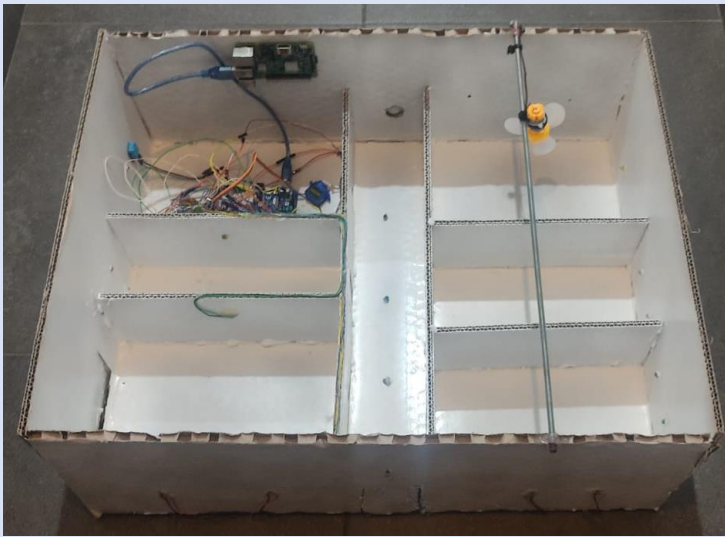


Domótica en Casa



Mario Rodríguez Rodríguez
[IES Zaidín Vergeles \(Granada\)](#)

Profesora: Aurora Pleguezuelos

2º Administración de
Sistemas Informáticos

1 INDICE

2	DEFINICION DEL PROYECTO.....	3
3	PALABRAS CLAVE	3
4	DESARROLLO DEL PROYECTO	4
	4.1.1 Raspberry Pi 3	
	4.1.2 Servidor Web	
	4.1.3 Script Python	
	4.1.4 Servidor de Base de Datos	
	4.1.5 Placa Arduino ATmega 2560	
	4.1.6 Código Arduino	
	4.1.7 Modulos y dispositivos de Arduino	
	4.1.8 Maqueta	
5	CONCLUSIONES	13
6	RECURSOS BIBLIOGRAFICOS Y PAGINAS WEB CONSULTADAS.....	14
7	ANEXOS	14
8	CONTACTO	14

2 DEFINICION DEL PROYECTO

Este proyecto consiste en una simulación del sistema domótico de una vivienda, este sistema se puede implementar en una vivienda, una oficina, un hotel, etc.

A este sistema se le podría añadir ciertos permisos para cada usuario y así poder evitar que terceros puedan alterar el funcionamiento, o para poder limitar el acceso a ciertos dispositivos de diferentes habitaciones.

Este sistema incluye:

- Luces en cada habitación.
- Ventilador.
- Sistema de Detección de movimiento con alarma.
- Sensor de Temperatura y Humedad.
- Puerta de Garaje automática.

El objetivo de este proyecto ha sido diseñar un sistema que permita al usuario controlar los dispositivos de su hogar de una manera fácil y sencilla.

[Volver al índice](#)

3 PALABRAS CLAVE

Palabras clave:

- Sistema Domótico.
- Arduino.
- Interfaz domótica.

[Volver al índice](#)

4 DESARROLLO DEL PROYECTO

Este proyecto se puede escalar a un nivel superior, añadiendo una aplicación con permisos personalizados para cada usuario y limitación de dispositivos.

También se podría controlar por medio de Bluetooth gracias al módulo HC-05 de Arduino.

Este sistema está constituido por:

- Raspberry PI 3.
- Servidor Web.
- Script Python.
- Servidor Base de datos.
- Placa Arduino Mega 2560.
- Código Arduino.
- Módulos de Arduino y Cableado.
- Diodos Arduino.
- Maqueta.

Raspberry PI 3

Se ha usado una Raspberry PI 3 Model B que será la que procesa toda la información del sistema.

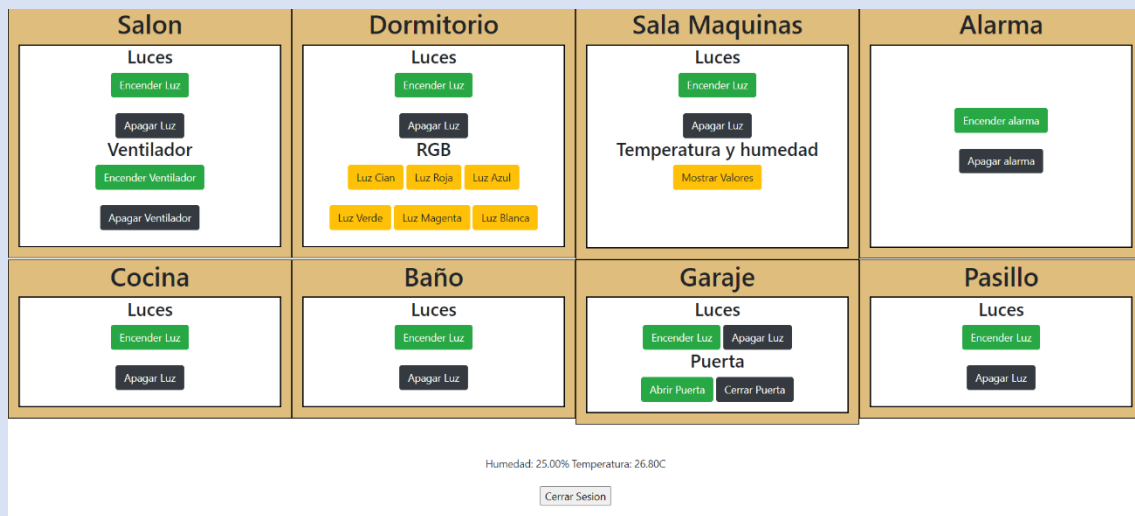
La Raspberry PI 3 cuenta con un procesador de 4 núcleos y 1'2 GHz y memoria de 1GB de RAM.

Esta Raspberry lleva alojado un servidor Web que contiene el código PHP necesario y un Script Python para la comunicación con una placa Arduino.

Servidor Web

Consiste en un servidor web con Apache y MariaDB.

Aquí tenemos la pagina web:



Al acceder a la web nos pide credenciales, las cuales se crean con un script que genera usuarios.

The screenshot shows a login form with a dark gray background. It contains two input fields: 'Nombre de usuario' and 'Contraseña'. Below the fields is an 'Enviar' button. The text 'Introduce el usuario' and 'Introduce la contraseña' is displayed above the respective input fields.

El script contiene lo siguiente:

```
<?php
require('conectar.php');
require('funcionComprobar.php');
$conexion = conectarBase();
$usuario = "$argv[1]";
$pass = Password::hash("$argv[2]");
$conexion->query("insert into usuarios values(NULL, '$usuario', '$pass');");
?>
```

Para comprobar la contraseña se usa la siguiente función:


```
<?php
class Password {
    const SALT = 'Pr0y3ct0Mario';
    public static function hash($password) {
        return hash('sha512', self::SALT . $password);
    }
    public static function verify($password, $hash){
        return ($hash == self::hash($password));
    }
}

?>
```

Se ha montado una pagina web que contiene botones para que el usuario interactúe con el sistema mediante PHP:

```
<form action="./DomoticaEnCasa.php" method="POST" style="padding:10 auto;text-align:center;background-color:white; border: 2px solid black" >
<h3>Luces</h3>
<input type="submit" name="salon" value="Encender Luz" class="btn btn-success">
<br>
<br>
<input type="submit" name="salon" value="Apagar Luz" class="btn btn-dark">
<h3>Ventilador</h3>
<input type="submit" name="salon" value="Encender Ventilador" class="btn btn-success">
<br>
<br>
<input type="submit" name="salon" value="Apagar Ventilador" class="btn btn-dark">
<br>
<br>
</form>
```

El código de esta pagina web consiste en formularios que envían el botón pulsado mediante POST.



The image shows a web form with a yellow border and a title 'Salon' in a large, bold, black font. Inside the form, there are two sections. The first section is titled 'Luces' in a bold, black font. Below it, there are two buttons: a green button with the text 'Encender Luz' and a dark gray button with the text 'Apagar Luz'. The second section is titled 'Ventilador' in a bold, black font. Below it, there are two buttons: a green button with the text 'Encender Ventilador' and a dark gray button with the text 'Apagar Ventilador'.

El mismo archivo procesa la información recibida y escribe lo que se ha pulsado en un archivo de texto, que posteriormente será el Script de Python el encargado de leerlo y enviarlo a Arduino.

```
if (isset($_POST['salon'])){
    if ($_POST['salon'] == "Encender Luz"){
        $archivo = fopen("fichero.txt", "w");
        fwrite($archivo, "encender_luz_salon");
        fclose($archivo);
    }
    if ($_POST['salon'] == "Apagar Luz"){
        $archivo = fopen("fichero.txt", "w");
        fwrite($archivo, "apagar_luz_salon");
        fclose($archivo);
    }
    if ($_POST['salon'] == "Encender Ventilador"){
        $archivo = fopen("fichero.txt", "w");
        fwrite($archivo, "encender_ventilador");
        fclose($archivo);
    }
    if ($_POST['salon'] == "Apagar Ventilador"){
        $archivo = fopen("fichero.txt", "w");
        fwrite($archivo, "apagar_ventilador");
        fclose($archivo);
    }
}
```

Script Python

Mediante este código se conecta mediante el puerto correcto (pasado al script como parámetro) Python con Arduino

```
arduino = serial.Serial(port="/dev/" + sys.argv[1], baudrate=9600, timeout=.1)
```

Este Script consiste en un bucle infinito que está continuamente leyendo un fichero de texto, y si la información recibida es correcta, la manda mediante una función a Arduino

```
while True:
    time.sleep(0.5)
    try:
        with open("fichero.txt") as f:
            firstline = f.readline().rstrip()
            print (firstline)
            value = write_read(firstline)
            time.sleep(0.5)

            print(value)
            f.close
            remove("fichero.txt")
    except:
        print()
```

```
def write_read(x):
    arduino.write(bytes(x, 'utf-8')) # MANDA A ARDUINO LO QUE SE HA ESCRITO AL ARCHIVO
    # QUE ES UN COMANDO PREVIAMENTE REGISTRADO EN ARDUINO
    time.sleep(1)
    data1 = arduino.readline() #AQUI LEE DATOS DESDE ARDUINO
    data = str(data1).replace("b","").replace("'", "")
    Humedad = data[0:6]
    Temperatura = data[len(data)-7:len(data)]
    sql = "INSERT INTO HistorialTemperatura values(NULL,%s,%s)"
    val = (Temperatura,Humedad)
    mycursor.execute(sql,val)
    mydb.commit()
#    time.sleep(1)
    time.sleep(0.05)
    return data
```

Para automatizar el proceso de ejecución de este script, se ha creado otro script Bash para ejecutar el script Python

```
#!/bin/bash
sleep 10
tty=$( dmesg | grep "ttyACM" | awk {'print $5'} | sed 's/://g')
#echo $tty
python3 /var/www/html/DomoticaEnCasa/DomoticaEnCasa/DomoticaEnCasa.py $tty
```

Servidor Base de Datos

Se ha montado un servidor MariaDB para almacenar la información que recibe el script Python de la temperatura y humedad.

El servidor tiene las siguientes tablas:

```
+-----+
| Tables_in_DomoticaEnCasa |
+-----+
| HistorialTemperatura      |
| usuarios                  |
+-----+
```

En la tabla HistorialTemperatura se van guardando los registros de temperatura y humedad que se van obteniendo para luego mostrarlo.

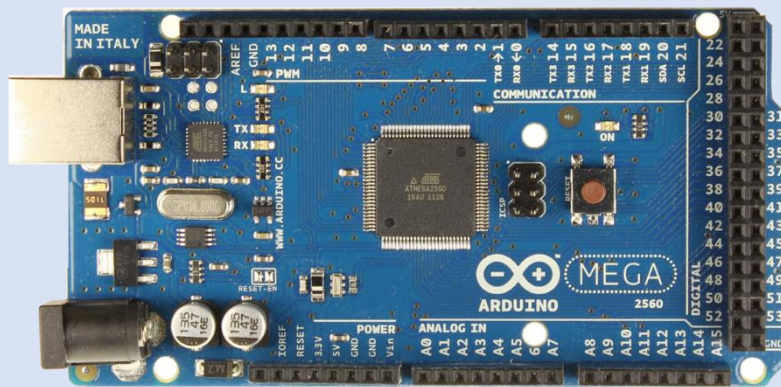
En la tabla usuarios están los usuarios permitidos para iniciar sesión.

Placa Arduino Mega 2560

Como placa controladora del sistema se ha usado una Arduino Mega 2560.

Esta placa tiene las siguientes características técnicas:

- Microcontrolador: ATmega2560
- Pines de E/S digital: 54
- Pines de entrada analógica: 16
- Memoria Flash: 256KB (8 de ellos se usan para el arranque)
- SRAM. 8KB
- EEPROM: 4KB
- Frecuencia reloj: 16 MHz



Código Arduino

La placa Arduino se programa mediante el Lenguaje de Programación de Arduino que envía y recibe información mediante el puerto Serial y los pines de esta placa.

Aquí hay algunos ejemplos de código Arduino que han sido usados en este sistema:

- Para incluir las librerías necesarias:

```
#include <SPI.h>
#include <Config.h>
#include <EasyBuzzer.h>
#include <DHT.h>
#include <Servo.h>
#define DHTPIN 2
#define DHTTYPE DHT11
```

- Para definir los pines:

```
int red_light_pin= 30;
int green_light_pin = 31;
int blue_light_pin = 32;
int ventilador = 33;
float humedad = 0;
float temperatura = 0;
int ledRojo = 38;
int ledVerde = 39;
int ledSalon = 42;
int ledDormitorio = 43;
int ledCocina = 44;
int ledGaraje = 45;
int ledBano = 46;
int ledMaquinas = 47;
int ledPasillo = 48;
const byte pinBuzzer = 3;
const int PIRPin = 5;
```

- Para definir un comando que le pasará después la Raspberry por la comunicación Serial:

```
if(comando=="encender_alarma"){  
  if (alarma==false){  
    digitalWrite(ledRojo, HIGH);  
    digitalWrite(ledVerde, LOW);  
    alarma = true;  
    EasyBuzzer.singleBeep(10,2);  
    delay(1000);  
    EasyBuzzer.stopBeep();  
  }  
}
```

- Esta parte de código lo que hace es que si la variable "comando" y la alarma esta apagada, cambia el estado de la alarma a encendida, cambia las luces y emite un pitido para avisar de que ya se ha activado la alarma

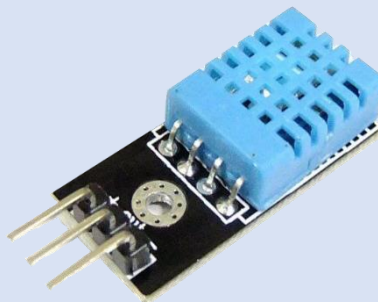
Módulos y Dispositivos de Arduino

Sensor infrarrojo pasivo (PIR)



Este sensor, mide la radiación infrarroja de los objetos, y envía la señal a Arduino.

Sensor DHT11



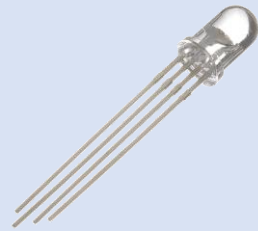
Este sensor mide la temperatura y humedad del ambiente

Buzzer



Este dispositivo se usa para emitir un pitido cuando se le manda energía (en este caso cuando se detecte algún movimiento con el sensor).

Diodos Led



Se han usado 2 tipos de Diodos Led (Diodos comunes, y un diodo RGB que permite cambiar su color según las necesidades del usuario).

Ventilador

El ventilador que se ha usado es un simple ventilador que tiene 2 pines (Positivo y Negativo), que esta conectado a un pin digital que le manda la señal de encendido/apagado

Servo SG90



Es un pequeño motor que permite controlar su ángulo de posición de forma precisa, este servomotor puede rotar desde 0° hasta 180°.

Maqueta

Para hacer la maqueta se han usado cajas de cartón y se ha hecho una simulación de una casa.

Para unir el cartón se ha usado silicona caliente.

En esta maqueta se ha colocado todo el sistema, ubicándose la parte que procesa en una sala llamada sala de máquinas.

Las conexiones se han realizado mediante cables soldándolos a los dispositivos y conectándolos a los pines digitales de Arduino.

Con el uso de un taladro, se han hecho los diferentes orificios para pasar el cableado del sistema.

Para hacer las puertas se ha usado un cúter cortando las puertas permitiendo que se puedan abrir y cerrar.

[Volver al índice](#)

5 CONCLUSIONES

Para la elaboración del proyecto lo más complicado que me ha resultado ha sido el cableado del sistema, ya que han sido muchos dispositivos.

Este sistema se puede implantar a escala real sin mucha inversión y de forma remota, gracias a los nuevos dispositivos que tenemos que permiten conectarse de manera remota, y así no sería necesario un cableado de toda la vivienda.

[Volver al índice](#)

6 RECURSOS BIBLIOGRAFICOS Y PAGINAS WEB CONSULTADAS

- proyectoarduino.com
- luisllamas.es
- tutorialspoint.com
- stackoverflow.com

[Volver al índice](#)

7 ANEXOS

- Código Python (DomoticaEnCasa.py)
- Código PHP (DomoticaEnCasa.php)
- Código Arduino (DomoticaEnCasa.ino)
- Script Arranque (scriptInicio.sh)

[Volver al índice](#)

8 CONTACTO

- Github
github.com/Br0xt0r
- Correo electrónico
mariorbrb1@gmail.com

[Volver al índice](#)