

Aplicación web interactiva para monitoreo, análisis y control de procesos con Arduino y Raspberry.

Web application for monitoring, analysis and control of processes with Arduino and Raspberry.

Rubén Herrera Galicia (1).
Instituto Tecnológico de Tuxtla Gutiérrez.
h_galicia24@hotmail.com.

Raúl Moreno Rincón (2), I. T. de Tuxtla Gutiérrez, rmoreno@ittg.edu.mx

Manuel de Jesús Hernández Gutiérrez (3), egresado I.T. de Tuxtla Gutiérrez, memeghz92@outlook.com

Miriam Roxana Vázquez Flores (4), estudiante Instituto Politécnico Nacional, miriamroxana26@gmail.com

Tania Berenice Palacios Molina (5), estudiante Instituto Politécnico Nacional, pamtani_28@hotmail.com

Artículo recibido en junio 23, 2015; aceptado en octubre 14, 2015.

Resumen.

El trabajo aquí presentado está enfocado a describir los avances en el desarrollo de una aplicación web interactiva que permite monitorear, analizar y controlar, con placas de desarrollo Arduino, Raspberry y un dispositivo móvil, a un sistema automatizado. El problema se aborda de manera genérica para que la solución defina las bases para ser aplicada a distintos procesos de monitoreo, análisis, y control. Derivado de ello se describe un sistema genérico que para la demostración de su utilidad se aplicó al monitoreo del voltaje y la corriente, y el control de las cargas activadas en una casa residencial con fines de ahorro energético.

Palabras clave: Arduino, Control de Procesos, Aplicación web interactiva, Raspberry.

Abstract.

The paper presented here focuses on describing the advances in the development of an interactive web application for analyzing monitoring and controlling based on Arduino, Raspberry Boards and a mobile device. The problem is approached in general form for defining a basic solution to be applied to different monitoring analyzing and controlling processes. Resulting therefrom it was developed a general system that to demonstrate his utility was applied to monitoring voltage and current, and active load control for energy saving in homes.

Keywords: Arduino, Processes Control, Interactive web application, Raspberry.

1. Introducción.

Redes Ethernet.- Hay un alto potencial para las aplicaciones enfocadas a mejorar la operatividad y la sustentabilidad. Y existe un creciente interés por aplicaciones que integren sensores y actuadores con las técnicas de redes. Las aplicaciones básicas son de automatización y detección de fallas. Los trabajos innovadores son de; monitoreo, análisis de la calidad de la energía, regulación de la respuesta a la demanda eléctrica, integración de fuentes renovables, y ahorro de energía.

Mihaela Albu *et al.* (Albu et al., 2010) desarrollaron un sistema de monitoreo de una micro red de corriente directa para transferencia de energía, incluyendo mediciones y comunicación, realizando una aplicación segura y de control intuitivo que no requiere software instalado en la computadora del cliente, excepto por un explorador web. Se combinó el uso de bases de datos de código abierto y la versatilidad del código PHP. La micro red está comunicada con el controlador a través de Ethernet y el protocolo de comunicación es TCP/IP. El ingreso de comandos por un usuario remoto es de forma indirecta a través de una página web alojada en el servidor.

Ilche Georgievski *et al.* (Georgievski et al., 2012) implementaron un sistema de control y monitoreo del consumo de energía eléctrica basado en redes inteligentes para edificios. El sistema realiza una esquematización de la activación de dispositivos, con el objeto de que la menor cantidad de dispositivos sean activados al mismo tiempo. Además toma en cuenta el momento en el cual el costo de la energía eléctrica es menor para realizar dicha activación, traduciéndose esto en optimización de costos. El sistema ha sido puesto a prueba solamente mediante prototipos.

Kelly Sean Dieter Tebje *et al.* (Tebje et al., 2013) desarrollaron un sistema para monitorear y controlar el consumo de energía en una casa. Las operaciones principales son el manejo remoto y control de cargas, de una lámpara eléctrica o un calentador de agua. El sistema proporciona inteligencia ambiental para reducir el consumo de energía a través de la tecnología del Internet de las cosas. El servidor del sistema recopila la información enviada por los sensores y la almacena en una base de datos. Una vez que el usuario lo solicita, la información es procesada y mostrada vía Website. La información es presentada en periodos de días, semanas y meses y también puede ser observada gráficamente.

He Dawei *et al.* (Dawei et al., 2013) crearon un sistema de monitoreo no intrusivo, el cual es un sistema que divide diferentes tipos de dispositivos en dos categorías. El grupo uno está formado por dispositivos que demandan una alta carga para funcionar (bombas de agua, aire acondicionado), mientras que el grupo dos está formado por dispositivos que demandan una carga menor (TV, cafeteras, impresoras). Lo que hace este sistema es monitorear y controlar la activación y desactivación de los dispositivos dependiendo del grupo al que pertenecen, dando prioridad a los dispositivos del grupo dos para no ser desactivados. La prioridad se basa en que la desactivación de dispositivos del grupo dos tiene mayor probabilidad de causar inconformidad al usuario. Se han hecho investigaciones del uso del sistema, pero aún se tienen retos por superar.

Han Jinsoo *et al.* (Jinsoo et al., 2014) diseñaron un sistema inteligente para el manejo de energía en el hogar considerando el consumo de energía y la generación. Para esto usaron un medidor inteligente que mide el consumo de energía de los aparatos eléctricos de la casa. La información de la medición es enviada hacia un servidor web, para ser visualizada en la red. El protocolo de comunicación es zigbee. La generación de la energía que requieren los aparatos eléctricos se controla con un PLC. Para la generación de energía eléctrica, se usa la energía solar y la energía del viento.

Ricardo Torquato *et al.* (Torquato et al., 2014) propusieron una plataforma de simulación Monte Carlo para estudiar redes residenciales de baja tensión. El sistema se basa en dos componentes. El primero es un modelo de red de múltiples fases con flujo de potencia armónica. El segundo es un modelo que establece las características de operación de una carga a través del comportamiento de varias curvas. Lo que hace la plataforma es un monitoreo de distintos dispositivos durante 24 horas y posteriormente hace un análisis de las curvas de consumo energético usando la técnica de simulación Monte Carlo.

Michael Lorek *et al.* (Lorek et al., 2014) diseñaron un sensor para mediciones de potencia eléctrica en edificios. En este sistema una placa de desarrollo Raspberry Pi funciona como una estación base para recolectar la información de los dispositivos instalados en el panel. Las mediciones son cercanas a los valores ideales y el sistema mejora drásticamente las estimaciones reales de potencia.

Jianli Pan *et al.* (Pan et al., 2014) presentan un estudio de la investigación y el estado de la técnica en edificios y microrredes inteligentes. El estudio es una perspectiva acerca del Internet móvil y la eficiencia energética. Se analizan los métodos de tratamiento de edificios y microrredes aplicando tecnologías de red y de control que permiten reducir el uso innecesario de energía. Se hace énfasis en que la medición inteligente, en interacción con las redes

inteligentes, es útil para calcular la respuesta a la demanda. Tal interacción ofrece información de precios en tiempo real y ayuda a los administradores a ejecutar acciones para reducir el uso de energía durante las horas pico.

Suryadevara Nagender Kumar *et al.* (Kumar et al., 2015) desarrollaron un sistema inteligente de control y monitoreo en tiempo real para calcular el consumo de energía eléctrica en una casa. El sistema monitorea el voltaje y la corriente para calcular la energía consumida. El protocolo de comunicaciones es zigbee. La información recibida es almacenada en una base de datos. Se visualizan los valores de voltaje, corriente y energía en una página web. El objetivo es determinar los niveles de uso de electricidad en las horas pico para bajar el consumo de energía y mejorar su utilización.

Vivekananthan Cynthujah *et al.* (Cynthujah et al., 2015) propusieron un algoritmo para un administrador de energía en el hogar que en los cálculos usa precios en tiempo real. El objetivo del administrador es reducir el pago por concepto del consumo de energía. La primera fase del algoritmo es el monitoreo en tiempo real de los estados de los dispositivos controlables. En la segunda fase de programación se usan procesos de decisión de Markov para reducir el uso de electrodomésticos basado en el comportamiento estocástico del costo del consumo. La fase de control en tiempo real incorpora las incertidumbres de los precios y el perfil del consumo del dispositivo para ayudar a la optimización del costo del consumo eléctrico.

Objetivo general.- Diseñar e implementar una aplicación para una red inteligente comunicada vía Ethernet que permita monitorear, analizar y controlar, con placas de desarrollo Arduino, Raspberry y un dispositivo móvil, a un sistema automatizado. Para la demostración de la utilidad del sistema desarrollado se aplica este al monitoreo del voltaje y la corriente, y el control de las cargas activadas en una casa residencial con fines de ahorro energético.

2. Métodos.

Lo que en el presente trabajo se hace es combinar *Raspberry Pi* con *Arduino Ethernet* asignando el control del hardware a *Arduino* y utilizando a *Raspberry Pi* como un servidor para alojar a la aplicación web principal que incluye los enlaces a las páginas web de control alojadas en las placas *Arduino Ethernet*. La forma física de comunicación entre ambos dispositivos es a través de *Ethernet*. Para el diseño de las páginas web se usa HTML, CSS y PHP.

HTML define la forma y el contenido de la página web. CSS define la presentación y el estilo. PHP es un lenguaje de código abierto adecuado para el desarrollo de páginas web. Las paginas HTML con PHP incrustado ocupan menos código que las páginas HTML.

Con PHP no hay manera de que los usuarios conozcan el código ejecutado. PHP es *Javascript*, por tanto es ejecutado en el servidor generando HTML. El servidor web se configura para procesar los archivos HTML con PHP y el código generado es enviado al cliente. El cliente recibe el resultado de ejecutar el *script*, pero desconoce el código subyacente.

JavaScript.- Es el lenguaje de programación de los buscadores de páginas web, define los aspectos dinámicos y de comportamiento de una página web, altera el contenido de un documento, responde a las interacciones del usuario, y recibe y presenta información de manera remota. Para introducir aspectos dinámicos a una página web se usan las bibliotecas *Jquery*, *Ajax* y *Highcharts*.

JQuery.- Es una librería de componentes y funciones *JavaScript* desarrollada por John Resig. *JQuery* accede a los elementos de una página web, altera el contenido, modifica la apariencia, responde a interacciones del usuario, anima cambios, simplifica las tareas de *JavaScript*, y actualiza una página con valores de un servidor a través de intercambio asíncrono de información con XMLHttpRequest del conjunto de tecnologías *Ajax*.

Highcharts.- Es una biblioteca en lenguaje JavaScript. Esta biblioteca se usa para agregar gráficas interactivas a una página web. Las gráficas que soporta son de tipo líneal, áreas, y columnas. Highcharts modifica los ejes en tiempo

real y en combinación con *jQuery* y *Ajax* presenta gráficas en constante actualización con los valores del servidor, y los datos suministrados por el usuario. [http:// www.highcharts.com /products /highcharts](http://www.highcharts.com/products/highcharts).

Highcharts funciona sin clientes secundarios y requiere solo dos archivos: el núcleo *highcharts.js* y *jQuery* o *Ajax*. El ajuste de las opciones de configuración se hace a través de una estructura de notación de objetos JavaScript.

Manejo de bases de datos.- SQL se usa para el control de bases de datos, MySQL se usa para acceder a la base de datos desde un buscador, PHPMyAdmin se usa para la administración de bases de datos a través de Ethernet.

SQL.- Es el lenguaje estándar ANSI/ISO de control de bases de datos relacionales. SQL es un lenguaje capaz de acceder a cualquier sistema relacional comercial. Por ser un lenguaje declarativo, es suficiente indicar qué se quiere hacer. En cambio, en los lenguajes procedimentales es necesario especificar cómo se debe hacer la acción.

Las operaciones de SQL reciben el nombre de sentencias y están formadas por partes denominadas cláusulas. Trabajar con un lenguaje de programación anfitrión para dar soporte a SQL como si fuese un huésped se conoce con el nombre de SQL hospedado. Para esto se requiere un compilador que separe las sentencias del lenguaje de programación de las sentencias SQL.

MYSQL.- Es una herramienta básica para diseñar y programar bases de datos de tipo relacional. Cuenta con múltiples aplicaciones y aparece en la literatura como una de las más utilizadas por usuarios del medio. MySQL se usa como servidor al cual pueden conectarse múltiples usuarios y utilizarlo al mismo tiempo.

MySQL permite recurrir a bases de datos multiusuario a través de un buscador web y en diferentes lenguajes de programación que se adaptan a diferentes necesidades y requerimientos. MySQL es conocida por desarrollar alta velocidad en la búsqueda de datos e información.

PhpMyadmin.- Es una herramienta de software libre escrito en PHP que permite administrar MySQL a través de Internet. PhpMyAdmin es utilizado en operaciones de gestión de bases de datos. PhpMyAdmin se puede incorporar a través de una interfaz de usuario manteniendo la capacidad de ejecutar cualquier sentencia SQL. [http:// www.phpmyadmin.net/home_page/index.php](http://www.phpmyadmin.net/home_page/index.php).

Placas.- Raspberry Pi es una PC en miniatura. La PC ha sido desarrollada por la Fundación Raspberry Pi de Reino Unido con el objetivo de estimular la enseñanza de la informática básica en las escuelas. La tarjeta de desarrollo Raspberry Pi trabaja a una frecuencia de reloj 40 veces más rápida que Arduino uno. Además, Raspberry Pi tiene 128,000 veces más memoria RAM.

Raspberry Pi es una computadora que ejecuta un sistema operativo basado en Linux, soporta trabajo multitarea, incluye dos puertos USB y soporta conexión a Internet. Raspberry Pi es una computadora suficientemente potente, pero no lo suficiente para competir con una PC normal. Su principal desventaja es la lentitud en el manejo de imágenes. Pero es comprensible, pues carece de recursos significativos para el procesamiento de imágenes.

Arduino uno y Raspberry Pi son parecidas y es posible que se asuma que estas plataformas compiten por resolver problemas similares. Raspberry Pi es una computadora, que corre múltiples programas e incluso se programa con ella. Mientras que Arduino uno es un microcontrolador que corre solamente un programa, el que tenga cargado en su memoria.

Arduino uno es programado con aplicaciones basadas en C, pero no ejecuta un sistema operativo y no puede ser el sustituto de una computadora. Raspberry Pi es una plataforma apta para el desarrollo de aplicaciones domóticas y de control. Sin embargo, tiene un número limitado de entradas y salidas disponibles.

Raspberry Pi es superior a Arduino uno, pero solamente para cierto tipo de aplicaciones de software. La simplicidad de Arduino uno hace que éste sea una mejor opción para proyectos de hardware con tareas repetitivas de control. Arduino uno y Raspberry Pi son productos que se pueden combinar, y juntos pueden colaborar y ampliar el rango de posibilidades de una placa sola.

Metodología.- En la figura 1 se muestra un diagrama a bloques del sistema. Los sensores están conectados a una placa de desarrollo Arduino Ethernet, ver Fig. 2. El Arduino Ethernet número uno se encarga de adquirir las mediciones y enviarlas al Raspberry Pi vía Ethernet. La placa de Arduino Ethernet hace las mediciones, agrupa a los datos y los envía por paquetes al Raspberry Pi.

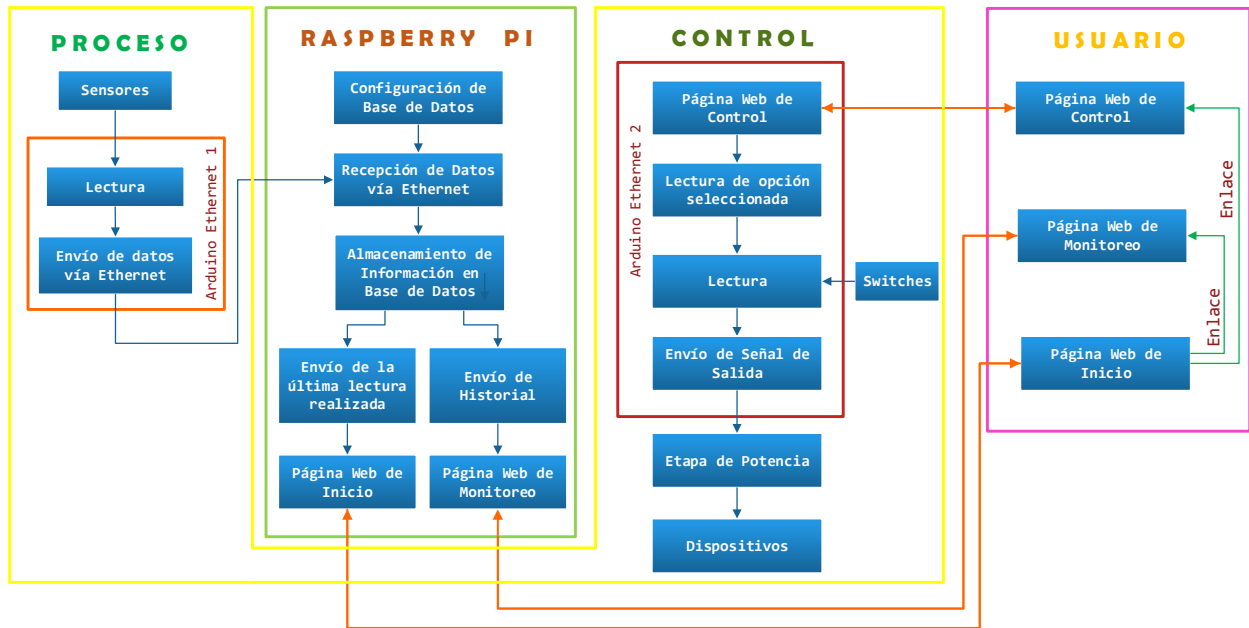


Figura 1. Diagrama a bloques; sistema de monitoreo y control.

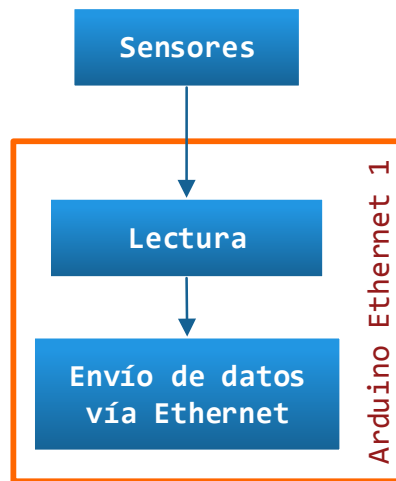


Figura 2. Diagrama a bloques; etapa de medición.

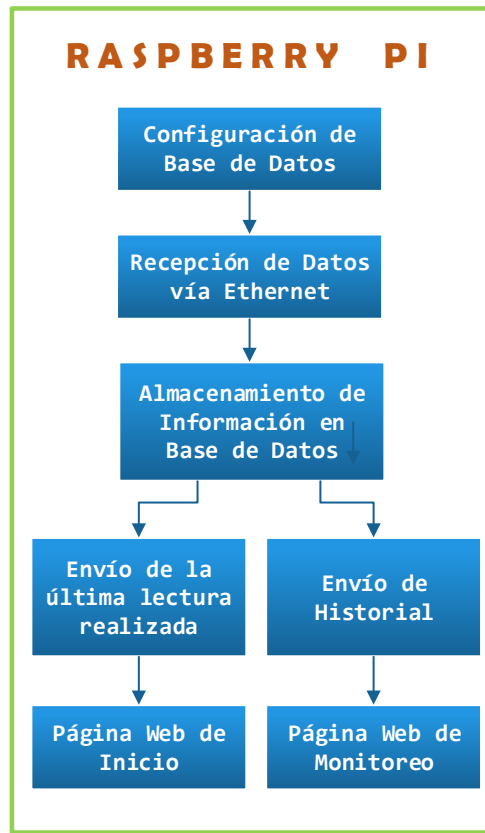


Figura 3. Diagrama a bloques; funciones del Raspberry Pi.

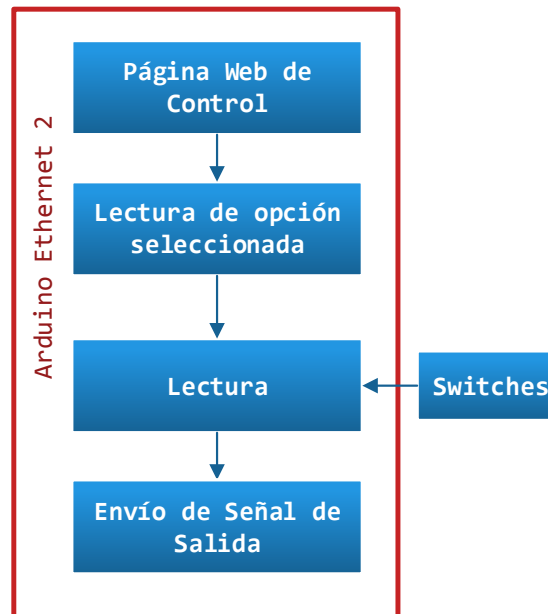


Figura 4. Diagrama a bloques; funciones del Arduino Ethernet dos.

El Raspberry Pi desempeña múltiples funciones, ver figuras 1 y 3; recibe los datos enviados por el Arduino Ethernet número uno correspondientes a las mediciones hechas con los sensores, desempeña la función de servidor web que muestra en la página web de inicio el valor de las últimas mediciones recibidas, mantiene actualizada a la base de datos, muestra el enlace para acceder a la página web de control, y aloja a la página web que permite al usuario observar el historial de los datos.

El Arduino Ethernet número dos se encarga de las tareas de control, ver figuras 1 y 4. Para ello aloja en su memoria una página web de control que permite al usuario seleccionar y transferir vía Ethernet los valores de referencia para el control. El usuario accede a la página web de control desde un enlace incluido en la página web de inicio, alojada en el Raspberry Pi. Además, el Arduino Ethernet dos se encarga de activar a los actuadores, cuando así se requiera.

El usuario intercambia información a través de las páginas web alojadas en el Raspberry Pi y en el Arduino Ethernet dos. Raspberry Pi actúa como cliente cuando recibe la última lectura de los sensores y actúa como servidor cuando muestra al usuario la última lectura de los sensores, o el historial de datos. El Arduino Ethernet dos actúa como cliente cuando recibe del usuario los valores de referencia y actúa como servidor cuando le muestra el estado de las variables de control.

3. Desarrollo.

Se implementaron los programas para las tareas presentadas en la figura 1. El código del programa cargado en Arduino Ethernet número uno se llama *Casa Inteligente Monitoreo*. El programa realiza la función de tomar las lecturas de los sensores y enviar dichos valores vía Ethernet mediante el método GET, ver código 1. El Raspberry Pi se encarga de recibir los datos de las mediciones con el programa *tabla.php*.

Código 1; Casa Inteligente Monitoreo. Envío de mediciones al servidor por el método GET.

```

Serial.println("connecting...");
if (client.connect(server,80)>0) {
    //Envío de los valores de los sensores al archivo tabla.php
    client.print("GET /Casa_Inteligente/tabla.php?temperatura="); client.print(tc);
    client.print("&rpm="); client.print(rpm);
    client.print("&voltaje="); client.print(supplyVoltage);
    client.print("&corriente="); client.print(Irms);
    client.println(" HTTP/1.0"); client.println("User-Agent: Arduino 1.0");
    client.println(); Serial.println("Conectado"); }
else Serial.print("fallo de conexion... ");
if (!client.connected()) Serial.println("desconectado!");
client.stop(); //Detener comunicacion
client.flush();

```

Tabla.php.- Este programa adquiere los valores de los sensores que son enviados por el Arduino Ethernet uno mediante el método GET y los inserta a la base de datos, llamada Tabla uno. Para acceder a la base de datos es necesario dar los permisos necesarios, importando el archivo de configuración *config.php* al programa *tabla.php*, ver código 2.

Config.php.- Este código se encarga de realizar la configuración de la base de datos, para lo cual es necesario declarar cuál es el usuario y contraseña de ingreso desde phpMyAdmin, así como el nombre de la base de datos, ver código 3.

Index.php.- Mediante este código se crea la página web de inicio y se incluye la programación para la adquisición de los valores medidos con los sensores. Así como la presentación de la fecha y la hora correspondientes. También en

este código se incluye la programación necesaria para la creación de los dos enlaces que permiten al usuario acceder a la página web de control y a la ventana de la presentación gráfica del historial, ver código 4.

Link.html.- Este código se encarga de implementar el enlace necesario para acceder a la página web de control, ver código 5. **Grafica.php.-** Se encarga de obtener las lecturas almacenadas en la base de datos y hacer una presentación gráfica del historial, ver código 6.

Casa_Inteligente_Control.- Este programa es el código cargado en Arduino Ethernet dos; se encarga de mostrar en un buscador la página web de control al recibir la petición HTTP, realiza la lectura de la opción seleccionada por el usuario, realiza la lectura de los interruptores activados, y finalmente realiza las comparaciones de los distintos casos para determinar la acción a realizar con los dispositivos, ver código 7.

Código 2; tabla.php. Leer los valores que llegan por GET.

```
<?php
    require("config.php");
    $temperatura = mysqli_real_escape_string($con,$_GET['temperatura']);
    $rpm = mysqli_real_escape_string($con,$_GET['rpm']);
    $voltaje = mysqli_real_escape_string($con,$_GET['voltaje']);
    $corriente = mysqli_real_escape_string($con,$_GET['corriente']);
    //Incertar los valores en Tabla1
    $query="INSERT INTO Tabla1(temperatura,RPM,voltaje,corriente)
    VALUES('".$temperatura."','".$rpm."','".$voltaje."','".$corriente."");
    mysqli_query($con,$query); //se ejecuta la instrucción
    mysqli_close($con);    ?>
```

Código 3; config.php. Configuración de la base de datos.

```
<?php
$dbhost = "localhost";
$dbuser = "root";
$dbpass = "hidrocentral";
$dbname = "Mecatronica";
$con = mysqli_connect($dbhost,$dbuser,$dbpass,$dbname);    ?>
```

Código 4; index.php. Aplicación web interactiva.

```
<HTML><head>
<META HTTP-EQUIV = "REFRESH" CONTENT="60;URL= ../Casa_Inteligente/index.php">
<!--[if lt IE 9]>
    <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
<![endif]-->
    <link rel="stylesheet" href="monitoreo/estilos.css">
    <link rel="stylesheet" href="monitoreo/jquery.ui.all.css">
    <script src="monitoreo/jquery-1.7.2.js"></script>
    <script src="monitoreo/jquery-ui.min.js"></script>
    <script src="monitoreo/jquery.ui.touch-punch.min.js"></script>
</head><body><title>Sistema de Monitoreo de una Casa Inteligente</title>
<div class="container"><?php
function conectarBD(){
    $server = "localhost";
    $usuario = "root";
```



```

$pass = "hidrocentral";
$BDB = "Mecatronica";
//variable que guarda la conexión de la base de datos
$conexion = mysqli_connect($server, $usuario, $pass, $BDB);
//Comprobación que la conexión ha tenido éxito
if(!$conexion){
    echo 'Ha sucedido un error inesperado en la conexión de la base de datos<br>'; }
//devolvemos el objeto de conexión para usarlo en las consultas
return $conexion; }
/*Desconectar la conexión a la base de datos*/
function desconectarBD($conexion){
//Cierra la conexión y guarda el estado de la operación en una variable
$close = mysqli_close($conexion);
//Comprobación que se ha cerrado la conexión correctamente
if(!$close){
    echo 'Ha sucedido un error inesperado en la desconexión de la base de datos<br>'; }
//devuelve el estado del cierre de conexión
return $close; }
//Devuelve un array multidimensional con el resultado de la consulta
function getArraySQL($sql){
//Creamos la conexión
$conexion = conectarBD();
//generamos la consulta
if(!$result = mysqli_query($conexion, $sql)) die();
$rawdata = array();
//guarda en un array multidimensional todos los datos de la consulta
$i=0;
while($row = mysqli_fetch_array($result)) {
//guarda en rawdata todos los vectores/filas que devuelve la consulta
    $rawdata[$i] = $row;
    $i++; } //Cerramos la base de datos
desconectarBD($conexion); //devolvemos rawdata
return $rawdata; } //Sentencia SQL
$sql = "SELECT temperatura,RPM,voltaje,corriente,tiempo from Tabla1 ORDER BY id DESC Limit 1;";
//Array Multidimensional
$rawdata = getArraySQL($sql); ?>
<div class="demo">
<h1 id="container">
<h1 align="right">  BIENVENIDO <align="center"></h1>
<p align="center" class="ui-state-default ui-corner-all ui-helper-clearfix" style="padding:10px;">
    <b><font size=5> SISTEMA DE MONITOREO DE UNA CASA INTELIGENTE</font></b>
</p><body bgcolor="#BDBDBD" text="#0101DF">
<table align="center" cellspacing="10" cellpadding="10" >
<tr><br><br><br><table align="left" cellspacing="20" cellpadding="10"><tr>
<td align="center"><b><font size=5>Temperatura Actual:</font></b> <h3><font color=orange> <?php for($i = 0
;$i<count($rawdata);$i++){ ?> <?php echo $rawdata[$i]["temperatura"];?> <?php } ?> </font></h3> </td>
<td align="center"><b><font size=5> Revoluciones por minuto:</font></b> <h3><font color=orange> <?php for($i = 0
;$i<count($rawdata);$i++){ ?> <?php echo $rawdata[$i]["RPM"];?> <?php } ?> </font></h3></td>
</tr><IMG SRC="Casa.png" width=380 height=280 align="left">
<tr>
<td align="center"><b><font size=5> Voltaje (V):</font></b> <h3><font color=orange><?php for($i = 0 ;$i<count($rawdata);$i++){ ?>
<?php echo $rawdata[$i]["voltaje"];?> <?php } ?> </font></h3></td>
<td align="center"><b><font size=5> Corriente(A):</font></b> <h3><font color=orange><?php for($i = 0 ;$i<count($rawdata);$i++){
?> <?php echo $rawdata[$i]["corriente"];?> <?php } ?> </font></h3></td>
</tr></table>
<font size=4 color='blue' > <form method="get" name="prf"
action="http://192.168.0.105"
target="_blank"
<input type="hidde" name="proxy"
value="random" /><br><br><br><br><br><br> <tr>
<div align="center"><b><input type="submit" style="width:8cm" value="Modificar Número de Dispositivos
Disponibles"</b></div></form>
<br><br><br><br><br><br><br>
<h3 align="right"><font size=4 color=green> Hora de lectura: <?php for($i = 0 ;$i<count($rawdata);$i++){ ?> <?php echo
$rawdata[$i]["tiempo"];?> <?php } ?> </h3></div>
<a href="..Casa_Inteligente/grafica.php" onclick="window.open(this.href);return false;">Ver Historial</a>

```

```
</html>
```

Código 5; link.php. Enlace a la página web de control.

```
<HTML><A href="Led_On_Off.php" >
<BR><P><FONT SIZE=3.5 FACE=ARIAL BLACK COLOR=RED><B>
MODIFICAR OPCIONES DE CONTROL<BR></B></FONT></A></HTML>
```

Código 6; grafica.php. Presentación grafica del historial.

```
<?php
function conectarBD(){
    $server = "localhost";
    $usuario = "root";
    $pass = "hidrocentral";
    $BD = "Mecatronica";
        //variable que guarda la conexión de la base de datos
    $conexion = mysqli_connect($server, $usuario, $pass, $BD);
    //Comprobación que la conexión ha tenido éxito
    if(!$conexion){
        echo 'Ha sucedido un error inesperado en la conexión de la base de datos<br>'; }
        //devolvemos el objeto de conexión para usarlo en las consultas
    return $conexion; }
    /*Desconectar la conexión a la base de datos*/
function desconectarBD($conexion){
    //Cierra la conexión y guarda el estado de la operación en una variable
    $close = mysqli_close($conexion);
    //Comprobación que se ha cerrado la conexión correctamente
    if(!$close){
        echo 'Ha sucedido un error inesperado en la desconexión de la base de datos<br>'; }
        //devuelve el estado del cierre de conexión
    return $close; }
    //Devuelve un array multidimensional con el resultado de la consulta
function getArraySQL($sql){
    //Implementar la conexión
    $conexion = conectarBD(); //generar la consulta
    if(!$result = mysqli_query($conexion, $sql)) die();
    $rawdata = array();
        //guardar en un array multidimensional los datos de la consulta
    $i=0;
    while($row = mysqli_fetch_array($result)) {
        //guarda en rawdata los vectores/filas que devuelve la consulta
        $rawdata[$i] = $row;
        $i++; } //Cierra la base de datos
    desconectarBD($conexion); //devuelve rawdata
    return $rawdata; } //Sentencia SQL
$sql = "SELECT temperatura,RPM,corriente,tiempo from Tabla1.";
    //Array Multidimensional
    $rawdata = getArraySQL($sql); //Adaptar el tiempo
for($i=0;$i<count($rawdata);$i++){
    $time = $rawdata[$i]["tiempo"];
    $date = new DateTime($time);
    $rawdata[$i]["tiempo"]=$date->getTimestamp()*1000; } ?>
<HTML><head><meta charset="utf-8">
<title>Grafica de Monitoreo</title>
<!-- Latest compiled and minified JavaScript -->
<script src="https://code.jquery.com/jquery.js"></script>
<!-- Importo el archivo Javascript de Highcharts directamente desde su servidor -->
<script src="http://code.highcharts.com/stock/highstock.js"></script>
<script src="http://code.highcharts.com/modules/exporting.js"></script>
```

```

<script type='text/javascript'>
$(function () {
  $(document).ready(function() {
    Highcharts.setOptions({
      global: {
        useUTC: false
      }
    });
    $('#container').highcharts('StockChart', {
      chart: {
        type: 'spline',
        animation: Highcharts.svg, // don't animate in old IE
        marginRight: 10,
        events: {
        },
        scrollbar: {
          barBackgroundColor: 'white',
          barBorderRadius: 7,
          barBorderWidth: 0,
          buttonBackgroundColor: 'gray',
          buttonBorderWidth: 0,
          buttonArrowColor: 'white',
          buttonBorderRadius: 7,
          rifleColor: 'gray',
          trackBackgroundColor: 'gray',
          trackBorderWidth: 1,
          trackArrowColor: 'black',
          trackBorderRadius: 7
        },
        navigator: {
          maskInside: false,
        },
        handles: {
          backgroundColor: 'gray',
          borderColor: 'white'
        },
        plotOptions: {
          series: {
            allowPointSelect: true
          }
        },
        rangeSelector: {
          selected: 1
        },
        title: {
          text: 'Grafica de Monitoreo'
        },
        xAxis: {
          type: 'datetime',
          tickPixelInterval: 150
        },
        legend: {
          enabled: true
        },
        exporting: {
          enabled: true
        },
        series: [{
          name: 'Temperatura (oC)',
          data: (function() {
            var data = [];
            <?php for($i = 0;$i<count($rawdata);$i++){
              data.push([<?php echo $rawdata[$i]["tiempo"];?>,<?php echo $rawdata[$i]["temperatura"];?>]);
            }
            <?php } ?> return data;
          })()
        },{
          name: 'Velocidad (RPM)',
          data: (function() {
            var data = [];
            <?php for($i = 0;$i<count($rawdata);$i++){
              data.push([<?php echo $rawdata[$i]["tiempo"];?>,<?php echo $rawdata[$i]["RPM"];?>]);
            }
            <?php } ?> return data;
          })()
        },{
          name: 'Voltaje (V)',
          data: (function() {
            var data = [];
            <?php for($i = 0;$i<count($rawdata);$i++){
              data.push([<?php echo $rawdata[$i]["tiempo"];?>,<?php echo $rawdata[$i]["voltaje"];?>]);
            }
            <?php } ?> return data;
          })()
        },{
          name: 'Corriente (A)',
          data: (function() {
            var data = [];
            <?php for($i = 0;$i<count($rawdata);$i++){
              data.push([<?php echo $rawdata[$i]["tiempo"];?>,<?php echo $rawdata[$i]["corriente"];?>]);
            }
            <?php } ?> return data;
          })()
        }
      ]
    });
  });
}
</script>
</head>
<BODY><BR><BR><BR><BR><BR><div id="container"></div></html>

```

Código 7; Casa_Inteligente_Control. Página web de control en Arduino Ethernet dos.

```

#include <SPI.h>
#include <Ethernet.h>
#include "EmonLib.h"

```

```

byte mac[]={0xDE,0xAD,0xBE,0xEF,0xFF,0xEE};
IPAddress ip(192,168,0,105);
EthernetServer servidor(80);
int out1=3; int out2=5; int out3=6; int out4=7;
String readString=String(30);
String switches,sw1,sw2,sw3,sw4,buffer;
int B; //Bandera
void setup() { Serial.begin(9600);
  Ethernet.begin(mac, ip); servidor.begin();
  pinMode(out1,OUTPUT); pinMode(out2,OUTPUT);
  pinMode(out3,OUTPUT); pinMode(out4,OUTPUT); }
void loop() { //Lectura de los estados de los interruptores
while(Serial.available()) {
  if(Serial.available()) {
    char d=Serial.read();
    switches += d;
    if (d==' ') {
      Serial.println(switches);
      if(switches.length() //Separación de la cadena obtenida
      { sw1=switches.substring(2,3); //Switch 1
        sw2=switches.substring(3,4); //Switch 2
        sw3=switches.substring(4,5); //Switch 3
        sw4=switches.substring(5,6); //Bomba }
      switches=""; } } }
//Conversion de String a Entero
char buf1[sw1.length()];
sw1.toCharArray(buf1,sw1.length()+1);
int sw1I=atoi(buf1);
char buf2[sw2.length()];
sw2.toCharArray(buf2,sw2.length()+1);
int sw2I=atoi(buf2);
char buf3[sw3.length()];
sw3.toCharArray(buf3,sw3.length()+1);
int sw3I=atoi(buf3);
char buf4[sw4.length()];
sw4.toCharArray(buf4,sw4.length()+1);
int sw4I=atoi(buf4);
//Declaración del cliente
EthernetClient cliente= servidor.available();
if(cliente) {
  boolean lineaenblanco=true;
//Cliente conectado
//Leer petición HTTP carácter a carácter
//Almacenar los caracteres en la variable readString
while(cliente.connected()) { //Verifica que el cliente se encuentre conectado
  if(cliente.available()) { //Verifica que el cliente se encuentre disponible
    char c=cliente.read();
    if(readString.length()<30) {
      readString.concat(c); //Cliente conectado
//Lectura de la petición HTTP carácter a carácter
} //Almacena los caracteres en la variable readString
if(c=='\n' && lineaenblanco) { //Si la petición HTTP ha finalizado
  int LED = readString.indexOf("LED=");
//Cabecera HTTP estándar; Creación página web
  cliente.println("HTTP/1.1 200 OK");
  cliente.println("Content-Type: text/html");
  cliente.println(); //Página Web en HTML
  cliente.println("<html>"); cliente.println("<head>");

```

```

cliente.println("<title> Dispositivos Disponibles </title>");
cliente.println("</head>");
cliente.println("<body bgcolor=#F5F5F5>");
cliente.println("<body width=100% height=100%>");
cliente.println("<center>");
cliente.println("<h1><center><P ><font size='+4' color='blue'>
    Dispositivos Disponibles </font></p></center></h1>");
cliente.println("<hr align='center' width='70%' size='5' noshade>");
cliente.println("<br>");
cliente.println("<form method=get>");
cliente.println("<b><center><font size='+3'>0 de 3 </font></p></center></b>");
cliente.println("<input type=radio name=LED value=0>");
cliente.println("<br><br><br><b><center><font size='+3'>1 de 3 </font></p></center></b>");
cliente.println("<input type=radio name=LED value=1>");
cliente.println("<br><br><br><b><center><font size='+3'>2 de 3 </font></p></center></b>");
cliente.println("<input type=radio name=LED value=2 >");
cliente.println("<br><br><br><b><center><font size='+3'>3 de 3 </font></p></center></b>");
cliente.println("<input type=radio name=LED value=3 >");
cliente.println("<br><br>");
cliente.println("<input type='submit' value='ENVIAR'></form>");
//Muestra el mensaje de acuerdo a la opción seleccionada y
//activa la bandera correspondiente al caso
if (readString.substring(LED,LED+5)=="LED=0") {
cliente.println("<font size='+4' color='red'>0 de 3</font>");    B=0;    }
else if (readString.substring(LED,LED+5)=="LED=1") {
cliente.println("<font size='+4' color='red'>1 de 3</font>");    B=1;    }
else if (readString.substring(LED,LED+5)=="LED=2") {
cliente.println("<font size='+4' color='red'>2 de 3</font>");    B=2;    }
else if (readString.substring(LED,LED+5)=="LED=3") {
cliente.println("<font size='+4' color='red'>3 de 3</font>");    B=3;    }
cliente.println("</body>");
cliente.println("</html>"); //Cierra la conexión con el cliente
cliente.stop();    readString=""; } } }
//Activación o desactivación de la bomba
//No existe condición para su funcionamiento
if(sw4I==1) digitalWrite(out4,HIGH);
else digitalWrite(out4,LOW);
//Dispositivos (3 Switches)
if(B==0){ //Bandera0 activa, 0 de 3, ningún dispositivo puede ser encendido
digitalWrite(out1,LOW);
digitalWrite(out2,LOW);
digitalWrite(out3,LOW);    }
else if(B==1){ //Bandera1 activa, 1 de 3 dispositivos puede ser encendido
if(sw1I==1) { //Switch1 activo
digitalWrite(out1,HIGH);    digitalWrite(out2,LOW);
digitalWrite(out3,LOW);    }
else if(sw2I==1){ //Switch2 activo
digitalWrite(out1,LOW);    digitalWrite(out2,HIGH);
digitalWrite(out3,LOW);    }
else if(sw3I==1) { //Switch3 activo
digitalWrite(out1,LOW);    digitalWrite(out2,LOW);
digitalWrite(out3,HIGH);    }
else { //Ningun switch activo
digitalWrite(out1,LOW);    digitalWrite(out2,LOW);
digitalWrite(out3,LOW);    } } }
else if(B==2){ //Bandera2 activa, 2 de 3 dispositivos pueden ser encendidos
if(sw1I==1){ //Switch1 activo, Prioridad 1
digitalWrite(out1,HIGH);

```

```

if(sw2I==1){ //Si el switch2 está activo
digitalWrite(out2,HIGH);      digitalWrite(out3,LOW);      }
else if (sw3I==1) { //Switch3 activo
digitalWrite(out2,LOW);      digitalWrite(out3,HIGH);      }
else { //Ni el switch 2 ni el 3 se encuentra activo
digitalWrite(out2,LOW);      digitalWrite(out3,LOW); } }
else if(sw2I==1) { //Switch 2 activo, Prioridad 2
digitalWrite(out1,LOW);      digitalWrite(out2,HIGH);
if(sw3I==1) digitalWrite(out3,HIGH); }
else digitalWrite(out3,LOW);
}
else if(sw3I==1) { //Switch 3 activo, Prioridad 3
digitalWrite(out1,LOW);
digitalWrite(out2,LOW);      digitalWrite(out3,HIGH); }
else { digitalWrite(out1,LOW);
digitalWrite(out2,LOW);      digitalWrite(out3,LOW); } }
else if(B==3) { //Bandera3 activa, 3 de 3, Cualquiera puede ser encendido
if(sw1I==1) digitalWrite(out1,HIGH);
else digitalWrite(out1,LOW);
if(sw2I==1) digitalWrite(out2,HIGH);
else digitalWrite(out2,LOW);
if(sw3I==1) digitalWrite(out3,HIGH);
else digitalWrite(out3,LOW); } }

```

Resultados.- En las figuras 5 y 6 se muestra la página web de inicio. La página muestra los últimos valores leídos de los sensores, la hora de la lectura, y dos enlaces. Un enlace para ir a la página de control y otro para ir a ver el historial de lecturas. En la figura 7 se muestran los valores almacenados en la Tabla uno de la base de datos. Las mediciones son actualizadas cada setenta milisegundos. En las figuras 8 y 9 se observa el historial de los datos leídos por los sensores. El registro de la Tabla uno del servidor Raspberry Pi se muestra en la página de monitoreo de forma gráfica usando *Highcharts*. El historial grafico se presenta haciendo uso del programa *grafica.php*.

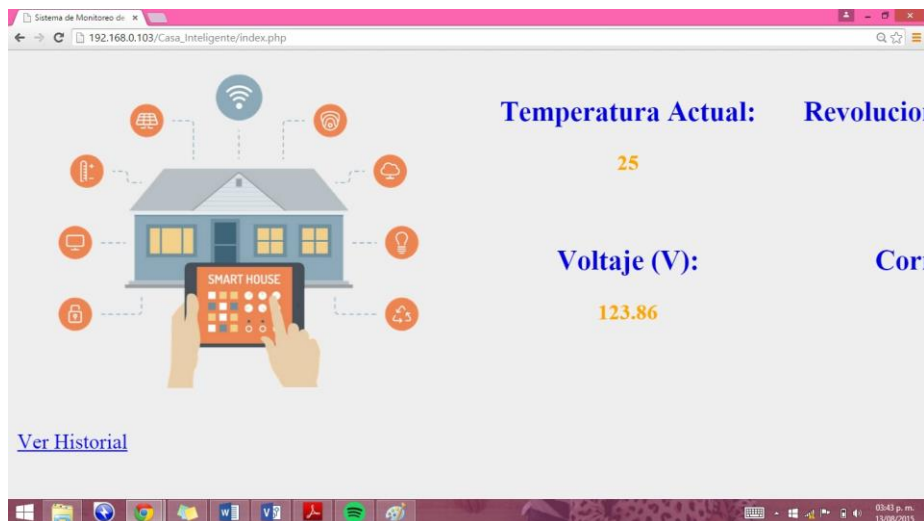


Figura 5. Página web de inicio; sistema de monitoreo y control.

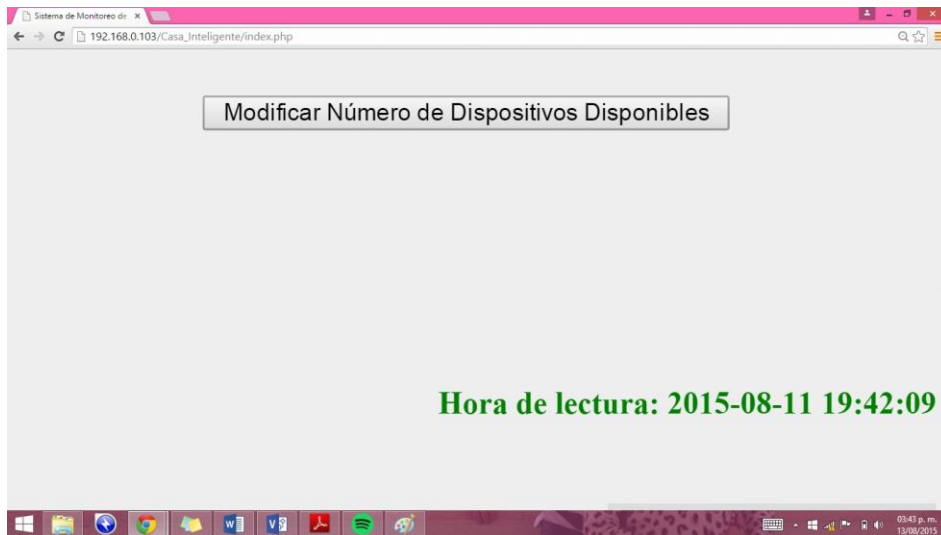


Figura 6. Página web de inicio; sistema de monitoreo y control.

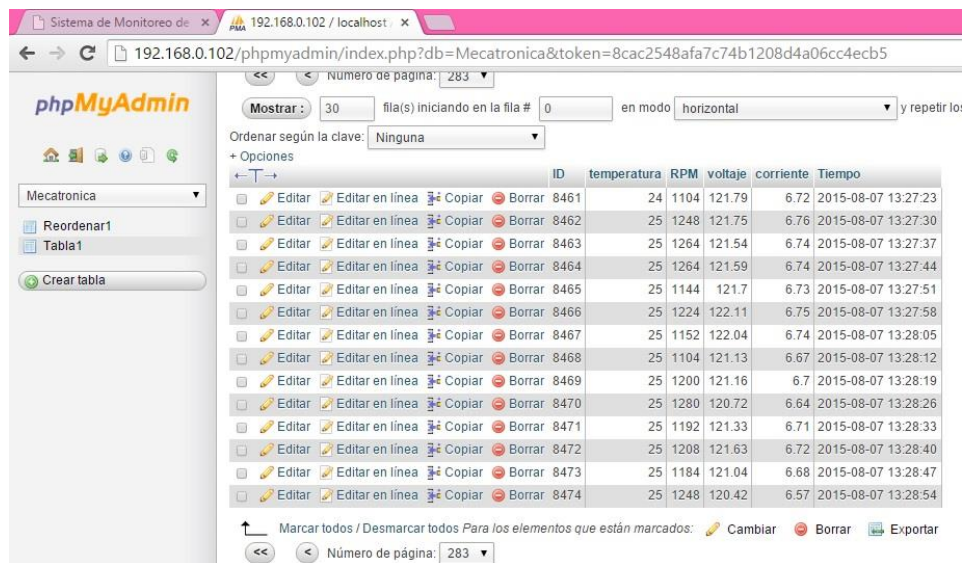


Figura 7. Gestión de la Tabla 1 con phpMyAdmin.

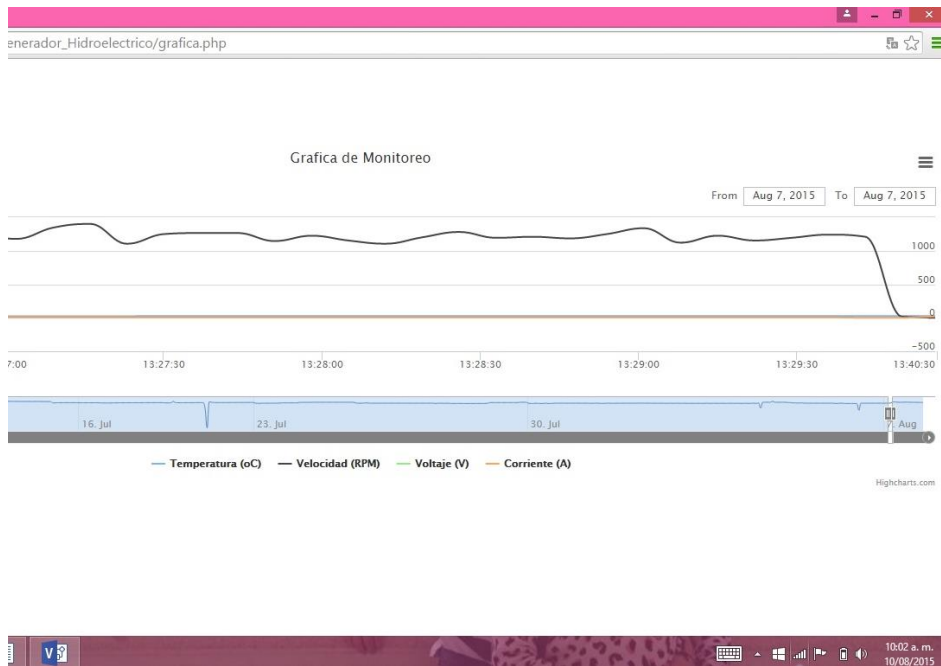


Figura 8. Gestión del historial de lecturas con el programa grafica.php.

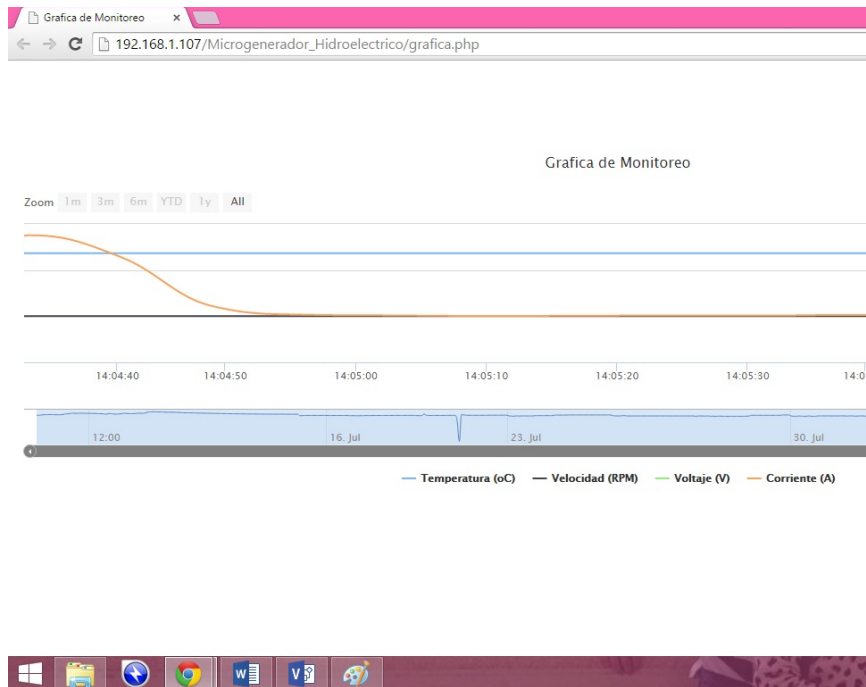


Figura 9. Gestión del historial de lecturas con el programa grafica.php.

En la figura 10 se muestra la página web de control. El código de programación se carga en Arduino Ethernet dos. En la página se puede seleccionar el número máximo de dispositivos que pueden ser activados al mismo tiempo. También en la página se puede seleccionar el modo de operación; manual o remoto. En modo manual los dispositivos se activan y desactivan de manera local. En modo remoto los dispositivos se activan y desactivan de manera remota presionando los botones incorporados a la página Web.

Durante el desarrollo del proyecto se solucionaron problemas derivados del uso de Linux, aplicaciones de software y lenguajes de programación. Se desarrolló una aplicación con *PhpMyadmin*, para la configuración del servidor en Raspberry Pi. Se subieron los datos del Arduino Ethernet número uno al servidor. Esta tarea se hizo con los lenguajes PHP y SQL. Se mandan cuatro datos de los sensores a una misma tabla.

Se diseñó un algoritmo para mandar los datos usando cadenas de caracteres. En Arduino Ethernet uno se forma la cadena de caracteres y se transmite. En Raspberry Pi se recibe la cadena de caracteres y se descodifica. El comportamiento y el historial de la base de datos se muestran usando la herramienta de software libre *Highcharts*. Con programación en PHP se adaptó la aplicación *Highcharts* para garantizar el acceso a los datos contenidos en la base de datos y mostrar el historial en forma de línea.

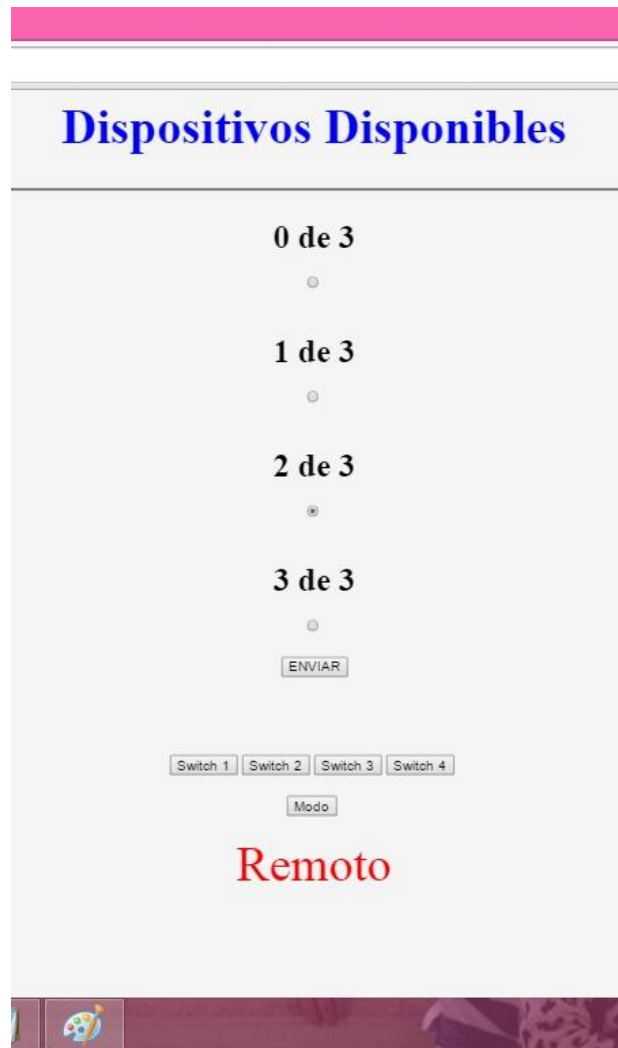


Figura 10. Vista del buscador mostrando la página Web de control.

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

Conclusiones.

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

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

Créditos.

Se reconoce a Miriam Roxana Vázquez Flores, Tania Berenice Palacios Molina, y Manuel de Jesús Hernández Gutiérrez el compromiso, desempeño y dedicación mostrados durante el desarrollo del presente proyecto.

Agradecimientos.

Se agradece a la División de Estudios de Posgrado e Investigación del Instituto Tecnológico de Tuxtla Gutiérrez todo el apoyo brindado en el desarrollo del presente proyecto a través del sistema de investigación interna de ese instituto.

Referencias Bibliográficas.

- Albu, M., Kyriakides, E., Chicco, G., Popa, M., Nechifor, A. (2010).** Online Monitoring of the Power Transfer in a DC Test Grid. *IEEE Transactions on Instrumentation and Measurement*, vol. 59, n° 5, May, 1104-1118.
- Cynthujah, V., Yateendra, M., Fangxing, L. (2015).** Real-Time Price Based Home Energy Management Scheduler. *IEEE Transactions on Power Systems*, vol. 30, n° 4, July, 2149-2159.
- Dawei, H., Weixuan, L., Nan, L., Ronald, G., Harley, T., Habetler, G. (2013).** Incorporating Non-Intrusive Load Monitoring Into Building Level Demand Response. *IEEE Transactions on Smart Grid*, vol. 4, n° 4, December, 1870-1877.
- Georgievski, I., Viktoriya, D., Andrea, G., Nguyen, T., Lazovik, A., Aiello, M. (2012).** Optimizing Energy Costs for Offices Connected to the Smart Grid. *IEEE Transactions on Smart Grid*, vol. 3, n° 4, December, 2273-2285.
- Jinsoo, H., Chang-Sic, C., Wan-Ki, P., Ilwoo, L., Sang-Ha, K. (2014).** Smart Home Energy Management System Including Renewable Energy Based on ZigBee and PLC. *IEEE Transactions on Consumer Electronics*, vol. 60, n° 2, May 2014, pp. 198-202.

Kumar, S. N., Chandra, M. S., Sean, D., Tebje, K., Pal, S. G. S. (2015). WSN-Based Smart Sensors and Actuator for Power Management in Intelligent Buildings. *IEEE/ASME Transactions on Mechatronics*, vol. 20, n° 2, April, 564-571.

Lorek, M., Chraim, F., Pister, K. S. J., Lanzisera, S. (2014). COTS-Based Stick-On Electricity Meters for Building Submetering. *IEEE Sensors Journal*, vol. 14, n° 10, October, 3482-3489.

Pan, J., Jain, R., Subharthi, P. (2104). A Survey of Energy Efficiency in Buildings and Microgrids using Networking Technologies. *IEEE Communications Surveys & Tutorials*, vol. 16, n° 3, 1709-1731.

Tebje, K. S. D., Suryadevara, N. K., Mukhopadhyay, S. C. (2013). Towards the Implementation of IoT for Environmental Condition Monitoring in Homes. *IEEE Sensors Journal*, vol. 13, n° 10, October, 3846-3853.

Torquato, R., Shi, Q., Xu, W., Freitas, W. (2014). A Monte Carlo Simulation Platform for Studying Low Voltage Residential Networks. *IEEE Transactions on Smart Grid*, vol. 5, n° 6, November, 2766-2776.

Información de los autores



Dr. Rubén Herrera Galicia obtuvo el título de Doctor en Ciencias Técnicas con especialidad en electrónica por la *Technical University of Warsaw*, Polonia. Es profesor de tiempo completo adscrito al departamento de Ingeniería Eléctrica y Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez y Jefe de Proyectos de Investigación de Ingeniería Eléctrica. Es investigador desde 1985 y líder de la línea de investigación Control de Procesos del programa de Ingeniería Electrónica.



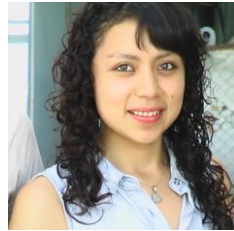
Raúl Moreno Rincón es Maestro en Ciencias en Ingeniería Electrónica egresado del Instituto Tecnológico de Toluca, Maestro en Educación Superior por la Universidad Autónoma de Chiapas e Ingeniero en Comunicaciones y Electrónica por la ESIME-IPN, en la ciudad de México, D.F. Es profesor de tiempo completo en el área de Ingeniería Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez y Jefe de Proyectos de Docencia de Ingeniería Electrónica. Es investigador desde 1999 y Colabora en la línea de investigación de “Robótica” de Ingeniería electrónica, en donde ha realizado proyectos como; sistema de alarma para personas con deficiencia auditiva basado en XBEE, robot de cafetería, araña hexápoda.



Manuel de Jesús Hernández Gutiérrez es egresado de la carrera de Ingeniería Electrónica del Instituto Tecnológico de Tuxtla Gutiérrez. en el año 2014, participante en el evento nacional de innovación tecnológica del Tecnológico Nacional de México llegando hasta la etapa regional con el proyecto “Control de procesos basado RaspBerry y Arduino vía Ethernet”.



Miriam Roxana Vázquez Flores es alumna del noveno semestre de la carrera de Ingeniería Mecatrónica en la Unidad Profesional Interdisciplinaria de Ingeniería y Tecnologías Avanzadas del Instituto Politécnico Nacional, promedio actual de 9.84, estudió el octavo semestre en Polonia.



Tania Berenice Palacios Molina es alumna del noveno semestre de la carrera de Ingeniería Mecatrónica en la Unidad Profesional Interdisciplinaria de Ingeniería y Tecnologías Avanzadas del Instituto Politécnico Nacional, promedio actual de 9.73, estudió el octavo semestre en Polonia.