



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

DISEÑO DE UN SISTEMA DE CONTROL HORARIO MEDIANTE RECONOCIMIENTO DE HUELLA

Alumno: Salvador Trujillo Diaz

Tutor: Prof. D. Manuel Ángel Gadeo Martos
Depto.: Ingeniería de Telecomunicación

RESÚMEN

En este Trabajo de Fin de Grado se han analizado posibles tecnologías para adoptar una solución con la cual satisfacer la necesidad del Real Decreto-ley 8/2019 de mantener un almacenamiento del registro de la jornada laboral.

Además del cumplimiento primordial anterior, exige ser una solución geolocalizada y con alertas de cumplimiento jornada laboral en el servidor, y con posibilidad de ser alimentado por batería.

De esta manera, se analiza y se toma la decisión de implementación que aporte más ventajas para llegar al objetivo del proyecto y que opte por ser una solución económica.

También se ha elaborado un manual de usuario para el uso del dispositivo en el registro de la jornada laboral, y para el procedimiento de registro de usuarios, editar y eliminar.

ABSTRACT

In this Final Degree Project we have analysed possible technologies to adopt a solution with which to satisfy the need of Real Decreto-ley 8/2019 to keep a storage of the working day record.

In addition to the aforementioned primary compliance, it must be a geolocalised solution with working day compliance alerts on the server, and with the possibility of being battery-powered.

In this way, the implementation decision is analysed and taken that provides the most advantages to achieve the project's objective and that opts to be an economical solution.

A user manual has also been developed for the use of the device in the working time registration, and for the procedure of user registration, editing and deleting.

ÍNDICE

RESÚMEN.....	2
ABSTRACT	3
ÍNDICE	4
ÍNDICE DE FIGURAS.....	7
1 MEMORIA	9
1.1 INTRODUCCIÓN	9
1.2 ANTECEDENTES.....	9
2 OBJETIVOS	10
3 MATERIALES Y TECNOLOGÍAS A UTILIZAR.....	11
3.1 TECNOLOGÍAS DE REDES DE SENSORES	11
3.1.1 <i>Introducción</i>	<i>11</i>
3.1.2 <i>Tecnologías inalámbricas.....</i>	<i>12</i>
3.1.3 <i>Bluetooth 4.x/Bluetooth low energy (BLE)</i>	<i>12</i>
3.1.4 <i>Zigbee.....</i>	<i>12</i>
3.1.5 <i>Red de telefonía móvil.....</i>	<i>13</i>
3.1.6 <i>LoRaWan.....</i>	<i>14</i>
3.1.7 <i>802.11/Wi-Fi</i>	<i>14</i>
3.2 ELECCIÓN DE LA TECNOLOGÍA INALÁMBRICA.....	15
3.2.1 <i>Conclusión.....</i>	<i>15</i>
3.3 TECNOLOGÍAS DE IMPLEMENTACIÓN DEL SISTEMA	15
3.3.1 <i>Introducción</i>	<i>15</i>
3.3.2 <i>Arduino</i>	<i>15</i>
3.3.3 <i>NodeMCU (ESP8266)</i>	<i>16</i>
3.3.4 <i>NodeMCU-32 (ESP-32).....</i>	<i>17</i>
3.3.5 <i>Raspberry Pi.....</i>	<i>17</i>
3.3.6 <i>Módulos ESP-XX.....</i>	<i>18</i>
3.3.7 <i>Módulo GSM/GPRS SIM800L.....</i>	<i>19</i>
3.3.8 <i>Módulo GSM/GPRS SIM900.....</i>	<i>19</i>
3.4 ELECCIÓN DE LA TECNOLOGÍA DE IMPLEMENTACIÓN	20
3.4.1 <i>Conclusión.....</i>	<i>20</i>
SENSORES BIOMÉTRICOS.....	22
3.4.2 <i>Sistema biométrico.....</i>	<i>22</i>
3.4.3 <i>Características de un sistema biométrico</i>	<i>23</i>
3.4.4 <i>Tipo de sistema biométrico a emplear</i>	<i>23</i>
3.4.5 <i>Funcionamiento sensores de huella dactilar</i>	<i>24</i>

3.4.6	<i>Tipos de sensores de huella dactilar</i>	25
3.4.6.1	<i>Sensores ópticos reflexivos</i>	25
3.4.6.2	<i>Sensores ópticos transmisivos</i>	26
3.4.6.3	<i>Sensor de huella térmico</i>	26
3.4.6.4	<i>Sensores capacitivos</i>	27
3.5	ELECCIÓN DE SENSOR BIOMÉTRICO	28
3.5.1	<i>Conclusión</i>	28
3.6	TECNOLOGÍAS DE COMUNICACIÓN Y DE LADO DEL SERVIDOR	29
3.6.1	<i>Introducción</i>	29
3.6.2	<i>Sistema de gestión de bases de datos</i>	29
3.6.2.1	<i>MYSQL</i>	29
3.6.2.2	<i>PostgreSQL</i>	30
3.6.2.3	<i>Microsoft SQL Server</i>	30
3.6.2.4	<i>Conclusión</i>	30
3.6.3	<i>Servidores web</i>	31
3.6.3.1	<i>Microsoft Internet Information Services</i>	31
3.6.3.2	<i>Servidor web Apache</i>	31
3.6.3.3	<i>Nginx</i>	31
3.6.3.4	<i>Conclusión</i>	32
3.6.4	<i>Herramienta de despliegue y lenguaje de programación de servidor</i>	32
3.6.4.1	<i>WampServer</i>	32
3.6.4.2	<i>Lenguaje de lado de servidor PHP</i>	32
3.6.5	<i>Servidor de hosting</i>	33
3.6.5.1	<i>000webhost</i>	33
3.6.5.2	<i>Awardspace</i>	33
3.6.5.3	<i>Infinityfree</i>	33
3.6.5.4	<i>Clever cloud</i>	33
3.6.5.5	<i>Conclusión</i>	34
3.7	HERRAMIENTAS DE DESARROLLO IDE	35
3.7.1	<i>Visual Studio Code</i>	35
3.7.2	<i>Arduino IDE</i>	36
3.8	MATERIAL HARDWARE ADICIONAL PARA EL DISEÑO DEL SISTEMA	37
3.8.1	<i>Pantalla OLED 0.96</i>	37
3.8.2	<i>Batería LiPo 3.7V 1500mA</i>	37
3.8.3	<i>Mini DC-DC 5V</i>	38
3.8.4	<i>Cargador USB regulador voltaje a 3.7V</i>	39
4	IMPLEMENTACIÓN	40
4.1	IMPLEMENTACIÓN EN EL SERVIDOR WEB	40

4.1.1	Registro de un dispositivo/nodo	40
4.1.2	Registro de usuarios.....	41
4.1.3	Eliminación de usuarios.....	44
4.1.4	Editar usuarios.....	44
4.1.5	Consulta de marcajes.....	45
4.1.5.1	Algoritmo cumplimiento jornada laboral.....	46
4.2	IMPLEMENTACIÓN DEL NODO INALÁMBRICO	47
4.2.1	Inicio de parámetros esenciales.....	47
4.2.2	Conexión a la red Wi-Fi.....	48
4.2.3	Conexión a la red GPRS	50
4.2.4	Registro de usuarios.....	51
4.2.5	Eliminación de usuarios.....	53
4.2.6	Marcaje de los usuarios.....	55
4.2.7	Geolocalización del dispositivo	56
5	CONCLUSIONES Y LÍNEAS DE FUTURO	57
5.1	CONCLUSIONES	57
5.2	LÍNEAS DE FUTURO	58
5.2.1	Hardware	58
5.2.2	Software	58
6	PRESUPUESTO.....	59
7	BIBLIOGRAFIA	59
8	ANEXOS.....	61
8.1	MANUAL DE USUARIO	61
8.1.1	Fichaje de entrada.....	61
8.1.2	Fichaje de salida.....	63
8.1.3	Registro de usuario	64
8.1.4	Borrado de usuario	68

ÍNDICE DE FIGURAS

Figura 1 Estructura básica de un nodo WSN	11
Figura 2 Resumen estadístico de uso de tecnologías para redes de sensores inalámbricas	12
Figura 3 Arduino UNO	16
Figura 4 NodeMCU	16
Figura 5 ESP-32	17
Figura 6 Raspberry Pi	18
Figura 7 Ejemplo de modelo ESP-01	18
Figura 8 Módulo SIM800L	19
Figura 9 SIM900	20
Figura 10 Esquema pinout del NodeMCU	21
Figura 11 Esquema pinout del SIM800L	21
Figura 12 Tipos de sistemas biométricos	22
Figura 13 Ejemplo de captura dactilar	23
Figura 14 Procesamiento de una huella	24
Figura 15 Sensor óptico Adafruit [10]	25
Figura 16 Sensor óptico transmisor	26
Figura 17 Sensor de huella térmico	26
Figura 18 Sensor de huella capacitivo	27
Figura 19 Ejemplo conexión con NodeMCU	28
Figura 20 Ejemplo de interfaz Visual Studio Code	35
Figura 21 Ejemplo interfaz Arduino IDE [4]	36
Figura 22 Display OLED 0.96	37
Figura 23 Batería LiPo 3.7V	37
Figura 24 Regulador voltaje step up 5V	38
Figura 25 Cargador USB para batería 3.7V	39
Figura 26 Ejemplo de representación de la solución	40
Figura 27 Modal del formulario para crear dispositivo	41
Figura 28 Vista de gestión de usuarios	41
Figura 29 Modal registro de usuario	42
Figura 30 Comprobación de ID disponible mediante procedimiento almacenado dentro del rango del sensor de huella	42
Figura 31 Proceso de registro huella en servidor	43
Figura 32 Ejemplo de borrado en el servidor web	44
Figura 33 Vista consulta de marjajes de usuarios	45
Figura 34 Vista consulta de marjajes de usuarios desde gestor de base de datos	45
Figura 35 Ejemplo código de consulta de marjajes con filtro	46
Figura 36 Ejemplo sencillo de aplicación de una regla para alertas	46
Figura 37 Ejemplo de implementación sin soldar en placa board	47
Figura 38 Librerías usadas NodeMCU	47
Figura 39 Credenciales Wi-Fi, token nodo y URL API web	48
Figura 40 Conectar el nodo a una red Wi-Fi	49
Figura 41 Conexión a red GPRS 1	50
Figura 42 Conexión a red GPRS 2	51
Figura 43 Ejemplo comprobación registro usuario con red GSM	51
Figura 44 Comprobación nuevo usuario a registrar	52
Figura 45 Código que informa haciendo una petición al servidor del registro de un usuario en el sensor	53
Figura 46 Comprobación de usuario a eliminar	54

<i>Figura 47 Función que envía ID al servidor para hacer un fichaje.....</i>	<i>55</i>
<i>Figura 48 Obtener datos de latitud y longitud con Google API Geolocation</i>	<i>56</i>
<i>Figura 49 Ejemplo de datos de dispositivo geolocalizado</i>	<i>56</i>
<i>Figura 50 Huella dactilar reconocida y correcta</i>	<i>61</i>
<i>Figura 51 Marcaje de entrada registrado en el servidor web.....</i>	<i>61</i>
<i>Figura 52 Huella dactilar no reconocida.....</i>	<i>62</i>
<i>Figura 53 Marcaje de salida registrado en el servidor web</i>	<i>63</i>
<i>Figura 54 Tabla administración usuarios.....</i>	<i>64</i>
<i>Figura 55 Registro con datos de usuario.....</i>	<i>64</i>
<i>Figura 56 Registrado en el servidor y a falta de confirmar en el sensor</i>	<i>65</i>
<i>Figura 57 Registro en el sensor</i>	<i>65</i>
<i>Figura 58 Mensaje para retirar dedo</i>	<i>66</i>
<i>Figura 59 Registro hecho correctamente</i>	<i>67</i>
<i>Figura 60 Vista con usuario registrado en el servidor.....</i>	<i>67</i>
<i>Figura 61 Modal confirmación borrar usuario.....</i>	<i>68</i>
<i>Figura 62 Mensaje de borrado usuario</i>	<i>68</i>

1 MEMORIA

1.1 Introducción

Este proyecto surge de la necesidad a efectos del Real Decreto-ley 8/2019 que pretende garantizar el cumplimiento de los límites de la jornada laboral, creando una obligatoriedad del registro de datos de la jornada laboral de los empleados durante un periodo de 4 años, sin importar el tipo de método o solución se adopte, de esta manera se crea un marco de seguridad jurídica que beneficia a los empleados y empresas y además facilita el trabajo de control por parte de la Inspección de Trabajo y Seguridad Social.

Se describe la implementación de los dispositivos y tecnologías inalámbricas que se van a utilizar para conseguir adoptar un sistema de control horario mediante huella dactilar, además de la implementación del servidor web que se usará tanto para consultar datos, como para añadir, editar y eliminar.

1.2 Antecedentes

Cómo se ha descrito en el apartado anterior, el problema surge en su mayoría porque no existen medidas implementadas en ningún caso o son rudimentarias (búsqueda de papeles, firmas, desconocimiento de la hora exacta de llegada...), y en cualquiera de los casos no se da solución a uno de los problemas principales por lo que surgió la nueva ley que es pagar las horas extras al no estar correctamente recogido el inicio y fin de la jornada laboral de los empleados.

Es un problema recurrente en España del que cada empresa debe tomar medidas, y la solución que aquí se plantea es una de ellas.

En la presente se persigue controlar la asistencia, en entornos que incluso pudieran ser rurales con escasez de redes de comunicaciones y, además, al ser un dispositivo portátil con batería su disposición geográfica puede ir variando, se planteara más adelante un pequeño análisis del porqué de las tecnologías utilizadas.

2 OBJETIVOS

La solución deberá ser capaz de hacer fichajes de forma segura e inequívoca con una fiabilidad y facilidad de uso que no dé lugar a ser una tarea tediosa para los usuarios.

En este apartado se explica con un breve resumen los objetivos principales del proyecto.

- Estudio de la tecnología de redes de sensores inalámbricos y elección del tipo de nodo a emplear.
- Configurar y programar un nodo para realizar la captura de información (señal de sensores GPS y de identificación biométrica) y su envío a un servidor de comunicaciones.
- Programar un servidor web para facilitar el almacenamiento y acceso a la información capturada por los nodos sensores (señal GPS e identificación biométrica), así como la gestión de los nodos sensores de la red.
- Diseño y programación en el servidor de un algoritmo que permita identificar situaciones de incumplimiento de la normativa vigente y dispare los correspondientes mensajes de alarma.
- Diseño y montaje de una red experimental de sensores inalámbricos para la monitorización de información (señal GPS e identificación biométrica).
- Realización de experimentos para comprobar el funcionamiento de la red experimental diseñada.

3 MATERIALES Y TECNOLOGÍAS A UTILIZAR

3.1 Tecnologías de redes de sensores

3.1.1 Introducción

Las redes de sensores inalámbricas (Wireless Sensor Networks, o simplemente WSN), son constituidas por plataformas de hardware con limitaciones en cuanto a recursos de memoria, procesamiento y energía, es por esto que las WSN suelen trabajar con protocolos específicos para este tipo de dispositivos, de tal manera que a la hora de presentar los datos o resultados al usuario u otro tipo de sistema perteneciente a los protocolos de la pila TCP/IP, sea necesario que se deban usar técnicas de hardware o software de intermediario entre una red WSN y una red TCP/IP.

Vamos analizar las tecnologías y plantear dentro de nuestras posibilidades, que puede ser lo más conveniente para alcanzar nuestro objetivo, y si, atendiendo a lo anterior podemos incluirlo o no como una solución WSN.

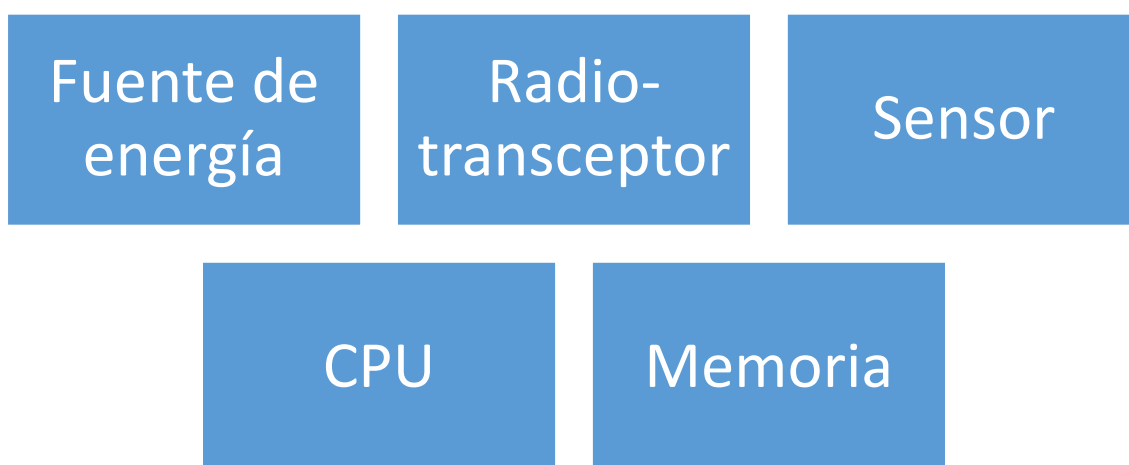


Figura 1 Estructura básica de un nodo WSN

Una red de sensores inalámbrica puede estar formada por nodos, routers, gateways y estaciones base.

3.1.2 Tecnologías inalámbricas

Para transmitir datos de sensores que bien pueden ser ambientales, meteorológico, por acciones físicas o circunstancias concretas es necesario hacer uso de una tecnología inalámbrica por su versatilidad y necesidad de movilidad de los dispositivos, vamos a hacer un estudio de las tecnologías o estándares inalámbricos más importantes que debemos tener en cuenta para el sistema que debemos implementar según nuestras necesidades.

	Local Area Network Short Range Communication	Low Power Wide Area (LPWAN) Internet of Things	Cellular Network Traditional M2M
	40%	45%	15%
😊	Well established standards In building	Low power consumption Low cost Positioning	Existing coverage High data rate
😞	Battery Live Provisioning Network cost & dependencies	High data rate Emerging standards	Autonomy Total cost of ownership
	Bluetooth 4.x Wi-Fi	LoRa	3G / H+ 4G

Figura 2 Resumen estadístico de uso de tecnologías para redes de sensores inalámbricas

3.1.3 Bluetooth 4.x/Bluetooth low energy (BLE)

- **Alcance:** rango de 30-50 metros, hasta 100 metros en buenas circunstancias(BLe> 100 metros). (futuro se prevé 500m – 1km de rango)
- **Velocidad de transferencia:** 1-3 Mbps(B4.x); 125kbps-2Mbps (BLe);
- **Arquitectura:** cliente-servidor
- **Tamaño paquetes:** pequeño
- **Proceso de conexión:** eficiente.
- **Precio:** muy barato
- **Consumo energético:** bajo-medio
- **Versatilidad:** Fácil uso, comparte espectro con Wi-Fi que suele estar presente y en uso en cualquier lugar pudiendo provocar problemas en la comunicación

3.1.4 Zigbee

- **Baja velocidad de transferencia:** 250Kbps.
- **Bajo consumo.**
- **Seguridad.**
- **Alta capacidad** de nodos en red.
- **Alcance:** 10-100m.

3.1.5 Red de telefonía móvil

- **Alcance:** rango de cobertura de áreas de kilómetros
- **Velocidad de transferencia:**
- **Tipos:**
 1. **GSM - CSD (2G):** hasta 9'6 kbps en subida y bajada - Prácticamente en desuso y se está comunicando mientras se usa.
 2. **GSM - GPRS (2'5G):** hasta 80 kbps en bajada y 20 kbps en subida
 3. **GSM - EDGE (2'75G):** hasta 236 kbps en bajada y 59 kbps en subida
 4. **3G - UMTS (3G) -** de 64 a 384 kbps de subida y bajada
 5. **3G - HSPA (HSDPA+HSUPA) (3'5G) -** hasta 7'2 Mbps de subida y bajada
 6. **3G - HSPA+ (3'75G) -** Hasta 22 Mbps de subida y bajada
 7. **4G - LTE -** La velocidad máxima en 4G es de 75 Mbps en bajada y 25 Mbps en subida.
- **Consumo energético:** muy alto
- **Precio:** muy alto
- **Versatilidad:** Facilidad de uso, sujeto a problemas de cobertura por causas meteorológicas.

Cualquiera de los módulos que se conectan a la red hacen uso de servidores **LBS** (Location Based Services) con lo cuál no sería necesario usar un dispositivo que incluya GPS o un módulo GPS aparte ya que para el nivel del proyecto es más que suficiente.

3.1.6 LoRaWan

- **Alcance:** rango de 10km hasta más de 20 kilómetros (condiciones muy favorables)
- **Consumo energético:** muy bajo
- **Velocidad de transferencia:** 0.3- 50kbps
- **Precio:** bajo (módulos de LoRa baratos desde 10€ y si hay cobertura de Gateway en la zona, si no lo hay será necesario instalar una o las necesarias y su precio ronda los 100 €)
- **Versatilidad:** Complejidad en el uso, aún es un estándar en desarrollo con documentación en desarrollo

3.1.7 802.11/Wi-Fi

Teniendo en cuenta el estándar IEEE 802.11 b/g/n que suele ser el más extendido y barato que es vendido en módulos y placas Wi-Fi:

- **Alcance:** rango de 100 metros hasta varios kilómetros (con repetidores)
- **Consumo energético:** bajo-medio
- **Velocidad de transferencia:** 10-100+ Mbps (600Mbps máximo teórico)
- **Precio:** muy bajo
- **Versatilidad y facilidad de uso:** es un estándar muy utilizado con mucha documentación al respecto y con compatibilidad muy extendida.

3.2 Elección de la tecnología inalámbrica

3.2.1 Conclusión

Teniendo en cuenta que con la mayoría de tecnologías son necesarias gateways o una estación base ya que se comunican con protocolos distintos a TCP/IP y después de analizar las facetas más importantes de cada una se opta por usar la tecnología **GSM/GPRS** y **Wi-Fi** ya que eso facilitaría el no tener que usar una estación base, en el caso de que la hubiera, si está fallase no sería posible que ninguno de los nodos sensores pudiera trabajar correctamente al necesitar de la estación para el envío de datos.

De esta manera son independientes y si alguno falla será problema del nodo en cuestión y no repercutirá en los demás, algunos de los motivos principales para el uso de las dos tecnologías son estos:

- **Bajo coste**, la tecnología Wi-Fi es barata y en el caso de GPRS hoy en día resulta más barato y accesible.
- **Velocidad de transferencia**, más que suficiente.
- **Alta fiabilidad**, Wi-Fi implementa seguridad y fiabilidad, GPRS tiene mayor estabilidad y cobertura que otros estándares de redes móviles.
- **Consumo energético bajo**: En los dos casos, mientras uno funciona el otro estará apagado o en modo bajo consumo y son del orden de mA en funcionamiento normal.

3.3 Tecnologías de implementación del sistema

3.3.1 Introducción

Aunque existen muchos tipos de microcontroladores, aquí se van a mostrar sólo los considerados y más conocidos que abarcan perfectamente la posibilidad de usarse en nuestra solución, pero vamos a analizar que necesitamos realmente y nos ofrece más ventajas.

3.3.2 Arduino

Arduino es una plataforma electrónica de código abierto basada en hardware y software de fácil uso. Las placas Arduino son capaces de leer entradas en un sensor y convertirlo en una salida, activando un motor, encendiendo un LED. Para hacerlo, se utiliza el lenguaje de programación Arduino (basado en Wiring), y el Software Arduino (IDE), basado en Processing.

Arduino ofrece un par de ventajas frente a otros microcontroladores que también se envuelven en un paquete y son las siguientes:

- **Multiplataforma**. El software Arduino (IDE) funciona en los sistemas operativos Windows, Macintosh OSX y Linux. La mayoría de los sistemas de microcontroladores se limitan a Windows.
- **Fácil de utilizar**. Cualquiera puede usarla, aun teniendo pocos conocimientos de programación, electrónica.
- **Plataforma abierta**. Es de código abierto, por lo que tiene muchas posibilidades de desarrollo y es escalable.

- **Alta flexibilidad.** Se puede trabajar en todas las plataformas informáticas, o casi todas. Es de código abierto y cuenta con muchas herramientas para facilitar su uso y crear grandes cosas con ello.
- **Un bajo coste.** Por poco dinero puedes hacerte con algún kit de Arduino con los componentes necesarios para iniciarte en él.



Figura 3 Arduino UNO

3.3.3 NodeMCU (ESP8266)

NodeMCU es una pequeña placa Wifi lista para usar en cualquier proyecto. Está montada alrededor del ya conocido ESP8266 [7] y expone todos sus pines en los laterales. Además, ofrece más ventajas como la incorporación de un regulador de tensión integrado, así como un puerto USB de programación. Se puede programar con LUA o mediante el IDE de Arduino.

Dispone de una extensa comunidad y documentación.

Una ventaja con respecto a otros módulos que integran el ESP2866 es que todos sus pines están disponibles en el exterior.



Figura 4 NodeMCU

3.3.4 NodeMCU-32 (ESP-32)

El SoC (System On a Chip) ESP32 de Espressif Systems es la evolución del ESP8266, diseñado para superar a su antecesor en capacidad de procesamiento y conectividad, integra un potente microcontrolador con arquitectura de 32 bits, conectividad Wi-Fi y Bluetooth. El SoM (System on Module) ESP-WROOM-32 fabricado por Espressif integra en un módulo el SoC ESP32, memoria FLASH, cristal oscilador y antena WiFi en PCB.

La plataforma ESP32 permite el desarrollo de aplicaciones en diferentes lenguajes de programación, frameworks, librerías y recursos diversos. Los más comunes a elegir son: Arduino (en lenguaje C++), Esp-idf (Espressif IoT Development Framework) desarrollado por el fabricante del chip, Simba Embedded Programming Platform (en lenguaje Python), LUA. Al trabajar dentro del entorno Arduino podremos utilizar un lenguaje de programación conocido y hacer uso de un IDE sencillo de utilizar, además de hacer uso de toda la información sobre proyectos y librerías disponibles en internet.



Figura 5 ESP-32

3.3.5 Raspberry Pi

La placa Raspberry se utiliza, como Arduino, en entornos de robótica o domótica, pero también como servidor de archivos. Es otra opción dentro del IoT y es muy interesante cuando el objetivo es procesar y tratar muchos datos. Cualquiera de ellos, Arduino o Raspberry, ofrece fórmulas eficaces para multitud de proyectos, pero todavía es difícil establecer su límite al estar en constante evolución.

Viene además con las tecnologías inalámbricas necesarias para comenzar a conectar proyectos:

- 11n Wireless LAN
- Bluetooth 4.0
- Bluetooth Low Energy (BLE)

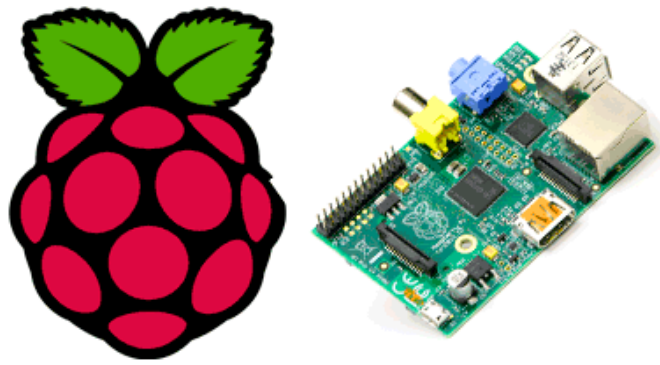


Figura 6 Raspberry Pi

3.3.6 Módulos ESP-XX

Son módulos que incorporan la funcionalidad Wi-Fi, son de bajo coste y pueden ir también incorporados en algunos microcontroladores como en el caso del esp8266.

Podemos incorporarlo en la tecnología que queramos usar para desarrollar el proyecto o bien buscarlo ya soldado en una placa de desarrollo.

En cualquier caso, se tratan de módulos muy de bajo coste que rondan entre 2-9€.

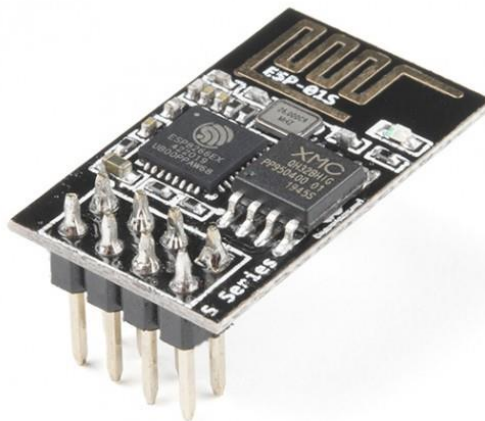


Figura 7 Ejemplo de modelo ESP-01

3.3.7 Módulo GSM/GPRS SIM800L

Es un módulo que hace uso de la tecnología de red móvil que permite la transmisión GPRS, enviar y recibir SMS, además de recibir y hacer llamadas de voz. Es el más barato y también ampliamente utilizado por su tamaño y facilidad.

- Fuente de voltaje: 3.7V – 4.2V.
- Voltaje recomendado: 4V.
- Consumo de corriente:
- sleep mode < 2.0mA.
- idle mode < 7.0mA.
- GSM transmission (avg): 350 mA.
- GSM transmission (peek): 2000mA.



Figura 8 Módulo SIM800L

3.3.8 Módulo GSM/GPRS SIM900

Al igual que el anterior es un módulo que hace uso de la tecnología de red móvil que permite la transmisión GPRS, enviar y recibir SMS, además de recibir y hacer llamadas de voz.

Es una tarjeta compacta, que además incluye cambio de nivel TTL el dispositivo tiene disponibles pines para comunicarse con microcontroladores con niveles lógicos (RX y TX) 3.3V y 5V, con lo que ofrece más facilidad de uso.

Además, es compatible con todos los modelos de Arduino con el formato UNO, es decir, que la puedes controlar con otros microcontroladores también.

Ideal para sistemas remotos, comunicación recursiva, puntos de control, mandar mensajes de texto a móviles, etc.



Figura 9 SIM900

3.4 Elección de la tecnología de implementación

3.4.1 Conclusión

Tomando en cuenta las necesidades y características de los microcontroladores se considera utilizar los dispositivos “NodeMCU” y el módulo “SIM800L”.

NodeMCU:

- Bajo coste, desde 3 euros puede estar disponible.
- Gran cantidad de documentación disponible en la red.
- Código abierto considerando Lua y Arduino, puede programarse como si fuera un Arduino.
- Contiene el módulo de ESP8266 para conexión Wi-Fi.
- Flexibilidad, alta calidad.
- USB-TTL incluido y Plug & Play
- Regulador de tensión 3.3V integrado perfecto para alimentar el sensor biométrico y módulos adicionales que la mayoría trabajan a este voltaje.

Y lo más importante, el procesamiento y características es suficiente con lo cual facilitan un consumo energético bajo.

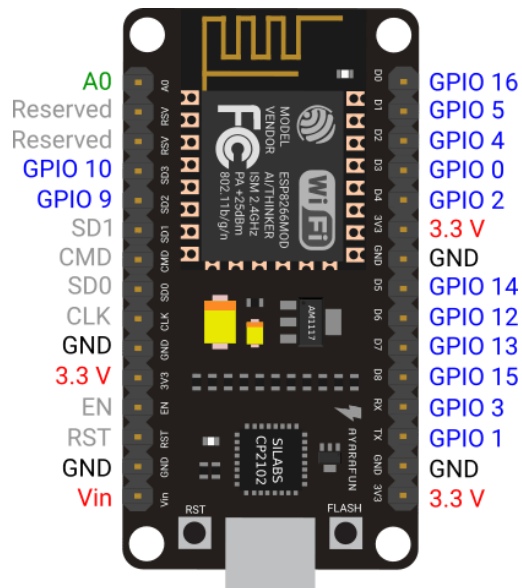


Figura 10 Esquema pinout del NodeMCU

SIM800L:

- Bajo coste, desde 2€ con antenas incluidas
- Documentación de comandos AT, soporte fabricante y gran comunidad programadores
- Voltaje: 3,7-4,2 V, con lo que es posible alimentarlo con una batería a la vez que el NodeMCU
- Cumple con el cometido de poder hacer peticiones a través de internet con GPRS (comandos AT para hacer peticiones HTTP [5])
- Y también es posible geolocalizar el dispositivo simplemente usando servicios **LBS**



Figura 11 Esquema pinout del SIM800L

Sensores biométricos

3.4.2 Sistema biométrico

Los sistemas biométricos [6] se utilizan para la identificación automática de personas mediante el uso de características físicas del individuo o de su comportamiento. Estas pueden ser su cara, el iris de los ojos o sus huellas dactilares: Son rasgos únicos e intransferibles de cada persona.

Las ventajas de un sistema biométrico son evidentes:

- Una identificación segura y única del individuo.
- El "código" de identificación es intransferible. Solamente la persona autorizada es identificada como tal.
- El código biométrico ni se puede perder ni se puede olvidar, pues la persona autorizada siempre lo lleva consigo.
- Un sistema de huellas dactilares tiene coste cero en consumibles y mantenimiento.

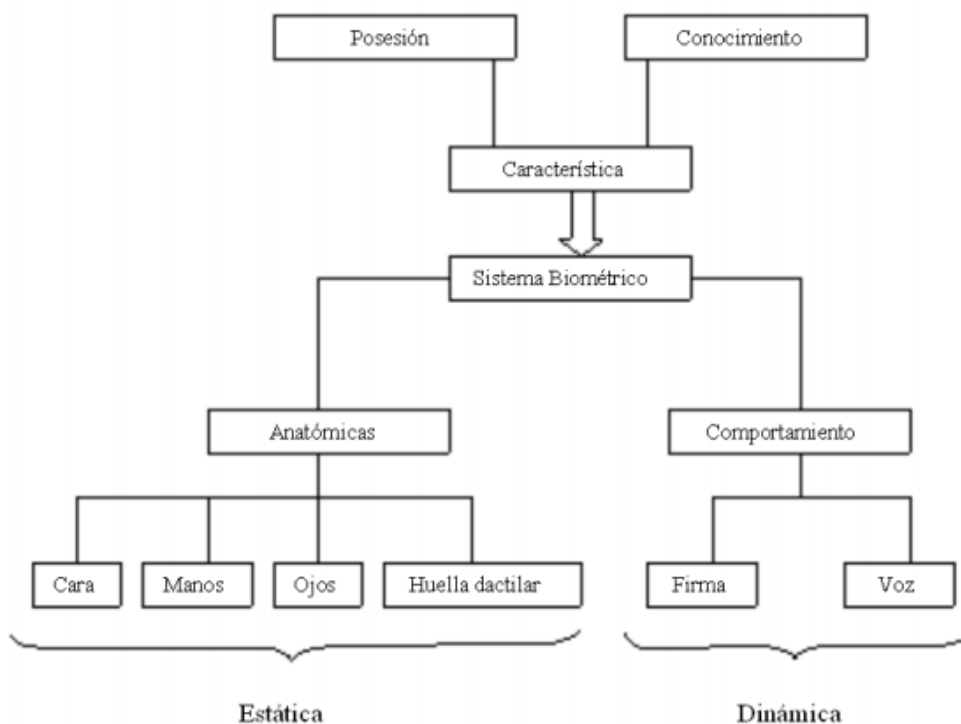


Figura 12 Tipos de sistemas biométricos

3.4.3 Características de un sistema biométrico

Las características básicas que un sistema biométrico para identificación personal debe cumplir son: desempeño, aceptabilidad y fiabilidad. Las cuales apuntan a la obtención de un sistema biométrico con utilidad.

- **DESEMPEÑO:** Esta característica se refiere a la exactitud, la rapidez y la robustez alcanzada en la identificación de individuos por parte del sistema biométrico. Su objetivo es comprobar si el sistema posee una exactitud y rapidez aceptable con un requerimiento de recursos razonable.
- **ACEPTABILIDAD:** Indica el grado en que la gente está dispuesta a aceptar un sistema biométrico en su vida diaria. Dicho sistema no debe representar peligro alguno para los usuarios por lo cual deberá ser un sistema de fácil uso y que inspire confianza a los usuarios finales.
- **FIABILIDAD:** Esta característica refleja cuán difícil es burlar al sistema. Para que el sistema biométrico sea fiable cien por ciento debe reconocer características de una persona viva, pues es posible crear dedos de látex, grabaciones digitales de voz, prótesis de ojos, entre otros, para burlar la seguridad del sistema y obtener acceso al lugar deseado.
- **UNIVERSALIDAD:** Todos los usuarios tienen que tener la característica que solicitar el sistema biométrico en cuestión.
- **UNIVOCIDAD:** La característica a medir debe tener carácter único y distintivo, ninguna otra persona debe tenerlo.

3.4.4 Tipo de sistema biométrico a emplear

De forma muy resumida y según los apartados anteriores el proyecto se centrará en usar el tipo de sistema biométrico estático, ya que usaremos una característica física univoca que es la huella dactilar del usuario.

A partir de aquí, en el siguiente apartado vamos a ver un resumen del funcionamiento de un sensor biométrico.



Figura 13 Ejemplo de captura dactilar

3.4.5 Funcionamiento sensores de huella dactilar

El procedimiento de comparación de huellas en un sensor es prácticamente el mismo sea cual sea el tipo de sensor, al colocar el dedo se obtiene una imagen de la huella y se calcula el patrón, después se hace una comparación con los patrones que están almacenados en la memoria (1: N).

Para no analizar por completo la imagen se usan las **minucias** que son unos rasgos específicos de las huellas. Se utilizan algoritmos muy complejos para reconocer y analizar estos rasgos, para obtener una coincidencia sólo necesita un número de patrones que haya en común y depende del algoritmo aplicado.

Cuando se registra una huella se genera un **template** y respecto a este existirá un **nivel de confianza o de coincidencia** cuando colocamos el dedo (**query**), esto ocurre porque difícilmente volverá a suceder exactamente el mismo resultado si colocamos el dedo ahora mismo a cuando lo volvamos a colocar, depende de muchos factores como la posición del dedo, presión aplicada... Además de ciertos patrones que pueden ser comunes entre varios **templates**.

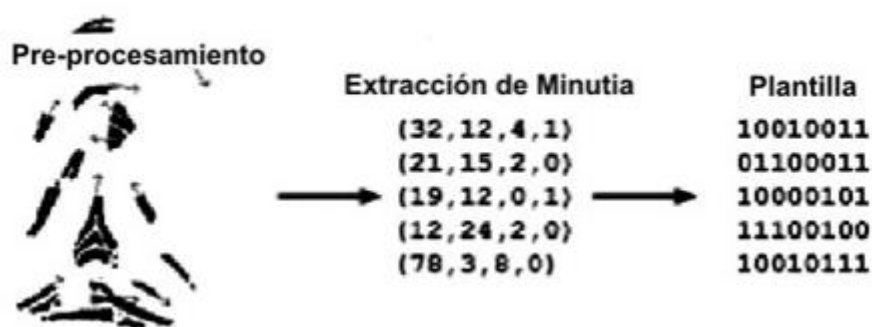


Figura 14 Procesamiento de una huella

3.4.6 Tipos de sensores de huella dactilar

Existen más tipos, pero con estos principales se analizará que dispositivos hay en el mercado utilizables, disponibles y que sea de bajo coste para trabajar con la tecnología escogida en el apartado anterior.

3.4.6.1 Sensores ópticos reflexivos

Este tipo de sensor se basa en una superficie de cristal o prisma que se ilumina con un diodo LED. Al poner el dedo sobre el cristal, la luz del LED es absorbida, las zonas oscuras que producen las crestas de las huellas y la luz resultante se registran como una imagen, que identifica la huella de la persona.

Con este tipo de sensor es necesario tocar la superficie y es propenso a error si hay suciedad o la piel está deteriorada, pero presenta ventaja que es de un coste muy bajo.



Figura 15 Sensor óptico Adafruit [10]

3.4.6.2 Sensores ópticos transmisivos

En este tipo de sensor se diferencia del anterior en que no es necesario tocar la superficie del dispositivo, la luz del sensor pasa por medio del dedo, desde la punta hasta el fin de la huella mientras una cámara captura una imagen de la huella dactilar



Figura 16 Sensor óptico transmisor

3.4.6.3 Sensor de huella térmico

Este tipo de sensor lo que hacen es detectar el calor del dedo, presentan una imagen en diversas escalas de grises y su calidad es muy buena a pesar de que el dedo este sucio, piel deteriorada o con el dedo húmedo.



Figura 17 Sensor de huella térmico

3.4.6.4 Sensores capacitivos

El sensor es un circuito integrado de silicio cuya superficie está cubierta por un gran número de elementos transductores (o píxeles), con una resolución típica de 500 dpi. Cada elemento contiene dos electrodos metálicos adyacentes. La capacidad entre los electrodos, que forma un camino de realimentación para un amplificador inversor, se reduce cuando el dedo se aplica sobre dicha superficie: se reduce más cuando detecta crestas y menos cuando detecta el espacio entre ellas-

El sensor es susceptible a las descargas electrostáticas. Estos sensores sólo trabajan con pieles sanas normales, ya que no son operativos cuando se utilizan sobre pieles con zonas duras, callos o cicatrices. La humedad, la grasa o el polvo también pueden afectar a su funcionamiento.



Figura 18 Sensor de huella capacitivo

3.5 Elección de sensor biométrico

3.5.1 Conclusión

Una vez hemos elegido el microcontrolador que vamos a utilizar, en nuestro caso fue el NodeMCU, podemos empezar a elegir también el tipo de sensor que nos permita trabajar con la placa.

Vamos a optar por el sensor de huella del tipo óptico reflexivo **FZ2904** compatible con **Adafruit** que es muy versátil y fácil de conseguir y además tiene librerías que funcionan con Arduino y con NodeMCU.

Este sensor nos ofrece y es ideal para realizar un sistema con protección de datos por medio de la huella dactilar, realiza procesamiento digital de imágenes (**DSP**) internamente, y compara en su base de datos y actualiza y la capacidad de procesamiento es realmente rápida.

Además, la mayoría se alimentan a 3.3 V lo cual significa que es perfecto para alimentarlo a través de los pines del NodeMCU ya que están hechos para alimentar módulos que normalmente necesitan de este tipo de alimentación.

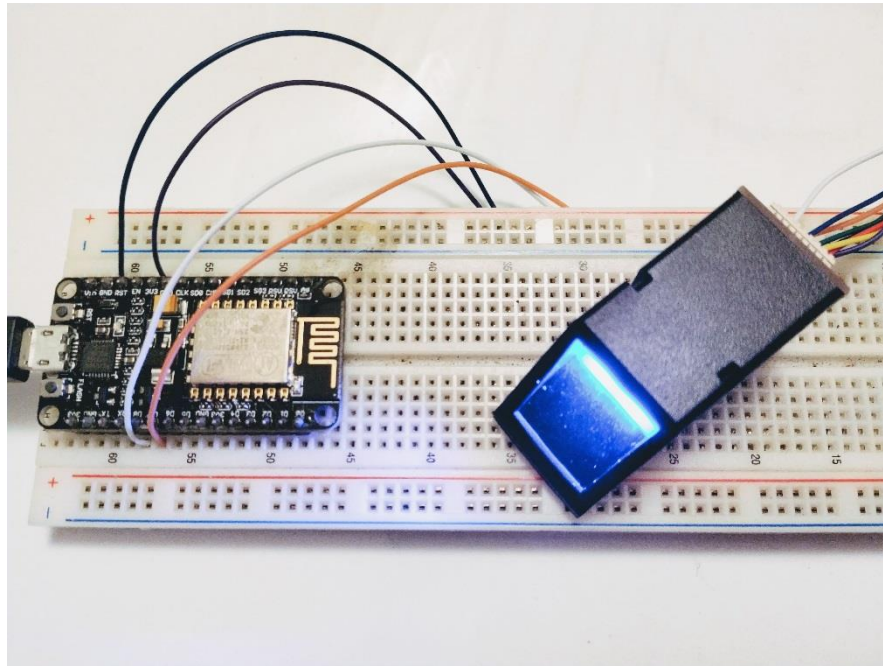


Figura 19 Ejemplo conexión con NodeMCU

3.6 Tecnologías de comunicación y de lado del servidor

3.6.1 Introducción

Vamos a hacer un breve resumen de las tecnologías que vamos a utilizar en este proyecto con la finalidad de poder tener comunicación en un entorno local y posteriormente para en producción, poder comunicarse a través de internet.

3.6.2 Sistema de gestión de bases de datos

Es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos. Los usuarios pueden acceder a la información usando herramientas específicas de consulta y de generación de informes, o bien mediante aplicaciones al efecto.

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe.

Generalmente se accede a los datos mediante lenguajes de consulta, lenguajes de alto nivel que simplifican la tarea de construir las aplicaciones. También simplifican las consultas y la presentación de la información. Un SGBD (Sistema Gestor de Base de Datos) permite controlar el acceso a los datos, asegurar su integridad, gestionar el acceso concurrente a ellos, recuperar los datos tras un fallo del sistema y hacer copias de seguridad. Las bases de datos y los sistemas para su gestión son esenciales para cualquier área de negocio, y deben ser gestionados con esmero.

A continuación, vamos a presentar los más usados que son en su mayoría completamente relacionales

3.6.2.1 MYSQL

Es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo, y una de las más populares en general junto a Oracle y Microsoft SQL Server, todo para entornos de desarrollo web.

La mayoría de sus ventajas son gracias a su recorrido, sencillo de usar, bajo coste de implementación, facilidad de configuración e instalación, rápido en las transacciones y la más importante es la gran documentación y comunidad de apoyo que hay detrás.

3.6.2.2 PostgreSQL

Es un sistema de gestión de base de datos relacional orientado a objetos y libre, publicado bajo la licencia BSD (Berkeley Software Distribution).

Es dirigido por una comunidad de desarrolladores que trabajan de forma libre y/o apoyada por organizaciones comerciales.

Entre sus ventajas están:

- Su instalación y uso es gratis
- Disponibilidad para multiplataforma y diferentes servidores webs como Apache [2]
- Fácil configuración
- Sistema de alta fiabilidad y robustez
- Control de concurrencias multiversión (MVCC)

3.6.2.3 Microsoft SQL Server

Microsoft SQL Server es la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos.

Su principal característica es la alta disponibilidad con una conmutación muy rápida sin usar muchos recursos del sistema gracias a funciones de memoria integrados en los motores de base de datos, mejora la flexibilidad y su uso. Otras características son:

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporte de procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el
- uso de comandos DDL y DML gráficamente.

3.6.2.4 Conclusión

Se decide utilizar **MySQL** ya que es de los más usados gracias a su fácil instalación, administración y bajo costo en infraestructura, así como en licencias.

Además, nos brinda de seguridad, potencia y profesionalidad para el proyecto ya que es de los más documentados.

Así que teniendo en cuenta a MySQL como gestor de base de datos, vamos a describir a continuación en el siguiente apartado la tecnología que usaremos para la implementación en local para el servidor web.

3.6.3 Servidores web

Los servidores web (también conocidos como servidores HTTP) son un tipo de servidores utilizados para la distribución (entrega) de contenido web en redes internas o en Internet.

Son los encargados de servir todo lo que se pide desde un dispositivo cuando alguien visita el servicio web.

En resumen, un servidor web es un equipo informático que satisface peticiones de otras máquinas (clientes) de forma remota que trabaja usando el modelo cliente-servidor.

3.6.3.1 Microsoft Internet Information Services

Este tipo de servidor web está desarrollado por Microsoft y, en el caso de que se necesite integrar herramientas de Microsoft, sería una alternativa recomendable por la compatibilidad que obtendríamos.

Se ejecuta con Windows gracias a la tecnología IIS (Internet Information Services) y es compatible con páginas programadas en ASP (Active Server Pages) o .NET.

3.6.3.2 Servidor web Apache

Apache HTTP Server es un software de servidor web gratuito y de código abierto para plataformas Unix con el cual se ejecutan el 46% de los sitios web de todo el mundo. Es mantenido y desarrollado por la Apache Software Foundation.

Les permite a los propietarios de sitios web servir contenido en la web. Es uno de los servidores web más antiguos y confiables, con la primera versión lanzada hace más de 20 años, en 1995.

Tiene múltiples ventajas, como varias funcionalidades y soporte y es completamente portable y multiplataforma.

Es el más documentado ya que tiene una comunidad muy grande y existe de hace más de 20 años.

3.6.3.3 Nginx

Al igual que con Apache estaríamos ante un software de código abierto y multiplataforma compatible con Window, IOS, Linux, etc. Tiene una versión de pago además de la gratuita.

Este tipo de servidor web es conocido por su buen funcionamiento cuando tiene que gestionar un número alto de visitas simultáneas, ya que los usuarios no perciben retraso en la carga de la página, aunque se esté produciendo ese acceso concurrente.

3.6.3.4 Conclusión

Se decide utilizar Apache por la facilidad de despliegue, funcionalidad y que lo ofrecen todos los servicios de hosting de forma predeterminada.

3.6.4 Herramienta de despliegue y lenguaje de programación de servidor

3.6.4.1 WampServer

WAMP es un acrónimo que significa Windows, Apache, MySQL y PHP. Es un stack o conjunto de soluciones de software que significa que cuando instalas WAMP, estás instalando Apache, MySQL y PHP en tu sistema operativo (Windows en el caso de WAMP). Aunque puede instalarlos por separado, por lo general son empaquetados.

Con lo cual ya hemos dicho rápidamente en que tecnologías vamos a trabajar: Apache (servidor web), MySQL (servidor de base de datos), PHP (lenguaje programación de servidor).

Esto facilita el empezar a trabajar rápidamente desde cero en la implementación del servidor web.

A continuación, vamos a describir de forma breve Apache y PHP que vienen ya con WampServer [1].

3.6.4.2 Lenguaje de lado de servidor PHP

PHP es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web

El código PHP suele ser procesado en un servidor web por un intérprete PHP implementado como un módulo, un daemon o como un ejecutable de interfaz de entrada común (CGI). En un servidor web, el resultado del código PHP interpretado y ejecutado que puede ser cualquier tipo de datos, como el HTML generado o el resultado de una operación formaría la totalidad o parte de una respuesta HTTP. Existen diversos sistemas de plantillas, sistemas de gestión de contenidos y frameworks que pueden emplearse para organizar o facilitar la generación de esa respuesta.

3.6.5 Servidor de hosting

3.6.5.1 000webhost

Es un proveedor de servicios de hosting que tiene un servicio de hosting gratuito para que los usuarios puedan usarlo para probar el servicio y hacer pruebas piloto de sus proyectos.

Pertenece a la compañía de Hostinger que tiene un prestigio y es de las más conocidas como proveedor de servicios web.

Nos ofrece subdominio gratuito con certificado SSL ya funcionando, un servidor web Apache con MySQL.

3.6.5.2 Awardspace

Awardspace es una empresa de hosting con un recorrido muy amplio fundada en 2003, se caracteriza por ser una de las pocas empresas que ofrece hosting gratuito además de los planes de pago, dando la posibilidad de probar el servicio o que una persona con pocos recursos pueda usarlo.

3.6.5.3 Infinityfree

Es un proveedor de hosting gratuito que ofrece 99,9% de tiempo de actividad, dominio gratuito, SSL gratuito, almacenamiento ilimitado.

Pero con muchas limitaciones en cuanto a recursos, es una buena opción para actualizar a un plan de pago y probar su servicio con más detenimiento.

3.6.5.4 Clever cloud

Clever cloud es una empresa que ofrece servicios en la nube y también máquinas virtuales dispuestas a funcionar como servidores de diferentes propósitos.

Lo interesante que ofrece es que con solo registrarse y con un click nos ofrece una base de datos MySQL totalmente gratis y segura para producción que además cada noche hace backups y mantiene totalmente tus datos a salvo y con soporte al usuario disponible.

3.6.5.5 Conclusión

Se utilizará para que pueda ser accesible a través de internet un hosting web, en concreto “**000webhost**” que es gratuito y cumple para el desarrollo del proyecto y en él finalmente sólo lo usaremos para tener un script que será el que actúe de “API” para las peticiones del dispositivo.

Aunque podemos subir también y dejar accesible la web para la gestión del sistema de control horario, lo dejaremos en local para así posteriormente mostrar que se han usado todas las herramientas de desarrollo aquí explicadas en este apartado.

Sin embargo, para la base de datos MySQL usaremos aparte otro servicio, ya que la disponibilidad de las bases de datos de “000webhost” no suele ser muy adecuada y es propensa a fallos.

“**Clever Cloud**” nos ofrece una base de datos MySQL totalmente gratis y disponible desde cualquier lugar [11] lo suficiente para el proyecto y con una disponibilidad del 99%, respecto a la de “**000webhost**” que es más propensa a fallos con un 97.103% de disponibilidad (agosto 2021) [12].

3.7 Herramientas de desarrollo IDE

3.7.1 Visual Studio Code

Es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para depuración, control de Git integrado, resaltado de sintaxis, finalización de código inteligente, fragmentos de código y refactorización de código.

También es personalizable, de modo que los usuarios pueden cambiar el tema del editor, los métodos abreviados de teclado y las preferencias. Es gratuito y de código abierto.

Visual Studio Code [3] es una herramienta que tiene soporte nativo para gran variedad de lenguajes, entre ellos PHP, Javascript y etiquetado HTML, CSS que será lo que usaremos.

Goza de un soporte técnico formidable pues debido a su frecuente uso por la comunidad de desarrolladores, se puede encontrar fácilmente documentación y ayuda en foros y sitios relacionado

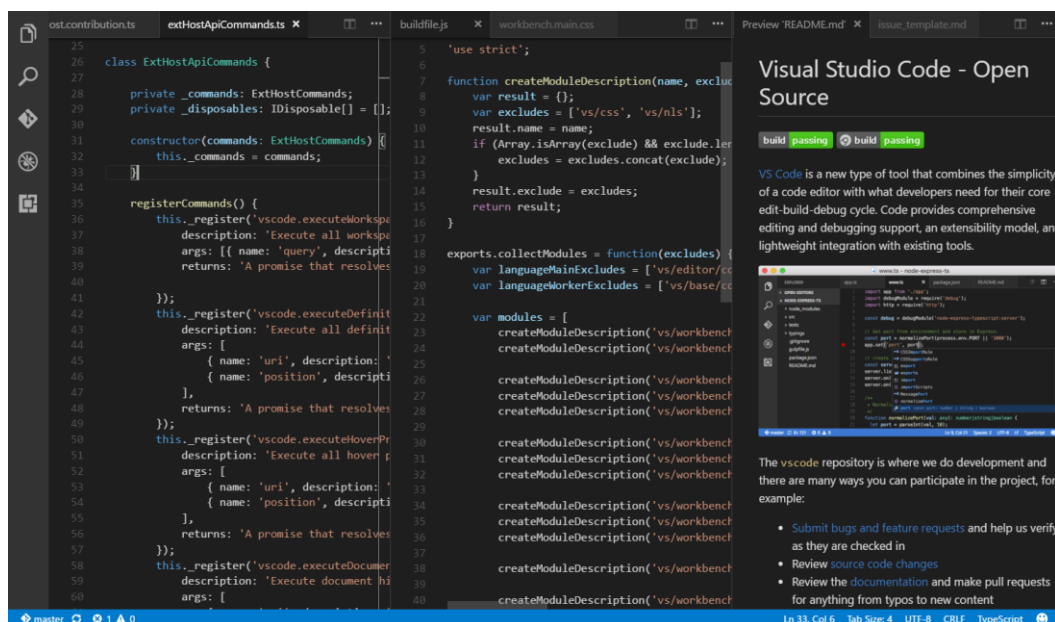


Figura 20 Ejemplo de interfaz Visual Studio Code

3.7.2 Arduino IDE

El entorno de desarrollo integrado (IDE) de Arduino es una aplicación multiplataforma (para Windows, macOS, Linux) que está escrita en el lenguaje de programación Java. Se utiliza para escribir y cargar programas en placas compatibles con Arduino, pero también, con la ayuda de núcleos de terceros, se puede usar con placas de desarrollo de otros proveedores.

El código fuente para el IDE se publica bajo la Licencia Pública General de GNU.

El IDE de Arduino admite los lenguajes C y C ++ utilizando reglas especiales de estructuración de códigos.³ El IDE de Arduino suministra una biblioteca de software del proyecto Wiring, que proporciona muchos procedimientos comunes de E/S.



Figura 21 Ejemplo interfaz Arduino IDE [4]

3.8 Material hardware adicional para el diseño del sistema

3.8.1 Pantalla OLED 0.96

Vamos a usar para tener algún tipo de feedback hacia el usuario con una pantalla OLED [8] que además es de muy bajo consumo, muy barata y muy ampliamente utilizada también.

De esta manera podremos informar al usuario si hay algún fallo o si ha conseguido fichar correctamente e incluso si hay algún problema y el sensor no está funcionando correctamente.



Figura 22 Display OLED 0.96

3.8.2 Batería LiPo 3.7V 1500mA

Utilizaremos una batería para el proyecto que mantenga en funcionamiento la solución un tiempo prologando en el que no haya forma de alimentación externa.



Figura 23 Batería LiPo 3.7V

3.8.3 Mini DC-DC 5V

Este pequeño dispositivo lo usaremos para poder alimentar el NodeMCU ya que no es posible alimentarlo directamente con una batería de 3.7V, se subirá voltaje a 5V para que el regulador de tensión interno del NodeMCU funcione correctamente. Es muy barato (0.24€) y cumple con una eficiencia muy buena.

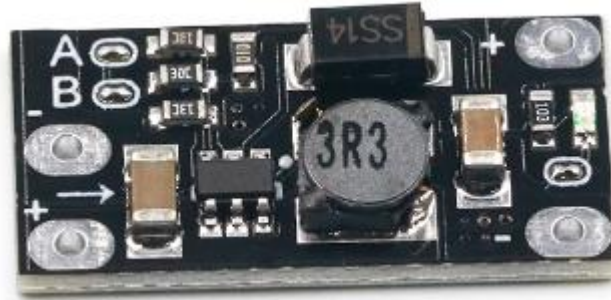


Figura 24 Regulador voltaje step up 5V

3.8.4 Cargador USB regulador voltaje a 3.7V

Este pequeño dispositivo lo usaremos para cargar la batería cuando se necesite mediante un cable micro USB. Puede proporcionar 1 A al circuito mientras está cargando también la batería.

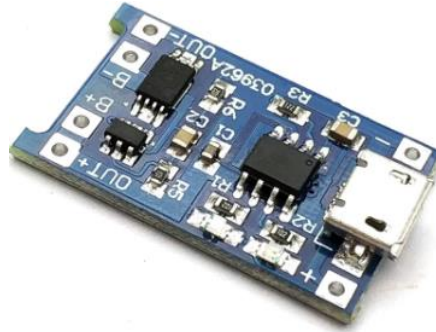


Figura 25 Cargador USB para batería 3.7V

4 IMPLEMENTACIÓN

En este capítulo, veremos cómo se ha implementado y el procedimiento normal para desplegar un nodo y permitir su funcionamiento correctamente.

Las funciones que se nombren del sketch de Arduino IDE se detallarán más adelante.

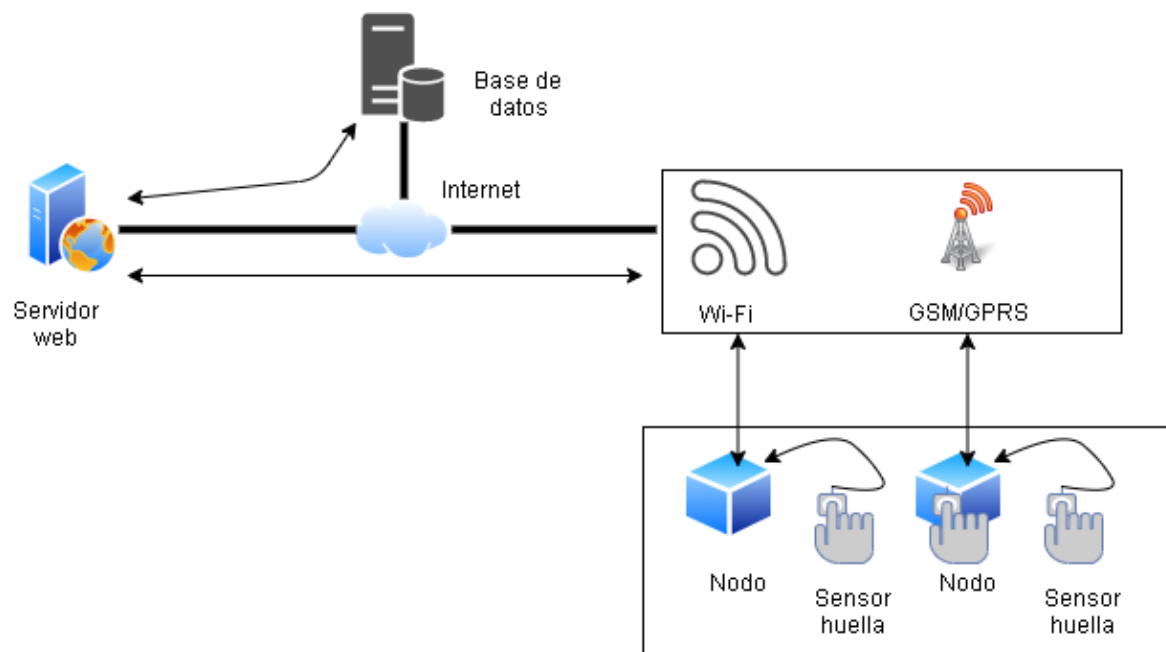


Figura 26 Ejemplo de representación de la solución

4.1 Implementación en el servidor web

Al no usar un framework, tenemos PHP y HTML en el mismo archivo. Así que no hay distinción de ningún tipo entre archivos. También usamos javascript/jquery para hacer peticiones AJAX y recuperar datos de tablas o hacer llamadas a ficheros para consultas.

4.1.1 Registro de un dispositivo/nodo

Antes de registrar usuarios en un nodo, debemos de crearlo, y para ello nos vamos a la vista de **devices.php** y creamos un dispositivo añadiéndole un nombre y un departamento o área de trabajo algo que lo distinga perfectamente, por ejemplo, si queremos hacer registro de la jornada laboral del “Departamento de I+D” le pondremos ese nombre para correctamente identificarlo de manera más clara y luego poder asociar y registrar usuarios fácilmente a la hora de seleccionar el departamento. Al hacer click en crear dispositivo se llamará a **dev_conf.php** que es el que hará la validación del formulario y hará la query para insertar en la base de datos.

Figura 27 Modal del formulario para crear dispositivo

Cuando se crea un dispositivo, se genera un **token** que debe ser único, este se añadirá en el código del sketch de forma manual para a la hora de hacer peticiones HTTP al servidor que actúa de API, la información que se devuelva y las operaciones que se hagan correspondan con ese sensor.

Antes de registrar un usuario con su huella en el sensor debemos tener la posibilidad de poder hacer un registro del usuario con datos que nos permitan identificar la huella de ese usuario en el sensor y que se almacenen en el servidor MySQL, para así poder tener registro de marcas y podamos saber a quién pertenecen.

4.1.2 Registro de usuarios

Para el registro de usuarios tendremos la vista **ManageUsers.php**.

La cuál se muestra los usuarios que están registrados. El dispositivo NodeMCU comprobará cada cierto tiempo si hay nuevos registros (**CheckNuevoID ()**) y si lo hay comenzará el proceso de registro de la huella dactilar para ese usuario.

GESTIÓN USUARIOS							
Añadir							
FINGER_ID	NOMBRE	ROL	CÓDIGO USUARIO	EMAIL	FECHA DE REGISTRO	DEPARTAMENTO	ACCIONES
1	Salva2	rol	953000001	std@gmail.com	2021-03-23 00:00:00	Domitorio	 

Figura 28 Vista de gestión de usuarios

Aquí la parte del registro se hace pulsando en el botón añadir y aparece el modal **registrarUsuario** con el siguiente formulario:

Registrar datos de usuario

1 Area de usuario

Area/departamento:

Selecione un departamento

2 Info usuario

Nombre de usuario...

Codigo de usuario...

Email de usuario...

3 Info adicional

Cargo que desempeña...

Cerrar Guardar

Figura 29 Modal registro de usuario

Si nos fijamos en el ejemplo anterior en el apartado de dispositivos, dijimos de crear un “Departamento de I+D”, si lo creamos correctamente y lo seleccionamos, podremos crear un usuario para ese dispositivo. Al hacer click en **Guardar** se llamará al script **manage_users_conf.php** es el que hará validaciones del formulario y calculará con un procedimiento almacenado en la base de datos que ID hay disponible y ejecutará las consultas junto con los datos introducidos en el formulario.

```

manage_users_conf.php > ...
$stmt = $conn->prepare("SET @dep = ?");
$stmt->bind_param('s', $dev_name);
$stmt->execute();

$stmt = $conn->prepare("SET @newID = ''");
$stmt->execute();

// Mi sensor tiene maximo 127 huellas en memoria, nos aseguramos que solo podamos añadir en el rango de valores 1-127
$result = $conn->query('call IDMAX_127(@dep, @newID)');

$r = $conn->query('SELECT @newID as new_ID');

if ($row = $r->fetch_assoc()){

    $fingerid = $row["new_ID"];
    $stmt->close();

}

else{
    echo "No es posible añadir más usuarios al dispositivo: Memoria completa";
    exit();
}

```

Figura 30 Comprobación de ID disponible mediante procedimiento almacenado dentro del rango del sensor de huella

En el loop del sketch se comprueba cada cierto tiempo si hay un nuevo registro de usuario para este sensor con la función **CheckNuevoID ()**, si la petición le devuelve una ID, significa que hay un registro y ese ID será el que se usará para registrar a ese usuario en el sensor.

Una vez esté hecho el registro correctamente en el sensor y guardado en su base de datos interna, el NodeMCU mandará una petición HTTP de confirmación con la función **confirmarNuevo (id)** al servidor web de tal manera que se complete el registro en la web del usuario.

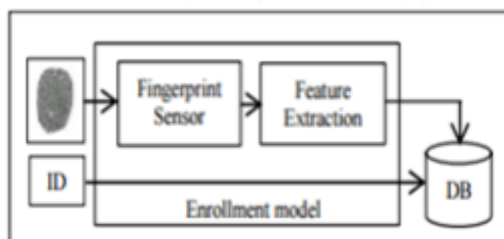


Figura 31 Proceso de registro huella en servidor

4.1.3 Eliminación de usuarios

Para la eliminación de usuarios se marca un campo llamado **del_finger** que por defecto está a '0' y se cambia a 1.

Cuando el NodeMCU compruebe si hay usuarios borrados en el servidor web con la función **CheckBorrarID ()**, el servidor le responderá con el 'ID' de huella dactilar del usuario y ese mismo 'ID' se borra del servidor y el sensor también lo borra de su memoria/base de datos.

```
if (isset($_POST['delete'])) {  
    $fingerid = $_POST['fingerid'];  
    $sql = "SELECT IDFingerPrint FROM usuarios WHERE IDFingerPrint=".$_fingerid;  
    $result = mysqli_stmt_init($conn);  
    if (!mysqli_stmt_prepare($result, $sql)) {  
        echo "SQL_Error_Select";  
        exit();  
    }  
    else{  
        mysqli_stmt_execute($result);  
        $result1 = mysqli_stmt_get_result($result);  
        if ($row = mysqli_fetch_assoc($result1)) {  
            $sql="UPDATE usuarios SET del_finger=1 WHERE IDFingerPrint=".$_fingerid;  
            $result = mysqli_stmt_init($conn);  
            if (!mysqli_stmt_prepare($result, $sql)) {  
                echo "SQL_Error_delete";  
                exit();  
            }  
            else{  
                mysqli_stmt_execute($result);  
                echo "¡Usuario eliminado correctamente!";  
                exit();  
            }  
        }  
        else{  
            echo "¡Refresque el listado, para comprobar que no haya sido borrado!";  
            exit();  
        }  
    }  
}
```

Figura 32 Ejemplo de borrado en el servidor web

4.1.4 Editar usuarios

A la hora de editar usuarios solo repercute en información en el servidor web, podemos modificar los datos del usuario, así como nombre, email, código de usuario y su rol.

No es posible cambiar el 'ID' ya que debe ser el mismo que en el sensor al igual que el departamento que define al dispositivo donde ese usuario está registrado y puede hacer marcajes.

4.1.5 Consulta de marcajes

En esta vista de la web llamada **Userslog.php** se puede consultar y filtrar por fechas, horas, departamentos, incluso usuarios. Las alertas se han hecho en el servidor web a modo de comprobar la funcionalidad de una forma sencilla y con un ejemplo de jornada laboral de 8 horas con una tolerancia de 5 min en el registro de la jornada.

Administración de usuarios Dispositivos

REGISTRO DE MARCAJES

Filtro de log/ Exportar a Excel

ID	NOMBRE	CODIGO DE USUARIO	ID FINGERPRINT	FECHA DE MARCADO	HORA DE ENTRADA	HORA DE SALIDA	DISPOSITIVO	ALERTA
1	Salva2	953000001	1	2021-08-02	11:26:36	19:29:09	Dormitorio	OK
1	Salva2	953000001	1	2021-08-02	08:33:00	20:00:00	Dormitorio	Horario de jornada superado
1	Salva2	953000001	1	2021-08-15	16:50:02	00:00:00	Dormitorio	Horario de jornada incompleto

Figura 33 Vista consulta de marcajes de usuarios

En la siguiente figura podemos ver además la información de localización que se registró en ese día en el dispositivo y se usó en estos marcajes que no se muestra en la consulta web personalizada.

ID	IDUsuario	FechaInOut Fecha de marcado del usuario	HoraEntrada Hora de marcado de entrada del usuario	HoraSalida Hora de marcado de salida del usuario	Tiempo_E_S	lat_marcaje	lng_marcaje
105	1	2021-08-02	11:26:36	19:29:09	08:02:33	38.0468407	-4.0389018
106	1	2021-08-02	08:33:00	20:00:00	08:11:00	38.0468407	-4.0389018
123	1	2021-08-15	16:50:02	00:00:00	NULL	38.0397186	-4.0413332

Figura 34 Vista consulta de marcajes de usuarios desde gestor de base de datos

```

$sql = "SELECT u.ID,u.NombreUsuario,u.CodigoUsuario, u.IDFingerPrint,u.disp_dep,ul.FechaInOut,ul.HoraEntrada,ul.HoraSalida,
ul.Tiempo_E_S from usuarios u
INNER JOIN usuarios_logs ul ON u.ID=ul.IDUsuario WHERE ".$SESSION['searchQuery']."," ORDER BY id DESC";
$result = mysqli_stmt_init($conn);
if (!mysqli_stmt_prepare($result, $sql)) {
    echo '<p class="error">SQL Error</p>';
}
else{
    mysqli_stmt_execute($result);
    $result1 = mysqli_stmt_get_result($result);
    if (mysqli_num_rows($result1) > 0){
        while ($row = mysqli_fetch_assoc($result1)){
            <?>
            <TR>
            <TD><?php echo $row['ID'];></TD>
            <TD><?php echo $row['NombreUsuario'];></TD>
            <TD><?php echo $row['CodigoUsuario'];></TD>
            <TD><?php echo $row['IDFingerPrint'];></TD>
            <TD><?php echo $row['FechaInOut'];></TD>
            <TD><?php echo $row['HoraEntrada'];></TD>
            <TD><?php echo $row['HoraSalida'];></TD>
            <TD><?php echo $row['disp_dep'];></TD>
            <TD><?php
            $tiempo=strtotime('1970-01-01'.$row['Tiempo_E_S']);
            if($tiempo > 8.08*60*60)//8 horas de jornada con 5 min de tolerancia
            echo "<div style='background-color:red;'><p style='color:orange'>Horario de jornada superado</p></div><br>";
        }
    }
}

```

Figura 35 Ejemplo código de consulta de marcajes con filtro

4.1.5.1 Algoritmo cumplimiento jornada laboral

Como hemos explicado en el apartado anterior a modo de comprobar la funcionalidad de una forma sencilla y con un ejemplo de jornada laboral de 8 horas con una tolerancia de 5 min en el registro de la jornada. Este sería el algoritmo sencillo a expensas de modificaciones y mejora en el futuro.

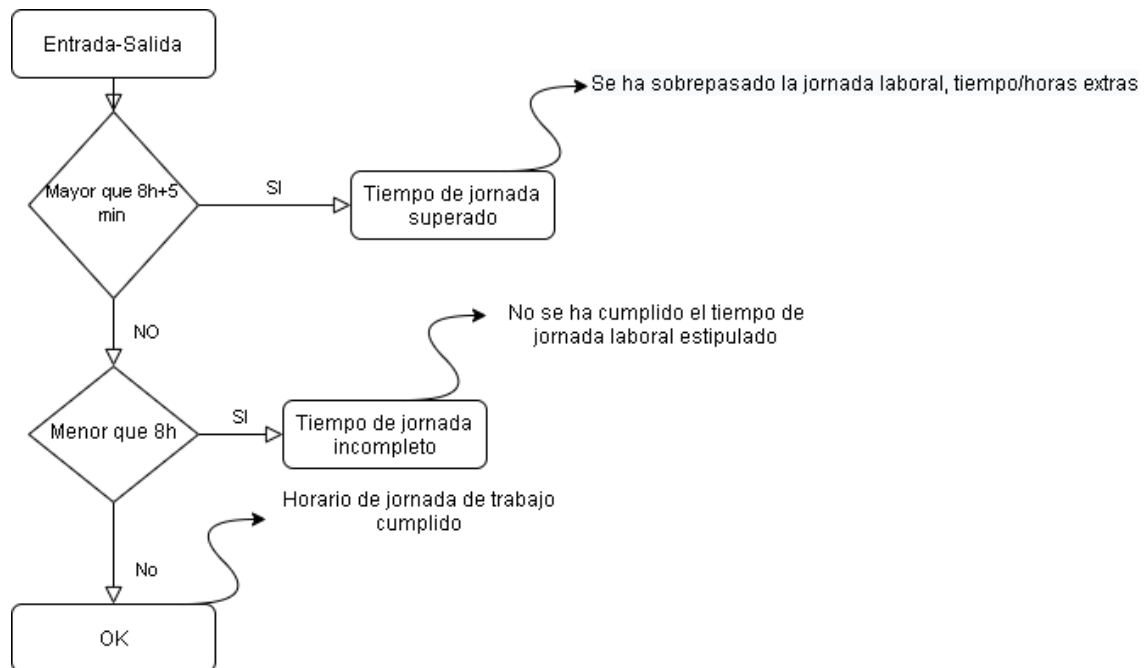


Figura 36 Ejemplo sencillo de aplicación de una regla para alertas

4.2 Implementación del nodo inalámbrico

Vamos a explicar cómo hemos hecho anteriormente con la implementación del servidor web de forma muy resumida, desde el punto de vista del nodo y detallando algunas funciones del código del sketch.

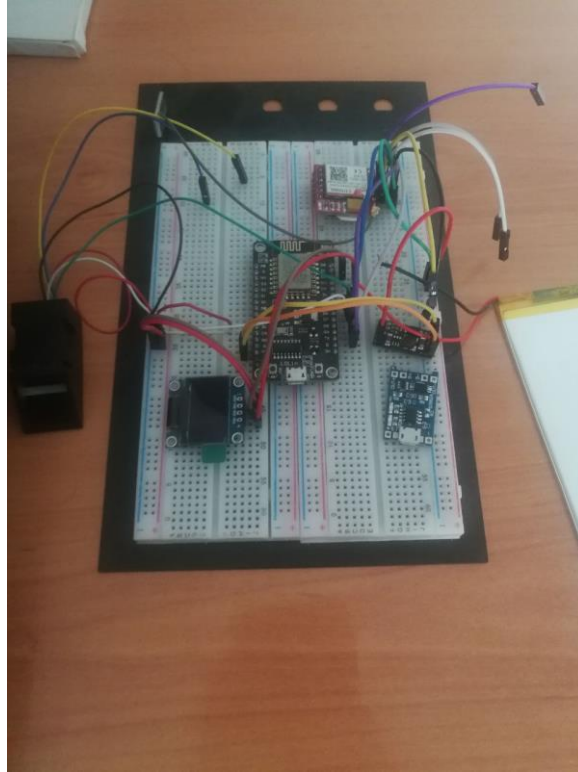


Figura 37 Ejemplo de implementación sin soldar en placa board

4.2.1 Inicio de parámetros esenciales

Aquí podemos ver las librerías que se van a usar que son más que suficientes.

```
//NodeMCU-----  
#include <Adafruit_Fingerprint.h>  
#include <SPI.h>  
#include <Wire.h>  
#include <ESP8266WiFi.h>  
#include <SoftwareSerial.h>  
#include <ESP8266HTTPClient.h>  
#include <WifiLocation.h>  
#include "Utils.hpp"  
  
//OLED-----  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>
```

Figura 38 Librerías usadas NodeMCU

4.2.2 Conexión a la red Wi-Fi

Para la red Wi-Fi definimos unas constantes con las credenciales, también podemos ver el token del dispositivo para que pueda hacer peticiones al servidor y además la dirección web que apunta al script que hace de “API” para nuestros nodos.

```
#define OLED_RESET    0 // Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

WiFiClient client;

const char* ssid      = "TP-Link_E7EC";
const char* password = "          ";

const char* device_token = "01fe0c5149c62 ";

String getData, Link;
String URL = "https://tfgsalva.000webhostapp.com/api/getdata";
String results; //Para modulo GSM

int tipo = 0; // WIFI=1 , GSM=2
int FingerID = 0;
uint8_t id;

#ifdef __AVR__ || defined(ESP8266) && !defined(__AVR_ATmega2560__)
```

Figura 39 Credenciales Wi-Fi, token nodo y URL API web

La función para conectar a una red inalámbrica se llama **ConnectWiFi_STA**, para darle tiempo al módulo wifi en caso de tardanza o problemas de cobertura que hagan más lenta la conexión le damos unos 15 segundos para intentar conexión Wi-Fi al punto de acceso, en el caso que no sea posible pasará a intentar conectar con el módulo GSM.

```
int recon = 0;

while (WiFi.status() != WL_CONNECTED && recon < 30) //15 segundos
{
    delay(500);
    //Serial.print('.');
    recon++;
} //Informamos de la conexión WIFI en el caso que haya conexión, s.
if (WiFi.isConnected()) {
    //arranque con un soft reset de la red
    Serial.println("AT+CFUN=0"); // <--- Modo minima funcionalidad
    //Serial.println("Conectado a red WIFI");

    display.clearDisplay();
    display.setTextSize(2);           // Normal 1:1 pixel scale
    display.setTextColor(WHITE);      // Draw white text
    display.setCursor(8, 0);          // Start at top-left corner
    display.print(F("Conectado \n"));
    display.drawBitmap( 33, 15, Wifi_connected_bits, Wifi_connected
    display.display();
    return 1;
}
```

Figura 40 Conectar el nodo a una red Wi-Fi

4.2.3 Conexión a la red GPRS

La función para conectar a la red GPRS es la misma que se usa para Wi-Fi **ConnectWiFi_STA**, en el caso que no sea posible por red Wi-Fi pasará a intentar conectar con el módulo GSM.

No ha sido posible usar librerías que funcionaran perfectamente con el modulo así que se ha simplificado el proceso enviando los comandos AT necesarios para cada función de la manera más sencilla posible y que no tome mucha demora en el dispositivo.

Si el dispositivo que en ese momento estará en modo de mínima funcionalidad, responde al comando “AT” con un **OK**, se puede proceder a habilitarlo al modo full function (**AT+CFUN=1**) [9].

A modo de resumen, se aplica la APN necesaria para la conexión GPRS, en este caso es una tarjeta SIM de Simyo, se habilita GPRS con “**AT+CIICR**”, se comprueba que tiene asignada IP con el comando “**AT+CIFSR**” y los siguientes dos comandos son para asegurar la conexión a la red GPRS y por último usamos **AT+HTTPINIT** para tener preparado el dispositivo para las peticiones al servidor

```
else if(results.indexOf("OK")>= 0 ) {
    //Serial.println("No disponible para conectarse a red WIFI:");
    //Serial.println(WiFi.status()); // Para depurar y ver codigo de estatus en caso de :
    WiFi.mode(WIFI_OFF);

display.clearDisplay();
display.setTextSize(1);           // Normal 1:1 pixel scale
display.setTextColor(WHITE);      // Draw white text
display.setCursor(0, 0);          // Start at top-left corner
display.print(F("Conectando... \n"));
display.setCursor(0, 50);
display.setTextSize(2);
display.print("GPRS net");
display.drawBitmap( 73, 10, Wifi_start_bits, Wifi_start_width, Wifi_start_height, WHITE);
display.display();

Serial.println("AT+CFUN=1"); //Habilitamos modo por defecto (full function)
delay(20000); //Vamos a darle 15 segundos para iniciarse y conectar a la red GSM

Serial.println("AT+CSTT=orangeworld");
delay(2000);
Serial.println("AT+CIICR");
delay(4000);|
```

Figura 41 Conexión a red GPRS 1

```

Serial.println("AT+CIFSR");
delay(5000);
Serial.println("AT+SAPBR=3,1,\"APN\", \"orangeworld\"");
delay(3000);
//toSerial();
Serial.println("AT+SAPBR=1,1");
delay(2000);
//toSerial();
Serial.println("AT+HTTPIINIT");
//AT+HTTPSSL para habilitar HTTPS pero para pruebas y demostracion del proyecto se dejará con el puerto 80
return 2;

```

Figura 42 Conexión a red GPRS 2

Ejemplo que se repite para todas las funciones (registrar, eliminar, confirmar, marcajes) estando con la conexión GSM. Solo cambian en la manera de extraer la respuesta del servidor para informar al usuario por el display.

```

} else{ // petición con conexion gsm

getData = "?Get_Fingerid=get_id&device_token=" + String(device_token);
Link = URL + getData;

Serial.println("AT+HTTPPARA=\"URL\", \""+Link+"\"");
delay(2000);
Serial.println("AT+HTTPACTION=0");
delay(3000);
Serial.read();
Serial.println("AT+HTTPREAD");
delay(2000);

results=updateSerial(results);

int index = results.indexOf('[');
int index2 = results.indexOf(']');

```

Figura 43 Ejemplo comprobación registro usuario con red GSM

4.2.4 Registro de usuarios

En este caso y como mencionado en la parte del servidor, tenemos un trigger se dispara en un determinado tiempo y hará la comprobación de si hay un nuevo registro en el servidor de un usuario y proceder a registrar su huella en el sensor y esa función se llama **CheckNuevoID ()**.

Se comprueba si está conectado a la red Wi-Fi y se hace la petición servidor con los correspondientes parámetros, si la respuesta contiene “**nuevoID**” se recoge el número y se llama a la función de la librería de Adafruit Fingerprint llamada **getFingerprintEnroll ()** que es la que se encarga de empezar a funcionar en modo registro.

```

//*****Aquí comprobamos si hay algún ID recién añadido en el servidor para un usuario
//y pasamos a llamar a la función para registrar la huella en el dispositivo*****
void CheckNuevoID() {

    //Serial.println("Check to Add ID");
    if (WiFi.isConnected()) {
        HTTPClient http;    //Declare object of class HTTPClient
        //GET Data
        getData = "?Get_Fingerid=get_id&device_token=" + String(device_token);
        //GET methode
        Link = URL + getData;
        http.begin(Link); //initiate HTTP request,
        //    Serial.println(Link);
        int httpCode = http.GET(); //Send the request
        String payload = http.getString(); //Get the response payload

        if (payload.substring(1, 8) == "nuevoID") {
            String nuevoID = payload.substring(8);
            //Serial.println(nuevoID);
            id = nuevoID.toInt();
            http.end(); //Close connection
            getFingerprintEnroll();
        }
    }
}

```

Figura 44 Comprobación nuevo usuario a registrar

Dentro de la función **getFingerPrintEnroll ()** se ha introducido una función para confirmar en el servidor cuando se ha hecho con éxito el registro en el sensor.

Se llama **confirmarNuevo ()** y al igual que antes, es muy sencilla y solamente hace la petición GET al servidor con parámetro **confirm_id=id**

```
//*****Esta función actualizara el registro en la BBDD añadiendole una f
void confirmarNuevo(int id) {

    //Serial.println("confirm Adding");
    if (WiFi.status() == WL_CONNECTED) {
        HTTPClient http;    //Declare object of class HTTPClient
        //GET Data
        getData = "?confirm_id=" + String(id) + "&device_token=" + String(device_token);
        //GET methodode
        Link = URL + getData;

        http.begin(Link); //initiate HTTP request,
        //    Serial.println(Link);
        int httpCode = http.GET();    //Send the request
        String payload = http.getString();    //Get the response payload
        if (httpCode == 200) {
            display.clearDisplay();
            display.setTextSize(1.5);           // Normal 1:1 pixel scale
            display.setTextColor(WHITE);        // Draw white text
            display.setCursor(0, 0);            // Start at top-left corner
            display.print(payload);
            display.display();
        }
    }
}
```

Figura 45 Código que informa haciendo una petición al servidor del registro de un usuario en el sensor

4.2.5 Eliminación de usuarios

Esta función es exactamente igual que la de registro sólo que sirve para eliminar un usuario, una vez este eliminado desde la página web, y pase un tiempo, el dispositivo lo borrará del sensor. Se llama **CheckBorrarID ()**.

Se comprueba si está conectado a la red Wi-Fi y se hace la petición servidor con los correspondientes parámetros, si la respuesta contiene “**del-id**” se recoge el número y se llama a la función de la librería de Adafruit Fingerprint llamada **deleteFingerprint (id)** que es la que se encarga de eliminar de la memoria del sensor la huella dactilar asociada a ese ID.

```

//*****Esta función comprobará si hay huellas que hayan s
void CheckBorrarID() {

    //Serial.println("Check to Delete ID");
    if (WiFi.isConnected()) {
        HTTPClient http;    //Declare object of class HTTPClient
        //GET Data
        getData = "?DeleteID=check&device_token=" + String(device_token);
        //GET method
        Link = URL + getData;
        http.begin(Link); //initiate HTTP request,
        //    Serial.println(Link);
        int httpCode = http.GET();    //Send the request
        String payload = http.getString();    //Get the response payload

        if (payload.substring(0, 6) == "del-id") {
            String del_id = payload.substring(6);
            //Serial.println(del_id);
            http.end(); //Close connection
            deleteFingerprint( del_id.toInt() );
            delay(1000);
        }
        http.end(); //Close connection
    }
}

```

Figura 46 Comprobación de usuario a eliminar

Si todo ha salido bien recibiremos en el display del sensor un mensaje confirmando que se ha eliminado correctamente del servidor y del sensor con el mensaje “¡**borrado!**”. Si no es así habrá que comprobarlo desde el servidor o repetir el proceso.

4.2.6 Marcaje de los usuarios

En el **loop ()** del programa estamos continuamente ejecutando la función de la librería que se encarga de hacer **match** entre huellas, si ponemos el dedo en el sensor y coincide se nos se muestra en el display llamando a la función **PrintDisplay**, dentro de esta última función se ejecuta la de la siguiente figura **EnviarFingerID** y si la petición ha ido bien y se hace el marcaje en el servidor se recibirá la respuesta por el display dependiendo de si ha sido un marcaje de entrada (**login**), se recibe un saludo con tu nombre de usuario por pantalla, y si ha sido un marcaje de salida (**logout**), se recibe un adiós con tu nombre de usuario.

```
//*****Función para enviar el IDFingerPrint al servidor y marcar *****
void EnviarFingerID( int finger ) {
    //Serial.println("Sending the Fingerprint ID");
    if (WiFi.isConnected()) {
        HTTPClient http;    //Declare object of class HTTPClient
        //GET Data
        getData = "?FingerID=" + String(finger) + "&device_token=" + device_token;
        //GET methode
        Link = URL + getData;
        http.begin(Link); //initiate HTTP request    //Specify content-type header

        int httpCode = http.GET();    //Send the request
        String payload = http.getString();    //Get the response payload

        if (payload.substring(1, 6) == "login") {
            String user_name = payload.substring(6,payload.lastIndexOf("]"));
            //Serial.println(user_name);

            display.clearDisplay();
            display.setTextSize(2);                // Normal 2:2 pixel scale
            display.setTextColor(WHITE);           // Draw white text
            display.setCursor(15, 0);               // Start at top-left corner
        }
    }
}
```

Figura 47 Función que envía ID al servidor para hacer un fichaje

4.2.7 Geolocalización del dispositivo

Para geolocalizar el dispositivo por red Wi-Fi vamos a usar la librería **WifiLocation**, esto supone que debemos crearnos una cuenta de Google Cloud y conseguir una API key para poder hacer peticiones con el dispositivo.

Una vez se tenga y se ha comprobado que funciona, vamos a crear la función **GoogleApiLoc**

```
String GoogleApiLoc() {  
  
    const char* googleApiKey = "AIzaSyAe_kiLSxwwxILtWXIC42kJemBbiPBe3WE";  
  
    WifiLocation location (googleApiKey);  
  
    if (WiFi.isConnected()) {  
  
        setClock ();  
  
        location_t loc = location.getGeoFromWiFi();  
  
        String buf = String(loc.lat, 7) + "," + String(loc.lon, 7);  
        //Serial.println(buf);  
        return buf;  
  
    }  
}
```

Figura 48 Obtener datos de latitud y longitud con Google API Geolocation

Haremos otra función la cual comprobará que en la base de datos del servidor el registro de los datos de la localización del dispositivo está actualizado desde no más de 24 horas, así no hacemos demasiadas peticiones innecesarias a la API ya que es costoso en tiempo y energía.

En la base de datos se guardará MAC/Departamento, sus datos de geolocalización y última actualización.

Esta será la localización que se asociará a los usuarios cuando marquen en sus registros en la base de datos.

ID	idconexion_depart	lat	lng	last_update
1	CC:32:E5:B0:E7:EB/Dormitorio	38.0397186	-4.0413332	2021-08-25 18:42:58

Figura 49 Ejemplo de datos de dispositivo geolocalizado

5 CONCLUSIONES Y LÍNEAS DE FUTURO

5.1 Conclusiones

Se ha implementado mediante el sensor de huellas óptico y se ha conseguido una alternativa económica para una solución biométrica, que además gracias a un servidor web, nos permite registrar los datos, consultar, visualizar los marcajes y administrar los usuarios y dispositivos, con además las tecnologías más conocidas por todo el mundo y sin ningún tipo de framework, lo cual hace ver que no hace que no es necesario una gran infraestructura de hardware y software para adoptar una solución asequible.

Se cumplen gracias a ello, los objetivos técnicos dispuestos en el Apartado 2:

- Estudio de tecnologías y redes de sensores inalámbricos y sus conclusiones de elección del tipo de nodo a emplear
- Se ha configurado un nodo y conseguido comunicar y enviar los datos del sensor a un servidor de comunicaciones
- Se ha programado un servidor web para facilitar el almacenamiento y acceso a información capturada por los nodos sensores, así como una implementación de gestión de los sensores
- Se ha implementado un sencillo algoritmo para identificar las situaciones de incumplimiento de la normativa vigente y se informa con alertas en las consultas del sitio web
- Diseño y montaje de la red experimental con los sensores que cumpliera el funcionamiento de los anteriores objetivos
- Se han realizado experimentos para corroborar el funcionamiento de la red que se ha diseñado

5.2 Líneas de futuro

5.2.1 Hardware

- En cuanto a hardware siempre se puede mejorar el sensor por uno más efectivo o que permita guardar concretamente las huellas dactilares de forma encriptada en tu propia base de datos. Eso añadiría, por ejemplo, la posibilidad de una solución más dinámica como poder reconocer a un usuario que hizo un registro de su huella en un sensor y poder reconocerlo en otro sensor distinto y ubicado en cualquier otro lugar.
- Sustituir módulo GSM por uno de mejor calidad y eficiencia energética y que incluya posibilidad de GPS/GNSS.
- Implementar un microcontrolador o PCB que incluya Wi-Fi y GSM o sustituir esta última por alguna otra tecnología de red inalámbrica más eficiente.
- En el caso que se quisiera seguir con el proyecto, se puede pensar en mejorar las tecnologías o sustituirlas por otras menos costosas en precio y energía como es el caso de usar protocolo MQTT.

5.2.2 Software

- Hacer una migración completa de back-end y de front-end mediante un framework y tecnologías que puedan ser interesantes (Spring MVC, Symfony...) y que hagan que el mantenimiento de la aplicación sea muchísimo más sencillo.
- El uso de PlatformIO que se pretendía usar junto con Visual Studio Code, pero se optó por usar IDE oficial de Arduino, realmente es una herramienta muy versátil y facilita también la instalación de librerías y podemos aprovechar que en este IDE si puedes hacer uso de Intellisense C/C++ que ayuda increíblemente a programar en Arduino, ya que, en Arduino IDE debes estar consultando la documentación constantemente.

6 PRESUPUESTO

A continuación, se muestra un presupuesto aproximado de los procesos y del hardware.

Hardware	Unidades	Precio unitario en €	Precio total en €
Sensor FZ2904	1	6,94	6,94
Display OLED 0.96	1	0,30	0,30
NodeMCU	1	1,93	1,93
SIM800L	1	4,05	4,05
Batería Lipo 1500mA	1	2,97	2,97
Regulador USB 3.7 V	1	0,30	0,30
Mini DC-DC	1	0,30	0,30
Cableado	20	0,50	0,50
Placa board exp	1	1,80	1,80

El coste total para un solo nodo en funcionamiento es de 19,09€

Proceso	Horas	Precio unitario en €	Precio total en €
Diseño y programación	480	30	14400
Configuración	20	25	500
Montaje e instalación	2	20	40
Pruebas y corrección	15	20	300
Documentación	30	25	750

El coste total de los procesos de diseño y programación, configuración, montaje e instalación, pruebas y corrección y documentación es de

Coste	Precio en €
Hardware	19,09
Recursos humanos	15990
Total sin IVA	16009,09
Total con IVA (21%)	19366,99

El presente presupuesto para el proyecto asciende a la cantidad de **DIECINUEVE MIL TRESCIENTOS SESENTA Y SEIS EUROS CON NOVENTA Y NUEVE CENTIMOS**

7 BIBLIOGRAFIA

1. **WampServer, Apache, PHP, MySQL sous Windows**. [En línea] 2021
<https://www.wampserver.com/>
2. **Foundation, The Apache Software**. [En línea] 26 junio 2021. <https://httpd.apache.org/>
3. **Visual Studio Code, Microsoft. Visual Studio Code Website**. [En línea] 24 junio de 2015.
<https://code.visualstudio.com/>
4. **Arduino IDE, Arduino website**. [En línea] 28 Julio de 2021. <https://www.arduino.cc/>

5. **Fielding, R., and J. Reschke. Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230.** [En línea] junio de 2014. <https://www.rfc-editor.org/info/rfc7230>.
6. **César Tolosa Borja., y Álvaro Giz Bueno. Sistemas Biométricos.** [En línea] s.f. https://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia/Bioinformatica/web_BIO/Documentacion/Trabajos/Biometria/Trabajo%20Biometria.pdf
7. **ESP8266 Arduino Core, Documentation.** [En línea] 2017. <https://arduino-esp8266.readthedocs.io/en/latest/>
8. **Luis del Valle Hernández, SSD1306 pantalla OLED con Arduino y ESP8266 I2C.** [En línea] 2019 <https://programarfacil.com/blog/arduino-blog/ssd1306-pantalla-oled-con-arduino/>
9. **SIMCom A Company of SIM Tech, SIM800 Series_AT Command Manual_V1.09.** [En línea] 3 de agosto de 2015 https://www.elecrow.com/wiki/images/2/20/SIM800_Series_AT_Command_Manual_V1.09.pdf
10. **Adafruit learning system, lady ada, Adafruit Optical Fingerprint Sensor** [En línea] 30 de Julio de 2021 <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf>
11. **Clever cloud, One-Click MySQL Hosting,** [En línea] 29 de octubre de 2020 <https://www.clever-cloud.com/en/mysql-hosting>
12. **000webhost, website status** [En línea] 20 de mayo de 2019 <https://status.000webhost.com/784411699/calendar>
13. **Jefatura del Estado, Real Decreto-ley 8/2019, de 8 de marzo, de medidas urgentes de protección social y de lucha contra la precariedad laboral en la jornada de trabajo, BOE-A-2019-3481.** [En línea] 12 de marzo de 2019 <https://www.boe.es/eli/es/rdl/2019/03/08/8>

8 ANEXOS

8.1 Manual de usuario

Este manual trata de aclarar de manera muy resumida el uso a la hora de registrar su jornada laboral. Rigiéndonos en el artículo 34, apartado 9 del Real Decreto-ley 8/2019:

9. La empresa garantizará el registro diario de jornada, que deberá incluir el horario concreto de inicio y finalización de la jornada de trabajo de cada persona trabajadora, sin perjuicio de la flexibilidad horaria que se establece en este artículo [13].

No se han hecho registros de descanso ya que uno de los objetivos del proyecto era cumplir lo dispuesto en el Real Decreto-ley, con lo que no estamos exigidos a hacerlo.

8.1.1 Fichaje de entrada

Al inicio de la jornada laboral ponga el dedo y fíjese si recibe una imagen de una huella dactilar con un tick.



Figura 50 Huella dactilar reconocida y correcta

Si es así retírelo y verifique que le responde posteriormente con un **“Hola NombreUsuario”**.



Figura 51 Marcaje de entrada registrado en el servidor web

Si recibe una imagen de huella dactilar y una “X “es que ha fallado la verificación, significa que no lo reconoció bien retire el dedo y vuélvalo a intentar.



Figura 52 Huella dactilar no reconocida

Posibles problemas adicionales:

1. Puede ocurrir que aun así reciba por display el sensor responde que reconoció la huella, pero no le responde el servidor (figura 47) de manera que no se hizo el marcaje por algún problema en la conexión a internet. Vuélvalo a intentar hasta que reciba su mensaje de bienvenida, si no, no se habrá hecho ningún marcaje.
2. Si aun así continua con no reconocer pruebe a limpiarse los dedos, ya que puede malograr la captura de la imagen, y si aun así no procede se deberá verificar que el dispositivo está correctamente funcionando, tiene conexión a internet, y que su huella dactilar este correctamente registrada en el sensor y en el servidor.

8.1.2 Fichaje de salida

Para el fin de la jornada laboral ponemos el dedo y nos fijamos si recibimos una imagen de la huella dactilar con un tick (figura 47)

Si es así lo retiramos y verificamos que nos responde posteriormente con un “**Adios NombreUsuario**” de tal manera que se ha conseguido fichar sin problemas.



Figura 53 Marcaje de salida registrado en el servidor web

Si recibe una imagen de huella dactilar y una “X “(figura 48) es que ha fallado la verificación, significa que no lo reconoció bien retire el dedo y vuélvalo a intentar.

Los dos problemas adicionales anteriores son igualmente aplicables en el fichaje de salida.

Pueden hacerse varios fichajes al día, pero se recuerda que el sistema no está hecho para eso y que debe comentarse al administrador de la razón.

Y si por descuido se ha fichado erróneamente mal, también se deberá avisar para solucionarlo.

8.1.3 Registro de usuario

Entramos en la web, concretamente en la parte de “**Administración de usuarios**” que es **ManageUsers.php**

Pulsados en “**Añadir**”

Añadir							
FINGER ID	NOMBRE	ROL	CÓDIGO USUARIO	EMAIL	FECHA DE REGISTRO	DEPARTAMENTO	ACCIONES
1	Salva2	rol	953000001	std@gmail.com	2021-03-23 00:00:00	Dormitorio	 
2	Antonio2	Cliente	222222222	antonio@gmail.com	2021-03-25 00:00:00	Dormitorio	 
3	SalvadorTD	Cargo	595959595	std93@hotmail.es	2021-04-25 18:03:51	Dormitorio	 
1	fastasfas	rol	114444444	hola	2021-06-02 00:00:00	Departamento I+D	 

Figura 54 Tabla administración usuarios

Nos aparecerá un modal para introducir datos del usuario, el departamento donde va a tener que fichar, nombre usuario, código usuario/DNI que le identifique, email y el rol que desempeña.

Registrar datos de usuario

1

Area de usuario

Area/departamento:

Dormitorio

2

Info usuario

Ejemplo

oooooooooZ

ejemplo@ej.com

3

Info adicional

ejemplo

Cerrar

Guardar

Figura 55 Registro con datos de usuario

Una vez hecho nos aparecerá un mensaje de confirmación y se habrá registrado en la base de datos, y se mostrará así en la web:

GESTIÓN USUARIOS							
Nuevo usuario registrado!							
Añadir							
FINGER_ID	NOMBRE	ROL	CÓDIGO USUARIO	EMAIL	FECHA DE REGISTRO	DEPARTAMENTO	ACCIONES
1	Salva2	rol	114444446	salva@gmail.com	2021-03-23 00:00:00	Dormitorio	 
2	Antonio2	Cliente	222222222	antonio@gmail.com	2021-03-25 00:00:00	Dormitorio	 
3	SalvadorTD1	Cargo	595959599	st497@hotmail.es	2021-04-25 18:03:51	Dormitorio	 
1	fastahola	rol	114444446	hola	2021-06-02 00:00:00	Departamento I+D	 
4	Ejemplo	ejemplo	00000000Z	ejemplo@ej.com	Por confirmar	Dormitorio	 

Figura 56 Registrado en el servidor y a falta de confirmar en el sensor

Con fecha de registro **“Por confirmar”** que quiere decir que aún hay que registrar la huella en el sensor.

Nos dirigimos al dispositivo y nos mostrará después de un tiempo que está en modo registro como aparece en la imagen:



Figura 57 Registro en el sensor

Ponemos el dedo en el sensor y si ha capturado bien la imagen de la huella nos responderá con una huella y un tick como en la figura 47 y posteriormente con un mensaje que nos pide quitar el dedo:



Figura 58 Mensaje para retirar dedo

Nos volverá a pedir que pongamos el dedo como en la imagen de la figura 53 y si ha capturado bien la imagen de la huella nos responderá otra vez con un tick y posteriormente con el siguiente mensaje del servidor:



Figura 59 Registro hecho correctamente

Si refrescamos la página web, ya si tendremos confirmado el registro de la huella dactilar para ese usuario y podremos hacer marcajes sabiendo que pertenecen a ese usuario.

FINGER_ID	NOMBRE	ROL	CÓDIGO USUARIO	EMAIL	FECHA DE REGISTRO	DEPARTAMENTO	ACCIONES
1	Salva2	rol	114444446	salva@gmail.com	2021-03-23 00:00:00	Dormitorio	 
2	Antonio2	Cliente	222222222	antonio@gmail.com	2021-03-25 00:00:00	Dormitorio	 
3	SalvadorTD1	Cargo	895959599	st893@hotmail.es	2021-04-25 19:33:51	Dormitorio	 
1	farfahola	rol	114444446	hola	2021-06-02 00:00:00	Departamento I+D	 
4	Ejemplo	ejemplo	00000000Z	ejemplo@ej.com	2021-09-02 11:18:39	Dormitorio	 

Figura 60 Vista con usuario registrado en el servidor

8.1.4 Borrado de usuario

Para borrar un usuario entramos a **ManageUsers.php** y seleccionamos el usuario que queremos borrar con el icono , en este caso vamos a borrar el que se creó en el apartado 8.1.3

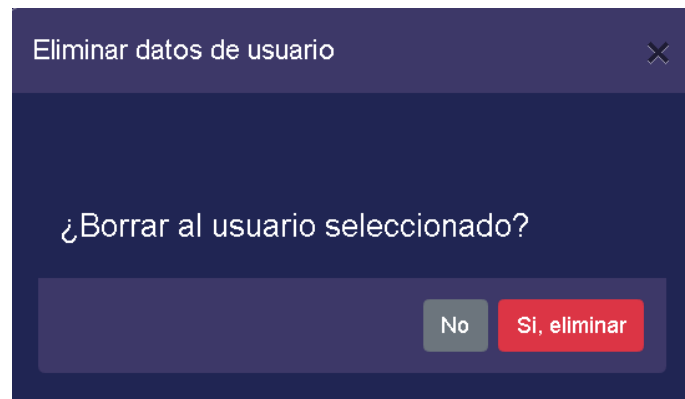


Figura 61 Modal confirmación borrar usuario

Nos pedirá confirmación para eliminarlo, pulsamos en **“Si, eliminar”** y se borrará del sensor y del servidor web cuando el sensor haga la comprobación. Lo podremos corroborar en el **index.php** que nos muestra una tabla con los usuarios registrados y que aún están registrados en el sensor biométrico.

¡Usuario eliminado correctamente!							
Añadir							
FINGER ID	NOMBRE	ROL	CÓDIGO USUARIO	EMAIL	FECHA DE REGISTRO	DEPARTAMENTO	ACCIONES
1	Salva2	rol	114444445	salva@gmail.com	2021-03-23 00:00:00	Dormitorio	 
2	Antonio2	Cliente	222222222	antonio@gmail.com	2021-03-25 00:00:00	Dormitorio	 
3	SalvadorTD1	Cargo	595959599	std93@hotmail.es	2021-04-25 18:03:51	Dormitorio	 
1	fasfahola	rol	114444446	hola	2021-06-02 00:00:00	Departamento I+D	 

Figura 62 Mensaje de borrado usuario