"IMPLEMENTACIÓN DE UN SERVIDOR L.A.M.P. PARA VISUALIZACIÓN DE DATOS - RASPBERRY S.O."

Trabajo Preparatorio N°12
Laboratorio de Sistemas Embebidos

Melanny Cecibel Dávila Pazmiño
Ingeniería en Telecomunicaciones
Facultad de Eléctrica y Electrónica
Quito, Ecuador
melanny.davila@epn.edu.ec

***Abstract*—En el presente documento se presenta el sustento teórico para comprender el funcionamiento de Sense Hat al momento de usar Raspberry, con el fin de poder desarrollar aplicaciones amigables con el usuario.**

***Index Terms*—Raspberry, PHP, Mariadb, Apache, L.A.M.P.**

I. Introducción

LAMP es un acónimo de "Linux, Apache, MySQL y PHP", es decir, las cuatro tecnologías que conforman esta plataforma que corre desde el lado del servidor.

Gracias a LAMP se puede crear sitios web, aplicaciones, realizar testing de páginas dinámicas y estáticas, entre muchas otras cosas más. En inglés se lo conoce como LAMP stack, es decir, una ampliación de servicios y tecnologías que nos permiten una gracias a la otra, conformar la plataforma que necesitamos.

II. Objetivos

- Relacionar al estudiante con el uso de las aplicaciones de un servidor L.A.M.P.
- Realizar un conjunto de configuraciones e instalaciones de servicios, que permitan desarrollar una página web simple, para mostrar al usuario datos ambientales [2].

III. Cuestionario

A. Consultar los elementos que conforman un servidor L.A.M.P. Describir detalladamente cada uno de ellos.

El funcionamiento de un servidor L.A.M.P. es muy simple. Linux sirve como sistema operativo base para ejecutar el servidor web Apache. Este último no puede interpretar contenidos dinámicos, pero es aquí donde PHP entra a ejercer sus funciones de programación del lado del servidor. El proceso funciona entonces de la siguiente manera: Apache le envía un código fuente al intérprete PHP, incluyendo la información correspondiente sobre las acciones del visitante de la web, y permite el acceso a la base de datos MySQL. El resultado es devuelto a Apache y este se muestra finalmente en el navegador web del visitante [1].

Un servidor LAMP es la opción preferida por muchos por sus bajo coste y su alta disponibilidad. Además, sus componentes individuales pueden ser reemplazados fácilmente por aquellos con las mismas funciones. Como sistema operativo se puede usar, por ejemplo, Windows (WAMP) o MacOS (MAMP). En vez de Apache, es común utilizar nginx como servidor web y en cuanto a gestor de bases de datos, MySQL y MariaDB son muy similares. Otros lenguajes de programación compatibles son Perl, Ruby o Python [1].

- **Linux:** Linux es un núcleo de sistema operativo libre tipo Unix. Es el sistema operativo base de la plataforma, y puede ser cualquier tipo de sistema Gnu/Linux, donde podrán ejecutarse el resto de los componentes.
  Las aplicaciones para GNU/Linux se distribuyen en una variedad de formatos debido a la diversidad de métodos de manejo de paquetes de software, algunos más preparados para ser ejecutados que otros. Mientras que en sistemas como Windows o MacOS el usuario normalmente busca el software de terceros por su cuenta; las distribuciones GNU/Linux fueron pioneras en los repositorios de aplicaciones soportadas oficialmente por el sistema operativo, similar a las tiendas de aplicaciones modernas, donde el usuario acude a buscar el software que desea instalar [1].
- **Apache:** El servidor HTTP Apache es un servidor web libre y de código abierto, el más popular en cuanto a uso. Permite alojar y despachar las páginas web dinámicas y estáticas que son ejecutadas desde el lenguaje PHP.
  Ventajas:
  - Modular
  - Código abierto
  - Multi-plataforma
  - Extensible
  - Popular (fácil conseguir ayuda/soporte)

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web [3].
Este servidor web es redistribuido como parte de varios paquetes propietarios de software, incluyendo la base de datos Oracle y el IBM WebSphere application server. MacOS integra apache como parte de su propio servidor web y como soporte de su servidor de aplicaciones WebObjects. Es soportado de alguna manera por Borland en las herramientas de desarrollo Kylix y Delphi. Apache es incluido con Novell NetWare 6.5, donde es el servidor web por defecto, y en muchas distribuciones Linux.

- **MySQL/MariaDB:** MySQL es un Sistema de Gestión de Bases de Datos (SGBD) relacional, que por lo tanto utiliza SQL, multihilo y multiusuario del que se estiman más de un millón de instalaciones. De la misma manera, MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL (General Public License) [3].
  MySQL es muy utilizado en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python),
  y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.
  MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante monitorizar de antemano el rendimiento para detectar y corregir errores tanto de SQL como de programación. Diferencia entre MySQL Y MariaDB:En la práctica, MariaDB reemplaza directamente a la misma versión de MySQL (MySQL 5.1- MariaDB 5.1, MariaDB 5.2 y MariaDB 5.3 son compatibles. MySQL 5.5-MariaDB 5.5) [4].
  - **Mecanismos de almacenamiento:** Además de los mecanismos de almacenamiento estándar MyISAM, Blackhole, CSV, Memory y Archive, también se incluyen en la versión fuente y binaria de MariaDB los siguientes:
    * Aria (alternativa a MyISAM resistente a caídas)
    * XtraDB (reemplazo directo de InnoDB)
    * PBXT (en MariaDB 5.1, 5.2 y 5.3. Deshabilitada en 5.5)
    * FederatedX (reemplazo directo de Federated)
    * OQGRAPH — nuevo en 5.2
    * SphinxSE — nuevo en 5.2
    * IBMDB2I. Eliminada por Oracle de MySQL 5.1.55 pero se incluye en el código de MariaDB hasta la versión 5.5.
    * Cassandra, en MariaDB 10.0 (otros mecanismos no-sql se incluirán en MariaDB)
    * Sequence, aparecido con MariaDB 10.0.3
  - **Facilidad de uso:**
    * Proporciona estadísticas de índices y tabla, para lo que añade nuevas tablas en INFORMATION_SCHEMA y nuevas opciones a los comandos FLUSH y SHOW para identificar la causa en la carga del SGBD.
    * Los comandos ALTER TABLE y LOAD DATA INFILE dejan de ser opacos e informan del progreso.
    * La precisión para tipo de datos TIME, DATETIME, y TIMESTAMP ampliada al microsegundo.
    * Introducidas características estilo NoSQL, como HandlerSocket que proporciona acceso directo a tablas InnoDB saltándose la capa SQL.
    * Columnas dinámicas, que proporcionan al usuario columnas virtuales en las tablas.
    * Las subqueries funcionan correctamente.
  - **Prestaciones:**
    * El optimizador de MariaDB -que se encuentra en el núcleo de cualquier SGBD- funciona claramente más rápido con cargas complejas.
    * En la replicación se han introducido sustanciosas mejoras, por ejemplo el “group commit for the binary log” que acelera la replicación hasta el doble.
    * Eliminación de tablas. El acceso a tablas a través de views acelera el acceso.
  - **Testeo:**
    * Más juegos de test en la distribución.
    * Parches para los tests.
    * Distintas combinaciones de configuración y sistema operativo para los tests.
    * Eliminación de tests innecesarios, como ”no testar la característica X si no la he incluido en mi ejecutable”.
  - **Menos errores y alertas:**
    * Los juegos de testeo han permitido reducir los errores sin introducir nuevos.
    * Las alertas de compilación están relacionadas, y los desarrolladores las han intentado reducir.
- **PHP:** Sus sigla en inglés significan: “Hypertext Preprocessor” es un lenguaje de programación diseñado para producir sitios web dinámicos. PHP es utilizado en aplicaciones del lado del servidor, aunque puede ser usado también desde una interfaz de línea de comandos o como aplicación de escritorio. Es al mismo tiempo el servidor PHP que permite la ejecución de este tipo de scripts.

El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado —que puede ser cualquier tipo de datos, como el HTML generado o datos de imágenes binarias— formaría la totalidad o parte de una respuesta HTTP. Existen diversos sistemas de plantillas, sistemas de gestión de contenidos y frameworks que pueden emplearse para organizar o facilitar la generación de esa respuesta [4].

B. Consultar las líneas de comando necesarias para configurar un servidor L.A.M.P. en la máquina virtual de Raspberry S.O., y describir su funcionamiento. (Utilizar la versión de Raspbian Buster 10)

A continuación se presentan las líneas de código que permiten implementar el servidor L.A.M.P. La primera parte consiste en instalar el servidor de Apache.

```
sudo apt-get install apache2
sudo chown -R pi:www-data /var/www/html/
sudo chmod -R 770 /var/www/html/
```

Script 1. Implementación del servidor Apache2

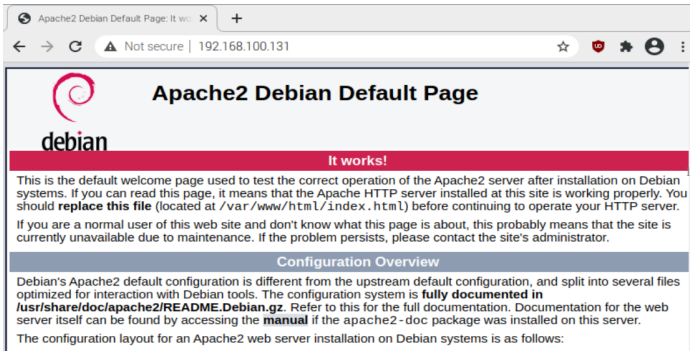

Fig. 1. Instalación del servidor Apache2

Posteriormente, se instaló MariaDB, que es el servidor de la base de datos.

```
sudo apt install mariadb-server-10.3 mariadb-client
    -10.3
sudo mysql --user=root

sudo nano /etc/mysql/my.cnf
service msqls restart
```

Script 2. Implementación del servidor de base de datos

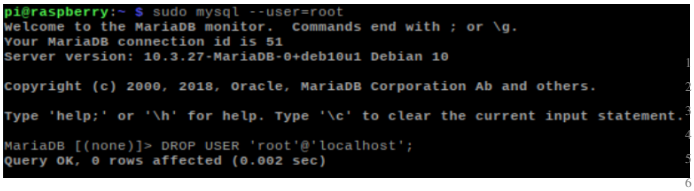

Fig. 2. Implementación de los comandos

De esta manera se realizó su implementación.

```
sudo apt-get install php
```

Script 3. Implementación del servidor de base de datos

Debido a que los repositorios para descargar e instalar phpmyadmin no se encontraban dentro del directorio con las direcciones de descarga, se procedió a descargar e instalar de manera manual tal como se indica en: [6]

Como primer paso se tiene que descargar el archivo con la aplicación de phpmyadmin:

```
wget https://files.phpmyadmin.net/phpMyAdmin
    /4.9.0.1/phpMyAdmin-4.9.0.1-all-languages.zip
```

Script 4. Descarga del archivo

Luego, descomprimir el archivo y mover lo que se obtenga al directorio `/usr/share/` creando a la vez un directorio que contenga a la aplicación.

```
unzip phpMyAdmin-4.9.0.1-all-languages.zip
sudo mv phpMyAdmin-4.9.0.1-all-languages /usr/share/
    phpmyadmin
```

Script 5. Creación del directorio

Luego, se debe hacer a un usuario propietario del servidor web:

```
sudo chown -R www-data:www-data /usr/share/
    phpmyadmin
```

Script 6. Creación del usuario propietario

Posteriormente se instalan los módulos de PHP requeridos y recomendados:

```
sudo apt install php-imagick php-phpseclib php-php-
    gettext php7.3-common php7.3-mysql php7.3-gd
    php7.3-imap php7.3-json php7.3-curl php7.3-zip
    php7.3-xml php7.3-mbstring php7.3-bz2 php7.3-
    intl php7.3-gmp
```

Script 7. Instalación de los módulos

Para que los cambios se apliquen se debe reiniciar el servidor apache2:

```
sudo systemctl restart apache2
```

Script 8. Reinicio del servidor

Para configurar phpmyadmin como tal se debe crear un archivo de configuración, para lo cual primero se abre un archivo con el nombre `phpmyadmin.conf` de la siguiente manera:

```
sudo nano /etc/apache2/conf-available/phpmyadmin.
    conf
```

Script 9. Apertura del archivo

Se debe colocar el siguiente texto dentro de ese archivo de configuración:

```
# phpMyAdmin default Apache configuration

Alias /phpmyadmin /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
    Options SymLinksIfOwnerMatch
    DirectoryIndex index.php

    <IfModule mod_php5.c>
        <IfModule mod_mime.c>
```

```
            AddType application/x-httpd-php .php
        </IfModule>
        <FilesMatch ".+\.php$">
            SetHandler application/x-httpd-php
        </FilesMatch>

        php_value include_path .
        php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
        php_admin_value open_basedir /usr/share/phpmyadmin/:/etc/phpmyadmin/:/var/lib/phpmyadmin/:/usr/share/php/php-gettext/:/usr/share/php/php-php-gettext/:/usr/share/javascript/:/usr/share/php/tcpdf/:/usr/share/doc/phpmyadmin/:/usr/share/php/phpseclib/
        php_admin_value mbstring.func_overload 0
    </IfModule>
    <IfModule mod_php.c>
        <IfModule mod_mime.c>
            AddType application/x-httpd-php .php
        </IfModule>
        <FilesMatch ".+\.php$">
            SetHandler application/x-httpd-php
        </FilesMatch>

        php_value include_path .
        php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
        php_admin_value open_basedir /usr/share/phpmyadmin/:/etc/phpmyadmin/:/var/lib/phpmyadmin/:/usr/share/php/php-gettext/:/usr/share/php/php-php-gettext/:/usr/share/javascript/:/usr/share/php/tcpdf/:/usr/share/doc/phpmyadmin/:/usr/share/php/phpseclib/
        php_admin_value mbstring.func_overload 0
    </IfModule>

</Directory>

# Disallow web access to directories that don't need it
<Directory /usr/share/phpmyadmin/templates>
    Require all denied
</Directory>
<Directory /usr/share/phpmyadmin/libraries>
    Require all denied
</Directory>
<Directory /usr/share/phpmyadmin/setup/lib>
    Require all denied
</Directory>
```

Script 10. Modificación del contenido del archivo

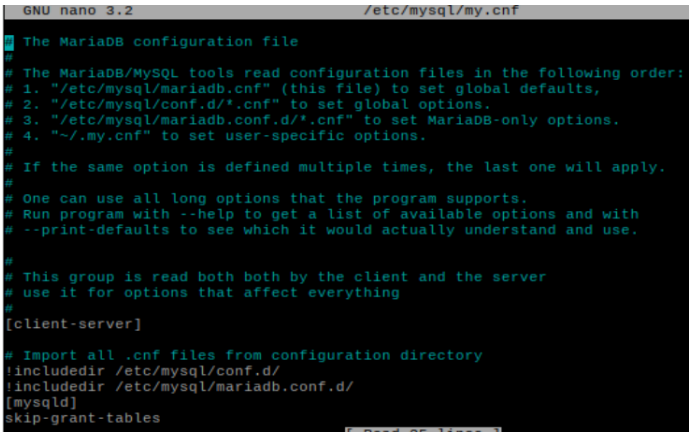

Fig. 3. Archivo de texto modificado

Se debe guardar y cerrar el archivo. Luego se debe activar la configuración *snippet* con el comando:

```
sudo a2enconf phpmyadmin.conf
```

Script 11. Activación de la configuración

Tambien se requiere crear una carpeta llamada *temp* para phpmyadmin con los comandos:

```
sudo mkdir -p /var/lib/phpmyadmin/tmp
sudo chown www-data:www-data /var/lib/phpmyadmin/tmp
```

Script 12. Creación de la carpeta

Finalmente, para que todos los cambios tenga efecto se debe reiniciar el servicio de nuevo con el siguiente comando:

```
sudo systemctl reload apache2
```

Script 13. Reinicio del servidor

Para ingresar a la página se debe colocar en el buscador la siguiente dirección:

```
direccion-ip-del-servidor/phpmyadmin
```

Script 14. Dirección para acceder a la página web

C. Crear una base de Datos de nombre "Registrador" en MySQL o MariaDB y administrarla a través de phpMyAdmin. Crear una tabla denominada "Variables_Meteorologicas" e introducir las variables ambientales del script realizado en la práctica anterior. Adquirir datos cada 60 segundos de la placa Sense HAT.

Creación de la base de datos con cada uno de los parámetros y tipos de datos que se mostrarán en la página web.

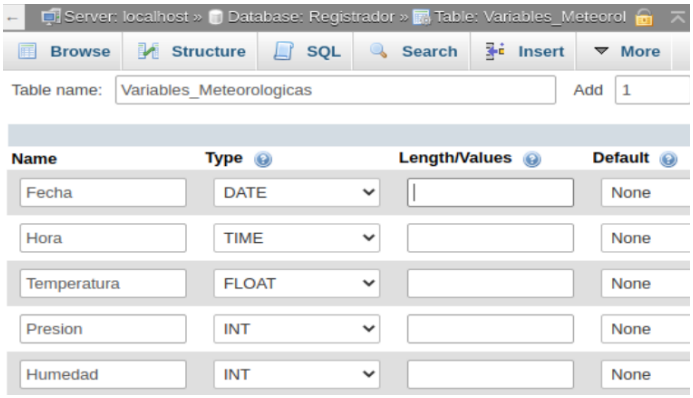

Fig. 4. Base de datos en PHPmyadmin

Para poder realizar esta actividad, se realizó un script en HTML con un bloque de código php incrustado en él, el próposito de este código es leer de la base de datos a la vez que se presenta la página web. Para que la página se muestre sin dar problema, todos los códigos en HTML se encuentran en el directorio `/var/www/html` junto con sus respectivas imágenes. Cabe destacar que para mostrar la tabla se usó CSS para agregar mas detalles como el fondo de la tabla o los títulos con sus diferentes colores y fuentes.

```
 GNU nano 3.2                    mostrar_datos.php

<?php
$conn = new mysqli ('localhost ', 'root ', '1234 ',
'Datos ');
$file = fopen ("/ home /pi/Programas Datos . txt ","r");
while (! feof ( $file )){
$content = fgets ( $file );
$carray = explode (",", $content );
list ( $Fecha ,$Hora , $Temperatura , $Humedad ,
$Presion )= $carray ;
$sql = " INSERT   INTO  'Datos ' ('Fecha ', 'Hora ',
 'Temperatura ', 'Presion ', 'HUmedad ')
VALUES  (' $Fecha ', '$Hora ', ' $Temperatura
', '$Presion ', '$Humedad ')";
$conn -> query ( $sql );
}
fclose ( $file )
?>
```

Fig. 5. Archivo mostrar_datos.php

```
#Escuela Politecnica Nacional
#CP-Sistemas Embebidos
#Practica 12
#Importacion de la libreria SenseHat, mean, math,
    time, MySQLdb, sys
from sense_emu import SenseHat
from statistics import mean
import math
import time
import MySQLdb
import sys
#Establecimiento conexion con la base de datos
def consultaDB(query):
    db = MySQLdb.connect(host='localhost', user='
    root', passwd='', db='Registrador')
    cur=db.cursor()
    cur.execute(query)
    db.commit()
    cur.close()
    db.close()
#Creacion del objeto
sense = SenseHat()
#Creacion de vectores en blanco para registrar
    variables
temp=[]
pres=[]
hume=[]
while True:
    for i in range(1,6):
        #Medicion de la temperatura
        temperatura = sense.get_temperature()
        #print(str(temperatura))
        temp.append(temperatura)
        #Medicion de la humedad
        humedad = sense.get_humidity()
        hume.append(humedad)
        #Medicion de la presion
        presion = sense.get_pressure()
        pres.append(presion)
        time.sleep(1)
    #Obtencion del promedio de las mediciones
    temperatura1=round(mean(temp),2)
    humedad1=math.floor(round(mean(hume),2))
    presion1=math.floor(round(mean(pres),2))
    #Vectores nuevamente en blanco
    temp=[]
    pres=[]
    hume=[]
    time.sleep(10)
    #Ingreso de los datos registrados
    if humedad is not None and temperatura is not
    None and presion is not None:

        consulta = 'INSERT INTO
    Variables_Meteorologicas(Fecha, Hora,
    Temperatura, Presion, Humedad) VALUES('"'+time.
    strftime("%Y/%m/%d")+'"','"'+time.strftime("%H:%M
    :%S")+'"',' +str(temperatura1)+','+ str(presion1)
     +','+ str(humedad1) + ')'
        print(consulta)
        consultaDB(consulta)
        time.sleep(1)
    else:
        print("Falla al obtener lectura. Pruebe de
    nuevo")
        sys.exit(1)
```

Script 15. Código implmentado para la lectura de las variables metorológicas

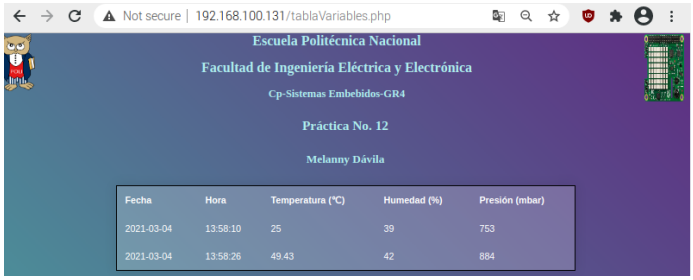

Fig. 6. Página web

Fig. 7. Base de datos en PHPmyadmin

Los scripts creados se presentan en el Anexo I.

D. En una empresa de fabricación de ropa se desea controlar los procesos de automatización en la elaboración de prendas de vestir, para lo cual se utilizará una plataforma de software libre (Raspberry Pi). Para el control de la producción se ha utilizado un código de barras el cual permite identificar la prenda elaborada. El código de barras a procesar está integrado por: lote, paquete, talla y cantidad. El lote estará identificado por los primeros 5 números del código de barras. El paquete estará identificado por los siguientes 2 números del código de barras. La talla estará identificada por las letras: XS, S, M, L y XL. Luego de lo cual existe un espacio y se presenta la cantidad, como se muestra en la figura 1. El código de barras será ingresado mediante el terminal o mediante el shell de Thonny, posteriormente al ingreso se presionará el botón central del joystick de la plataforma Sense Hat y se registrará hora y fecha de operación. Los resultados del sistema se presentarán en una página web mediante una tabla como se indica en la figura 2.

Para la elaboración de este literal, en cuanto a la codificación en Python se toma todo un string que correspondería al código

de barras ingresado, este es dividido entre varias variables que corresponden a cada uno de los campos indicados, se espera que se presione el botón del centro del joystick y se pide que se ingresen los datos nuevamente.

En cuanto a la codificación en HTML no hay mayor diferencia con el caso anterior mas que los colores de la tabla y de los textos de cabecera. Para este caso el código incrustado en php se encarga de leer los datos de un archivo de texto ya que esto presenta mayor facilidad de implementación que con una base de datos ya que se tenían problemas al tratar las variables tipo `VARCHAR` que serían las indicadas para manejar datos con números y caracteres mezclados. El código en php es muy simple, solo crea filas con datos mientras lee los archivos.

```
#Escuela Politecnica Nacional
#CP-Sistemas Embebidos
#Practica 12
#Importacion de la libreria SenseHat, time y sys
from sense_emu import SenseHat
import time
import sys
#Creacion del objeto
sense = SenseHat()
#Lazo while para leer los codigos de barras
while True:
    codigoDeBarras = input("Ingrese el codigo de barras\n")
    lote = codigoDeBarras[0:5:1]
    paquete = codigoDeBarras[5:7:1]
    talla = codigoDeBarras[7:9:1]
    cantidad = codigoDeBarras[9:len(codigoDeBarras):1]
    event = sense.stick.wait_for_event()
    hora = time.strftime("%H:%M:%S")
    fecha = time.strftime("%Y-%m-%d")
    mostrar = [codigoDeBarras+','+lote+','+paquete+','+talla+','+cantidad+','+hora+','+fecha]# Concateno
    #Escribir la informacion
    archivo = open('/var/www/html/codigosdebarras.txt','a')
    archivo.write(str(mostrar)[2:-2])
    archivo.write("\n")
    archivo.close()
    time.sleep(2)
```

Script 16. Líneas de código que permiten la lectura de un código de barras

Fig. 8. Página web

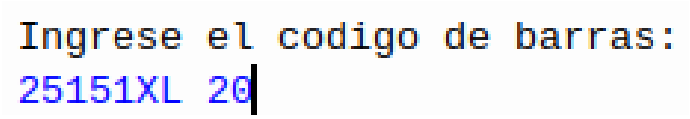

Fig. 9. Ingreso del código de barras

Los scripts creados se presentan en el Anexo II.

REFERENCES

[1] "Servidor LAMP: una solución económica para webs dinámica", IONOS Digitalguide. https://www.ionos.es/digitalguide/servidores/know-how/servidor-lamp-la-solucion-para-webs-dinamicas/ (accedido mar. 04, 2021).

[2] E. Espinosa, E. Tatayo, "IMPLEMENTACIÓN DE UN SERVIDOR L.A.M.P. PARA VISUALIZACIÓN DE DATOS -RASPBERRY S.O". C.P. SISTEMAS EMBEBIDOS, Accedido: feb. 26, 2021. [En línea].

[3] "About the Apache HTTP Server Project - The Apache HTTP Server Project". http://httpd.apache.org/ABOUT_APACHE.html (accedido mar. 04, 2021).

[4] S. B. Tosbling, "Servidor LAMP: ¿Qué es? Funcionamiento e Instalación paso a paso", Infranetworking, jun. 09, 2020. https://blog.infranetworking.com/servidor-lamp/ (accedido mar. 04, 2021).

[5] Cómo instalar en Ubuntu 18.04 la pila LAMP — Linux, Apache, MySQL y PHP. https://www.digitalocean.com/community/tutorials/como-instalar-en-ubuntu-18-04-la-pila-lamp-linux-apache-mysql-y-php-es

[6] X. Guoan, "Install phpMyAdmin with Apache (LAMP) on Debian 10 Buster", LinuxBabe, jul. 13, 2019. https://www.linuxbabe.com/debian/install-phpmyadmin-apache-lamp-debian-10-buster (accedido mar. 04, 2021).

ANEXO I

Código para mostrar la tabla con las variables meteorológicas

```
<head>
  <title>Literal 3</title>
</head>

<style>
html,
.a {
  text-align: center;
}
body {
  height: 100%;
}

body {
  margin: 0;
  background: linear-gradient(45deg, #49a09d, #5f2c82);
  font-family: sans-serif;
  font-weight: 100;
}

.container {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
}

table {
  width: 800px;
  border-collapse: collapse;
  overflow: hidden;
  box-shadow: 0 0 20px rgba(0,0,0,0.1);
}

th,
td {
  padding: 15px;
  background-color: rgba(255,255,255,0.2);
  color: #fff;
}

th {
  text-align: left;
}

thead {
```

```
  th {
    background-color: #55608f;
  }
}

tbody {
  tr {
    &:hover {
      background-color: rgba(255,255,255,0.3);
    }
  }
  td {
    position: relative;
    &:hover {
      &:before {
        content: "";
        position: absolute;
        left: 0;
        right: 0;
        top: -9999px;
        bottom: -9999px;
        background-color: rgba(255,255,255,0.2);
        z-index: -1;
      }
    }
  }
}
</style>
<body>
  <div class="a">
    <img src="buho.png" alt="universidad" style="float:left;
       width:60px;height: 100px;"/>
    <img src="sensehat.png" alt="sensehat" style="float:right;
       width:90px;height: 120px;"/>
    <h2 style="color:paleTurquoise; font-family:Simpson;" >
       Escuela Politecnica Nacional</h1>
    <h2 style="color:paleTurquoise; font-family:Simpson;" >
       Facultad de Ingenieria Electrica y Electronica</h2>
    <h3 style="color:paleTurquoise; font-family:Simpson;">
       Cp-Sistemas Embebidos</h3>
    <h4 style="color:paleTurquoise; font-family:Simpson;
       font-size:23px;">Practica No. 12</h4>
    <h5 style="color:paleTurquoise; font-family:Simpson;
       font-size:20px;">Melanny Davila<h5>
    <table style="border:1px solid black;margin-left:auto;
       margin-right:auto;">
      <tr>
      <th>Fecha</th>
      <th>Hora</th>
```

```
      <th>Temperatura (<span>&#8451;</span>)</th>
      <th>Humedad (<span>&#37;</span>)</th>
      <th>Presi n (mbar)</th>
      </tr>
        <?php
        $conn=mysqli_connect("localhost","Pi", "1234","Datos");
        if ($conn->connect_error)
        {
          die("Connection failed:".$conn-> connect_error);
        }

        $sql="SELECT * FROM Datos";
        $result = $conn-> query($sql);
        if ($result-> num_rows > 0){
          while($row = $result->fetch_assoc()){
            echo "<tr><td>". $row["Fecha"]."</td><td>". $row["
                Hora"]."</td><td>". $row["Temperatura"]."</td><
                td>". $row["Humedad"]."</td><td>". $row["
                Presion"]."</td></tr>";
          }
          echo "</table>";
        }
        else{
          echo "0 result";
        }
        $conn-> close();
        ?>
    </table>
  </div>
</body>
```

ANEXO II

Código para mostrar la tabla para los códigos de barras

```
<html lang="en">
<head>
  <meta charset="utf-8" />
  <title>Literal 4</title>
  <meta name="viewport" content="initial-scale=1.0;
      maximum-scale=1.0; width=device-width;">
</head>
<style>
  @import url(https://fonts.googleapis.com/css?family=
      Roboto:400,500,700,300,100);
  .a {
    position: relative;
    text-align: center;
  }
```

```
.centrado{
  position: relative;
  top:50%;
  left:50%;
  transform: translate(-50%,-50%);
}
body {
  background-color: #823F5D;
  font-family: "Roboto", helvetica, arial, sans-serif;
  font-size: 16px;
  font-weight: 400;
  text-rendering: optimizeLegibility;
}

div.table-title {
   display: block;
  margin: auto;
  max-width: 600px;
  padding:5px;
  width: 100%;
}

.table-title h3 {
   color: #fafafa;
   font-size: 30px;
   font-weight: 400;
   font-style:normal;
   font-family: "Roboto", helvetica, arial, sans-serif;
   text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.1);
   text-transform:uppercase;
}


/*** Table Styles **/

.table-fill {
  background: white;
  border-radius:3px;
  border-collapse: collapse;
  height: 320px;
  margin: auto;
  max-width: 600px;
  padding:5px;
  width: 100%;
  box-shadow: 0 5px 10px rgba(0, 0, 0, 0.1);
  animation: float 5s infinite;
}

th {
```

```
    color:#D5DDE5;;
    background:#1b1e24;
    border-bottom:4px solid #9ea7af;
    border-right: 1px solid #343a45;
    font-size:23px;
    font-weight: 100;
    padding:24px;
    text-align:left;
    text-shadow: 0 1px 1px rgba(0, 0, 0, 0.1);
    vertical-align:middle;
  }

  th:first-child {
    border-top-left-radius:3px;
  }

  th:last-child {
    border-top-right-radius:3px;
    border-right:none;
  }

  tr {
    border-top: 1px solid #C1C3D1;
    border-bottom-: 1px solid #C1C3D1;
    color:#666B85;
    font-size:16px;
    font-weight:normal;
    text-shadow: 0 1px 1px rgba(256, 256, 256, 0.1);
  }

  tr:hover td {
    background:#4E5066;
    color:#FFFFFF;
    border-top: 1px solid #22262e;
  }

  tr:first-child {
    border-top:none;
  }

  tr:last-child {
    border-bottom:none;
  }

  tr:nth-child(odd) td {
    background:#EBEBEB;
  }

  tr:nth-child(odd):hover td {
```

```
    background:#4E5066;
  }

  tr:last-child td:first-child {
    border-bottom-left-radius:3px;
  }

  tr:last-child td:last-child {
    border-bottom-right-radius:3px;
  }

  td {
    background:#FFFFFF;
    padding:20px;
    text-align:left;
    vertical-align:middle;
    font-weight:300;
    font-size:18px;
    text-shadow: -1px -1px 1px rgba(0, 0, 0, 0.1);
    border-right: 1px solid #C1C3D1;
  }

  td:last-child {
    border-right: 0px;
  }

  th.text-left {
    text-align: left;
  }

  th.text-center {
    text-align: center;
  }

  th.text-right {
    text-align: right;
  }

  td.text-left {
    text-align: left;
  }

  td.text-center {
    text-align: center;
  }

  td.text-right {
    text-align: right;
  }
```

```
160
161 </style>
162
163
164 <body>
165     <h2 style="color:darkTurquoise; font-family:Simpson;",
          class="a" >Escuela Politecnica Nacional</h1>
166     <img src="buho.png" alt="universidad" style="float:left;
          width:60px;height: 90px;"/>
167     <img src="bar.png" alt="sensehat" style="float:right;
          width:100px;height: 60px;"/>
168     <h2 style="color:darkTurquoise; font-family:Simpson;",
          class="a"  >Facultad de Ingenieria Electrica y
          Electronica</h2>
169     <h3 style="color:darkTurquoise; font-family:Simpson;",
          class="a">Cp-Sistemas Embebidos</h3>
170     <h4 style="color:darkTurquoise; font-family:Simpson;
          font-size:23px;", class="a">Practica No. 12</h4>
171     <h5 style="color:darkTurquoise; font-family:Simpson;
          font-size:20px;", class="a">Melanny Davila</h5>
172     <table style="border:1px solid black;margin-left:auto;
          margin-right:auto;">
173       <tr>
174       <th>Codigo<br>Barras</th>
175       <th>Lote</th>
176       <th>Paquete </th>
177       <th>Talla </th>
178       <th>Cantidad</th>
179       <th>Fecha</th>
180       <th>Hora</th>
181       </tr>
182       <?php
183       $file = fopen("codigosdebarras.txt", "r");
184       while(!feof($file)){
185         $content = fgets($file);
186         $carray = explode(",", $content);
187         list($Codigo,$Lote,$Paquete,$Talla,$Cantidad,$Fecha,
              $Hora)= $carray;
188         echo"<tr><td>$Codigo</td><td>$Lote</td><td>$Paquete</td
              ><td>$Talla</td><td>$Cantidad</td><td>$Fecha</td><
              td>$Hora</td></tr>";
189       }
190       echo "</table>";
191       fclose($file)
192       ?>
193     </table>
194
195 </body>
```