03
6. Cremades, Pablo y Castro, F.y.F.R.y.C.R.y.P.E.: Desarrollo de un monitor abierto de calidad del aire (MACA) (2013), Séptimo Encuentro de Investigadores y Docentes de Ingeniería. EnIDI. Los Reyunos, San Rafael. Mendoza, Argentina
7. Espressif: ESP8266EX Datasheet Version 4.3. <http://download.arduino.org/products/UNOWIFI/0A-ESP8266-Datasheet-EN-v4.3.pdf> (2015), online; último Acceso 25-11-2015
8. Fisher, D.K.: Open-source hardware is a low-cost alternative for scientific instrumentation and research. Modern Instrumentation 2012 (2012)

<sup>15</sup> <http://fich.unl.edu.ar/cim>

9. Fisher, D.K., Sui, R.: An inexpensive open-source ultrasonic sensing system for monitoring liquid levels. Agricultural Engineering International: CIGR Journal 15(4), 328–334 (2013)
10. Hicks, Steven D. y Aufdenkampe, A.K.y.M.D.S.: Creative uses of custom electronics for environmental monitoring at the christina river basin czo (2011), stroud Water Research Center, Avondale, Pennsylvania
11. Hicks, S.D., Aufdenkampe, A.K., Montgomery, D.S.: Sensor networks, dataloggers, and other handy gadgets using open-source electronics for the christina river basin czo (2012), stroud Water Research Center, Avondale, Pennsylvania
12. Hodes, D.A.: Darlington's contributions to transistor circuit design. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 46(1), 102–104 (1999)
13. Hund, Silja y Johnson, M.y.K.T.: Developing a hydrologic monitoring network in data-scarce regions using open-source arduino dataloggers. Agricultural Environmental Letters 2016 (2016)
14. LinkSprite: SIM900 GPRS-GSM Shield - LinkSprite Playgound. <http://linksprite.com/wiki> (2014), online; última visita 2014-06-18
15. MaximIntegrated: DS18B20 Programmable Resolution 1-Wire Digital Thermometer. <http://www.maximintegrated.com/en/products/analog/sensors-and-sensor-interface/DS18B20.html> (2014), online; último acceso 2014-11-24
16. Nordic: nRF24L01 Ultra low power 2.4GHz RF Transceiver IC. <http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01> (2015), online; último acceso 2015-10-25
17. van Rossum, G.: Python programming language. In: USENIX Annual Technical Conf. (2007)
18. Sommerville, I.: Ingeniería del Software. (7ma edición). Pearson Addison-Wesley, Madrid. España (2008)
19. Spreadtrum: SM5100B-D GSM/GPRS Module Specification.
20. Stallman, R.: Free Hardware and Free Hardware Designs. <https://www.gnu.org/philosophy/free-hardware-designs.html> (2016), online; último Acceso 2016-12-20