



Sistema FALTAS

Faltas del Alumnado con la Tecnología
AS608

Manual de usuario

Diseño de Sistemas Interactivos

Desarrolladores:

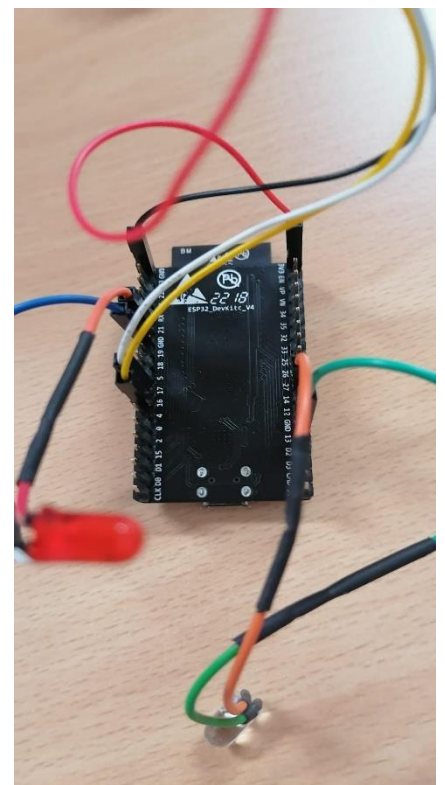
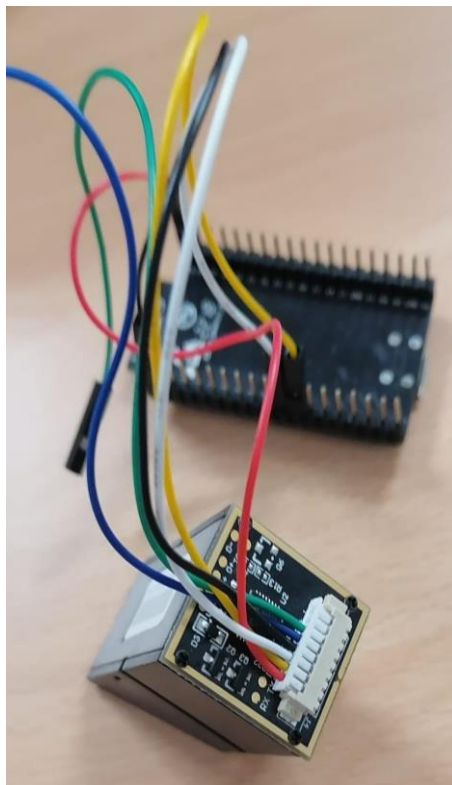
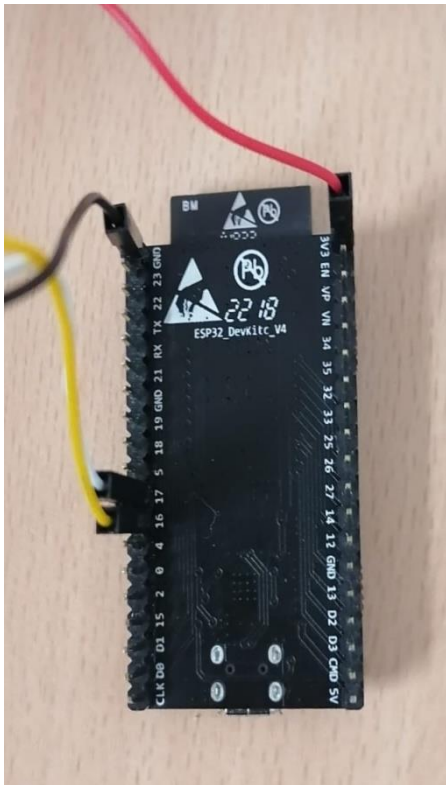
- José Alberto Sánchez-Camacho
- María Blanco González-Mohino
- Pablo Velasco Crespo

Fecha:

14 de diciembre de 2021

MANUAL DE USUARIO

Una vez descargados todos los archivos del proyecto lo que primero debemos hacer es conectar la *ESP32* al lector *AS608*, esto lo podemos encontrar en la documentación del *AS608* :



El led rojo debe ir en los pines 21 y GND; el pin amarillo debe ir en los pines 27 y GND.

Una vez realizadas las conexiones se debe cargar el archivo *registrar.ino* en la *ESP32* :



```
enroll_FPM Arduino 1.8.16
Archivo Editar Programa Herramientas Ayuda

enroll_FPM

#include <HardwareSerial.h>
#include <FPM.h>
#include <WiFi.h>
#include <WiFiMulti.h>
#include <HTTPClient.h>

HardwareSerial fserial(2);
FPM finger(sfserial);
FPM_System_Params params;
#define BUFF_SZ 768
uint8_t template_buffer[BUFF_SZ];
String DIR = "192.168.207.67";
WiFiMulti wifiMulti;
String huella = "";

void setup()
{
  Serial.begin(115200);
  fserial.begin(57600);
  wifiMulti.addAP("POCO", "Cacadevaca97");
  if (finger.begin()) {
    finger.readParams(&params);
    Serial.println("[ESP32] SENSOR ENCONTRADO");
  } else {
    Serial.println("[ESP32] SENSOR NO ENCONTRADO");
    while (1) yield();
  }
}

/* Use an Arduino if you want to connect the hardware module */

Subido

Leaving...
Hard resetting via RTS pin...

10 ESP32 Dev Module, Disabled, Default-4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, None en COM4
```

Después de cargar este archivo, debemos cargar la página web, archivo *index.php*. En él se deben rellenar los campos *Nombre*, *Apellidos* y *Curso* y darle a registrar, una vez le demos a registro la página nos mostrará cuando tenemos que quitar y poner el dedo para que lo lea (se tendrá que poner dos veces para que lo lea bien).



Registro

Nombre Alumno:

Primer Apellido:

Segundo Apellido:

Curso:

REGISTRAR RELOG

Ponga el dedo

Quitar dedo:

Registro

Nombre Alumno:

Primer Apellido:

Segundo Apellido:

Curso:

Quite el dedo

Una vez quitado el dedo la página pedirá volver a introducirlo. Una vez introducido la página mostrará si la huella se ha almacenado correctamente.

Registro

Nombre Alumno:

Primer Apellido:

Segundo Apellido:

Curso:

Huella almacenada

A nivel interno estos son los pasos que se realizarían hasta almacenar la huella en la base de datos.

[illegible]

Una vez almacenada la huella en la base de datos en la tabla *alumno* nos aparecerán los diferentes alumnos con su correspondiente nombre, curso y la huella registrada:

The screenshot shows the phpMyAdmin interface. The top navigation bar includes links for Examining, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Monitoring, and Triggers. The main content area displays a table named 'alumno' with the following columns: A_idAlumno, A_nombre, A_apellido, A_apellido2, A_idCurso, and A_puntajeExamen. The table contains 8 rows of data. The first row is highlighted, and the value in the A_puntajeExamen column (3,393.28) is enclosed in a red box. A red arrow points to this box with the text 'Huella de cada alumno'.

A_idAlumno	A_nombre	A_apellido	A_apellido2	A_idCurso	A_puntajeExamen
1	Jose Alberto	Seco	Sánchez-Camacho	1	3,393.28
2	Pablo	Velasco	Crespo	1	3,385.32
3	Maria	Blanco	González-Mohino	1	3,389.41
4	Andrea	Tarabuta	_	1	3,392.16
5	Lorenzo	Selas	Santos	1	3,3103.26
8	John	Doe	Smith	1	3,395.26

Una vez tenemos las diferentes huellas de cada alumno almacenadas en la base de datos es el momento de poner a funcionar nuestro controlador de faltas, para esto se deberá poner en funcionamiento nuestro controlador de la siguiente manera:

Debemos cargar en nuestra placa *ESP32* el archivo *autenticar.ino* el cual pasa la lista de los alumnos.

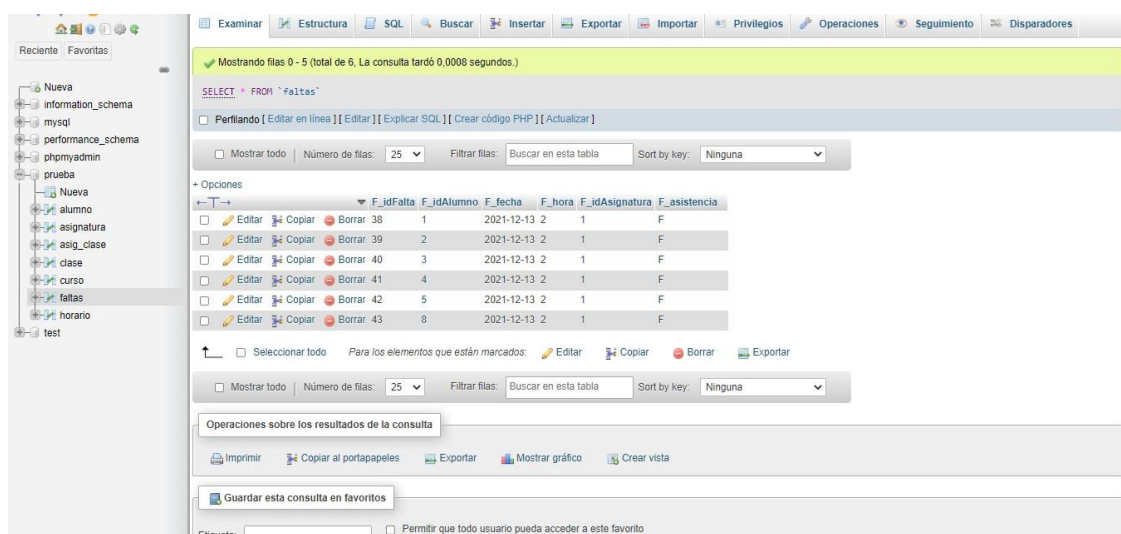
Nuestro controlador (*templateControler.py*) se iniciará de manera automática a primera hora de la mañana, este controlador implementado en *Python3* leerá los diferentes alumnos que tienen clase a una hora en específica y crea archivos tipo *.json* donde se almacenan el *idAlumno* y la *huella* de cada uno, si lo iniciásemos de manera manual nos encontraríamos algo parecido a esto:

```
C:\Windows\System32\cmd.exe

C:\xampp\htdocs>python templateControler.py
REGISTRO AÑADIDO -> idAlumno: 1, hora_clase: 2, idCurso: 1
REGISTRO AÑADIDO -> idAlumno: 2, hora_clase: 2, idCurso: 1
REGISTRO AÑADIDO -> idAlumno: 3, hora_clase: 2, idCurso: 1
REGISTRO AÑADIDO -> idAlumno: 4, hora_clase: 2, idCurso: 1
REGISTRO AÑADIDO -> idAlumno: 5, hora_clase: 2, idCurso: 1
REGISTRO AÑADIDO -> idAlumno: 8, hora_clase: 2, idCurso: 1

C:\xampp\htdocs>
```

Una vez cargado el archivo e iniciado nuestro controlador nuestra tabla de faltas se habrá iniciado con la asistencia del alumno en F hasta que un alumno llegase en horario de clase y pusiese el dedo en el correspondiente horario de asistencia.



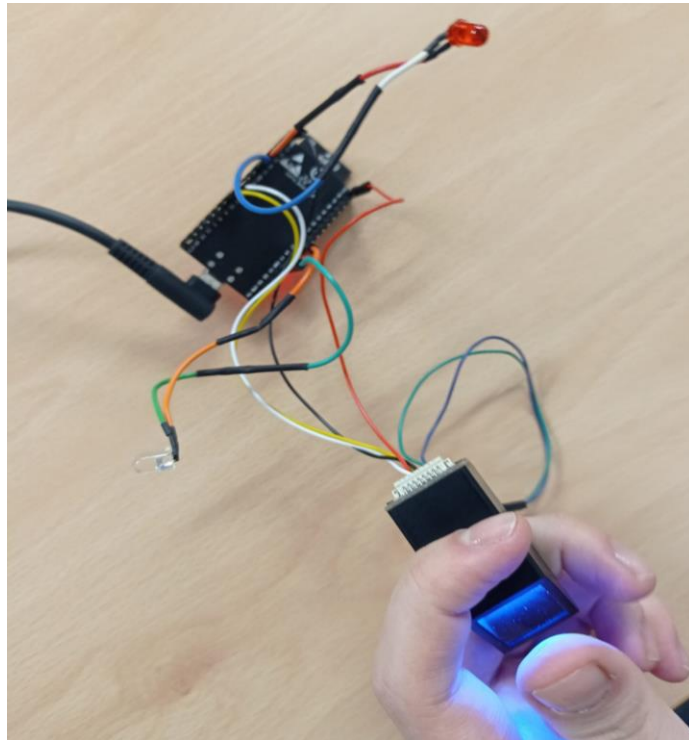
A continuación, la *ESP32* hace una lectura a los archivos *.json* almacenados en el servidor web y almacena las huellas en el lector *AS608* junto con los *idAlumnos* en un array para así identificar el *id* del lector con el *idAlumno*.

[illegible]

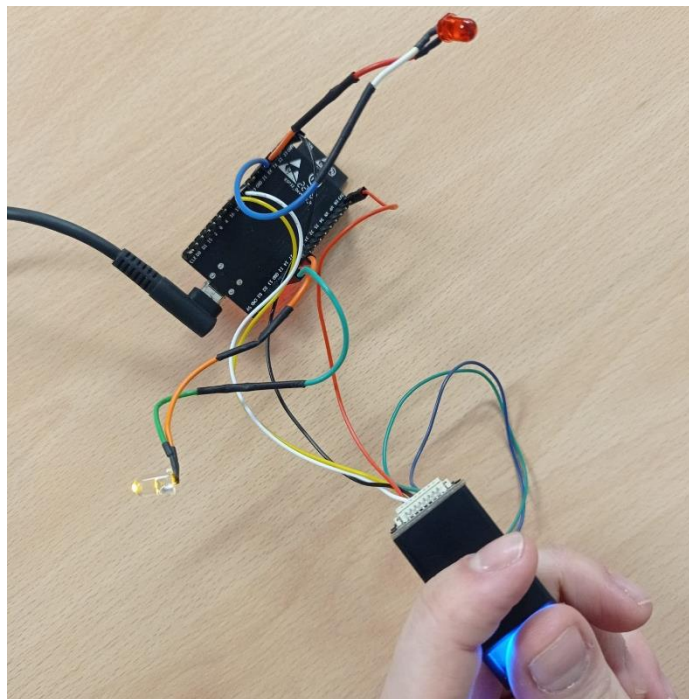
Una vez todo iniciado podemos detectar si un alumno pone el dedo, cuando el lector detecta un dedo lo compara con los diferentes *idAlumno* que hemos almacenado.

[illegible]

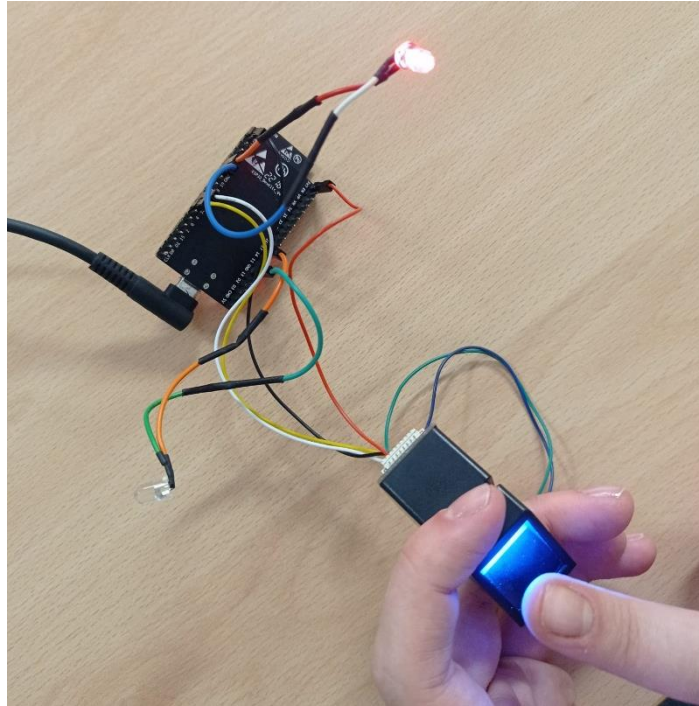
Visualización externa del funcionamiento:



Lector con los dos leds apagados esperando lectura.



Lector con led en amarillo como indicador de dedo detectado y almacenamiento de asistencia o retraso correcto.



Lector un led en rojo como señal de error

Cambio de configuración de la base de datos:

Si quisiésemos cambiar la configuración de la base de datos tendríamos que acceder al archivo *config.php* y modificar los diferentes datos con la información de nuestra base de datos.

```
1 <?php
2 // config.php
3 // Credenciales
4 $dbhost = "localhost";
5 $dbuser = "root";
6 $dbpass = "";
7 $dbname = "prueba";
8 // Conexión con la base de datos
9 $con = new mysqli($dbhost, $dbuser, $dbpass, $dbname);
10 ?>
```