



**C.P.R. Liceo “La Paz”
Proyecto fin de ciclo**

OTTOMATIC

Administración de Sistemas Informáticos en Red

**Autor : Yago Fernández Blanco
Tutor : Xabier Pérez Maestre**

Resumen

En este Proyecto de Fin de Ciclo de Administración de Sistemas Informáticos en Red (en adelante, ASIR) se ha trabajado junto a un servidor web raspberryPi de tipo LAMP, una aplicación web ("*Ottomatic*") para el manejo y control domótico de la residencia y conectado a través de un sistema mecánico (Arduino). Todo acceso a los controles está regulado y se registra toda la información de los usuarios que entran o salen de la aplicación.

Abstract

In this Project of Network Computer Systems Administration (hereinafter, ASIR) we have worked together with a raspberryPi web server of the LAMP type, a web application ("Ottomatic") for home automation management and control of the residence and connected through a mechanical system (Arduino). All access to the controls is regulated and all the information of the users that enter or leave the application is registered.

Palabras clave

El término **domótica** abarca el conjunto de técnicas y recursos para automatizar una vivienda, una muestra de eso sería la monitorización de los dispositivos de seguridad de una manera remota o el control de dispositivos eléctricos como puertas, luces o persianas.

La domótica a día de hoy supone uno de los mayores empujes al ámbito del **Internet de las Cosas**, más conocido por su nomenclatura anglosajona Internet of Things (IoT) y que se puede definir como la interconexión digital de los elementos cotidianos con Internet. Esta gran relevancia que está teniendo la domótica cada vez más en la actualidad no es algo tan extraño ya que las nuevas tecnologías lo que permiten es una mayor comodidad, seguridad y accesibilidad a nuestro hogar.

Como consecuencia surge la idea de *Ottomatic* que a través de una aplicación web nos permite manejar diversos elementos del hogar y aporta diversa información de este.

Estoy especialmente agradecido a mis padres y hermana que han tenido que vivir muy de cerca los momentos complicados del proyecto. Y me gustaría reconocer también la lluvia de ideas por parte de mi tutor que han sido de gran ayuda.

Contenido

Introducción.....	3
Objetivos.....	7
Estado del arte.....	11
Caso de estudio.....	15
Desarrollo del proyecto.....	19
Viabilidad tecno-económica.....	23
Conclusiones.....	27
Lineas abiertas de investigación.....	31
Bibliografía, referencias e índices.....	35
Bibliografía.....	35
Referencias.....	36
Índices.....	37

Introducción

Introducción

La idea principal nace de la necesidad, la necesidad de adaptarse a las tecnologías que van avanzando en nuestro día a día. Poniendo en práctica todos los conocimientos sobre informática poder elaborar un software estable y seguro para monitorizar y manejar diversos elementos del hogar.

Una aplicación web con una proyección de crecimiento importante, un servidor web basado en el principio de Open Source Hardware y una idea de proyecto muy gratificante de realizar.

Haciendo uso de una casa de muñecas se realiza la maqueta con la cual se harán las pruebas del funcionamiento del sistema domótico, usando un servomotor para las muertas, led para el sistema de luces, ventilador con motor DC y una pila de petaca de 4,5V para el suelo calefactable.

En el apartado del Servidor Web se utiliza una Raspberry Pi4 Model B con un sistema LAMP y la comunicación entre la maqueta y el servidor se realiza con un Arduino Uno Rev 3.

Objetivos

Objetivos

El objetivo principal era tener un server seguro y protegido ya que la información que pasará a través de él es sensible y podrá ser un problema importante si estuviera fácilmente accesible.

Como se mencionó previamente, se hace uso del principio Open Source Hardware ya que un objetivo clave era que todo el código realizado y los elementos hardware utilizados sean de acceso público a través de la cuenta de GitHub. Cuanto más software libre exista, las aplicaciones podrán crecer y mejorar de manera exponencial.

Otro objetivo era la simplicidad, una aplicación web sencilla al alcance y manejo de cualquiera para poder hacerla accesible a todo tipo de público por muy inexperto que pueda ser en el ámbito de las tecnologías.

Estado del arte

Estado del arte

Las tecnologías no para de crecer y día a día son cada vez más protagonistas en nuestras vidas haciéndose un hueco importante en nuestros hogares y facilitando nuestras tareas diarias.

A la vez que nace la “Domótica” aparece la “Inmótica” que se define de la misma manera que la primera pero orientada a las edificaciones, tales como, edificios de viviendas, museos, hoteles, bancos o industrias.

Tal es el avance en estos campos que gobiernos como Japón, EEUU, China, Canadá y algunos países de la Unión Europea gracias a la EA0026, una ley que certifica la instalación domótica por niveles, pueden llevar un control económico y energético del uso de estas tecnologías.

A principios del 2020, la empresa de automovilismo “Toyota” anunció que tenía el proyecto de construcción en las cercanías al Monte Fuji de una ciudad para sus empleados basada 100% en la Dómotica y la Inmótica llamada Woven City. Prometen que será un pequeño ejemplo de las ciudades en el futuro.

Esta no es la primera “ciudad del futuro” que nos presentan, en 2019 Alphabet (la matriz de Google) anunció la construcción de una ciudad inteligente en Toronto capaz de recortar un 89% los gases emitidos a la atmósfera y crear un impacto económico de 14.000.000.000\$ para 2040, con 44.000 puestos de trabajo directo. El problema vino con la cantidad de información que se iba a recoger a través de esto, desde las rutinas de la gente, a donde se encuentra cada ciudadano en cada momento o incluso lo que vamos a hacer en un tiempo.

Caso de estudio

Caso de estudio

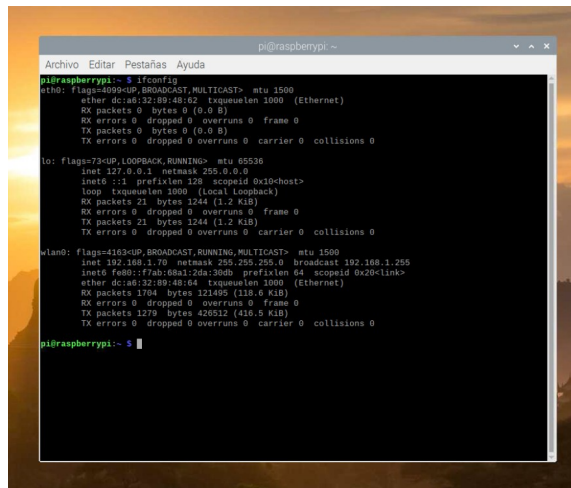
El proyecto se inicia con la idea de domotización completa de una casa transformándola en un hogar. Asegurando por completo su seguridad al mismo tiempo y convertirla en un lugar más protegido tras la domotización.

Gracias al conocimiento sobre el lenguaje de marcas, programación python y desarrollo web con php, la aplicación web es capaz de manejar los controles del sistema Arduino de una manera eficiente y segura.

Desarrollo del proyecto

Desarrollo del proyecto

Lo primero que se hizo fue la instalación del sistema operativo NOOBS en la Raspberry Pi4, la cual fue en entorno gráfico por que a pesar de que no ser necesario dicho entorno se hizo para ser más interactivo en caso de existir un futuro problema. Se configuró para poder acceder a el por SSH y por entorno VNC.

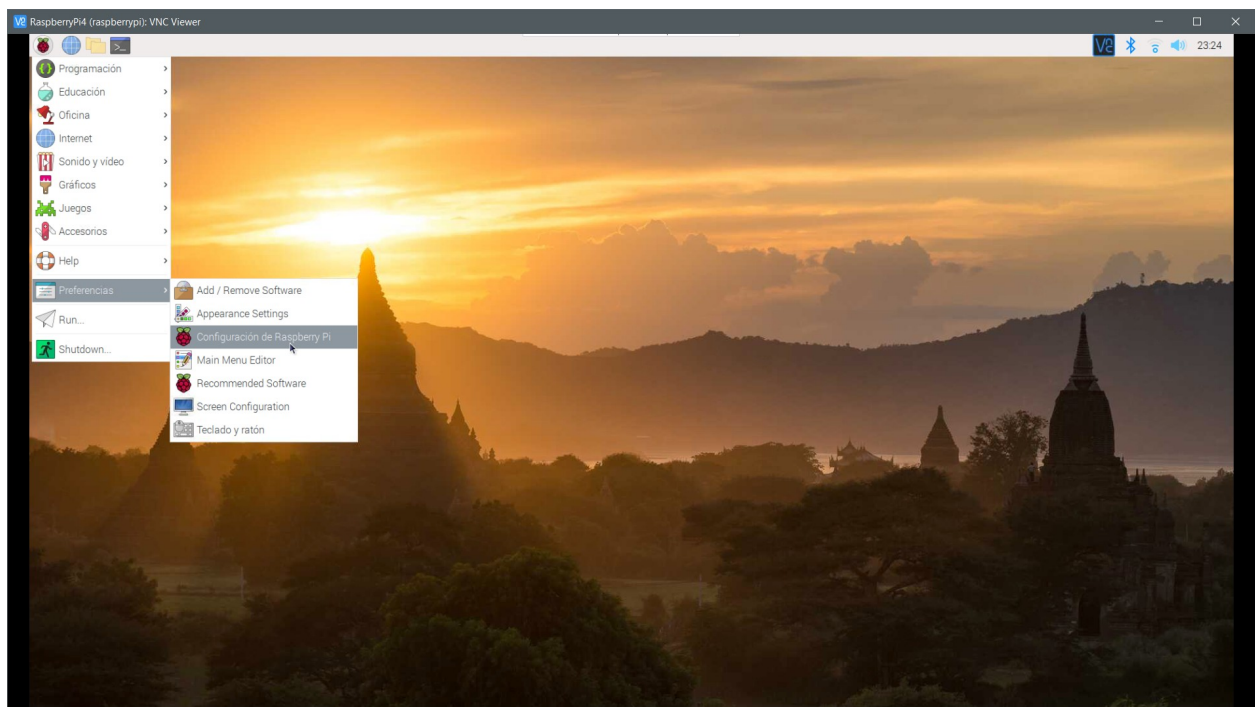


```
pi@raspberrypi:~$ ifconfig
eth0: flags=4096<UP,BROADCAST,MULTICAST> mtu 1500
    ether dc:a6:32:89:48:62 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

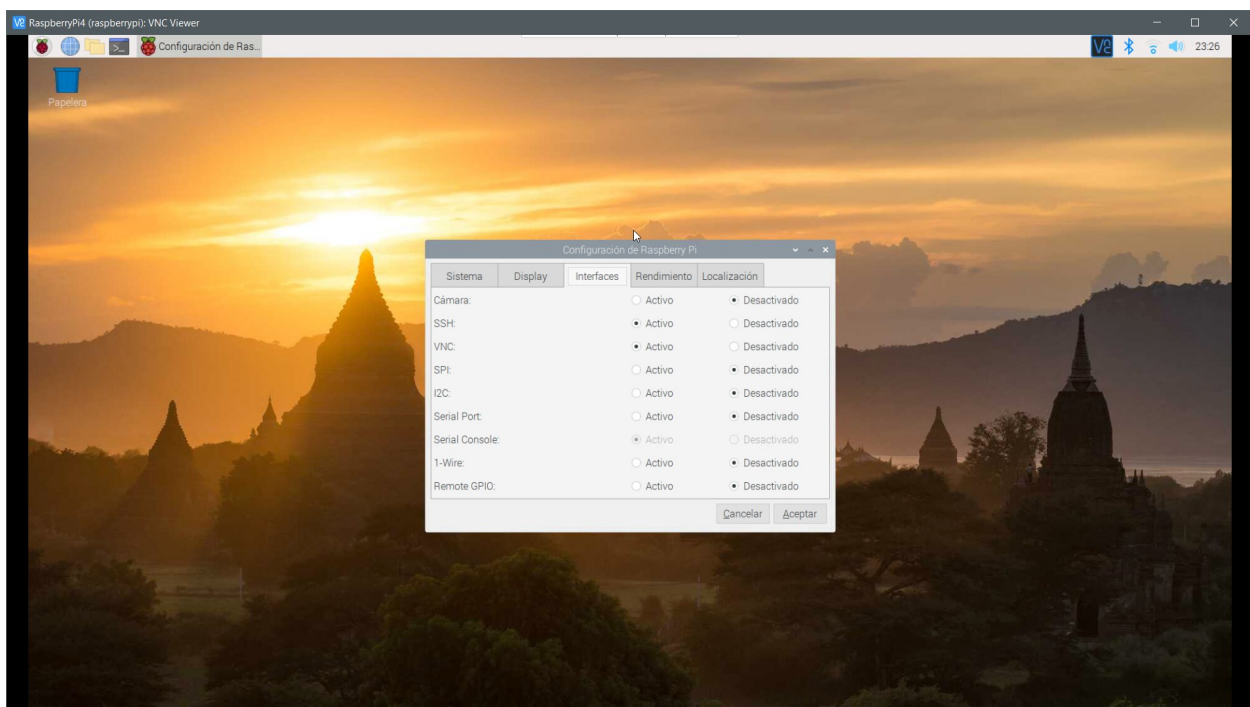
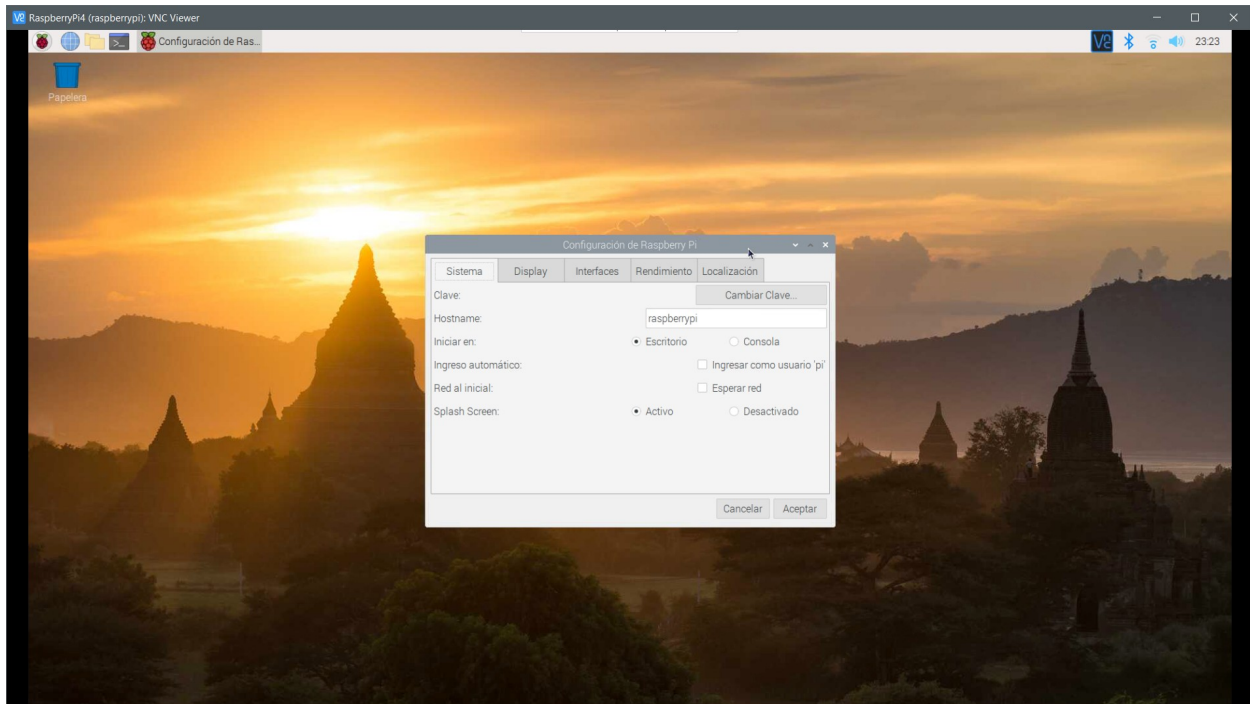
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 21 bytes 1244 (1.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 21 bytes 1244 (1.2 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.70 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::f7ab:68a1:2da:38db prefixlen 64 scopeid 0x20<link>
    ether dc:a6:32:89:48:14 txqueuelen 1000 (Ethernet)
    RX packets 1704 bytes 121495 (118.6 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1279 bytes 426312 (416.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~$
```



Desarrollo del proyecto



```

pi@raspberrypi:~$ ssh pi@192.168.1.70
Microsoft Windows [Versión 10.0.18363.836]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\yagof>ping 192.168.1.70

Haciendo ping a 192.168.1.70 con 32 bytes de datos:
Respuesta desde 192.168.1.70: bytes=32 tiempo=66ms TTL=64
Respuesta desde 192.168.1.70: bytes=32 tiempo=38ms TTL=64
Respuesta desde 192.168.1.70: bytes=32 tiempo=23ms TTL=64
Respuesta desde 192.168.1.70: bytes=32 tiempo=66ms TTL=64

Estadísticas de ping para 192.168.1.70:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
        (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 23ms, Máximo = 66ms, Media = 46ms

C:\Users\yagof>ssh pi@192.168.1.70
pi@192.168.1.70's password:
Linux raspberrypi 4.19.97-v7l+ #1294 SMP Thu Jan 30 13:21:14 GMT 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed May 13 23:28:07 2020 from 192.168.1.48
pi@raspberrypi:~$

```

Tras establecer la conexión de la Raspberry con el PC comienza la configuración del Servidor Web. Para ello se instalará Apache2, PHP7.3 y mysql.

Para el mysql usaremos MariaDB sin entorno gráfico ya que no nos aporta ninguna ventaja el entorno gráfico en este caso.

```

pi@raspberrypi:~$ sudo su
root@raspberrypi:/home/pi# apt-get update
Des:1 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0 kB]
Obj:2 http://archive.raspberrypi.org/debian buster InRelease
Des:3 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packages [13,0 MB]
Descargados 13,0 MB en 9s (1.398 kB/s)
Leyendo lista de paquetes... Hecho
root@raspberrypi:/home/pi# apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
  libmicrodms
Utilice «sudo apt autoremove» para eliminarlo.
Se actualizarán los siguientes paquetes:
  apt apt-utils libapt-inst2.0 libapt-pkg5.0
1 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 2.803 kB de archivos.
Se liberarán 9.216 B después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ftp.cica.es/mirrors/Linux/raspbian/raspbian buster/main armhf libapt-pkg5.0 armhf 1.8.2.1 [865 kB]
Des:2 http://ftp.cica.es/mirrors/Linux/raspbian/raspbian buster/main armhf libapt-inst2.0 armhf 1.8.2.1 [199 kB]
Des:3 http://ftp.cica.es/mirrors/Linux/raspbian/raspbian buster/main armhf apt armhf 1.8.2.1 [1.349 kB]
Des:4 http://ftp.cica.es/mirrors/Linux/raspbian/raspbian buster/main armhf apt-utils armhf 1.8.2.1 [399 kB]
Descargados 2.803 kB en 1s (2.925 kB/s)
Leyendo lista de cambios... Hecho.
(Leyendo la base de datos ... 156570 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libapt-pkg5.0_1.8.2.1_armhf.deb ...
Desempaquetando libapt-pkg5.0:armhf (1.8.2.1) sobre (1.8.2) ...
Configurando libapt-pkg5.0:armhf (1.8.2.1) ...
Preparando para desempaquetar .../libapt-inst2.0_1.8.2.1_armhf.deb ...
Desempaquetando libapt-inst2.0:armhf (1.8.2.1) sobre (1.8.2) ...
Preparando para desempaquetar .../archives/apt_1.8.2.1_armhf.deb ...
Desempaquetando apt (1.8.2.1) sobre (1.8.2) ...
Configurando apt (1.8.2.1) ...
(Leyendo la base de datos ... 156570 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../apt-utils_1.8.2.1_armhf.deb ...
Desempaquetando apt-utils (1.8.2.1) sobre (1.8.2) ...
Configurando libapt-inst2.0:armhf (1.8.2.1) ...
Configurando apt-utils (1.8.2.1) ...
Procesando disparadores para man-db (2.8.5-2) ...
Procesando disparadores para libc-bin (2.28-10-rpi1) ...
root@raspberrypi:/home/pi#

```

```
pi@raspberrypi: ~  
root@raspberrypi:/home/pi# apt-get install apache2
```

Comprobamos el funcionamiento de apache2.

```
pi@raspberrypi: ~  
Enabling conf charset.  
Enabling conf localized-error-pages.  
Enabling conf other-vhosts-access-log.  
Enabling conf security.  
Enabling conf serve-cgi-bin.  
Enabling site 000-default.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.  
Procesando disparadores para systemd (241-7-debianu4+rpil) ...  
Procesando disparadores para man-db (2.8.5-2) ...  
Procesando disparadores para libc-bin (2.28-10+rpil) ...  
root@raspberrypi:/home/pi# cd /var/www/html  
root@raspberrypi:/var/www/html# nano test.html
```

```
pi@raspberrypi: ~  
GNU nano 3.2 test.html  
  
<!DOCTYPE html>  
<html lang="es">  
  <head>  
    <meta charset="UTF-8">  
    <title>APACHE2</title>  
  </head>  
  <body>  
    <h1>PRIMERA PRUEBA APACHE2</h1>  
  </body>  
</html>
```

```
192.168.1.70/test.html x +  
← → ↺ ↻ 192.168.1.70/test.html ... ☆  
Nueva carpeta
```

APACHE2

PRIMERA PRUEBA APACHE2

Instalamos el PHP7.3

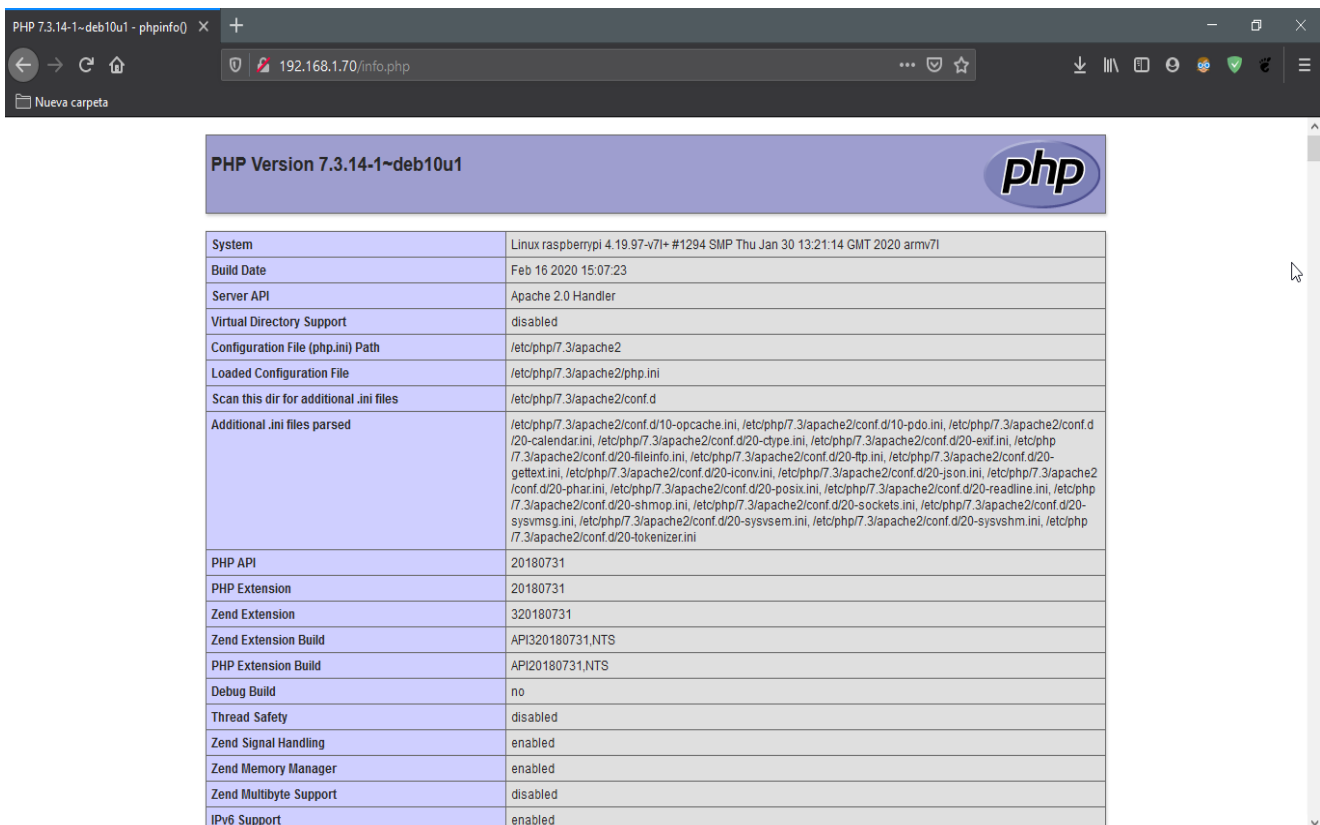
```
pi@raspberrypi: ~  
Enabling conf charset.  
Enabling conf localized-error-pages.  
Enabling conf other-vhosts-access-log.  
Enabling conf security.  
Enabling conf serve-cgi-bin.  
Enabling site 000-default.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.  
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.  
Procesando disparadores para systemd (241-7-deb10u4+rpil) ...  
Procesando disparadores para man-db (2.8.5-2) ...  
Procesando disparadores para libc-bin (2.28-10+rpil) ...  
root@raspberrypi:/home/pi# cd /var/www/html  
root@raspberrypi:/var/www/html# nano test.html  
root@raspberrypi:/var/www/html# apt-get install php7.3 libapache2-mod-php7.3
```

```
pi@raspberrypi: ~  
Creating config file /etc/php/7.3/mods-available/exif.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/fileinfo.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/ftp.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/gettext.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/iconv.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/pdo.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/phar.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/posix.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/shmop.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/sockets.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/sysvmsg.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/sysvsem.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/sysvshm.ini with new version  
  
Creating config file /etc/php/7.3/mods-available/tokenizer.ini with new version  
Configurando php7.3-opcache (7.3.14-1-deb10u1) ...  
Creating config file /etc/php/7.3/mods-available/opcache.ini with new version  
Configurando php7.3-json (7.3.14-1-deb10u1) ...  
  
Creating config file /etc/php/7.3/mods-available/json.ini with new version  
Configurando php7.3-readline (7.3.14-1-deb10u1) ...  
  
Creating config file /etc/php/7.3/mods-available/readline.ini with new version  
Configurando php7.3-cli (7.3.14-1-deb10u1) ...  
update-alternatives: utilizando /usr/bin/php7.3 para proveer /usr/bin/php (php) en modo automático  
update-alternatives: utilizando /usr/bin/phar7.3 para proveer /usr/bin/phar (phar) en modo automático  
update-alternatives: utilizando /usr/bin/phar.phar7.3 para proveer /usr/bin/phar.phar (phar.phar) en modo automático  
  
Creating config file /etc/php/7.3/cli/php.ini with new version  
Configurando libapache2-mod-php7.3 (7.3.14-1-deb10u1) ...  
  
Creating config file /etc/php/7.3/apache2/php.ini with new version  
Module mpm_event disabled.  
Enabling module mpm_prefork.  
apache2_switch_mpm Switch to prefork  
apache2_invoke: Enable module php7.3  
Configurando php7.3 (7.3.14-1-deb10u1) ...  
Procesando disparadores para man-db (2.8.5-2) ...  
root@raspberrypi:/var/www/html#
```

Comprobamos la instalación de PHP7.3

```
pi@raspberrypi: ~  
root@raspberrypi:/var/www/html# nano info.php
```

```
Seleccionar pi@raspberrypi: ~  
GNU nano 3.2 info.php  
<?php phpinfo();?>
```



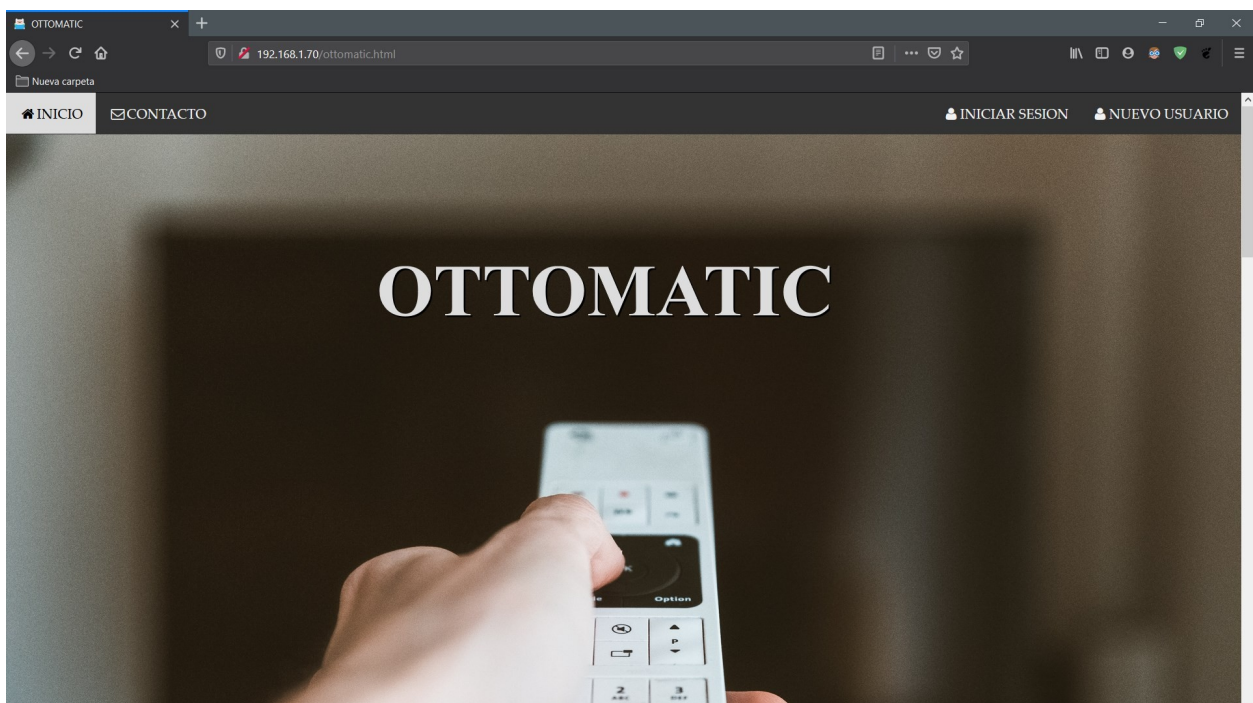
System	Linux raspberrypi 4.19.97-v7l+ #1294 SMP Thu Jan 30 13:21:14 GMT 2020 armv7l
Build Date	Feb 16 2020 15:07:23
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled

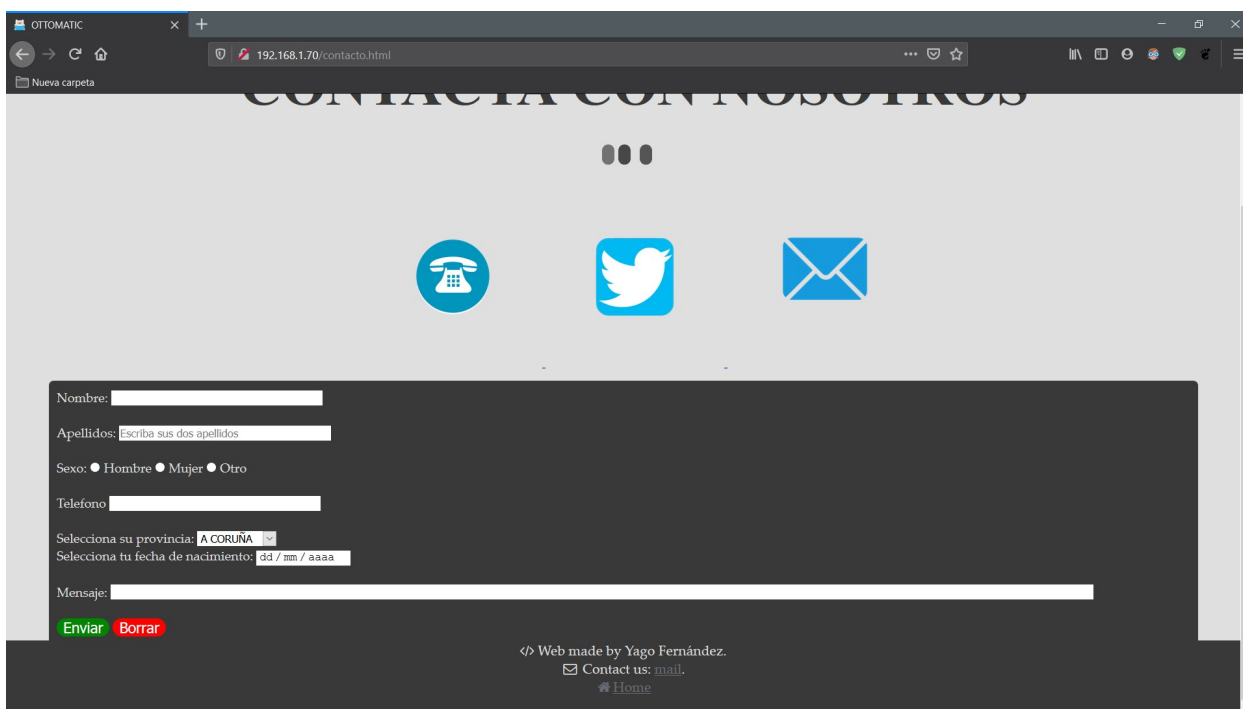
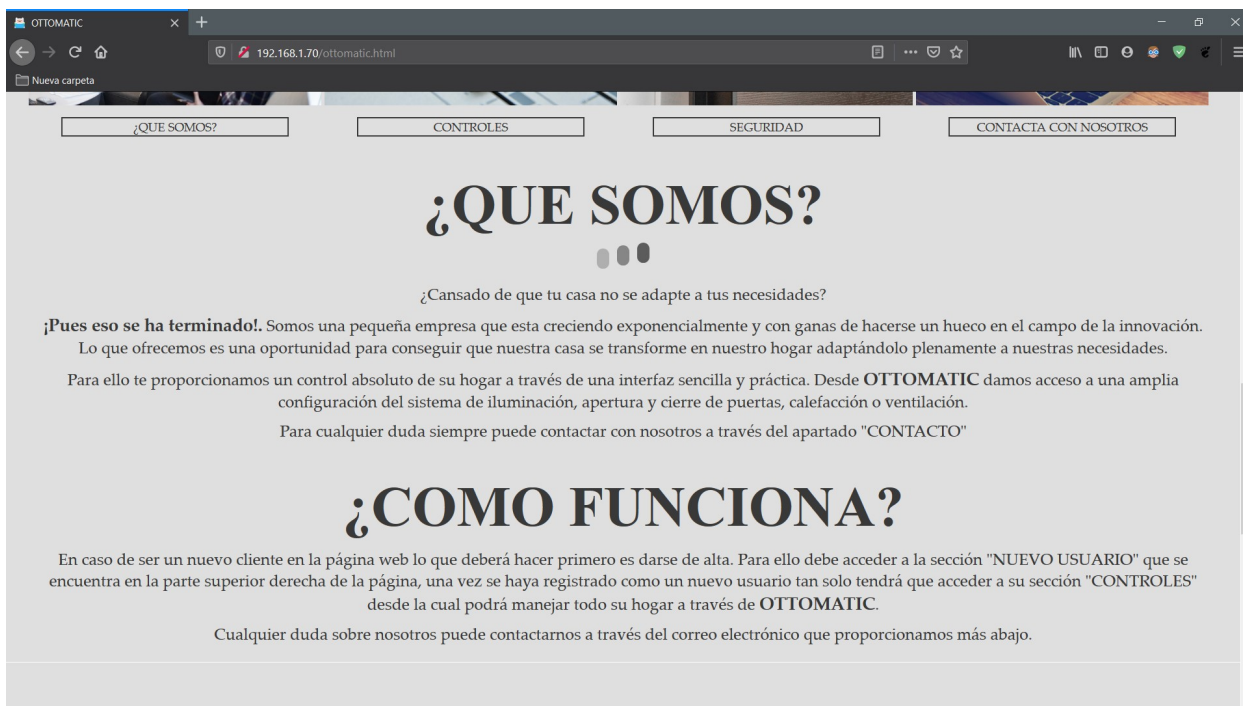
Instalamos y comprobamos el funcionamiento de mysql MariaDB

```
pi@raspberrypi: ~  
root@raspberrypi:/var/www/html# apt-get install mariadb-server-10.3
```

```
pi@raspberrypi: ~  
root@raspberrypi:/var/www/html# service apache2 restart  
root@raspberrypi:/var/www/html# mysql -u root -p  
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 47  
Server version: 10.3.22-MariaDB-0+deb10u1 Raspbian 10  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]>
```

Una vez tengamos el Servidor Web lo siguiente es diseñar la página web insertándola en la carpeta / var / www / html. A continuación se muestra como se ve la web.





The screenshot shows a web browser window with the address bar displaying `192.168.1.70/register/register.php`. The page has a dark header with navigation links: [INICIO](#), [CONTACTO](#), [INICIAR SESION](#), and [NUEVO USUARIO](#). The main content area has a light gray background with the title

Ingrese Sus Datos

. Below the title is a registration form with the following fields and labels:

- Ingrese su nombre de usuario
- Escriba su contraseña
- Repita su contraseña

At the bottom of the form is a button labeled **ENTRAR**. The footer of the page contains the text:

Web made by Yago Fernández.
Contact us: [mail](#).
[Home](#)

The screenshot shows a web browser window with the address bar displaying `192.168.1.70/login/login.php`. The page has a dark header with navigation links: [INICIO](#), [CONTACTO](#), [INICIAR SESION](#), and [NUEVO USUARIO](#). The main content area has a light gray background with the title

Ingrese Sus Datos

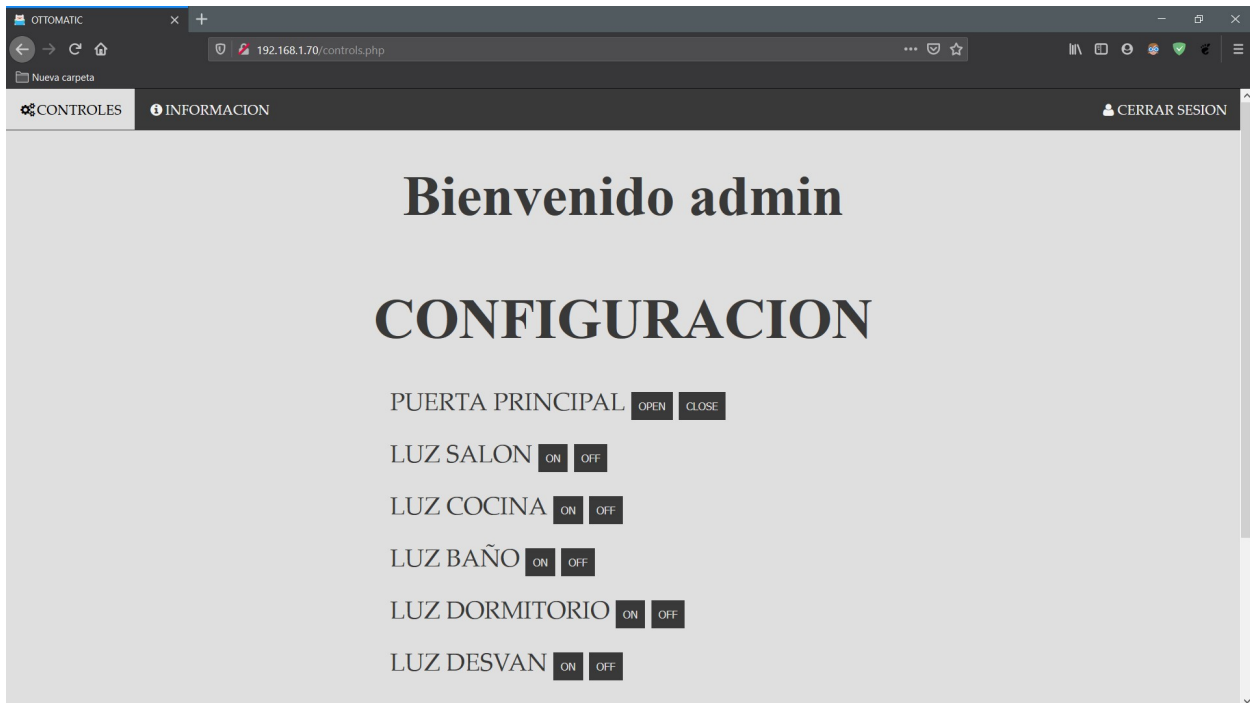
. Below the title is a login form with the following fields and labels:

- Usuario
- Contraseña

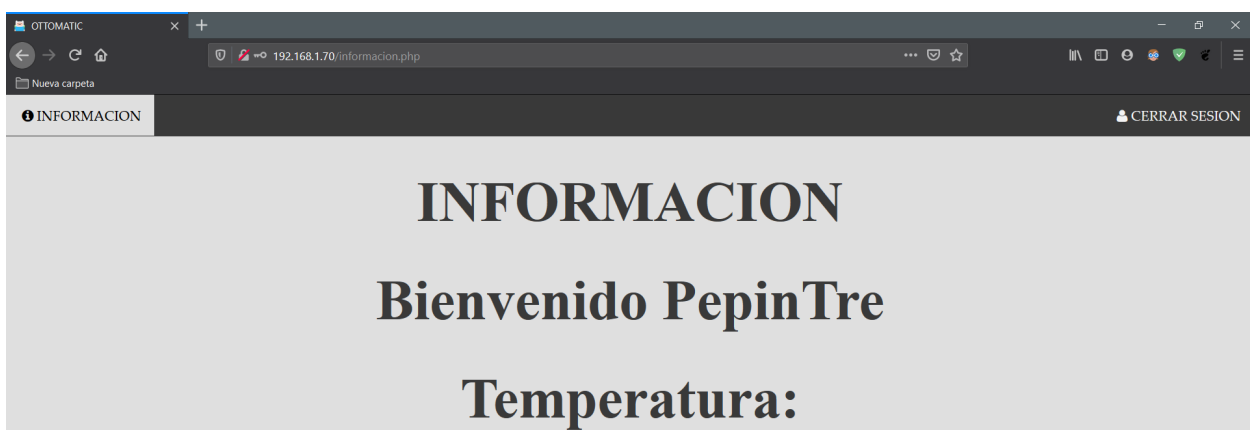
At the bottom of the form is a button labeled **ENTRAR**. The footer of the page contains the text:

Web made by Yago Fernández.
Contact us: [mail](#).
[Home](#)

Como Admin tendremos acceso a todos los controles y a la información correspondiente a nuestro hogar como temperatura o humedad.



Como otro usuario no tendremos acceso a los controles solo a la información.



Lo siguiente fue la creación de la base de datos de los logins para llevar un control de los usuarios y además la creación del usuario pi con permisos para gestionar toda esa información.

```

pi@raspberrypi:~$ mysql -u root -p
root@raspberrypi:/home/pi# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 57
Server version: 10.3.22-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create login;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'login' at line 1
MariaDB [(none)]> create database login;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| login      |
| mysql      |
| performance_schema |
+-----+
4 rows in set (0.001 sec)

MariaDB [(none)]> use login;
Database changed
MariaDB [login]> create table users
-> (
->   id_user INT NOT NULL auto_increment,
->   username VARCHAR(100) NOT NULL,
->   password VARCHAR(100) NOT NULL,
->   PRIMARY KEY (id_user));
Query OK, 0 rows affected (0.183 sec)

MariaDB [login]> select * from users;
Empty set (0.002 sec)

MariaDB [login]> INSERT INTO users (id_user,username,password)
-> VALUES (NULL,'admin','');
Query OK, 1 row affected (0.010 sec)

MariaDB [login]>

```

```

MariaDB [mysql]> create user pi identified by 'Toor.1234';
Query OK, 0 rows affected (0.001 sec)

MariaDB [mysql]> grant all privileges on *.* to pi with grant option;
Query OK, 0 rows affected (0.001 sec)

MariaDB [mysql]> flush privileges;
Query OK, 0 rows affected (0.001 sec)

MariaDB [mysql]> quit;
Bye
root@raspberrypi:/var/www/html# service apache2 restart
root@raspberrypi:/var/www/html# mysql -u pi -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 66
Server version: 10.3.22-MariaDB-0+deb10u1 Raspbian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| login      |
| mysql      |
| performance_schema |
+-----+
4 rows in set (0.001 sec)

MariaDB [(none)]> use mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MariaDB [mysql]> select user from user;
+-----+
| user |
+-----+
| pi    |
| phpmysqladmin |
| root |
+-----+

```

Desarrollo del proyecto

Para comunicar la base de datos con la web usamos `mysqli_connect`, un comando muy útil para php7.3. A continuación un ejemplo del código de registro de usuarios y de la conexión para el login.

Registro usuarios.

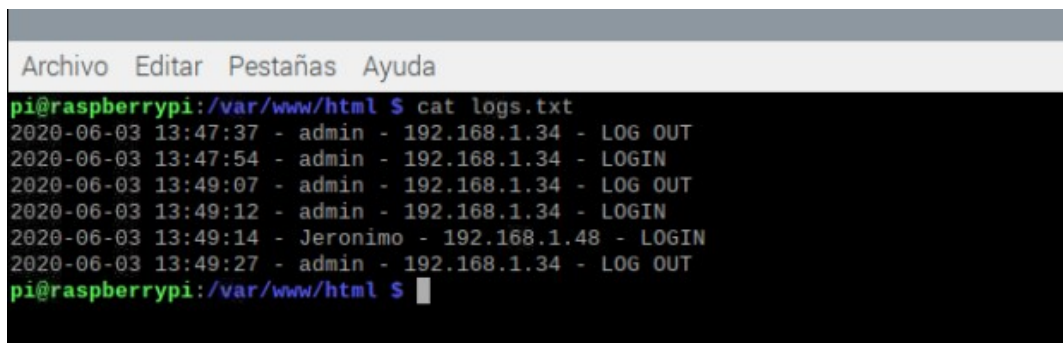
```
C:\Users\yagof\Desktop\Ottomatic\web\register\process_r.php - Sublime Text (UNREGISTERED)
< > process_r.php x
1 <?php
2 //cogemos los valores del login
3 $username = $_POST['user'];
4 $password = $_POST['pass'];
5 $password2 = $_POST['pass2'];
6
7
8 //conectamos a la base de datos
9 $enlace = mysqli_connect("localhost","pi","Toor.1234","login");
10
11 //consulta y comprobacion de usuarios
12 $username = mysqli_real_escape_string($enlace,$username);
13 $password = mysqli_real_escape_string($enlace,$password);
14 $password2 = mysqli_real_escape_string($enlace,$password2);
15 $result = mysqli_query($enlace,"select * from users where username = '$username'")
16 or die("No se pudo consultar la base de datos".mysqli_error());
17
18 $row = mysqli_fetch_array($result);
19 if ($row['username'] == $username) {
20     echo "NOMBRE DE USUARIO YA EXISTENTE";
21 } else {
22     if ($password != $password2) {
23         echo "LAS CONTRASEÑAS NO COINCIDEN";
24     } else {
25         mysqli_query($enlace,"insert into users (id_user,username,password) values (NULL,'$username','$password')");
26         echo "USUARIO CREADO CORRECTAMENTE";
27         header("Location:..\login\login.php");
28     }
29 }
30 //cerramos la conexion
31 mysqli_close($enlace);
32
33 ?>
```

Inicio de sesión

```
C:\Users\yagof\Desktop\Ottomatic\web\login\process.php - Sublime Text (UNREGISTERED)
< > process_r.php x process.php x
1 <?php
2 //cogemos los valores del login
3 $username = $_POST['user'];
4 $password = $_POST['pass'];
5
6
7 //conectamos a la base de datos
8 $enlace = mysqli_connect("localhost","pi","Toor.1234","login");
9
10 //consulta y comprobacion de usuarios
11 $username = mysqli_real_escape_string($enlace,$username);
12 $password = mysqli_real_escape_string($enlace,$password);
13 $result = mysqli_query($enlace,"select * from users where username = '$username' and password = '$password' ")
14 or die("No se pudo consultar la base de datos".mysqli_error());
15
16 $row = mysqli_fetch_array($result);
17 if ($row['username'] == $username && $row['password'] == $password ) {
18     session_start();
19     $_SESSION['username'] = $username;
20     $fichero = '/var/www/html/logs.txt';
21     $dataToLog = array (date("Y-m-d H:i:s"),$username,$_SERVER['REMOTE_ADDR'],'LOGIN');
22     $data = implode(" - ", $dataToLog);
23     $data .= PHP_EOL;
24     file_put_contents($fichero, $data, FILE_APPEND | LOCK_EX);
25     if ($username == "admin") {
26         header("Location:..\controls.php");
27     } else {
28         header("Location:..\informacion.php");
29     }
30 } else {
31     echo "ERROR CON EL USUARIO";
32 }
33 //cerramos la conexion
34 mysqli_close($enlace);
35
36
37 ?>
```

Lo siguiente por hacer es un gestor de Logs que lleve un control de quien entra y sale de la web con la información pertinente para eso se añade esta parte del código.

```
session_start();
$_SESSION['username'] = $username;
$fichero = '/var/www/html/logs.txt';
$dataToLog = array (date("Y-m-d H:i:s"),$username,$_SERVER['REMOTE_ADDR'],'LOGIN');
$data = implode(" - ",$dataToLog);
$data .= PHP_EOL;
file_put_contents($fichero, $data, FILE_APPEND | LOCK_EX);
```



```
Archivo  Editar  Pestañas  Ayuda
pi@raspberrypi:/var/www/html $ cat logs.txt
2020-06-03 13:47:37 - admin - 192.168.1.34 - LOG OUT
2020-06-03 13:47:54 - admin - 192.168.1.34 - LOGIN
2020-06-03 13:49:07 - admin - 192.168.1.34 - LOG OUT
2020-06-03 13:49:12 - admin - 192.168.1.34 - LOGIN
2020-06-03 13:49:14 - Jeronimo - 192.168.1.48 - LOGIN
2020-06-03 13:49:27 - admin - 192.168.1.34 - LOG OUT
pi@raspberrypi:/var/www/html $
```

Para completar el control de los usuarios dentro de la web hacemos uso de los `_SESSIONS`. Censurando el acceso dependiendo del tipo de sesión iniciada. Parte de código correspondiente a los controles.

```
<?php
    session_start();
    error_reporting(0);
    $varsesion = $_SESSION['username'];

    if ($varsesion == 'admin') {

    } else {
        echo 'Usted no tiene autorización';
        die();
    }
```

El siguiente paso es conectar la aplicación web con el Arduino que se encarga de la parte mecánica del proyecto.

Para ello lo primero es el código de Arduino el cual ejecutará las ordenes recibidas por la aplicación, a continuación se muestra el código correspondiente a la puerta y a la luz del salón.

```
door Arduino 1.8.12
Archivo Editar Programa Herramientas Ayuda

door

1 #include <Servo.h>
2
3 Servo servo;
4
5 int DOOR = 9;
6 char ESTADO = '0';
7 int ANGLE1 = 90; // servo posicion
8 int ANGLE2 = 0;
9
10 void setup() {
11   Serial.begin(9600);           //comunicacion
12   servo.attach(DOOR);           //door
13   servo.write(ANGLE1);
14 }
15 //
16 void loop() {
17   if (Serial.available()>0) {
18     ESTADO = Serial.read();
19     if (ESTADO == '1') {
20       servo.write(ANGLE2);
21     } else {
22       servo.write(ANGLE1);
23     }
24   }
25 }
```

```
livingroom Arduino 1.8.12
Archivo Editar Programa Herramientas Ayuda

livingroom

1 int LED = 3;
2 char ESTADO = '0';
3
4 void setup() {
5   Serial.begin(9600);           //comunicacion
6   pinMode(LED, OUTPUT);         //led
7 }
8
9 void loop() {
10  if (Serial.available()>0) {
11    ESTADO = Serial.read();
12    if (ESTADO == '1') {
13      digitalWrite(LED, LOW);
14    } else {
15      digitalWrite(LED, HIGH);
16    }
17  }
18 }
```


Para establecer la comunicación lo que sucederá es que desde el PHP se ejecutará un archivo python y este mandará información para que se ejecute el archivo Arduino correspondiente y responderá de la manera pertinente ejecutando la orden correcta.

```
if (isset($_POST['li_on']))
{
    exec('python /var/www/html/controls/livingroom/livingroom_on.py');
} elseif (isset($_POST['li_off'])) {
    exec('python /var/www/html/controls/livingroom/livingroom_off.py');
} elseif (isset($_POST['k_on'])) {
    exec('python /var/www/html/controls/kitchen/kitchen_on.py');
} elseif (isset($_POST['k_off'])) {
    exec('python /var/www/html/controls/kitchen/kitchen_off.py');
} elseif (isset($_POST['ba_on'])) {
    exec('python /var/www/html/controls/bathroom/bathroom_on.py');
} elseif (isset($_POST['ba_off'])) {
    exec('python /var/www/html/controls/bathroom/bathroom_off.py');
} elseif (isset($_POST['be_on'])) {
    exec('python /var/www/html/controls/bedroom/bedroom_on.py');
} elseif (isset($_POST['be_off'])) {
    exec('python /var/www/html/controls/bedroom/bedroom_off.py');
} elseif (isset($_POST['lo_on'])) {
```

Archivo correspondiente a door_open.py

```
import serial
import time
arduino = serial.Serial('/dev/ttyACM0', 9600)
time.sleep(2)
arduino.write(b'1')
arduino.close()
```

Con el comando “arduino upload” y el nombre del archivo subimos el archivo Arduino que queramos a nuestro Arduino UNO.

El siguiente paso ya sería el montaje final de la maqueta con todo el cableado, LEDs y servomotores de arduino.

Viabilidad técnico-económica

Viabilidad tecno-económica

El coste del proyecto no fue excesivamente caro, es cierto que la Raspberry no fue barata pero para el trabajo que realiza el precio no es desorbitado. Luego todo el material mecánico del Arduino es bastante económico incluso si no tienes prisa por recibir los elementos mecánicos como los servomotores o los motores DC puedes encontrarlos a precios muy económicos.

A la hora de patentar el proyecto para hacerlo utilitario al público creo que sería altamente rentable y una idea de negocio muy buena si a mayores se le añaden todos los proyectos explicados más tarde en el apartado de líneas de investigación.

Conclusiones

Conclusiones

Tras finalizar el proyecto puedo decir que el campo de la domótica es un tema con una capacidad de crecimiento muy grande y que de primeras puede asustarnos por el hecho de la cantidad de información que se podría obtener de nosotros.

Pero esta recolecta de información es casi inevitable en nuestro día a día en el cual estamos todo el tiempo conectado a aplicaciones de mensajería, redes sociales o juegos en el móvil que registran nuestra posición en todo momento. Es cierto que haciendo un pequeño esfuerzo podemos censurar esa información que se recoge de nosotros pero sino es por nosotros mismos esa información será también recogida por cercanos nuestros. Lo que quiero decir es que es un problema no ligado solo a la domótica sino a la sociedad y tecnologías actuales.

Gracias a este proyecto puedo decir que han mejorado mis habilidades de diseño web, programación y administración de bases de datos notablemente ya que me he visto obligado a si o si dominar estas secciones. Incluso gracias a todo esto he descubierto nuevas ramas de la tecnología que me interesan como la robótica.

Considero que ha sido un proyecto muy interesante y que aunque por momentos me he sentido bloqueado, una vez terminado y visto el resultado estoy orgulloso del trabajo. Puede que haya muchas cosas por pulir e incluso me gustaría continuar con él para añadir algunos detalles que se me han ido ocurriendo.

El trabajo con la Raspberry ha sido un lujo, siendo súper sencilla de utilizar y muy útil para muchas cuestiones.

En conclusión ha sido un proyecto muy interesante, que creo que engloba una idea con mucho futuro y además me ha servido para poner en práctica casi todos los conocimientos obtenidos en los últimos años.

Lineas abiertas de investigación

Lineas abiertas de investigación

Es un proyecto con una gran capacidad de expandirse, gracias en parte a su software libre y la gran cantidad de elementos compatibles con las tecnologías usadas.

Una de las principales lineas de investigación abiertas sería la recogida continua de información de los controles usados día a día y poder elaborar un perfil inteligente capaz de simular la habitabilidad de sus residentes aun cuando estos no estuviesen en casa haciéndola más segura y protegida.

Otra posible linea de investigación seria la portabilidad de la aplicación web a dispositivos móviles como una app.

También seria ideal la creación de una suscripción mensual a la hora de registrarse en la aplicación con un servicio personal para instalar y mantener en buen estado los dispositivos incorporados. Añadiendo a mayores un mercado par comprar diferentes módulos que añadir a nuestro hogar.

En cuanto al apartado de hardware existe un modulo del Arduino capaz de conectarse al servidor web a través de tecnología WiFi haciendo menos aparatoso el uso de cableado.

También se buscará acelerar la comunicación entre servidor y controlador físico de los elementos del hogar.

A través de PiCroft la creación de un asistente personal capaz de manejar los controles del hogar.

Bibliografía, referencias, índices y tablas

Bibliografía, referencias e índices

ARDUINO:

Arduino Official Tutorials → <https://www.arduino.cc/en/Tutorial/HomePage?from=Main.Tutorials>

Arduino Forum → <https://forum.arduino.cc/>

Makerspace → <https://www.makerspaces.com>

Circuito.io → <https://www.circuito.io/>

Luis Llamas → <https://www.luisllamas.es/controlar-arduino-con-python-y-la-libreria-pyserial/>

Aprendiendo Arduino → <https://aprendiendoarduino.wordpress.com/>

RASPBERRY:

Pedro Pablo Moral → <https://www.pedropablomoral.com/raspberrypi/>

RedesZone → <https://www.redeszone.net/raspberry-pi/>

GeekTheory → <https://geekytheory.com/>

PYTHON:

JavatPoint → <https://www.javatpoint.com/>

LearnPython → <https://www.learnpython.org/es/>

PHP:

PhpNet → <https://www.php.net/docs.php>

JavatPoint → <https://www.javatpoint.com/>

DUDAS VARIAS:

StackOverflow → <https://stackoverflow.com/>