



**Universidad Politécnica de Juventino  
Rosas**

**Ingeniería en Redes y  
Telecomunicaciones**



## Proyecto Integrador

**”SCAN”**

**Sistema de control de asistencia y nómina por medio de huella  
dactilar**

### **Integrantes**

**JOCSAN URIEL GUZMAN GASCA  
YAHIR RAFAEL AMBRIZ MARTINEZ  
ARLETH MELISSA NUÑEZ MARTINEZ**

**Santa Cruz de Juventino Rosas, Gto.  
3 de abril de 2025**

# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Capítulo 1</b>	<b>2</b>
2.1. Objetivo General . . . . .	2
2.2. Objetivos Específicos . . . . .	2
2.3. Planteamiento del Problema . . . . .	3
2.4. Justificación . . . . .	3
<b>3. Capítulo 2</b>	<b>4</b>
3.1. Marco teórico . . . . .	4
3.1.1. Control de Asistencia . . . . .	4
3.1.2. Sistemas Biométricos y su Aplicación . . . . .	4
3.1.3. Desarrollo de Aplicaciones Web para Control de Asistencia . . . . .	4
3.1.4. Programación y Desarrollo de Software . . . . .	5
3.1.5. Microcontroladores y Conectividad . . . . .	5
<b>4. Capítulo 3</b>	<b>6</b>
4.1. Antecedentes . . . . .	6
<b>5. Capítulo 4</b>	<b>8</b>
5.1. Desarrollo . . . . .	8
5.1.1. Sensor . . . . .	8
5.1.2. Código . . . . .	10
5.1.3. Base de datos . . . . .	14
5.1.4. Código util.php . . . . .	17
<b>6. Capítulo 5</b>	<b>19</b>
6.1. Resultados . . . . .	19

<b>7. Capitulo 6</b>	<b>20</b>
7.1. Conclusiones . . . . .	20
<b>8. Capitulo 7</b>	<b>21</b>
8.1. Referencias . . . . .	21

# Introducción

Este proyecto es acerca de un sistema de control de entrada por medio de su huella digital, con la ayuda de un sensor de huellas y con el debido programa de Arduino IDE, en conjunto con un ESP32 se puede lograr que la huella quede registrada y de esa manera poder tener el control de acceso. En las empresas es de suma importancia el registro de sus trabajadores, identificarlos de alguna manera, llevar el control de sus horas de entrada y de sus horas de salida. Es por eso que también es necesario no solo controlar bien, sino optimizar el tiempo de estos registros y, a su vez, ser una empresa más productiva y sofisticada.

Este proyecto consistirá en varias etapas, es decir, el proceso se dividirá en partes para alcanzar los objetivos de manera adecuada y que se logre obtener los resultados acordes a precisamente los objetivos. En primera instancia se programará el código en Arduino IDE para que el sensor de huella pueda funcionar y que se logre el registro de una variedad de huellas que permitan el control de acceso a una empresa o lugar que requiera el uso de este tipo de control, y que se lleven a cabo los registros de cuando ingresan y salen los trabajadores de la empresa o lugar correspondiente.

Con el registro de huellas y el código funcionando, todos estos datos se guardarán en una base de datos, y en la segunda etapa se realizará una página web donde el usuario/trabajador pueda ver los registros de sus entradas y salidas, así como también el dinero que se puede descontar por retardos o sucesos de esa índole.

Este proyecto es relevante ya que no todas las empresas cuentan con un sofisticado control de acceso y normalmente se requiere el uso de una tarjeta ID o credencial identificadora. Por eso, con el uso de las huellas se podrá registrar todo acerca de entradas y salidas y hacer que el proceso sea más fácil y que el tiempo se pueda optimizar.

# Capítulo 1

## 2.1. Objetivo General

Desarrollar e implementar una página web con un sistema de registro de huellas digitales para empleados, que permita gestionar de manera eficiente el control de asistencia, registrando la hora de entrada y salida, así como el cálculo automático de salarios.

## 2.2. Objetivos Específicos

- Desarrollar una página web con una interfaz accesible para la gestión de empleados y su información laboral.
- Elaborar una base de datos eficiente que almacene la información de los empleados, sus horarios de trabajo y salarios.
- Implementar un módulo de autenticación dactilar para que los empleados puedan registrarse e identificarse de manera segura.

## 2.3. Planteamiento del Problema

En las empresas, es fundamental llevar un control preciso sobre los registros de entrada y salida de los empleados. Para ello, se suelen utilizar sistemas de identificación basados en credenciales, que los empleados deben pasar por un lector para registrar sus horas. Sin embargo, este proceso puede resultar lento, ya que el simple hecho de portar la credencial y pasarla por el dispositivo genera un retraso en la gestión de los registros.

Además, muchas veces no se tiene un control claro de las horas exactas de entrada y salida de los empleados. En muchos casos, los trabajadores no tienen visibilidad de sus registros de tiempo, lo que genera incertidumbre sobre las horas trabajadas y, por ende, sobre los descuentos que puedan aplicarse a su pago. Esta falta de transparencia provoca que los empleados deban recurrir al área de recursos humanos para consultar sus registros, lo que termina generando un proceso aún más tedioso y demorado.

Este panorama se traduce en una pérdida de tiempo tanto para los empleados como para los responsables de gestión, afectando la productividad general de la empresa. El proceso de verificación y control de las horas se vuelve ineficiente, lo que impacta negativamente en la gestión administrativa y en el desempeño de la organización.

## 2.4. Justificación

El proyecto tiene como objetivo la implementación de un sistema de control de asistencia digital mediante huellas dactilares, para optimizar el proceso de registro de entrada y salida de los empleados. Actualmente, las empresas enfrentan dificultades en el control eficiente de las horas de trabajo, debido a la dependencia de credenciales físicas y los retrasos generados por su uso. Esto no solo afecta la productividad de la organización, sino que también crea incertidumbre entre los empleados, quienes desconocen las horas exactas de su registro y los descuentos aplicados en su salario.

La solución planteada a través de este sistema biométrico tiene el potencial de mejorar significativamente la eficiencia operativa. Al eliminar la necesidad de credenciales físicas, se optimiza el tiempo de registro y se reducen los errores humanos. Además, el sistema permitirá a los empleados acceder a su historial de entradas y salidas, lo que mejora la transparencia y fomenta un entorno de confianza.

La implementación de esta tecnología no solo contribuirá a una mejor gestión administrativa, sino que también proporcionará un valor agregado al proceso, al integrar la biometría, lo que representa una tendencia creciente en los sistemas de seguridad y control de acceso. De este modo, el proyecto no solo responde a una necesidad inmediata de la empresa, sino que también se alinea con las mejores prácticas tecnológicas en el sector.

# Capítulo 2

## 3.1. Marco teórico

### 3.1.1. Control de Asistencia

El control de asistencia es un proceso fundamental en la gestión de recursos humanos de cualquier empresa, ya que permite registrar las horas trabajadas por los empleados y garantizar un cálculo preciso de salarios. Normalmente, este control se ha llevado a cabo mediante registros manuales o tarjetas de identificación, pero con los avances tecnológicos, los sistemas biométricos han cobrado relevancia por su seguridad y precisión.

### 3.1.2. Sistemas Biométricos y su Aplicación

Los sistemas biométricos son tecnologías que utilizan características físicas o de comportamiento de una persona para su identificación. Dentro de estos sistemas, el reconocimiento de huellas digitales es uno de los más utilizados debido a su fiabilidad, rapidez y accesibilidad. La huella dactilar es única para cada individuo y difícil de falsificar, lo que la convierte en una solución eficiente para el control de asistencia en entornos laborales.

### 3.1.3. Desarrollo de Aplicaciones Web para Control de Asistencia

Las aplicaciones web permiten la gestión remota y centralizada del control de asistencia, ofreciendo ventajas como:

- **Accesibilidad:** Los datos pueden ser consultados desde cualquier dispositivo con conexión a internet.
- **Automatización:** Registro automático de entradas y salidas y cálculo de horas trabajadas.

- **Interfaz amigable:** Facilita la interacción de los empleados y administradores con el sistema.

#### 3.1.4. Programación y Desarrollo de Software

Para la implementación del sistema de control de asistencia se utiliza Arduino IDE, un entorno de desarrollo integrado ampliamente utilizado en la programación de microcontroladores. Se emplean diversas bibliotecas, entre ellas Adafruit Fingerprint, que permite la interacción con el sensor de huellas dactilares, facilitando el almacenamiento, reconocimiento y comparación de huellas.

En etapas posteriores del desarrollo del proyecto, se incorporarán bibliotecas para la conectividad a internet, como WiFi.h, y librerías para la comunicación con la base de datos mediante protocolo HTTP.

#### 3.1.5. Microcontroladores y Conectividad

El ESP32 es un microcontrolador de alta capacidad con conectividad Wi-Fi y Bluetooth, ideal para aplicaciones de Internet de las Cosas. Su integración en este proyecto permite no solo la captura y almacenamiento de huellas, sino también la transmisión de datos a una base de datos alojada en un servidor web, ya que el uso de la conectividad Wi-Fi es fundamental para el almacenamiento y consulta de datos en tiempo real, facilitando la integración con una página web en la que los empleados puedan revisar sus registros de asistencia y cálculos salariales.



# Capítulo 3

## 4.1. Antecedentes

El Sistema de Control de Asistencia y nómina por Medio de Huella Dactilar es una solución tecnológica que permite registrar, controlar y gestionar la asistencia de los empleados mediante el uso de huellas dactilares como mecanismo de autenticación. Este tipo de sistema se utiliza comúnmente en empresas y organizaciones que buscan una forma más segura y eficiente de llevar un control preciso de la presencia de sus trabajadores y calcular sus horas laborales para la nómina. A continuación, te doy un resumen de los antecedentes y elementos clave de este tipo de proyecto:

- **Necesidad de Control de Asistencia:** Las organizaciones requieren un sistema eficaz para asegurar que los empleados lleguen a tiempo y trabajen las horas estipuladas. Tradicionalmente, muchos sistemas utilizaban tarjetas de control de asistencia o sistemas de registro manual, los cuales son propensos al fraude o a errores humanos.
- **Evolución hacia Tecnología Biométrica:** Con el avance de la tecnología, la biometría se convirtió en una de las mejores soluciones para autenticar a los empleados. Entre las formas más comunes de biometría, las huellas dactilares son muy populares debido a su facilidad de implementación y alta precisión. El uso de huellas dactilares elimina la posibilidad de suplantación, ya que son únicas e imposibles de replicar fácilmente.

### **Ventajas de la Tecnología Biométrica:**

- **Seguridad:** No hay posibilidad de fraude o error humano, ya que las huellas dactilares son únicas e intransferibles.
- **Eficiencia:** Registra la asistencia de los empleados de forma rápida y sin intervención manual.

- **Ahorro de costos:** Reduce la necesidad de personal para controlar manualmente las asistencias, y disminuye los errores en el cálculo de la nómina.
- **Automatización:** Facilita la integración con otros sistemas de recursos humanos, como la nómina, para calcular el salario de acuerdo con las horas trabajadas.

# Capitulo 4

## 5.1. Desarrollo

### 5.1.1. Sensor

Para este proyecto se está utilizando el uso de un microcontrolador ESP32, junto con el sensor de huellas modelo AS608 para la obtención de las huellas dactilares, así como también el uso de otras herramientas de software como lo son Arduino IDE.

El sensor AS608 es un módulo óptico de reconocimiento de huellas dactilares que se utiliza en sistemas de seguridad y control de acceso. Su funcionamiento se basa en la captura de imágenes de las huellas dactilares y su procesamiento para identificación o verificación.



Figura 5.1: Sensor AS608.

El sensor de huellas AS608 es un módulo óptico que captura y procesa imágenes de huellas dactilares. Funciona mediante una interfaz UART y tiene un procesador interno que permite el reconocimiento sin depender completamente del microcontrolador o computadora a la que está conectado.

Su funcionamiento comienza con la captura de la imagen de la huella mediante un sensor óptico, seguido de la conversión a datos digitales y la extracción de características únicas llamadas minucias. Luego genera una plantilla de la huella que puede ser almacenada en su memoria interna para futuras comparaciones.

Puede realizar tanto verificación uno a uno como identificación uno a muchos. Se comunica a través de UART con comandos específicos como registro de huella, búsqueda, eliminación y vaciado de la base de datos. Se integra fácilmente con ESP32 usando librerías como la de Adafruit.

Para la configuración del hardware del sistema de control de asistencias se selecciona el uso de los pines que se requieren conectar en el ESP32 del sensor de huellas digitales, las conexiones son:

Sensor R307:

- VCC - 5V del ESP32
- GND - GND del ESP32
- TX - Pin RX del ESP32
- RX - Pin TX del ESP32

Son las necesarias para el funcionamiento del sensor junto con el microcontrolador ESP32, quedando de esa manera las conexiones elegidas para el uso del hardware del que se está haciendo uso para la realización de este proyecto.

### 5.1.2. Código

Este código se diseñó para funcionar con un ESP32 y un sensor de huellas AS608 para registrar las huellas dactilares y enviarlas a un servidor web, donde se almacena la información en una base de datos.

```
1  #include <WiFi.h>
2  #include <HTTPClient.h>
3  #include <Adafruit_Fingerprint.h>
4
5  const char* ssid = "b54aa2";
6  const char* password = "273439736";
7
8
9  const char* serverURL = "http://144.24.22.106/util.php";
10
11 #define RX_PIN 16
12 #define TX_PIN 17
13
14
15 HardwareSerial mySerial(2);
16 Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
17
18 void setup() {
19     Serial.begin(115200);
20
21     Serial.print("Conectando a WiFi...");
22     WiFi.begin(ssid, password);
23     while (WiFi.status() != WL_CONNECTED) {
24         delay(500);
25         Serial.print(".");
26     }
27     Serial.println(" Conectado a WiFi");
28     delay(2000);
29
30     mySerial.begin(57600, SERIAL_8N1, RX_PIN, TX_PIN);
31     finger.begin(57600);
```

Figura 5.2: Código en Arduino IDE - Parte 1.

```

30 mySerial.begin(57600, SERIAL_8N1, RA_PIN, TA_PIN);
31 finger.begin(57600);
32
33 if (finger.verifyPassword()) {
34   Serial.println(" Sensor AS608 detectado correctamente");
35 } else {
36   Serial.println(" ERROR: No se detecta el sensor. Verifica conexiones.");
37   while (1) { delay(1); }
38 }
39 }
40
41 void loop() {
42   Serial.println("Coloca tu dedo en el sensor...");
43   int fingerprintID = registrarHuella();
44
45   if (fingerprintID >= 0) {
46     Serial.print(" Huella registrada con ID: ");
47     Serial.println(fingerprintID);
48     enviarHuella(String(fingerprintID));
49   } else {
50     Serial.println(" Error al registrar la huella.");
51   }
52
53   delay(3000);
54 }
55
56
57 int registrarHuella() {
58   int p = -1;
59
60   while (p != FINGERPRINT_OK) {

```

 Connection lost. Cloud sketch actions and u

Figura 5.3: Código en Arduino IDE - Parte 2.

```

57 int registrarHuella() {
58     int p = -1;
59     while (p != FINGERPRINT_OK) {
60         p = finger.getImage();
61     }
62     p = finger.image2Tz(1);
63     if (p != FINGERPRINT_OK) return -1;
64
65     Serial.println(" Coloca el mismo dedo nuevamente...");
66     p = -1;
67     while (p != FINGERPRINT_OK) {
68         p = finger.getImage();
69     }
70
71     p = finger.image2Tz(2);
72     if (p != FINGERPRINT_OK) return -1;
73
74     p = finger.createModel();
75     if (p != FINGERPRINT_OK) return -1;
76
77     int id = 1;
78     p = finger.storeModel(id);
79     if (p != FINGERPRINT_OK) return -1;
80
81     return id;
82 }
83 void enviarHuella(String huellaID) {
84     if (WiFi.status() != WL_CONNECTED) {
85         Serial.println("No hay conexión WiFi");
86         return;
87     }

```

Figura 5.4: Código en Arduino IDE - Parte 3.

```

82 }
83 void enviarHuella(String huellaID) {
84     if (WiFi.status() != WL_CONNECTED) {
85         Serial.println("No hay conexión WiFi");
86         return;
87     }
88
89     Serial.println("Ingresa el nombre para almacenar:");
90     while (Serial.available() == 0) {}
91     String nombre = Serial.readString();
92     nombre.trim();
93
94     HTTPClient http;
95     http.begin(serverURL);
96     http.addHeader("Content-Type", "application/x-www-form-urlencoded");
97
98     String postData = "huella=" + huellaID + "&nombre=" + nombre;
99     Serial.println("Enviando datos: " + postData);
100
101     int httpResponseCode = http.POST(postData);
102
103     if (httpResponseCode > 0) {
104         String response = http.getString();
105         Serial.println("Respuesta del servidor: " + response);
106     } else {
107         Serial.println("Error en la solicitud HTTP: " + String(httpResponseCode));
108     }
109
110     http.end();
111 }
112

```

Figura 5.5: Código en Arduino IDE - Parte 4.

El código comienza estableciendo la conexión a una red WiFi mediante el SSID y la contraseña proporcionados utilizando la biblioteca `WiFi.h` para conectar el ESP32 a la red WiFi. Si la conexión es exitosa, muestra un mensaje indicando que está conectado.

## Configuración del Sensor de Huellas AS608

Se configura el sensor de huellas AS608 utilizando la biblioteca `Adafruit_Fingerprint.h` y se establece la comunicación con el sensor a través de los pines RX (16) y TX (17) del ESP32 usando el puerto serie hardware (`HardwareSerial`). De esta manera, se verifica la correcta detección del sensor. Si no se detecta, el código se detiene con un mensaje de error que se observa en el código.

## Registro de Huellas

Dentro de la función `loop()`, el ESP32 espera que el usuario coloque su dedo en el sensor. De esta manera, la función `registrarHuella()` captura la huella, toma dos imágenes del dedo (una para crear el modelo y otra para verificarlo) y luego almacena el modelo de la huella en la memoria del sensor. Por lo tanto, si la huella se registra correctamente, se asigna un ID a la huella registrada.



## Envío de Datos al Servidor

Después de registrar una huella, el código solicita al usuario un nombre a través del monitor serial y utiliza la librería `HTTPClient.h` para enviar los datos del ID de la huella y el nombre al servidor en la URL proporcionada (`http://144.24.22.106/util.php`) el cual recibe los datos enviados y los procesa para almacenarlos en una base de datos. Luego, envía una solicitud al servidor incluyendo los datos con los parámetros huella y nombre. Cuando esto sucede, el servidor responde con una confirmación, que se muestra en el monitor serial.

## Envío de Datos HTTP

El ESP32 verifica si la conexión WiFi está activa antes de enviar la solicitud HTTP. Si la conexión es correcta, se muestra la respuesta del servidor indicando si la operación fue realizada o si hubo algún error.

### 5.1.3. Base de datos

En cuanto a base de datos, se está trabajando con una base de datos llamada `admin.default` creada en MySQL, donde se creó una tabla llamada `registro1`.

La tabla `registro1` es una tabla con campos para datos personales, huellas dactilares e `id` como llave primaria. El campo `id` sirve como identificador único para cada registro en la tabla. Al ser `AUTO_INCREMENT`, MySQL asigna automáticamente un valor único y secuencial a cada nueva fila insertada, evitando la necesidad de especificarlo manualmente. Además, gestiona la numeración automáticamente, optimizando la inserción de datos.

```
Windows PowerShell x melissanm@LAPTOP-F41MR4L x melissanm@LAPTOP-F41MR4 x + v
El comportamiento de finalización se puede configurar en la configuración avanzada del perfil. No volver a mostrar

affiliates. Other names may be trademarks of their respective owners.

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

mysql>
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| admin_default |
+-----+
2 rows in set (0.00 sec)

mysql> use admin_default;
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
mysql> USE admin_default;
Database changed
mysql> use information_schema;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

Figura 5.6: Base de datos en terminal.

```
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_admin_default |
+-----+
| registro |
| registro1 |
+-----+
2 rows in set (0.00 sec)

mysql> DESCRIBE registro1;
ERROR 2006 (HY000): MySQL server has gone away
No connection. Trying to reconnect...
Connection id: 320262
Current database: admin_default
```

Figura 5.7: Tabla registro1 en terminal.

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
numero_nomina	int(11)	YES		0	
nombre	varchar(50)	NO		NULL	
apellido_p	varchar(50)	NO		NULL	
apellido_m	varchar(50)	NO		NULL	
hora_e	datetime	NO		NULL	
hora_s	datetime	NO		NULL	
huella	varchar(255)	YES		NULL	

8 rows in set (0.01 sec)

mysql> client\_loop: send disconnect: Broken pipe

melissanm@LAPTOP-F41MR4LK: ~\$

Figura 5.8: Estructura de la tabla registro1.

#### 5.1.4. Código util.php

```
<?php
|
$servername = "localhost";
$username = "usuario";
$password = "contraseña";
$dbname = "base_de_datos";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect_error) {
    die("Conexión fallida: " . $conn->connect_error);
}

$huellaID = $_POST['huella'];
$nombre = $_POST['nombre'];

$sql = "INSERT INTO registro1 (huella, nombre) VALUES ('$huellaID', '$nombre')";

if ($conn->query($sql) === TRUE) {
    echo "Huella registrada correctamente";
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}

$conn->close();
?>
```

Figura 5.9: Código util.php.

En este sistema, el archivo `util.php` en el servidor cumple la función de intermediario entre el ESP32 y la base de datos MySQL. Es el encargado de recibir los datos enviados por el ESP32, procesarlos y almacenarlos en la base de datos.

Cuando el ESP32 registra una huella y un nombre, estos datos se envían a través de una solicitud HTTP POST a la URL del servidor donde está `util.php`. El ESP32 envía los valores de huella y nombre en formato `application/x-www-form-urlencoded`. Después, `util.php` recupera estos valores usando `$_POST['huella']` y `$_POST['nombre']`, y usa

PHP y MySQLi para establecer una conexión con la base de datos `admin.default` e insertar los datos en la tabla `registro1`.

## Capitulo 5

## 6.1. Resultados



Figura 6.1: Mensaje en el monitor serial.

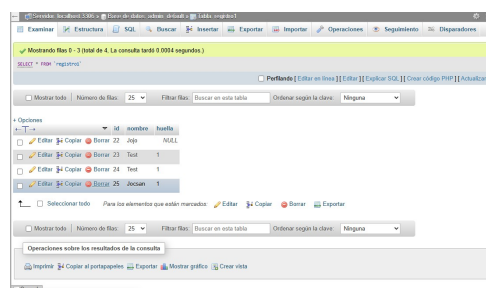


Figura 6.2: Datos registrados.

El mensaje en el monitor serial del ESP32 indica que el sistema de registro de huellas está funcionando correctamente y que los datos se han almacenado en la base de datos de acuerdo a los datos enviados.

# Capitulo 6

## 7.1. Conclusiones

# Capitulo 7

## 8.1. Referencias



# Bibliografía

- [1] Ayala, F. (2023, 23 octubre). Cómo utilizar el Lector de Huella Dactilar As608. UNIT Electronics. Disponible en: <https://blog.uelectronics.com/tarjetas-desarrollo/como-utilizar-el-lector-de-huella-d>
- [2] Equipo editorial, EtecÃ©. (2025b, febrero 20). *Servidor web - Concepto, características, tipos y ejemplos*. Concepto. Recuperado de: <https://concepto.de/servidor-web/>