

# Prototype of mini-weather station with cloud service through Arduino and PHP

Oscar Jhan Marcos Peña-Cáceres, Mg<sup>1</sup>, Manuel Alejandro More-More, Dr<sup>2</sup>, Rudy Espinoza-Nima, Mg<sup>3</sup>, Henry Silva-Marchan, Mg<sup>4</sup>, Jorge Yáñez-Palacios, Mg<sup>5</sup>

<sup>1</sup>Universitat de València, España, osjmarpe@alumni.uv.es

<sup>2,3</sup>Universidad Nacional de Piura, Perú, mmore@unp.edu.pe; respinozan@unp.edu.pe

<sup>4</sup>Universidad Nacional de Tumbes, Perú, hsilvam@untumbes.edu.pe

<sup>5</sup>Universidad de Guayaquil, Ecuador, jorge.yanezp@ug.edu.ec

*Abstract - Information technologies are the perfect medium for the development of multidisciplinary proposals that contribute to the development of digital media for the benefit of society on the tracking and monitoring of the weather. The objective of this research is to develop a prototype of a low-cost mini-weather station with cloud service through Arduino, PHP and MySQL. The methodological design comprises in the identification of technological resources, design, coding, testing and implementation. In this first approach the mini-meteorological station allows obtaining data on ambient temperature, relative humidity and atmospheric pressure. The tests were conducted during 2017,2018 and 2019 in order to assess the operability of the prototype in different seasons of the year. The data quality performance represents an accuracy of 93.76% for the environmental temperature variable, 99% on atmospheric pressure and 50% for relative humidity. It is recommended that a FC-37 rain sensor and a JL-FS2 anemometer to measure wind speed be integrated to the prototype in future research.*

**Keywords--** *Prototype, mini-weather station, cloud, Arduino, PHP.*

**Digital Object Identifier:** (only for full papers, inserted by LACCEI).  
**ISSN, ISBN:** (to be inserted by LACCEI).  
**DO NOT REMOVE**

# Prototipo de mini-estación meteorológica con servicio en la nube a través de Arduino y PHP

Oscar Jhan Marcos Peña-Cáceres, Mg<sup>1</sup>, Manuel Alejandro More-More, Dr<sup>2</sup>,  
Rudy Espinoza-Nima, Mg<sup>3</sup>, Henry Silva-Marchan, Mg<sup>4</sup>, Jorge Yáñez-Palacios, Mg<sup>5</sup>

<sup>1</sup>Universitat de València, España, osjmarpe@alumni.uv.es

<sup>2,3</sup>Universidad Nacional de Piura, Perú, mmore@unp.edu.pe; respinozan@unp.edu.pe

<sup>4</sup>Universidad Nacional de Tumbes, Perú, hsilvam@untumbes.edu.pe

<sup>5</sup>Universidad de Guayaquil, Ecuador, jorge.yanezp@ug.edu.ec

**Resumen**– Las tecnologías de la información son el medio perfecto para el desarrollo de propuestas multidisciplinarias que coadyuven al desarrollo de medios digitales en beneficio de la sociedad sobre el seguimiento y monitoreo del clima. El objetivo de esta investigación es desarrollar un prototipo de mini-estación meteorológica a bajo costo con servicio en la nube a través de Arduino, PHP y MySQL. El diseño metodológico comprende en la identificación de recursos tecnológicos, diseño, codificación, pruebas e implementación. En este primer acercamiento la mini-estación meteorológica permite obtener datos de temperatura ambiental, humedad relativa y presión atmosférica. Las pruebas se realizaron durante los años 2017,2018 y 2019 con la finalidad de valorar la operatividad del prototipo en diferentes estaciones del año. El desempeño de la calidad de los datos representa una precisión del 93.76% para la variable temperatura ambiental, 99% sobre la presión atmosférica y 50% para humedad relativa. Se recomienda que en futuras investigaciones al prototipo se integre un sensor de lluvia FC-37 y un anemómetro JL-FS2 para medir la velocidad del viento.

**Palabras clave**-- Prototipo, mini-estación meteorológica, nube, Arduino, PHP.

## I. INTRODUCCIÓN

El cambio climático hoy en día es uno de los tópicos que ha empezado a ganar terreno y ser promovido por la comunidad científica para analizar, evaluar y recomendar sobre posibles eventos o desastres que puedan ocurrir en un espacio geográfico. En mucho de los casos las instituciones técnico científicas [1] son las encargadas de monitorear y hacer seguimiento sobre el comportamiento del clima [2]. Sin embargo, la adquisición de equipos meteorológicos es limitada por los elevados costos, situación que impedita el acceso a información en tiempo real. La tecnología Arduino, es más que una plataforma de prototipo electrónica de código abierto [3] o también definida como una placa hardware libre que incorpora un microcontrolador reprogramable y una serie de pines-hembra que permiten allí de forma muy sencilla y cómoda ejecución de diferentes sensores y actuadores [4]. Mediante esta tecnología es posible desarrollar proyectos de bajo costo [5] enmarcados en los sectores como educación, salud, agricultura, minería, hidrología, meteorología y particularmente el medio ambiente. La capacidad que tiene Arduino para integrarse con otras tecnologías es amplia y gracias a ello es posible transmitir datos en tiempo real, dando

sinergia y dinamización en la previsión de resultados que pueda generar el dispositivo. La incorporación de nuevos componentes se ha hecho más visible dando impulso a innovaciones y proyectos coloquiales orientados a soluciones inmediatas, económicas y efectivas con un impacto benéfico para la sociedad.

La construcción del prototipo de mini-estación meteorológica es de contexto transversal, ya que involucra su funcionamiento en zonas rurales o urbanas donde se carece de este tipo de instrumentos y su operatividad puede estar sujeta a espacios internos como externos con la finalidad de conocer el estado del clima. Entre las principales variables meteorológicas, se conoce la temperatura ambiental, humedad relativa y presión atmosférica. Para obtener este tipo de datos, se recurre a sensores como, BMP085 [6] y DHT22 [7] donde su integración es accesible y administrable a la tecnología Arduino.

Para el diseño del prototipo, se considera hacer uso del software Fritzing, por ser un programa libre y automatizable para el diseño electrónico que permite simular y verificar errores comunes que se puedan encontrar durante la etapa desarrollo [8]. Para la codificación se emplearán técnicas de programación orientadas al lenguaje de programación C++, nativo de Arduino para la lectura de los sensores y PHP como medio de recepción y envío de datos, lenguaje de programación que permite un rápido desarrollo de aplicaciones web [9]. Por otro lado, se utilizará MySQL para el almacenamiento de los datos por considerarse un sistema open source de administración de base de datos [10].

Es importante conocer que medios tecnológicos son compatibles para este tipo de estudios o en su mayoría, similares en uso. La integración de Arduino, PHP y MySQL, permite desarrollar proyectos amplios en conocimiento y sobre todo que cumplan las expectativas de los usuarios. En este contexto, el prototipo de mini-estación meteorológica, tendrá la funcionalidad de visualizar datos en tiempo real y generar información histórica de carácter técnico científico que oriente a una eficiente toma de decisiones a las autoridades, profesionales y academia en la planificación de actividades prospectivas que beneficien a la población y se impulse el

componente prospectivo en materia de gestión del riesgo de desastres.

Finalmente, el objetivo del presente estudio es desarrollar un prototipo de mini-estación meteorológica a bajo costo con servicio en la nube a través de Arduino, PHP y MySQL, con la finalidad de obtener de datos en tiempo real de temperatura ambiental, humedad relativa y presión atmosférica. Además de ello, será posible la clasificación de los datos para conocer un comportamiento histórico de cada variable meteorológica conllevando a un análisis y a la formulación de informes técnicos científicos o boletines meteorológicos. Por otro lado, los datos obtenidos brindaran resultados, como, el promedio de temperatura máxima y mínima, valor máximo y mínimo de humedad relativa y presión atmosférica durante el día.

## II. METODOLOGÍA

La metodología abarca cuatro fases que comprende, la identificación de recursos tecnológicos, diseño, codificación, pruebas e implementación.

### 2.1. Identificación de recursos tecnológicos

En muchas oportunidades es difícil identificar recursos tecnológicos cuya integración sea accesible y administrable mucho más aún cuando el estudio tiene como finalidad obtener datos de temperatura ambiental, humedad relativa y presión atmosférica en tiempo real, además de sobreestimar el promedio, valor máximo y valor mínimo de cada una las variables meteorológicas antes mencionadas.

Tal como se mencionó párrafos atrás, Arduino es un medio tecnológico compuesto por el lenguaje de programación C++, adaptado para establecer el proceso de compilación en los microcontroladores. Arduino es una herramienta de entrenamiento electrónico de hardware libre, diseñada con el firme propósito de facilitar el aprendizaje electrónico y su factor clave ha sido su representación de proyectos de código abierto, no sólo porque el software que es gratuito, sino porque las placas Arduino están a disposición de todo el mundo lo que permite gestionar e impulsar proyectos de esta naturaleza. El lenguaje de programación PHP es uno de los elementos técnicos orientado al desarrollo rápido de aplicaciones web [9], o en su defecto alojadas en la nube donde en muchos de los casos este tipo de aptitudes representa mejora en los procesos que se buscan optimizar gracias a las tecnologías de la información y comunicaciones [11]. Además, tiene posibilidades de conexión a bases de datos, es escalable y portátil [12]. Por otro lado, el almacenamiento de los datos será posible a través MySQL, por considerarse un administrador de base de datos [13]. PHP y MySQL, representan una accesible integración en el desarrollo de servicios en la nube, lo que da cabida a la construcción de aplicaciones web [14], dinámicas, administrables y eficientes [15] que satisfagan las necesidades de organizaciones,

empresas y sociedad con un enfoque transversal en su ejecución.

### 2.2. Diseño

En esta fase, se precisa los microcontroladores requeridos para la construcción del prototipo, así como el diseño para su ensamblaje. Adicionalmente se describen aspectos como accesibilidad, calidad y costo. Ejes fundamentales para el diseño de la mini-estación meteorológica. En la Tabla 1, se detalla el tipo de microcontrolador-sensor requerido y las cantidades que se necesitaran para su desarrollo.

TABLA I  
MICROCONTROLADORES EMPLEADOS

Nº	Microcontroladores	Cantidad
1	Arduino UNO	1
2	Shield Ethernet W5100	1
3	Barómetro - BMP085	1
4	Sensor de Humedad - Temperatura - DHT22	2

Es importante mencionar que de los cuatro componentes que se describen en la Tabla 1, Arduino UNO, es el eslabón principal. Este microcontrolador permite cumplir con la integración de los demás microcontroladores-sensores, sin él no sería posible el desarrollo del prototipo [16]. Mientras que la tarjeta electrónica Shield Ethernet W5100, le permite a un Arduino tener conectividad al protocolo TCP/IP, es decir tener la posibilidad que este tenga accesibilidad a un servicio en la nube [17]. El Barómetro BMP085, se encarga de medir la presión atmosférica, posee un muy bajo consumo de energía, además de proporcionar un excelente rendimiento [18]. Sus características, pueden llevar a cabo una estimación del clima. Finalmente, el sensor DHT22, es un microcontrolador que mide la temperatura y la humedad [19]. Su desempeño permite obtener datos eficientes y precisos [20]. Cada uno de los microcontroladores antes mencionados representan costos bajos [21] para su adquisición y han ganado campo en la industria por ser confiables. La Tabla 2, describe las características de Arduino UNO.

TABLA II  
CARACTERÍSTICAS DE ARDUINO UNO

Descripción	Propiedad
Voltaje de Operación	5V
Pines Digitales	14
Pines PWM	6
Pines de Entrada Análogas	6
Corriente DC por cada pin I/O	20 mA
Corriente DC en pin de 3.5V	50 mA
Memoria Flash	32 KB
Memoria SRAM	2 KB
Memoria EEPROM	1 KB
Velocidad de Reloj	16 MHZ

Conocer las características de cada uno de los microcontroladores permite conocer las bondades y las limitaciones que tiene cada uno de estos medios. La Tabla 3,

expresa los accesorios que adicionalmente se requieren para el desarrollo, operatividad y funcionamiento del prototipo.

TABLA III  
ACCESORIOS REQUERIDOS

N°	Accesorios	Cantidad
1	Cables de conexión para Arduino, 20 cm, macho-hembra	25
2	Soldadura 2 mt	1
3	Cable de Red + RJ45	1
4	Carcasa de prototipo	1

Después de haber identificado los recursos, características y accesibilidad, se diseñó y ensambló los microcontroladores mediante el software Fritzing, por ser una herramienta dinámica, libre y con un ambiente sencillo, fácil e inmediato que permite implementar de forma segura, concreta y automatizada el modelo del circuito [22], tal como se indica en la Fig. 1.

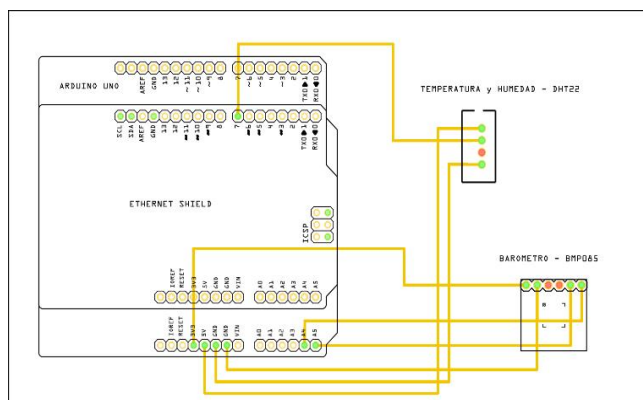


Fig. 1 Diagrama del circuito

### 2.3. Codificación

En apartado se dan a conocer las pautas para codificar los microcontroladores que se mencionaron en la Tabla 1, desde la incorporación de librerías hasta la descripción de instrucciones que permitirán su operatividad y funcionamiento. Para ello se recurre al Entorno de Desarrollo Integrado (IDE) Arduino [23], el cual brinda la posibilidad de programar microcontroladores e interactuar en tiempo real con entradas y salidas.

Para este caso la Fig. 2, representa 35 líneas de código, donde las primeras cinco líneas están relacionadas a la definición de librerías que integran al Arduino UNO, Shield Ethernet W5100 y los microcontroladores para la toma de datos de temperatura ambiental, humedad relativa y presión atmosférica. La línea 6 y 7, precisa la dirección MAC del hardware Shield Ethernet W5100. Es importante tener en cuenta que el chip controlador W5100, se configura con la misma librería que viene por defecto en el IDE de Arduino. Cuando el Shield Ethernet W5100 recibe o transmite datos se

visualiza un parpadeo de color verde y tiene una presencia de 100 megabits/s para su funcionamiento.

Después de haber incorporado las librerías de los microcontroladores, se procedió a parametrizar las entradas del sensor de temperatura, humedad y presión atmosférica con la finalidad de tener claro los pines que estarían reservados por cada uno de los sensores. Entre las líneas 9 y 20 de la Fig. 3, se puede observar la declaración de algunas variables e intervalos que forman parte de la configuración de los sensores antes mencionados.

Es imprescindible saber que para programar en Arduino se dispone de un *setup* y *loop*. En el *setup* se determinan todas las configuraciones básicas para la puesta en marcha del *loop*, que algunos de los casos también son conocido como bucle donde se ejecuta un número de veces dependiendo las instrucciones establecidas en el script.

En el *setup*, se ha definido inicializando el Módulo Shield Ethernet W5100 y los tres sensores del prototipo. Por otro lado, ha sido necesario definir variables para obtener los datos relacionados a la temperatura, humedad y presión atmosférica. Para la inicialización de los microcontroladores se proyectó 115200 milisegundos lo que corresponde a 2 minutos de espera para su carga y de esta manera reconozca cada una de sus propiedades que dispone cada sensor para su funcionamiento, así como lo describen las líneas 22-35 de la Fig 2.

```

1 #include <DHT.h>
2 #include <Ethernet.h>
3 #include <SPI.h>
4 #include <SFE_BMP180.h>
5 #include <Wire.h>
6 byte mac[] = { 0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x01 };
7 EthernetClient client;
8
9 #define DHTPIN 7
10 #define DHTTYPE DHT22
11 DHT dht(DHTPIN, DHTTYPE);
12
13 SFE_BMP180 bmp180;
14 double PresionNivelMar = 1013.25;
15 long previousMillis = 0;
16 unsigned long currentMillis = 0;
17 long interval = 250000;
18 float tx = 0;
19 float h = 0;
20 String data;
21
22 void setup() {
23   Serial.begin(115200);
24
25   if (Ethernet.begin(mac) == 0) {
26     Serial.println("Failed to configure Ethernet using DHCP");
27   }
28
29   dht.begin();
30   bmp180.begin();
31   delay(10000);
32   h = (float) dht.readHumidity();
33   tx = (float) dht.readTemperature();
34   data = "";
35 }

```

Fig. 2 Codificación del bloque N° 01

Una vez configurado el setup y encaminado la operatividad del microcontrolador Arduino UNO, Shield Ethernet W5100 y los sensores, pasamos al loop, aquel bucle que ejecuta de forma continua la captura de datos sobre los sensores que se han establecido. Entre las líneas 37 y 67 de la Fig.4, se puede ver la asignación de variables, donde *h* corresponde a humedad relativa, *tx* a temperatura ambiental y la variable *P*, asociada a la presión atmosférica.

Para conocer el estado de cada uno de los sensores, es decir, si están activos o inactivos, se recurre a las estructuras de control. De esta manera se asegura que los sensores estén activos y listos para realizar el envío de los datos que han sido capturados. Los datos son recibidos por un servicio en la nube el cual se encarga de almacenarlos en variables globales para después procesarlas [24] y registrarlas en la tabla correspondiente de la base de datos. Hasta llegar aquel proceso es necesario precisar que el éxito de esta tarea dependerá que tan bien ha sido definido los métodos de envíos de datos, como se conoce el lenguaje de programación PHP, dispone del método GET y POST, siendo los más comunes y empleados [25]. El método GET se encuentra relacionado a la recepción o recuperación de datos de forma visible a través de URLs, mientras que el método POST se utiliza para enviar datos de forma que no pueden ser vistos por el usuario y su propósito es almacenar o actualizar datos. Este tipo de métodos conllevan a salvaguardar la integridad de los datos dependiendo la tarea y el fin establecido con la finalidad de evitar inconsistencias durante el proceso.

```

37 void loop() {
38   char status;
39   double T, P, A;
40   currentMillis = millis();
41   if (currentMillis - previousMillis > interval) {
42     previousMillis = currentMillis;
43     h = (float) dht.readHumidity();
44     tx = (float) dht.readTemperature();
45   }
46
47   status = bmp180.startTemperature();
48   if (status != 0)
49   {
50     delay(status);
51     status = bmp180.getTemperature(T);
52     if (status != 0)
53     {
54       status = bmp180.startPressure(3);
55       if (status != 0)
56       {
57         delay(status);
58         status = bmp180.getPressure(P, T);
59         if (status != 0)
60         {
61           A = bmp180.altitude(P, PresionNivelMar);
62         }
63       }
64     }
65   }
66 }
67 }

```

Fig. 3 Codificación del bloque N° 02

Después de haber discernido el porqué de variables, validación de los sensores y definición métodos de envío como recepción de datos. La Fig. 4, a partir de la línea 70, da a conocer la creación de la variable *DATA*, cuya función es concatenar las variables *tx*, *h* y *P*, siendo estas las que almacenan el valor capturado por los sensores de temperatura ambiental, humedad relativa y presión atmosférica. Entre la línea 75 y 84, se observa la conexión del servicio en la nube que recibe los valores capturados haciendo empleo del método POST. El nexa para enviar los datos es la URL, que se encuentra en la línea 75, la misma que puede variar, en algunos casos puede ser una dirección IP en caso no tener un dominio propio.

```

70 data = "temp1=" + String(tx) +
71 "&hum1=" + String(h) +
72 "&presion=" + String(P) +
73 "&temp2=" + String(T);
74
75 if (client.connect("geomap.sirad.org.pe", 80)) {
76   Serial.print(data);
77   client.println("POST /index_ar.php HTTP/1.1");
78   client.println("Host: geomap.sirad.org.pe");
79   client.println("Content-Type: application/x-www-form-urlencoded");
80   client.print("Content-Length: ");
81   client.println(data.length());
82   client.println();
83   client.print(data);
84 }
85
86 if (client.connected()) {
87   client.stop();
88   Serial.println("Termino");
89 }
90
91 delay(60000);
92 }

```

Fig. 4 Codificación del bloque N° 03

La Fig. 4, ha demostrado que las técnicas de programación son dinámicas e integradoras sobre diversas plataformas y entornos de trabajo. Al haber comprendido cómo funciona el envío de datos de Arduino a un servicio en la nube, analizamos la Fig. 5, cuyo script esta referido a recibir los datos que son enviados a través del método POST, para su procesamiento y posterior almacenamiento de los datos. Las primeras doce líneas, se encuentran relacionadas en definir la zona horaria y la conexión de la base de datos a través de MySQL. Se precisa que las instrucciones *date\_default\_timezone\_set*, *mysqli* y *query*, cumplen un papel fundamental ya que es posible determinar el tiempo de envío de los datos que se caracteriza en saber la fecha y hora actual que los datos son recepcionados. Mientras que la instrucción *mysqli* desempeña el rol de establecer la conexión a través del paradigma orientado a objetos. Finalmente, entre las líneas 14 y 29, se crean las seis variables que cumplen la función de almacenar los datos que son enviados desde los microcontroladores además de precisar datos de fecha y hora para que posteriormente en la línea 29 a través de la instrucción *query*, se realice la inserción de los datos a la tabla de nombre *captura\_datos*.



```

1 k?php
2 date_default_timezone_set('America/Lima');
3 $host="localhost";
4 $user="adminweb_root";
5 $pass="*****";
6 $base="adminweb_geomap";
7 $con= new mysqli($host,$user,$pass,$base);
8
9 if ($con -> connect_errno) {
10     echo "Error en la Conexion:" . $mysqli -> connect_error;
11     exit();
12 }
13
14 $temperatura =$ _POST['temp1'];
15 $humedad     =$ _POST['hum1'];
16 $presion     =$ _POST['presion'];
17 $temp2       =$ _POST['temp2'];
18 $hora        =$ _POST['hora'];
19 $fecha       =$ _POST['fecha'];
20
21 $valores="('";
22     '$temperatura',
23     '$humedad',
24     '$presion',
25     '$temp2',
26     '$fecha',
27     '$hora')";
28
29 $insert=$con->query("INSERT INTO captura_datos VALUES $valores");
30 >>

```

Fig. 5 Codificación del bloque N° 04

Como se ha visto en el párrafo anterior, para el almacenamiento de los datos existe una tabla de nombre *captura\_datos*, cuya estructura consta de seis campos los que se visualizan en la Fig. 6.

```

1 CREATE TABLE `captura_datos` (
2   `idtemperatura` int(11) NOT NULL,
3   `temp` varchar(20) NOT NULL,
4   `humedad` varchar(20) NOT NULL,
5   `presion` varchar(12) NOT NULL,
6   `fecha` varchar(12) NOT NULL,
7   `hora` varchar(12) NOT NULL
8 ) ENGINE=MyISAM DEFAULT CHARSET=utf8;

```

Fig. 6 Estructura de la tabla: captura\_datos

## 2.4. Pruebas e implementación

En este último apartado de la sección metodológica se pretende dar a conocer el desempeño mediante la puesta en marcha del prototipo de mini-estación. Las pruebas iniciales se basaron en dejar funcionando y operando el prototipo durante doce horas continuas. Para ello el envío de datos al servicio en la nube se realizaba cada 60 segundos, logrando obtener 720 registros en la base de datos con datos completos de temperatura ambiental, humedad relativa y presión atmosférica, fecha y hora de envío.

En este primer acercamiento se resalta que no se requiere un amplio ancho de banda para que el prototipo tenga un eficiente desempeño. En cuanto a la integridad de los datos, fueron sólidos de acuerdo al tipo de dato que corresponde a cada campo de la tabla. Se precisaron algunas diferencias mínimas sobre los valores de temperatura ambiental y presión atmosférica. Mientras que los datos de humedad relativa se encontraban muy dinámicos esto debido a que el sensor tenía que calibrarse y precisar una altura considerable para obtener datos confiables sobre humedad relativa. Otros de los factores importantes es la ubicación, no es lo mismo que prototipo

tome datos al interior de un ambiente como al exterior, pues influye mucho en su comportamiento, por lo que se hace hincapié que este tipo de pruebas siempre deben ejecutarse en espacios abiertos y a una altura mínima de 3 metros.

La calibración del sensor de humedad relativa, estuvo sujeta al empleo de un termohigrómetro, instrumento que se encarga de comprobar las condiciones climáticas en áreas internas de un ambiente. Mientras que para áreas externas se recurrió a comparar los datos que emitía el prototipo y una estación meteorológica marca Davis por un lapso de 60 minutos. Teniendo en cuenta este tipo de aptitudes y mejoras para el prototipo de mini-estación meteorológica, se encuentra en condiciones de ser implementada y operada en cualquier zona geográfica que disponga del servicio de energía eléctrica e internet, requerimientos mínimos para su puesta en marcha. En la Fig. 7, se puede apreciar el prototipo ensamblado y recubierto con una caja de exterior cuyo material es de plástico e impermeable frente a la ocurrencia de precipitaciones pluviales o la fuerte radiación solar que pueda afectar la integridad, funcionamiento y operatividad del equipo. Es importante precisar que en este primer acercamiento los materiales empleados disponen una caracteriza accesible y pueden ser fortalecidos con otro tipo de recubrimiento que proteja los microcontroladores, evitando alterar la toma de los datos por parte de los sensores integrados.

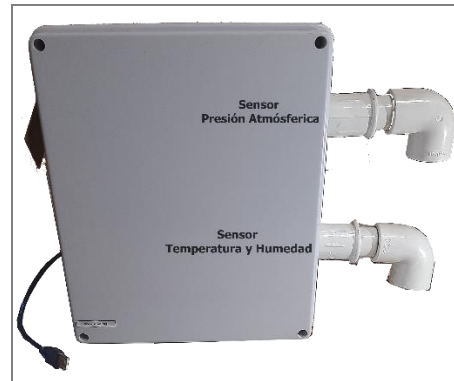


Fig. 8 Mini-estación Meteorológica

## III. RESULTADOS

Finalmente, la integración de los microcontroladores y los medios tecnológicos han permitido el desarrollo del prototipo de mini-estación meteorológica cuya operatividad le permite la captura de datos de temperatura ambiental, humedad relativa y presión atmosférica por un intervalo de 60 segundos de manera continua. Este intervalo puede ser alterado dependiendo la necesidad y el tipo de análisis que se pretenda realizar. Durante su implementación el prototipo ha sido instalado en diversas áreas geográficas de la región Piura, Perú, con la finalidad de asegurar su capacidad, comportamiento e integridad de los datos.

La Tabla 4, describe los resultados obtenidos durante un periodo de 5 días en los años 2017, 2018 y 2019. Su desempeño permite conocer promedios diarios y en espacios de tiempo sobre la temperatura ambiental, humedad relativa y presión atmosférica. En su primera aproximación el prototipo ha demostrado equilibrio, solides y un buen estado de funcionamiento.

**TABLA IV**  
REGISTRO DE MINI-ESTACIÓN METEOROLÓGICA – PROMEDIO DE TEMPERATURA, HUMEDAD Y PRESIÓN ATMOSFÉRICA DURANTE 5 DÍAS EN LOS AÑOS 2017, 2018 Y 2019.

Fecha	Horario	Promedio		
		Temperatura	Humedad	Presión Atmosférica
23/10/2017	10:13:00 12:33:00	28.42	14.39%	1009.45
15/01/2018	03:48:47 05:42:26	26.62	1%	1006.00
05/02/2018	03:40:14 05:21:30	27.14	1%	1003.27
23/10/2018	12:04:42 12:41:59	26.4	3%	1010.15
15/06/2019	01:34:06 01:42:06	31.39	1%	1007.44

Por otro lado, la Tabla 4, hace hincapié que un producto de esta naturaleza no solo debe validarse en espacios de tiempos cortos o temporales. Se debe tener en cuenta que durante el año existen diversas estaciones del clima, siendo estas las que aseveran y dan firmeza a la integridad de los datos a conocer el comportamiento de los sensores en un clima cálido o frío. Es por ello que en la Tabla 5, se precisa la unidad de medida y los límites físicos que cada uno de los sensores. Ambas características profundizan el modo de lectura de los datos, el campo de acción y la capacidad de soporte que puede tener el prototipo de acuerdo al escenario o espacio geográfico que se implemente.

**TABLA V**  
LÍMITES FÍSICOS DE LAS VARIABLES METEOROLÓGICAS

Variable	Unidad	Rango
Temperatura de aire	°C	-35/55
Humedad relativa del aire	%	0/100
Presión Atmosférica	hPa	700/1080

Con la finalidad de evaluar la integridad de los datos que se expresan en la Tabla 4. Los resultados fueron analizados y comparados con estaciones meteorológicas de uso especializado que por lo general las instituciones técnico científicas administran. Fue necesario que el prototipo de mini-estación meteorológica se instale en un radio de tres metros paralelo a la estación meteorológica automática de modelo Davis Vantage Pro 2 [26]. Esta actividad se realizó para generar una simetría de los resultados y conocer que tan dinámicos han sido los datos. Una de las cualidades importantes que se debe tener en cuenta es que cada estación meteorológica tiene configurado el intervalo de toma de datos, por lo que fue necesario parametrizar el intervalo en un rango

de 60 segundos para la toma de los datos. La Tabla 6, ilustra los resultados obtenidos por la estación meteorológica Davis Vantage Pro 2. El intervalo de tiempo es el mismo de la Tabla 4 y se presencia un acercamiento similar en cuanto a los datos de temperatura ambiental y presión atmosférica, mientras que para humedad relativa con algunas diferencias lejanas.

**TABLA VI**  
REGISTRO DE ESTACIÓN METEOROLÓGICA AUTOMÁTICA DE COMPARACIÓN - PROMEDIO DE TEMPERATURA, HUMEDAD Y PRESIÓN ATMOSFÉRICA DURANTE 5 DÍAS EN LOS AÑOS 2017, 2018 Y 2019

Fecha	Horario	Promedio		
		Temperatura	Humedad	Presión Atmosférica
23/10/2017	10:13:00 12:33:00	29.45	25%	1010
15/01/2018	03:48:47 05:42:26	28.65	13%	1007
05/02/2018	03:40:14 05:21:30	29.1	5%	1004.5
23/10/2018	12:04:42 12:41:59	27.5	5%	1011
15/06/2019	01:34:06 01:42:06	33.01	1%	1006.69

Recopilar datos meteorológicos de otros dispositivos, es una de las buenas maneras rápidas y accesibles que atribuye a conocer la capacidad y calidad de un producto que busca ser industrializado. Este tipo de intercambios, aporta no solo a validar datos o resultados, si no a conocer el dinamismo que puede tener cada uno de los sensores dependiendo la estación del año en la que se obtienen los datos meteorológicos.

Es importante señalar que este a tipo de equipos se le debe realizar un mantenimiento periódico, ya que por lo general se encuentran expuesto a escenarios que limiten su desempeño. Algunas de las eventualidades más frecuentes se centran en los cortes de energía, lluvia y vientos. Aspectos que de manera progresiva pueden alternar el comportamiento de este tipo de equipos por lo que se enfatiza, un constante mantenimiento y soporte del mismo. Finalmente, las Figuras, 8, 9 y 10 expresan el desenvolvimiento que tuvo cada uno de los sensores en comparación a la estación meteorológica Davis Vantage Pro 2.

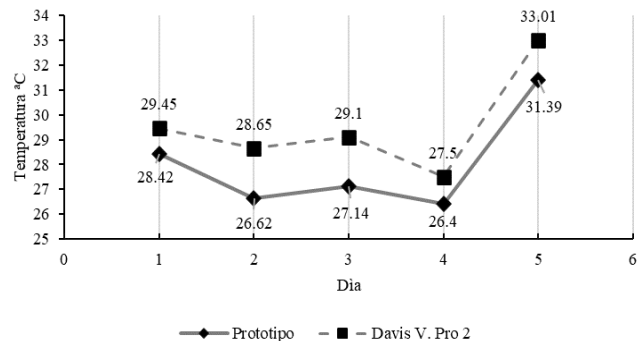


Fig. 8 Promedio de la temperatura entre la mini-estación meteorológica y la estación meteorológica Davis Vantage Pro 2.

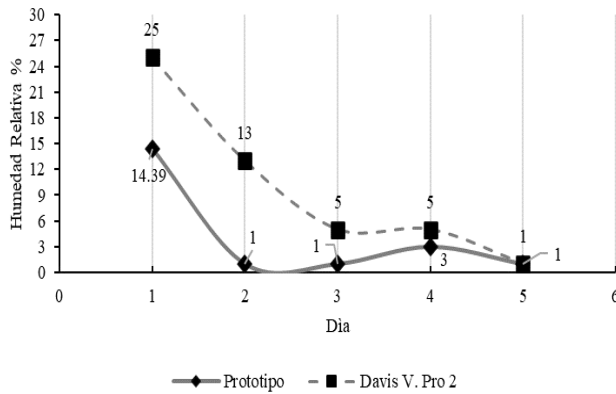


Fig. 9 Promedio de la humedad relativa entre la mini-estación meteorológica y la estación meteorológica Davis Vantage Pro 2.

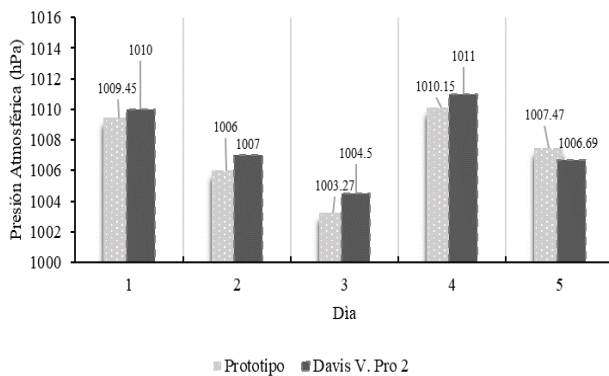


Fig. 10 Promedio de la presión atmosférica entre la mini-estación meteorológica y la estación meteorológica Davis Vantage Pro 2.

Las figuras antes mencionadas reflejan márgenes de error mínimos. Un 6.24% para los datos de temperatura ambiental, 1% presión atmosférica y 50% humedad relativa. Realizar la calibración y renovación constante de los sensores es una de las mejores experiencias y buenas prácticas para encontrar la precisión de los datos. Sin este tipo de tareas no sería posible disponer información de calidad.

Por otro lado, el área geográfica no es limitación para el desarrollo e implementación de este tipo de dispositivos. Mientras se tenga acceso a los servicios de energía eléctrica e internet, la operatividad y funcionamiento del prototipo estará asegurada, considerando que no requiere de una amplia inversión económica [27]. Dispositivos de esta naturaleza contribuyen a fortalecer la generación de data histórica [28] en áreas geográficas donde no se dispone de este tipo de instrumentos y que urge para conocer el estado del clima en sus tres componentes esenciales que son la temperatura ambiental, presión atmosférica y humedad relativa.

#### IV. FUTURAS INVESTIGACIONES

La propuesta ha demostrado tener una capacidad aceptable en operatividad y funcionamiento, sin embargo, esta situación no exime a desarrollar nuevos estudios que coadyuven a reducir aún brechas pendientes y abordar nuevas líneas de investigación multidisciplinarias. Entre las futuras investigaciones se considera, desarrollar un nuevo prototipo donde se integre un sensor FC-37 para la detección de lluvia [29] y un Anemómetro JL-FS2 cuya función es medir la velocidad del viento [30]. Estos dos instrumentos de medición contribuirán en ampliar el contenido de información. Datos que pueden ser empleados para transmitir contenido informativo a la población de un espacio geográfico como complemento a los medios disponibles. Otra de las investigaciones relevantes, se orienta en aplicar técnicas de inteligencia artificial, para desarrollar modelos predictivos en base a la información histórica que cada uno de los instrumentos de medición ha ido generando en una escala de tiempo. Este tipo de modelos permitirá disponer de pronósticos que orienten a la población y a las autoridades a tomar medidas prospectivas en beneficio de cada ciudad.

#### V. CONCLUSIONES

El prototipo ha brindado resultados favorables, tanto en la calidad de los datos, así como la estabilidad del dispositivo. Los sensores han llegado a tener una vigencia de uso en diversos intervalos de tiempo por más de cuatro años, manteniendo su eficacia y buen desempeño.

La interacción entre la tecnología Arduino y PHP, ha permitido obtener información en tiempo real y disponer de una precisión del 93.76% para la variable temperatura ambiental, 99% de asertividad para la presión atmosférica y 50% de exactitud para la humedad relativa. La implementación de este tipo de servicios en la nube coadyuva a reducir brechas digitales en los espacios lejanos de una ciudad por lo que se requiere impulsar este tipo de buenas prácticas para la industria que fortalezcan a las instituciones públicas, técnico científicas, academia, organizaciones y sociedad en general, logrando un mejor índice de conocimiento sobre la variabilidad climática y enriquecer la toma de decisiones ante eventos adversos que puedan dañar la integridad física de la población.

Es necesario profundizar en estudios de esta naturaleza que le permitan al ser humano disponer de nuevos medios tecnológicos para adoptar decisiones frente a los efectos del cambio climático y prever la integridad de las personas.

#### AGRADECIMIENTOS

Al Gobierno Regional Piura, Municipalidad Provincial de Piura y Universidad Nacional de Piura, por facilitar el acceso a sus estaciones meteorológicas con la finalidad de fortalecer y validar el desempeño del prototipo de mini-estación meteorológica.



## REFERENCIAS

- [1] L. Hernandez, Y. Calderon, H. Martinez, A. Pranolo, and I. Riyanto, "Design of a system for detection of environmental variables applied in data centers," *Proceeding - 2017 3rd Int. Conf. Sci. Inf. Technol. Theory Appl. IT Educ. Ind. Soc. Big Data Era, ICSITEch 2017*, vol. 2018-Janua, pp. 389–395, Jul. 2017, doi: 10.1109/ICSITECH.2017.8257144.
- [2] S. Rubiños *et al.*, "Design of a low-cost automatic weather station for capturing and improving the processing of variables obtained in real time," *Proc. LACCEI Int. Multi-conference Eng. Educ. Technol.*, vol. 2022-July, 2022, doi: 10.18687/LACCEI2022.1.1.433.
- [3] F. G. Russo, V. C. Fonteriz, D. C. Caputo, P. G. Cavataio, N. E. A. Ferré, and G. A. Receloglu, "Application of arduino microcontrollers and data processing using MatLab applied to the internal combustion engine test," *Proc. LACCEI Int. Multi-conference Eng. Educ. Technol.*, 2020, doi: 10.18687/LACCEI2020.1.1.434.
- [4] Ó. Torrente Artero, *Arduino Curso Práctico de formación*, Primera Ed. Mexico, 2013. Accessed: Jan. 17, 2023. [Online]. Available: [https://cienciasvirtuales.com/wp-content/uploads/2016/11/Arduino\\_Curso\\_Practico\\_de\\_Formacion.pdf](https://cienciasvirtuales.com/wp-content/uploads/2016/11/Arduino_Curso_Practico_de_Formacion.pdf)
- [5] A. Ramos Mejía and A. de J. Hernández-Martínez, "Un equipo de electroanálisis base Arduino y su uso en laboratorios de bajo costo," *Educ. Química*, vol. 33, no. 4, pp. 14–26, Oct. 2022, doi: 10.22201/FQ.18708404E.2022.4.0.82198.
- [6] A. Dickow and G. Feiertag, "A framework for calibration of barometric MEMS pressure sensors," *Procedia Eng.*, vol. 87, pp. 1350–1353, 2014, doi: 10.1016/J.PROENG.2014.11.716.
- [7] I. G. M. N. Desnanjaya, A. A. G. B. Ariana, I. M. A. Nugraha, I. K. A. G. Wiguna, and I. M. U. Sumaharja, "Room Monitoring Uses ESP-12E Based DHT22 and BH1750 Sensors," *J. Robot. Control*, vol. 3, no. 2, pp. 205–211, Mar. 2022, doi: 10.18196/JRC.V3I2.11023.
- [8] A. Faiña, "Learning Hands-On Electronics from Home: A Simulator for Fritzing," *Lect. Notes Networks Syst.*, vol. 530 LNNS, pp. 404–413, 2023, doi: 10.1007/978-3-031-15226-9\_38.
- [9] C. L. Vidal, L. L. López, J. A. Rojas, and M. M. Castro, "Desarrollo de sistema web de reclutamiento y selección y de directivos por competencias mediante PHP codeigniter 3.0," *Inf. Technol.*, vol. 28, no. 2, pp. 203–212, Apr. 2017, doi: 10.4067/S0718-07642017000200021.
- [10] P. Wadichar, A. Khandait, Y. Chavhan, and B. Bohra, "IoT Based Smart Energy Monitoring System," *Int. J. Innov. Eng. Sci.*, vol. 7, no. 5, p. 8, May 2022, doi: 10.46335/ijies.2022.7.5.2.
- [11] L. C. B. Osorio, S. E. C. Mostacero, O. R. A. Moreno, and J. F. P. Torres, "Aplicación Web de Apoyo Social para Mejorar el Estrés Laboral en una Empresa Comercial de Chapén," *Proc. LACCEI Int. Multi-conference Eng. Educ. Technol.*, vol. 2021-July, 2021, doi: 10.18687/LACCEI2021.1.1.313.
- [12] C. M. Zapata and J. J. Chaverra, "Un entorno para la generación automática de código bajo el patrón MVC a partir de esquemas preconceptuales," *DYNA*, vol. 79, no. 176, pp. 56–63, Dec. 2012.
- [13] M. E. de Oliveira, R. L. Z. de Oliveira, P. A. R. Trujillo, J. A. C. Trigoso, and A. R. B. Tech, "Development of an automated thermal monitoring and control system for a conventional aviary using IoT technology," *RISTI - Rev. Iber. Sist. e Tecnol. Inf.*, no. Special Issue 44, pp. 49–62, Aug. 2021.
- [14] K. Islam, F. Alam, A. I. Zahid, M. M. Khan, and M. Inamabbasi, "Internet of Things- (IoT-) Based Real-Time Vital Physiological Parameter Monitoring System for Remote Asthma Patients," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022, doi: 10.1155/2022/1191434.
- [15] L. Bravo and B. Marín, "Herramienta para la generación automática de software en PHP," *CIBSE 2016 - XIX Ibero-American Conf. Softw. Eng.*, pp. 458–467, Apr. 2016.
- [16] Y. B. Sundaresan and M. A. Saleem Durai, "An arduino based smart real time web application for server room monitoring and appliances control," *Int. J. Pharm. Technol.*, vol. 8, no. 3, pp. 18471–18481, Sep. 2016.
- [17] G. M. Tamilselvan, V. Ashishkumar, S. Jothi Prasath, and S. Mohammed Yusuff, "IoT based automated water distribution system with water theft control and water purchasing system," *Int. J. Recent Technol. Eng.*, vol. 7, no. 4, pp. 151–156, 2019.
- [18] C. Jun, M. Sanyi, Z. Tao, and Y. Shuqiang, "Design of digital pressure measuring system based on BMP085," *ACM Int. Conf. Proceeding Ser.*, Sep. 2018, doi: 10.1145/3284557.3284724.
- [19] E. K. Kpobi, B. A. K. Foli, K. A. Agyekum, and G. Wiafe, "Development of a Raspberry Pi-Based Remote Station Prototype for Coastal Environment Monitoring," *Remote Sens. Earth Syst. Sci.*, vol. 5, no. 1–2, pp. 14–25, Jun. 2022, doi: 10.1007/S41976-021-00053-2.
- [20] C. H. Lee, M. Ramasamy, S. Deivasigamani, and M. K. A. Ahamed Khan, "IoT Based Farming System," *Adv. Transdiscipl. Eng.*, vol. 30, pp. 1059–1069, Dec. 2022, doi: 10.3233/ATDE221132.
- [21] G. Solano, F. Lama, J. Terrazos, and J. Tarrillo, "Weather station for educational purposes based on Atmega8L," *Proc. 2017 IEEE 24th Int. Congr. Electron. Eng. Comput. INTERCON 2017*, Oct. 2017, doi: 10.1109/INTERCON.2017.8079728.
- [22] M. M. M. A. Kader *et al.*, "Automotive Mechanical Vehicle Starter," *J. Phys. Conf. Ser.*, vol. 2107, no. 1, Dec. 2021, doi: 10.1088/1742-6596/2107/1/012021.
- [23] T. Anbalagan *et al.*, "A novel examination of limonene detection using plastic fiber optic sensors and the tapered approach," *Bull. Electr. Eng. Informatics*, vol. 12, no. 2, pp. 807–814, Apr. 2023, doi: 10.11591/EEI.V12I2.4379.
- [24] J. C. Tipán-Alcívar, J. Párraga-álava, and J. Herrera-Tapia, "Monitoring of Air Pollution Levels with Long Range Transmission Technology," *RISTI - Rev. Iber. Sist. e Tecnol. Inf.*, vol. 2022, no. E47, pp. 157–170, Jan. 2022.
- [25] M. M. Amin, "Interoperabilitas perangkat lunak menggunakan restful web service," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 4, no. 1, pp. 14–22, Jan. 2018, doi: 10.26594/REGISTER.V4I1.1129.
- [26] E. A. Abioye *et al.*, "IoT-based monitoring and data-driven modelling of drip irrigation system for mustard leaf cultivation experiment," *Inf. Process. Agric.*, vol. 8, no. 2, pp. 270–283, Jun. 2021, doi: 10.1016/J.INPA.2020.05.004.
- [27] E. Serrano-Pérez, "Arduino-based low-cost electronic textiles to

introduce electric circuits and programming.” *Rev. Cuba. Fis.*, vol. 36, no. 2, pp. 110–113, Dec. 2019.

- [28] M. M. Miranda-Ramos, A. A. Ortiz, and L. A. Moreno, “Monitoring system using XBee and GSM technology for climate supervision in plantain production,” *Inf. Technol.*, vol. 31, no. 6, pp. 69–76, 2021, doi: 10.4067/S0718-07642020000600069.
- [29] V. Asanza, R. E. Pico, D. Torres, S. Santillan, and J. Cadena, “FPGA based meteorological monitoring station,” *2021 IEEE Sensors Appl. Symp. SAS 2021 - Proc.*, Aug. 2021, doi: 10.1109/SAS51076.2021.9530151.
- [30] M. N. Hidayat, F. Ronilaya, I. H. Eryk, and R. R. Kusuma, “Integration of the Weather Station Monitoring System in the Wind Power Plant Prototype,” *Proc. - IEIT 2022 2022 Int. Conf. Electr. Inf. Technol.*, pp. 181–185, 2022, doi: 10.1109/IEIT56384.2022.9967849.