

Software para Robots

Grupo B3

19 DICIEMBRE

Integrantes: Chen Xin Pan Wang UO276967
Mateo Rico Iglesias UO277172
Eduardo Blanco Bielsa UO285176



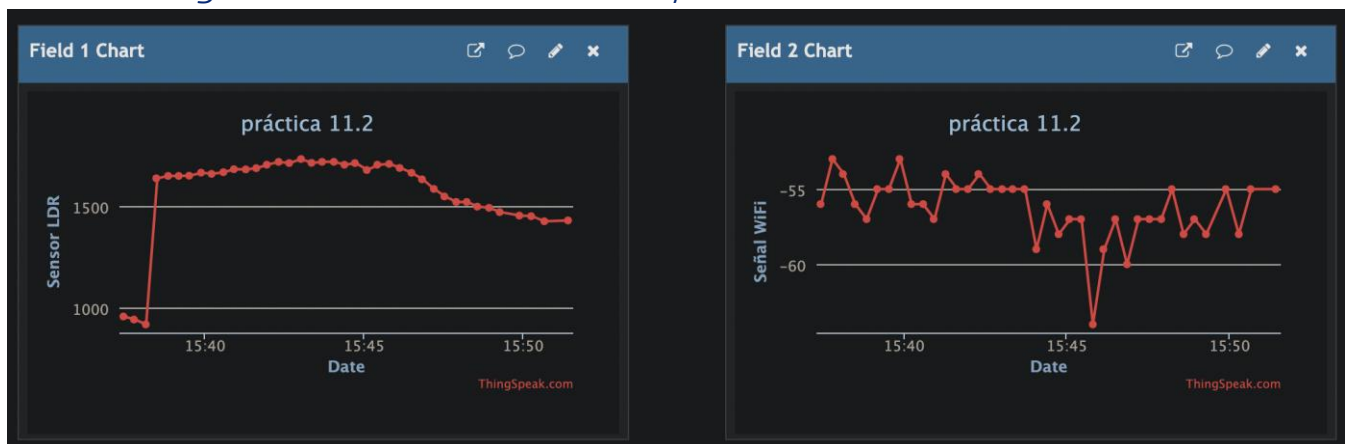
Práctica 1 1

(ThingSpeak 1 1.2) Hola Mundo en ThingSpeak (0,5 Punto) ☒

Proceso:

(Se ha trabajado con PPlatform.io en VSCode)

- Primero se buscó información sobre cómo conectar el LOLIN D32 por WIFI -> <https://www.upesy.com/blogs/tutorials/how-to-connect-wifi-access-point-with-esp32>
- Después se intentó conectar el Arduino al d32 para hacer que el Arduino obtuviese los datos del ldr y enviárselos al d32, pero después de profundas búsquedas, se averiguó que no comparten los mismos voltajes (5 y 3.3V respectivamente). Por tanto, se necesitaría un conversor entre medias. *NOTA: se encontró una supuesta manera de conectarlo, que es conectar ARDUINO—RX -> D32—TX, ARDUINO—TX -> D32—RX, D32 - GND (pero no funcionaba).*
- Después de varias comeduras de cabeza, se nos ocurrió mirar si el d32 tenía pines analógicos (y así realizar él todo el trabajo), y si tenía. Por tanto, hicimos que el d32 obtuviese directamente los datos y los enviase a la api de ThingSpeak, obteniendo dos gráficos: la **temperatura** (lo que se pedía), y el **tráfico de red** (para comprobar que el d32 seguía funcionando correctamente).



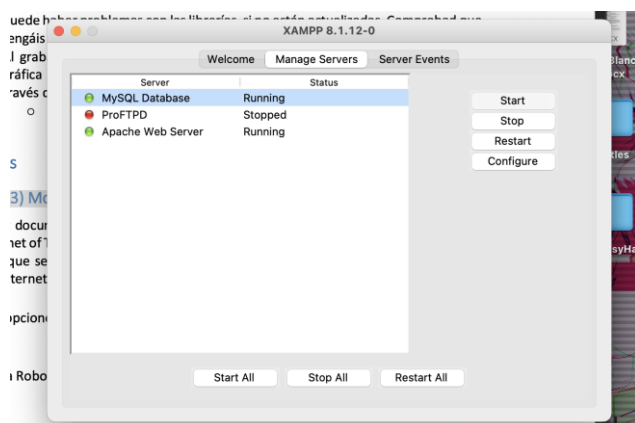
(gráfica real tomada)

- Para la funcionalidad de los Talkbacks, se ha configurado un led blanco externo al d32, conectado a un pin digital que lo activa, creando una petición POST a la API de ThingSpeak (en la que creamos un array de comandos de tipo TURN_ON y TURN_OFF). En función del comando que sea (pues se va analizando la cola de comandos), se enciende o se apaga el led (con un cooldown de 20 segundos para no sobrecargar la API)
- Se cogió de la API de Mathworks una función POST estándar para crear las peticiones

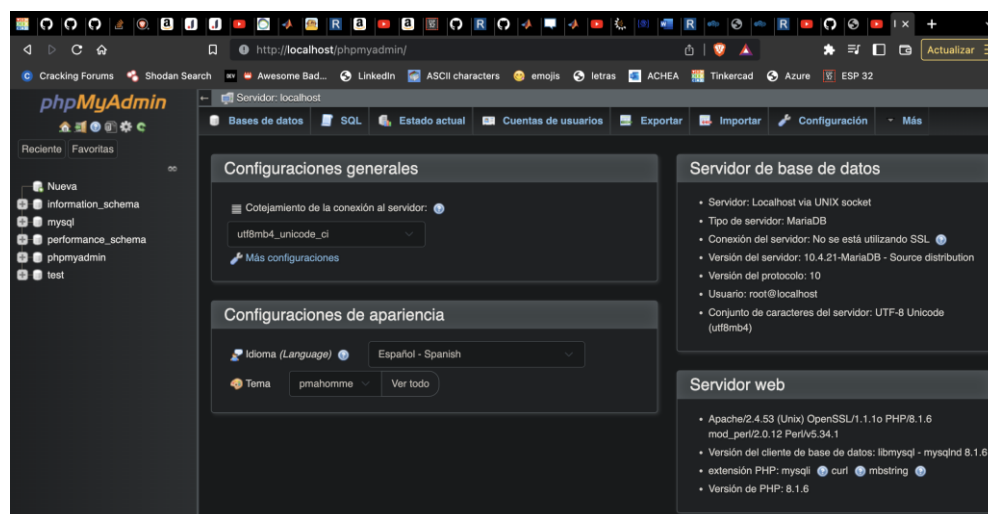
(IoTServer1 1.3) Montar Servidor IoT (1,5 puntos) ☒Hecho por Eduardo Blanco Bielsa

Nota: no se incluye vídeo de este ejercicio, pues sería muy extenso. A cambio, se ofrece una documentación detallada.

- Primero averigüé si se podía conectar el dht 11 a 3.3V. Lo comprobé en esta página y vi que sí se podía -> <http://www.esp32learning.com/code/esp32-and-dht11-sensor-example.php>
- Después revisé un proyecto que encontré por Internet que me sirvió como base e inspiración para crear el servidor de IOT -> <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>
- Decidí cambiar de idea, y en vez de accionar dos leds, mostrar la humedad y la temperatura del ambiente (así era más original).
- Para la base de datos, tuve que utilizar XAMPP -> <https://www.youtube.com/watch?v=fsQoA7NMhsU> y <https://microcontrollerslab.com/esp32-mysql-database-php/>
- Lo descargué y cree una base de datos MYSqL y un servidor apache



- Luego cree la base de datos dentro del panel de control de php:



Bases de datos

Crear base de datos

- Cree una tabla para los valores tomados por el dht 11

⚠ No se han encontrado tablas en la base de datos.

Crear nueva tabla

Nombre de la tabla:
 Número de columnas:

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar
2	temperatura	varchar(30)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar
3	humedad	varchar(30)	utf8mb4_general_ci		No	Ninguna			Cambiar Eliminar
4	created_at	timestamp			No	current_timestamp()			Cambiar Eliminar

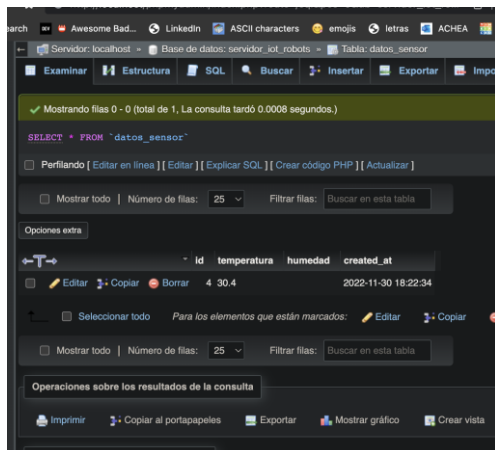
- Cré un script en php para conectar los datos del Arduino a la base de datos -> <https://microcontrollerslab.com/esp32-mysql-database-php/>

```

connect.php

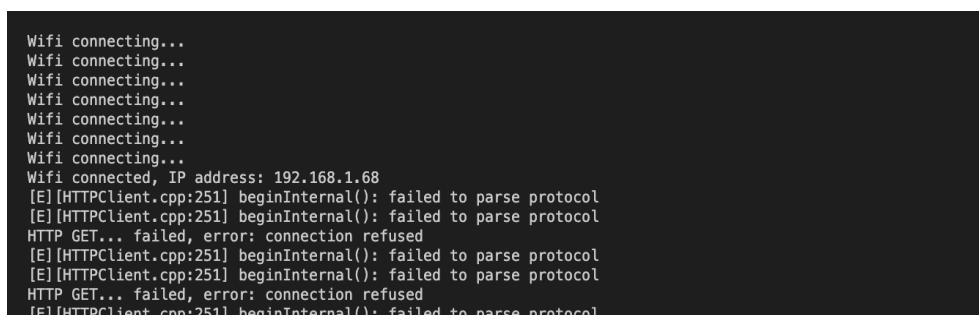
3
4 <?php
5
6 $dbname = 'servidor_iot_robots';
7 $dbuser = 'root';
8 $dbpass = '';
9 $dbhost = 'localhost';
10
11 $connect = @mysqli_connect($dbhost,$dbuser,$dbpass,$dbname);
12
13 if(!$connect){
14     echo "Error: " . mysqli_connect_error();
15     exit();
16 }
17
18 echo "Connection Success!<br><br>";
19
20 $temperatura = $_GET["temperatura"];
21 $humedad = $_GET["humedad"];
22
23
24 $query = "INSERT INTO datos_sensor(temperatura, humedad) VALUES ('$temperatura', '$humedad')";
25 $result = mysqli_query($connect,$query);
26
27 echo "Insertion Success!<br>";
28
29 ?>
30 </body>
31 </html>
  
```

- Probé que el script funcionase:

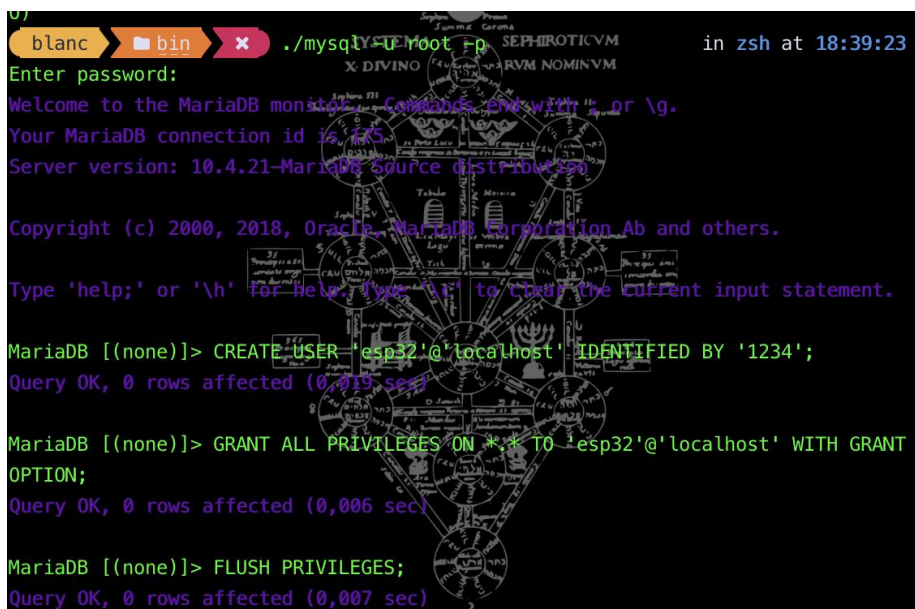


Todo correcto

- Me surgió un error de permisos:




- Para solucionarlo, tuve que crear un usuario en MariaDB con todos los privilegios:



- Se buguea bastante en mac, estuve bastante rato buscando una solución para relanzar los Daemon httpd y encontré esto ->
<https://stackoverflow.com/questions/4582504/xampp-apache-webserver-localhost-not-working-on-mac-os>
- Conseguí que funcionase, cambiando la uri de la petición que le enviamos a la base de datos (introduciendo la ip de mi mac (pues la base de datos es local))
- He aquí unas imágenes del diseño:

D32 Web Server







































Software para Robots 2022, Eduardo Blanco Bielsa

 Temperatura **22.70** °C

 Humedad **95.00** %

Detector a tiempo real

- He aquí la base de datos (el d32 escribe la temperatura, la humedad, y la fecha de inserción del dato (que inserta cada segundo)):

							id	temperatura	humedad	created_at
<input type="checkbox"/>		Editar		Copiar		Borrar	1	34.5	89	2022-11-30 23:15:38
<input type="checkbox"/>		Editar		Copiar		Borrar	2	21.2	94	2022-11-30 23:19:35
<input type="checkbox"/>		Editar		Copiar		Borrar	3	21.4	74	2022-11-30 23:20:38
<input type="checkbox"/>		Editar		Copiar		Borrar	4	21.4	74	2022-11-30 23:20:39
<input type="checkbox"/>		Editar		Copiar		Borrar	5	21.2	64	2022-11-30 23:20:40
<input type="checkbox"/>		Editar		Copiar		Borrar	6	21.2	64	2022-11-30 23:20:41
<input type="checkbox"/>		Editar		Copiar		Borrar	7	21.2	75	2022-11-30 23:20:43
<input type="checkbox"/>		Editar		Copiar		Borrar	8	21.2	75	2022-11-30 23:20:45
<input type="checkbox"/>		Editar		Copiar		Borrar	9	21.2	75	2022-11-30 23:20:46
<input type="checkbox"/>		Editar		Copiar		Borrar	10	21.2	75	2022-11-30 23:20:47
<input type="checkbox"/>		Editar		Copiar		Borrar	11	21.2	71	2022-11-30 23:20:49
<input type="checkbox"/>		Editar		Copiar		Borrar	12	21.2	71	2022-11-30 23:20:50
<input type="checkbox"/>		Editar		Copiar		Borrar	13	21.2	74	2022-11-30 23:20:51
<input type="checkbox"/>		Editar		Copiar		Borrar	14	21.2	74	2022-11-30 23:20:52
<input type="checkbox"/>		Editar		Copiar		Borrar	15	21.2	75	2022-11-30 23:20:53
<input type="checkbox"/>		Editar		Copiar		Borrar	16	21.2	75	2022-11-30 23:20:55
<input type="checkbox"/>		Editar		Copiar		Borrar	17	21.1	74	2022-11-30 23:20:56
<input type="checkbox"/>		Editar		Copiar		Borrar	18	21.1	74	2022-11-30 23:20:57

Anotaciones:

Toda actividad que lleve un ☒ es porque ha sido entregada correctamente.

Si se desea consultar, dejamos el enlace del correspondiente repositorio en github a través del cual fuimos trabajando durante el transcurso de la asignatura:

<https://github.com/gitblanc/SR>



Escuela de
Ingeniería
Informática
Universidad de Oviedo

