GENERALITAT VALENCIANA
CONSELLERIA D'EDUCACIÓ, INVESTIGACIÓ, CULTURA I ESPORT

INSTITUT D'EDUCACIÓ SECUNDÀRIA
Tirant lo Blanc
GANDIA

instal·lacions automatismes informàtica telecomunicacions
electricitat i electrònica

Proyecto Final Ciclo Grado Superior

Alumno: Josep María Villalba Fernández

Tutor: Arturo Aparicio Gimeno

2CFSN

Unió Europea
Fons Social Europeu

Indice

Unió Europea
Fons Social Europeu

Introducción

El proyecto consiste en crear un sistema capaz de distinguir entre usuarios, y que tenga un proceso sencillo para agregar usuarios, acceder a un registro, y también poder modificar los usuarios de la propia base de datos.

- En este documento se contempla una explicación e información de los componentes utilizados en mi proyecto de final de curso, que estará basado en una placa ESP32, un lector de huellas, una pantalla LCD 2x16 y un lector de Tarjetas SD.

- También se hablará brevemente del proceso de funcionamiento del código creado.

- Fuentes utilizadas.

- y se adjunta una conclusión personal tanto del producto como el desarrollo del mismo código.

La elección de estos componentes ha sido hecha por mi tutor del proyecto, con la premisa de poder implementar sistemas de bajo consumo energético como es la placa ESP32, y a un bajo coste de fabricación, para este proyecto, se ha tenido en cuenta el uso de código basado en Arduino (C++).

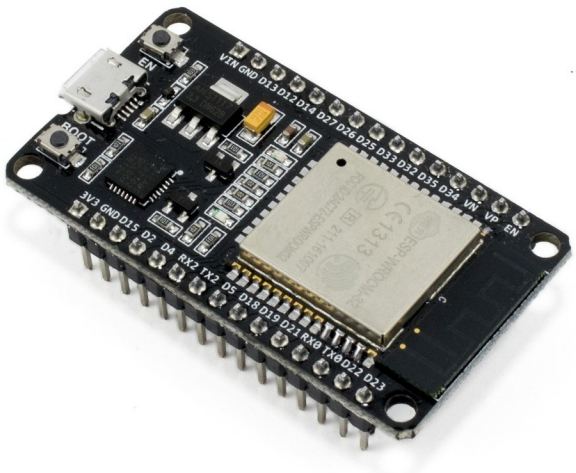
Unió Europea
Fons Social Europeu

Componentes

Placa esp32

La placa ESP32 es un módulo de microcontrolador altamente versátil y de bajo consumo de energía desarrollado por Espressif Systems. Esta placa destaca por su capacidad para integrar conectividad Wi-Fi y Bluetooth de manera nativa, lo que la convierte en una opción ideal para proyectos de IoT (Internet de las cosas) y aplicaciones de comunicación inalámbrica.

El ESP32 cuenta con un procesador dual-core Tensilica LX6 con una frecuencia de reloj de hasta 240 MHz, proporcionando un rendimiento considerablemente mayor en comparación con otros microcontroladores de su clase. Además, incluye 520 KB de memoria SRAM y hasta 16 MB de memoria flash externa, lo que permite almacenar y ejecutar programas más grandes y complejos.

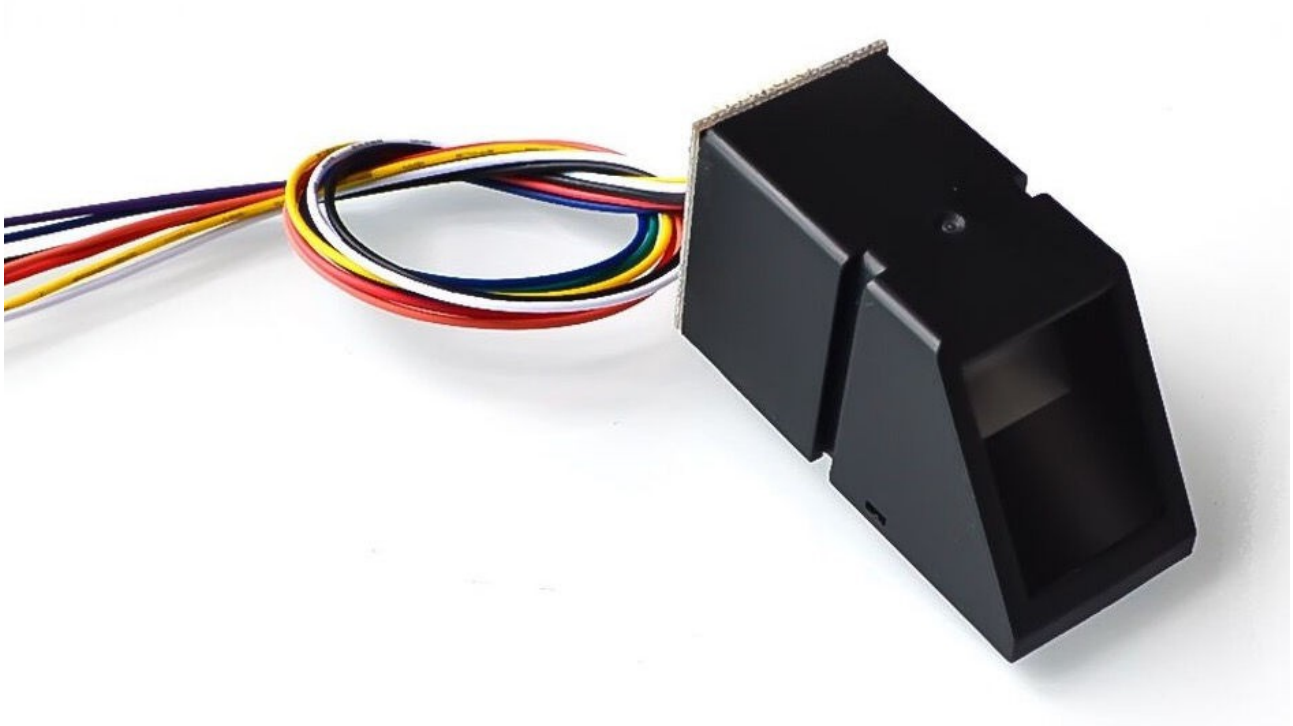


Unió Europea
Fons Social Europeu

Lector de huellas

El lector de huellas dactilares para Arduino es un módulo electrónico que permite la autenticación de usuarios mediante la biometría de sus huellas dactilares.

Se compone de un sensor óptico o capacitivo que captura la imagen de la huella digital y un procesador integrado que realiza la comparación y el análisis de las características únicas de la huella. El módulo también incluye una memoria no volátil para almacenar las posibles 127 huellas dactilares registradas.



Unió Europea
Fons Social Europeu

Pantalla LCD 2x16

Una pantalla LCD genérica 2x16 es un módulo de visualización alfanumérica que ofrece dos filas de 16 caracteres cada una, permitiendo mostrar un total de 32 caracteres. Este tipo de pantalla es comúnmente utilizado en proyectos electrónicos que necesitan una forma sencilla y económica de presentar información, datos o mensajes al usuario en tiempo real.

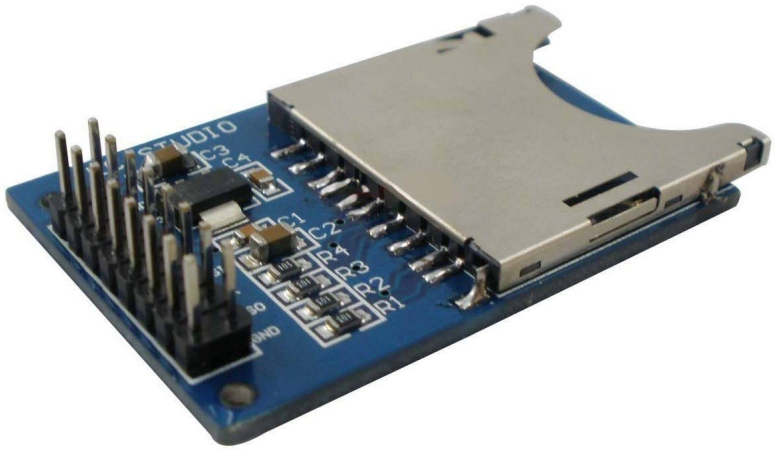
Esta conectará a una placa ESP32 a través del protocolo I2C, que es una interfaz de comunicación serie que requiere únicamente dos pines: uno para el reloj (SCL) y otro para los datos (SDA). Esta conexión simplifica el cableado y ahorra pines en comparación con las interfaces paralelas, facilitando su integración en proyectos con múltiples componentes.

Unió Europea
Fons Social Europeu

Lector de tarjetas SD

El lector de tarjetas SD es un módulo que permite al usuario disponer de un acceso a una tarjeta SD (Secure Data) y además interactuar con sus archivos mediante una librería.

Se conectará a la placa Arduino mediante el protocolo SPI (Serial Peripheral Interface), una interfaz de comunicación serie que utiliza cuatro pines: MOSI (Master Out Slave In), MISO (Master In Slave Out), SCK (Serial Clock) y CS (Chip Select). Estos pines permiten la transmisión de datos entre el Arduino y la tarjeta SD, así como el control de la selección de dispositivos en caso de que haya múltiples módulos SPI conectados al mismo bus.



Unió Europea
Fons Social Europeu

Ordenador

He usado mi propio ordenador portátil, pero cualquier ordenador capaz de mover el IDE (Entorno de desarrollo de Arduino) es también totalmente valido, por motivos de comodidad y compatibilidad, es preferible usar un ordenador que disponga de puertos USB normales directamente, debido a que usar adaptadores u otros, puede dar problemas en los puertos COM de Arduino, y por ende, problemas con las funciones Serial.

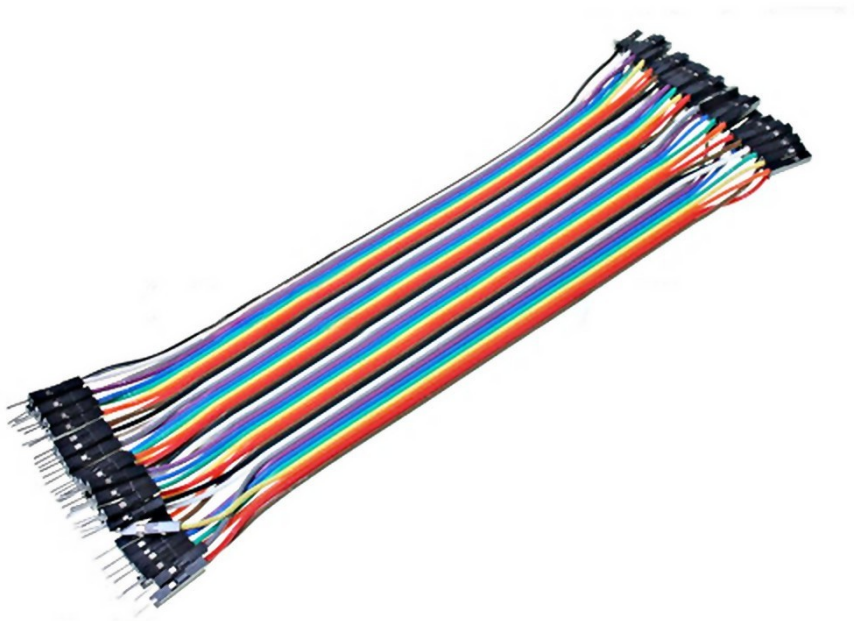
El entorno de desarrollo de Arduino que he usado es la versión 1.8.19.

Componentes adicionales

Cables Pin

Los cables pin, son simples conectores de cobre, normalmente ya preparados para usar (plug-and-play) , vienen soldados internamente a 2 conectores, siendo Macho u Hembra, según el modelo.

Estos cables son usados para conectarse en una PlacaBoard para un conexionado sencillo de montar y modificar, vienen en varios colores para una identificación más sencilla.



Unió Europea
Fons Social Europeu

PlacaBoard

La placa board es un elemento excelente para montar sistemas y circuitos sencillos, y que visualmente son muy intuitivos, en mi caso he usado 2 dado que por temas de diseño, la placa ESP32 no era compatible con solo 1

Esta compuesta principalmente por plástico endurecido que es aislante eléctricamente, y internamente tiene pistas de un metal conductor para poder crear tus ‘’empalmes’’ con los cables pin.

Unió Europea
Fons Social Europeu

Cables USB

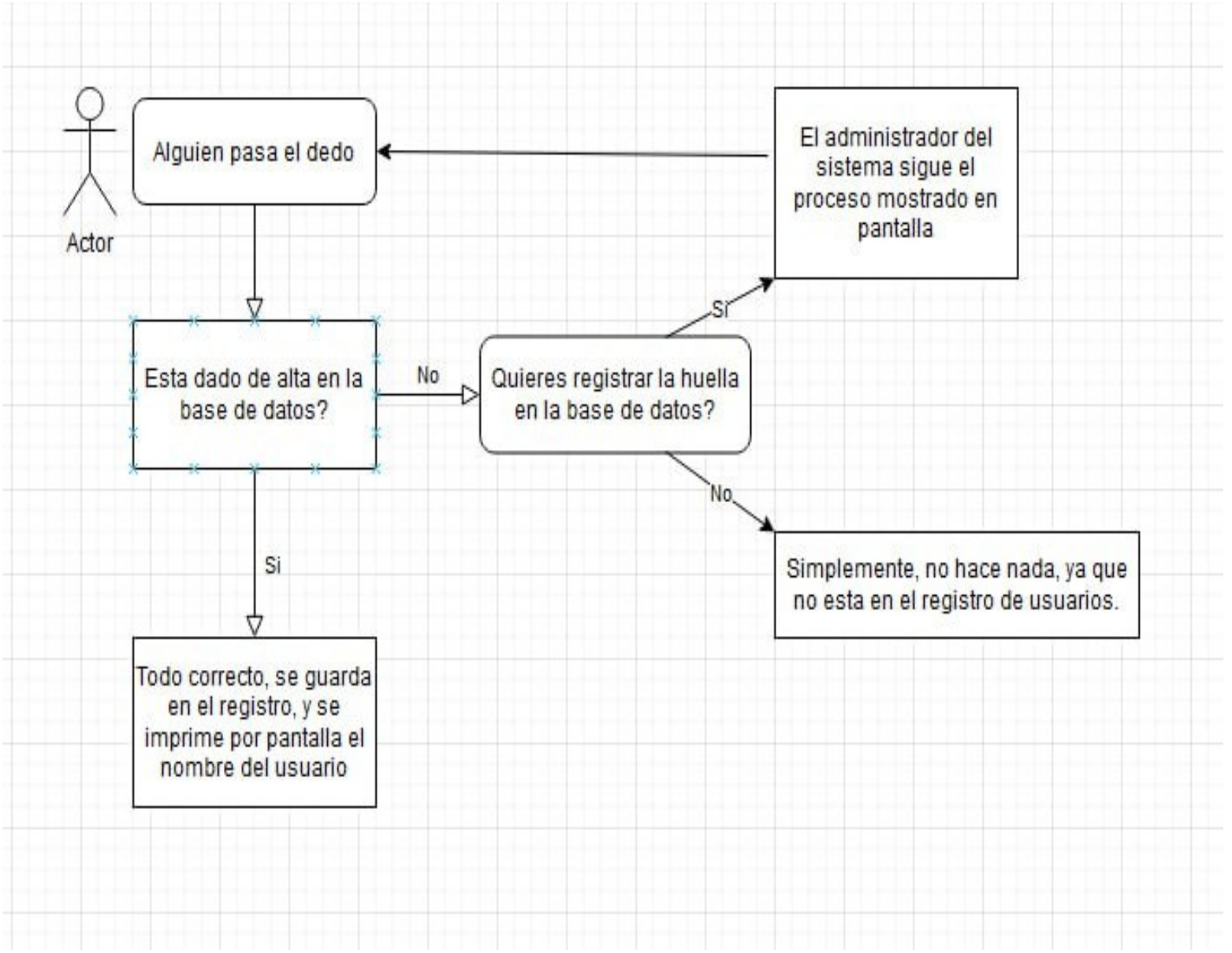
El cable USB a micro USB es un tipo de cable que permite la conexión y transferencia de datos entre dispositivos electrónicos y/o la carga de baterías.

En un extremo del cable, cuenta con un conector USB estándar (USB-A). En el otro extremo, tiene un conector micro USB (micro USB-B), más pequeño y delgado.

Idea general

Mi proyecto tiene como idea principal el uso de un lector de huellas, conectado a una placa ESP32 que funcionará como un pequeño control de acceso y así tener un control el acceso a una empresa de manera segura.

Para ello, he desarrollado un programa que sigue este Diagrama de elecciones



Flujo de datos y funciones

Inicio y Setup

En un principio lo que hace el programa es cargar las librerías necesarias, aparte de conectarse al wifi, lo he programado para que sea un proceso auto-bloqueante hasta que acceda a una wifi, el código esta pensado para que solo tengas que des-comentar las credenciales de wi-fi necesarias.

Los primeros pasos del programa al ejecutarse, es comprobar las conexiones, como ver que IP tiene al acceder a internet, comprobar si hay una SD y demás componentes, en caso de que alguno falle o se desconecte, mandaría una mensaje de error.

Una vez hecho todos los ‘’preparativos’’ entra en modo espera en el loop, esperando una de estas acciones*:

- A que alguien pase un dedo.
- Alguien desde un terminal escribe ‘Menu’.

*Comentario importante: a pesar de que el programa es auto-suficiente sin interacciones exteriores, se recomienda usar un puerto COM desde un terminal Arduino, para poder aprovechar más funcionalidades.

Unió Europea
Fons Social Europeu

Loop

Si pasas el dedo pueden suceder 2 cosas:

Caso 1. Si estas registrado y en la base de datos (Archivo de la SD para tu nombre);

- El lector lee tu huella.

- La compara con la base de datos interna del modulo.

- Si estas registrado, simplemente buscara tu ID asignado (no confundir con el nombre) y lo comprueba en la base de datos de ID, de ahi rescatará tu nombre, y lo mostrará en pantalla dando a entender que puedes pasar.

Caso 2. No estas registrado;

- La pantalla LCD mostrará como acceso no valido.

- El monitor serie, pedirá al administrador del sistema, si quiere registrar la huella.

- Pones la contraseña que este asignada al administrador del sistema (1234) en mi caso, y tienes un intervalo de 10 segundos y 1 intento para ponerla.

- Si es correcto, simplemente, se llamará la función de registro, donde asignas una ID a la huella, y sigues el proceso en pantalla

En cualquier momento, el administrador del sistema puede escribir la palabra Menu por comandos en el monitor serie, y gracias a eso se ejecutara un sencillo menú (bloqueante de procesos), que ahí puedes asignar el ID de la huella, a un nombre, que esta información se guardará en el archivo SD.

La codificación es la siguiente en el archivo de la SD:

‘’XXX_nombre’’ que esto equivale a por ejemplo: 001_Pepito.

Codigo

Aquí se explicará lo más importante y se va generalizar sobre el código, por que el código en su totalidad, será adjuntado con este documento.

En esta primera imagen, se pueden ver las librerías usadas para el proyecto, entre las que se destacan las de Adafruit para el lector, y la LiquidCrystal I2C para la pantalla.

```
//Añadimos las librerias de la SD
#include <SPI.h>
#include <SD.h>

#include <Wire.h>
#include <WireSlave.h>

#include <Adafruit_Fingerprint.h>

//Añadimos las librerias que usaran i2C para la pantalla LCD
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);

//añadimos la librerias del Wi-fi
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>

#include <HardwareSerial.h>
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial2);
```


Unió Europea
Fons Social Europeu

En el Setup, vamos a establecer los inicios de la pantalla LCD, una pequeña comprobación de la SD (saber que esta conectada si no, se bloqueará el programa) , y también empezará el proceso de conectarse al Wi-fi para obtener la Hora y fechas, y nos mostrará por Serial la IP y demás información, que sirve para depurar posibles errores en la conexión.

```
void setup()
{
  Serial.begin(9600);

  lcd.init();                      // initialize the lcd
  lcd.backlight();
  lcd.setCursor(3,0);
  lcd.print("Iniciandome");



  //Conectamos con el lector micro SD. Si no conecta, el Esclavo no respoderá
  if(!SD.begin(SD_CS))
      {
        Serial.println("Card Mount Failed");
        while(true){};
      }
  delay(1000); //este delay si se borra, peta.
  finger.begin(57600);
  delay(100);

  WiFi.begin(ssid, passwordwifi);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi conectado");
  Serial.print("Dirección IP: ");
  Serial.println(WiFi.localIP());

  timeClient.begin();

}
```

Con este código, podemos mostrar por la pantalla (Serial y LCD) la hora actual.

Principalmente, para no colapsar el sistema, se usa la función de Millis() de Arduino, para tener un contador entre ciclos, lo cual, nos refrescará la información mostrada.

```
unsigned long currentMillis = millis(); // Obtiene el tiempo actual en milisegundos

  if (currentMillis - previousMillis >= interval)
  {
    // Guarda el tiempo actual
    previousMillis = currentMillis;
    timeClient.update();

    // Obtención de la hora actual en formato UNIX
    unsigned long epochTime = timeClient.getEpochTime();

    // Verifica si el tiempo epoch es válido antes de continuar
    if (epochTime > 1500000000)
      { // Comprueba si la fecha es válida (después del 14 de julio de 2017)
        // Conversión de la hora UNIX a una estructura de tiempo en C
        struct tm *timeinfo;
        timeinfo = localtime((time_t *)&epochTime);

        // Conversión de la estructura de tiempo a una variable de tipo "char" en el formato deseado
        char fechaHora[17];
        strftime(fechaHora, sizeof(fechaHora), "%d-%m-%Y %H:%M", timeinfo);

        // Impresión de la fecha y la hora actual
        imprimirMensaje(fechaHora);
      }
    else
      {
        Serial.println("Sincronizando...");
        Serial.print("Servidor NTP: ");
        Serial.println(ntpServer);
        delay(1000);
      }
  }
```

Aquí la función usada para la pantalla LCD y Serial.

```
void imprimirMensaje(const char *mensaje)
    {
      Serial.println(mensaje);
      lcd.clear();
      lcd.print(mensaje);
    }
```

Este es el código del LOOP que principalmente, se dedica a comparar los dedos, y estará comprobando si hay una huella, el intervalo entre ciclos tiene un delay de 50, (un valor más alto hace que falle, el detector del dedo, y un valor más pequeño haría que se ejecute demasiado rápido los procesos)

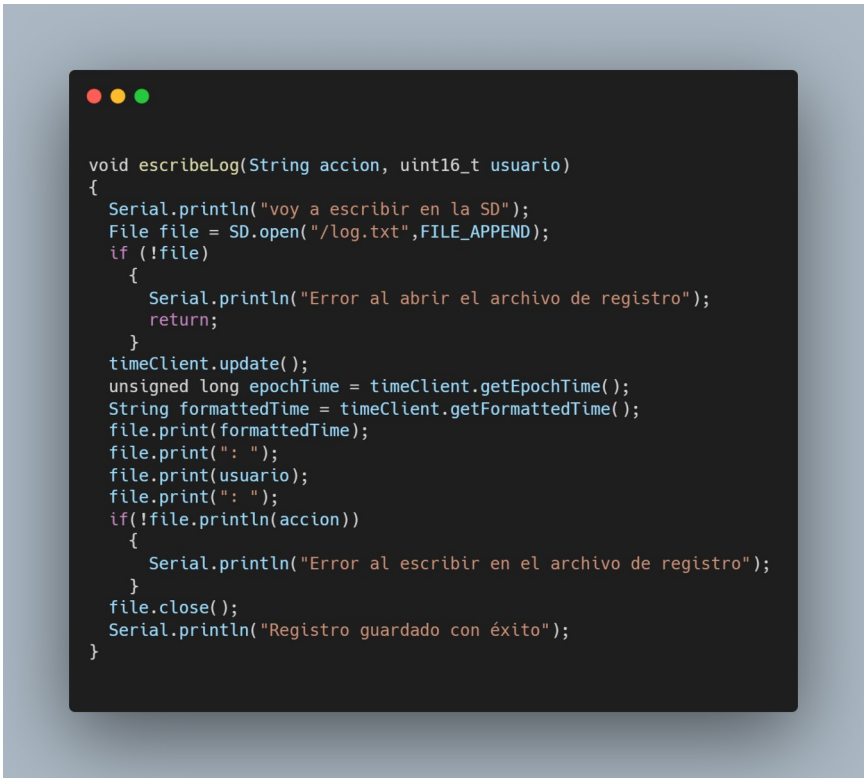
```
uint8_t result = finger.getImage();
  if (result == FINGERPRINT_OK)
  {
    Serial.println("Imagen capturada");
    result = finger.image2Tz();

    if (result == FINGERPRINT_OK)
    {
      result = finger.fingerFastSearch();
      if (result == FINGERPRINT_OK)         // La huella coincide con una huella guardada
      {

        char mensaje[50];
        snprintf(mensaje, sizeof(mensaje), "Acceso Valido");
        imprimirMensaje(mensaje);

        String y = String(finger.fingerID);
        while(y.length() < 3)
          {
            y = "0" + y;
          }
        Serial.println(y); //Aqui vemos el ID en pantalla codificado en XXX

        String nombre = buscarNombrePorID(y);
          if (nombre != "")
          {
            Serial.print("Nombre encontrado: ");
            Serial.println(nombre);

            // Muestra el mensaje en la pantalla LCD
            lcd.clear(); // Limpia la pantalla
            lcd.setCursor(0, 0); // Coloca el cursor en la primera columna de la primera fila
            lcd.print("Bienvenido ");
            lcd.setCursor(0, 1); // Coloca el cursor en la primera columna de la segunda fila
            lcd.print(nombre);
          }
          else
          {
            Serial.println("ID no encontrada.");
          }


        escribeLog("Entro",finger.fingerID);
      }
      else if (result == FINGERPRINT_NOTFOUND)     // La huella no coincide con ninguna huella guardada
      {

      requestPassword(); // Solicita la contraseña
      }
      else
      {
        imprimirMensaje("Error al buscar la huella.");
      }
    }
  }
  else if (result == FINGERPRINT_NOFINGER)
    {
      // No hacer nada si no hay dedo
    }
  else if (result == FINGERPRINT_PACKETRECIEVEERR)
    {
      imprimirMensaje("Error de comunicación");
    }
  else if (result == FINGERPRINT_IMAGEFAIL)
    {
      imprimirMensaje("Error al capturar la imagen");
    }
  else
    {
      imprimirMensaje("Error desconocido");
    }

  delay(50); // Puedes ajustar este delay para controlar la frecuencia de lectura del sensor
```


Unió Europea
Fons Social Europeu

Funciones

Ahora comentaré algunas funciones usadas para poder hacer algunos procesos.

Esta función cumple el rol de principalmente crear un registro secuencial en orden, con los parámetros que le pases.



```
void escribeLog(String accion, uint16_t usuario)
{
  Serial.println("voy a escribir en la SD");
  File file = SD.open("/log.txt",FILE_APPEND);
  if (!file)
    {
      Serial.println("Error al abrir el archivo de registro");
      return;
    }
  timeClient.update();
  unsigned long epochTime = timeClient.getEpochTime();
  String formattedTime = timeClient.getFormattedTime();
  file.print(formattedTime);
  file.print(": ");
  file.print(usuario);
  file.print(": ");
  if(!file.println(accion))
    {
      Serial.println("Error al escribir en el archivo de registro");
    }
  file.close();
  Serial.println("Registro guardado con éxito");
}
```

Función sencilla, que permite enviar el mismo mensaje, por Serial y por LCD a la vez

```
void imprimirMensaje(const char *mensaje)
    {
      Serial.println(mensaje);
      lcd.clear();
      lcd.print(mensaje);
    }
```


Unió Europea
Fons Social Europeu

Esta es posiblemente de las funciones que más problemas me ha dado, debido al funcionamiento del Serial con el ESP32, esta función permite crear un bloqueante de procesos, uno con temporizador para la contraseña, y otro sin tiempo, para escribir.

```
void requestPassword()
{
  const unsigned long passwordTimeout = 10000; // Tiempo límite para ingresar la contraseña en
milisegundos
  unsigned long startTime = millis();
  bool isTimedOut = false;
  int attempts = 0;
  const int maxAttempts = 1;


  Serial.println("Huella no valida");
  Serial.println("Introducir contraseña");
  while (!isTimedOut && attempts < maxAttempts)
  {
    if (Serial.available() > 0)
    {
      String inputPassword = Serial.readStringUntil('\n');
      inputPassword.trim();
      if (inputPassword == password) // La contraseña es correcta, agregar la nueva huella
      {

        Serial.println("Que ID?");
        id = readnumber();
          if (id == 0)
            {// ID #0 not allowed, try again!
               return;
            }
          char mensaje[50];
          snprintf(mensaje, sizeof(mensaje), "añadiendolo a la ID: %d", id);
          Serial.println(mensaje);
          Serial.print("Enrolling ID #");
          Serial.println(id);
          Serial.flush();

          //-------------------------------------proceso de añadirlo a la SD/nombres------------------
          String y = String(id);
          while(y.length() < 3)
          {
            y = "0" + y;
          }


          String nombre;
          Serial.println("Ahora escribe el nombre: ");
            while (!Serial.available())
            {
              delay(100); // Delay para evitar un bucle rápido de ejecución.
            }
            nombre = Serial.readStringUntil('\n');
            nombre.trim(); // Limpia cualquier espacio en blanco antes y después del nombre
            Serial.println(nombre);
            escribenombres(y, nombre);



        while (!  getFingerprintEnroll() );
        break;
      }
      else
      {
        Serial.println("Contraseña incorrecta.");

        attempts++;
      }
    }


    if (millis() - startTime >= passwordTimeout)// Comprueba si ha pasado el tiempo límite
    {
      isTimedOut = true;
    }
  }

  if (isTimedOut)
  {
    Serial.println("Tiempo Excedido");
  }
  else if
  (attempts >= maxAttempts)
  {
    Serial.println("Intentos Excedidos");;
  }
}
```

Con esta función se puede rescatar el nombre del archivo SD, por su Identificador codificado de manera: “ XXX_Nombre “

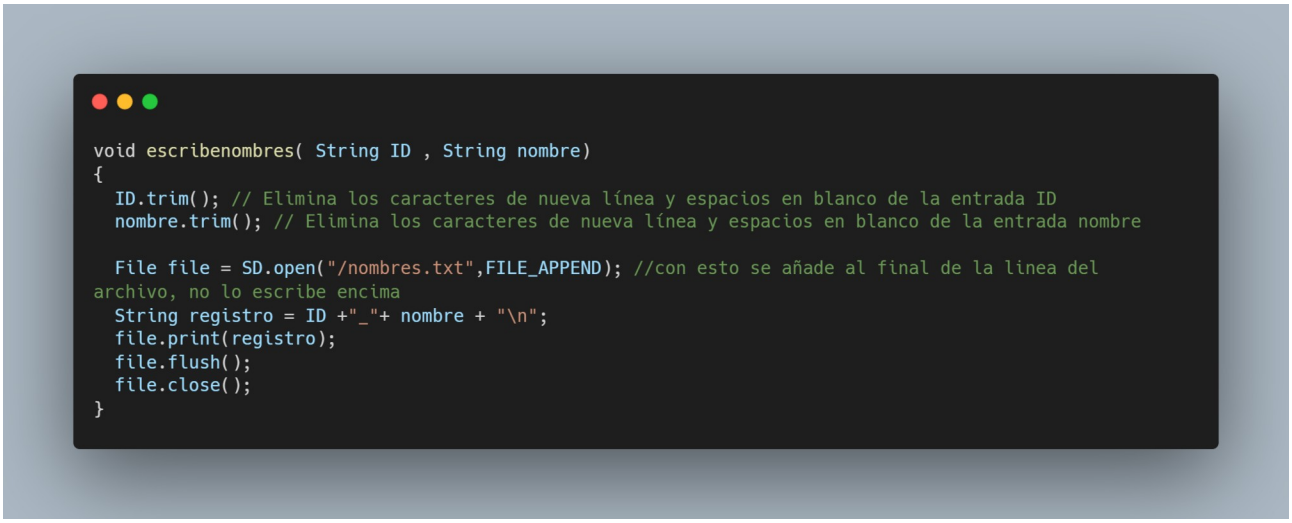
```
String buscarNombrePorID(String ID)
{
  File file = SD.open("/nombres.txt", FILE_READ);
  if (file) {
    while (file.available()) {
      String registro = file.readStringUntil('\n');
      int indiceGuion = registro.indexOf('_');
      String idActual = registro.substring(0, indiceGuion);
      if (idActual == ID) {
        file.close();
        return registro.substring(indiceGuion + 1, registro.length());
      }
    }
    file.close();
  }
  return "";
}
```

Esta función permite codificar el nombre en el archivo nombres.txt de la SD , hay que recordar que se debe codificar de manera: “ XXX_nombre “

por ejemplo:

001_Pepito

002_Fulanito



```
void escribenombres( String ID , String nombre)
{
  ID.trim(); // Elimina los caracteres de nueva línea y espacios en blanco de la entrada ID
  nombre.trim(); // Elimina los caracteres de nueva línea y espacios en blanco de la entrada nombre

  File file = SD.open("/nombres.txt",FILE_APPEND); //con esto se añade al final de la linea del
archivo, no lo escribe encima
  String registro = ID +"_"+ nombre + "\n";
  file.print(registro);
  file.flush();
  file.close();
}
```

Conclusión

Mi opinión más sincera y veraz, es que es un proyecto muy útil de cara a un planteamiento de un proceso, con este proyecto, he podido observar que sistema de procesos es más sencillo de cara al usuario.

¿Cómo deducimos que es más rápido y sencillo para el usuario?

Pondré mi ejemplo: en el desarrollo de este proyecto, se tuvo que plantear quitar el menú inicial, y con ello, llegar a un punto más sencillo y comprensible de cara a un usuario que solo usará el sistema sin necesitar saber que hacer.

En versiones anteriores, el proceso inicial era llamar un menú en cada ciclo, y que el usuario tuviera que seleccionar la opción deseada (registrar, entrar con su huella, etc) este proceso daba algunos problemas como bloquear el programa muy continuamente, y era poco practico, se descartó la idea, y se siguió el consejo del tutor en cuestión de simplificar todo el proceso.

El resumen de esta experiencia, es que puedo sentir personalmente orgullo de crear algo funcional y que en términos generales, podría considerarse muy sencillo, pero considero que lo sencillo no es malo, ya que algo tan sencillo como un palo con una piedra, nos diferencia de los animales.

Posibles mejoras

Una mejora que se podría implementar en este proyecto, sería el método de guardado de información de los usuarios, cambiando el sistema de la SD, por una base de datos MySQL alojada principalmente en un servidor interno de apache.

Para comunicar el sistema de nuestra ESP32 con la base de datos, lo más sencillo sería hacer un sistema que se comunique C++ → Python → MySQL

Las bases de datos de SQL son muy sencillas de usar y sobretodo tienes la información detallada en formato tabla normalmente y cada linea de información esta identificada por un número secuencial incremental, que se le llama KEY , con eso te aseguras que si hay 2 usuarios con el mismo nombre, realmente estarán separados 100% por una key distinta.

Unió Europea
Fons Social Europeu

Presupuesto

Este presupuesto contempla los precios encontrados a fecha de escritura de este documento, es posible que los precios cambien con el tiempo.

Componente	Precio Ud
Placa ESP32	8,81 €
Pantalla LCD 2x16	0,60 €
Lector Huellas	8,03 €
Lector SD	0,72 €
Cables PIN	0,41 €
PlacaBoard	2,45 €
Cable micro-USB a USB	4,90 €
Total:	25,92 €

Los links de compra se adjuntarán en el apartado de referencias

Referencias

Información

Unió Europea
Fons Social Europeu

1. Tutor: Arturo Aparicio Gimeno.

2. Compañero de Trabajo: Davil Holstarich.

3. https://stackoverflow.com (Algunos ejemplos de código).

4. https://github.com (Principal fuente de las librerías en su mayoría).

5. Archivos proporcionados del profesor de cómo usar y montar los componentes.

6. https://chat.openai.com/?model=gpt-4 (Depuración de código y sobretodo corrección de Syntaxis).

7. https://carbon.now.sh/?bg=rgba%28171%2C+184%2C+195%2C+1%29&t=vscode&wt=none&l=text%2Fx-c%2B%2Bsrc&width=680&ds=true&dsyoff=20px&dsblur=68px&wc=true&wa=true&pv=56px&ph=56px&ln=false&fl=1&fm=Hack&fs=14px&lh=133%25&si=false&es=2x&wm=false&code= (Web utilizada para crear las imágenes ilustrativas del código en este documento).

8. https://inputmakers.com/componentes/conectar-pantalla-lcd-a-arduino-por-conexion-i2c

9. https://github.com/fmalpartida/New-LiquidCrystal

Links de compra

Placa ESP 32

Pantalla LCD 2x16

Lector de Huellas Digitales

Lector Tarjetas SD

Cables PIN

PlacaBoard

Cable Micro-USB a USB