



UNIVERSIDAD TECNOLÓGICA ISRAEL

TRABAJO DE TITULACIÓN EN OPCIÓN AL GRADO DE:

INGENIERO EN ELECTRÓNICA DIGITAL Y TELECOMUNICACIONES

TEMA: Diseño e implementación de un biométrico inteligente que envíe SMS de alerta a los empleados antes de su hora de entrada.

AUTOR: CARLOS ANDRES IZA SEVILLA

TUTOR: ING. LUIS MONTOYA LARA, MGs.

AÑO: 2018

Carrera:	Electrónica Digital y Telecomunicaciones
Autor/a del TT:	Carlos Andrés Iza Sevilla
Tema del TT:	Diseño e implementación de un biométrico inteligente que envíe SMS de alerta a los empleados antes de su hora de entrada.
Articulación con la Línea de Investigación Institucional:	Automatización y control de procesos
Su línea de Investigación Institucional:	Electrónica digital y microprocesadores.
Fecha de presentación del plan:	Noviembre 2017

DECLARACIÓN

Yo, Carlos Andrés Iza Sevilla declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedo los derechos de propiedad intelectual correspondiente a este trabajo a la Universidad Tecnológica Israel, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Carlos Andrés Iza Sevilla

CERTIFICACIÓN DEL AUTOR

En mi calidad de tutor del trabajo de titulación certifico:

Que el trabajo de titulación **“Diseño e implementación de un biométrico inteligente que envíe SMS de alerta a los empleados antes de su hora de entrada.”** Presentado por el Sr. Carlos Andrés Iza Sevilla, estudiante de la carrera de Electrónica Digital y Telecomunicaciones, reúne los requisitos y méritos suficientes para ser sometido a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito, agosto 18 de 2018

TUTOR

Ing. Luis Montoya, Mgs.

AGRADECIMIENTO

Agradezco a Dios por haberme permitido cumplir esta meta tan anhelada, a mi familia que siempre me estuvo apoyando y por ultimo pero no menos importante al personal docente de la Facultad de Ingeniería Electrónica Digital y Telecomunicaciones de la Universidad Tecnológica Israel por todo el conocimiento adquirido.

DEDICATORIA

Este logro alcanzado dedico a mi esposa, a mis hijos, a mis padres y a todos mis familiares que con su apoyo moral me incentivaron a llegar a este objetivo tan anhelado en mi carrera profesional.

RESUMEN

Para las empresas es muy importante reducir el porcentaje de ausentismo y atrasos, ya que el no contar con uno de sus colaboradores reduce la productividad y eficacia de sus servicios, lo cual conlleva no tener fidelidad por parte de sus clientes y esto es un causal de pérdidas económicas.

La importancia del diseño e implementación del proyecto de un biométrico que alerte a los colaboradores antes de su hora de entrada ayudará de mucho en la reducción de atrasos y además beneficiará en dar un registro de entradas y salidas del personal veraz y confiable, ya que este tipo de dispositivo registra información con la presencia de los empleados.

En este escrito se presenta información relevante de los componentes que conforman el Biométrico inteligente, el desarrollo e implementación del dispositivo para el punto de trabajo asignado.

Palabra Clave: Biométrico, Arduino, GSM, Pantalla LCD, Reloj en Tiempo Real

ABSTRACT

For companies it is very important to reduce the percentage of absenteeism and delays, since not having one of their collaborators reduces the productivity and efficiency of their services, which entails not having fidelity on the part of their clients and this is a cause of economic losses.

The importance of the design and implementation of a biometric project that alerts employees before their time of entry will help in the reduction of arrears and also benefit in giving a record of entries and departures of the staff and reliable, since this Device type records information with the presence of employees.

In this document you will find relevant information on the components that make up the Smart Biometric, the development and implementation of the device for the assigned work point.

Keyword: Biometric, Arduino, GSM, LCD Display, Real Time Clock

INTRODUCCIÓN

El método de reconocimiento de individuos por medio de las características de su cuerpo se viene utilizando desde hace muchos años atrás, los pioneros en utilizar la biometría fueron los chinos en el siglo XIV, la biometría desde sus inicios se ha utilizado para el reconocimiento de criminales, pero con el pasar del tiempo se ha ido creando dispositivos electrónicos que son utilizados hasta la actualidad para el acceso de personal exclusivo en áreas restringidas, y también son utilizados en las empresas para controlar el registro de la hora de ingreso y salida del personal que pertenece a las nóminas de cada una de ellas, la razón por la que se vio la necesidad de elaborar este proyecto es porque en la agencia de CNT E.P. de Quicentro Sur se lleva el registro de asistencias manualmente y es vulnerable a ser manipulado según la conveniencia del empleado.

- **Antecedentes de la situación de objeto de estudio.**

La agencia de Quicentro Sur perteneciente a CNT EP, se encuentra situada en la ciudad del D.M. de Quito al sur de la ciudad, entre las calles Quitumbe Ñan y Moran Valverde, en esta agencia no se cuenta con control de asistencia electrónico, razón por la cual nace la necesidad de implementar un dispositivo biométrico.

Para la puesta en marcha de este proyecto se revisó varias citas bibliográficas con temas similares, pero el que más se asemeja al tema a desarrollarse pertenece a Baldeon (2016), quien realizó la implementación de un reloj biométrico, en la escuela Luis Pallares de la parroquia de Yaruquí, este dispositivo en la actualidad es de gran ayuda para controlar los ingresos y salidas de los docentes de la escuela.

Para solventar el problema de atrasos y evitar que el registro de asistencias se lo haga manualmente en la agencia de Quicentro Sur, se propuso el diseño e implantación de un biométrico inteligente que envíe SMS, 10 minutos antes de la hora de entrada.

▪ **Planteamiento del Problema**

El personal de la agencia de CNT E.P de Quicentro Sur recurrentemente llega atrasado a su lugar de trabajo, lo que genera que al arranque de turno haya mucho acumulo de personas en la sala de atención al cliente, lo cual causa que el nivel de servicio baje considerablemente, pero el problema central para esta investigación es que al llevar un sistema de asistencia tradicional, existe la probabilidad de que los registros sean manipulados a conveniencia de cada colaborador.

El diseño e implementación de un biométrico inteligente que envíe SMS de alerta a los empleados antes de su hora de entrada, beneficiara a que se lleve un control veraz de la asistencia de los empleados y ayudará a que los miembros del equipo de esta agencia no lleguen atrasados, porque deben registrar su hora de entrada en el dispositivo biométrico.

▪ **Objetivo General**

Diseñar e implementar un biométrico inteligente que envíe SMS de alerta a los empleados antes de su hora de entrada.

▪ **Objetivos Específicos**

- Determinar el estado del arte en el diseño de biométricos inteligentes.
- Definir los elementos electrónicos que van a ser utilizados en el biométrico inteligente.
- Diseñar el sistema electrónico del dispositivo basado en microcontroladores.
- Desarrollo del algoritmo de programación que regirá el funcionamiento del biométrico inteligente.
- Determinar la mejor tecnología para el envío de SMS de alerta.

TABLA DE CONTENIDOS

CAPITULO I. FUNDAMENTACION TEORICA	1
1.1. Historia de la biometría	1
1.2. Modelos de biométricos	3
1.2.1. Biométricos de huellas dactilares	3
1.2.2. Biométrico facial	5
1.2.3. Biométrico de iris	6
1.2.4. Biométrico de voz	8
1.3. Componentes Electrónicos	10
1.3.1. Arduino	10
1.3.1.1. ¿Qué es Arduino?	10
1.3.1.2. Tipos de placas Arduino	10
1.3.1.3. Arduino Mega 2560	14
1.3.1.4. Microcontrolador Atmega2560	15
1.3.1.5. Plataforma de Arduino	16
1.3.2. Sensor finger print FPM10A	17
1.3.3. Módulo Tiny RTC DS 1307	18
1.3.4. Módulo GSM-GPRS-SMS-SIM900	19
1.3.5. Pantalla LCD 20X4:	21
1.3.6. Módulo SD CARD	21
1.3.7. Teclado 4X4	22
CAPITULO II. MARCO METODOLÓGICO	24
2.1. Modelo de investigación	24
2.2. Técnica de recolección de datos	26
CAPITULO III. PROPUESTA	27
3.1. Diseño de hardware	28
3.1.1. Etapas de entrada	28
3.1.2. Etapas de salida	30
3.2. Diseño del software	32
3.3. Presupuesto y materiales	33
3.4. Ventajas que presta dispositivo biométrico	33
CAPITULO IV. IMPLEMENTACIÓN	35
4.1. Desarrollo	35
4.1.1. Desarrollo de software	35

4.1.1.1. Arduino versión 1.6.5.....	35
4.1.2. Desarrollo de hardware.	38
4.1.2.1. Desarrollo de montaje del dispositivo.....	38
4.1.2.2. Proteus 8.5	39
4.2. Implementación.....	40
4.2. Pruebas de funcionamiento.....	46
4.3. Análisis de resultados	51
CONCLUSIONES	53
RECOMENDACIONES	54
BIBLIOGRAFIA	55
GLOSARIO	58
ANEXOS	60

LISTA DE FIGURAS

Figura 1. Método minucias.....	3
Figura 2 . Método correlación.	3
Figura 3. Etapas de registro y verificación dactilar.....	4
Figura 4. Reconocimiento dactilar.....	4
Figura 5. Reloj biométrico huella digital.	4
Figura 6. Puntos claves de la estructura de tejidos duros del rostro – inamovibles.....	5
Figura 7. Etapas de la identificación por reconocimiento facial.	5
Figura 8. Sistema de reconocimiento facial.....	6
Figura 9. Biométrico facial.	6
Figura 10. Etapas de la identificación por reconocimiento de iris.	7
Figura 11. Mapeo del iris del ojo para los sistemas de reconocimiento del iris.....	7
Figura 12. Lector biométrico de iris.	8
Figura 13. Etapas en el procesamiento de señal de voz.	9
Figura 14. Sistema de reconocimiento de voz.	9
Figura 15. Sistema BioCloser de reconocimiento de voz.	9
Figura 16. Placa Arduino Mega2560.	15
Figura 17. Microcontrolador ATMEGA2560.....	15
Figura 18. Logo de arranque de plataforma Arduino	16
Figura 19. Inclusión de librerías en Arduino.....	16
Figura 20. Lector de huella digital	17
Figura 21. Módulo RTC DS 1307.	18
Figura 22. Módulo GSM-GPRS-SMS-SIM900.....	20
Figura 23. Pantalla LCD 20x4.....	21
Figura 24. Módulo SD CARD.	22
Figura 25. Teclado 4X4.	22
Figura 26. Diagrama estructural de bloques con software y hardware del dispositivo.....	28
Figura 27. Diseño etapa de control proceso.....	29
Figura 28. Dispositivos de etapa de salida.	30
Figura 29. Diseño hardware	31
Figura 30. Flujograma de funcionamiento del programa.	32
Figura 31. Programación Arduino versión 1.6.5.....	35
Figura 32. Compilación de programación.....	36
Figura 33. Error en la compilación.	36
Figura 34. Compilación exitosa.	37
Figura 35. Programa subiendo a tarjeta Arduino.....	37
Figura 36. Programa subido con éxito.	38
Figura 37. Verificación de funcionamiento Arduino MEGA 2466	38
Figura 38. Comprobación Arduino MEGA 2560 con SIM 900.....	39
Figura 39. Comprobación Arduino MEGA 2560 con sensor FPM10A.	39
Figura 40. Proteus 8.5.....	40
Figura 41. Montaje de regletas hembras	41
Figura 42. Soldadura de montaje de regletas hembras.....	41
Figura 43. Montaje completo PCB	42

Figura 44. Soldadura de montaje completo.	42
Figura 45. Montaje de RTC y módulo SD CARD	43
Figura 46. Módulo GSM-GPRS-SMS-SIM900 con SIM CARD CNT EP.	43
Figura 47. Montaje y conexión.	44
Figura 48. Montaje provisional.	44
Figura 49. Carcasa para dispositivo	45
Figura 50. Lugar asignado para dispositivo	45
Figura 51. Implementación dispositivo.	46
Figura 52. Comprobación de arranque.	46
Figura 53. Arranque del dispositivo.	47
Figura 54. Digitación de clave.	47
Figura 55. Validación de clave.	48
Figura 56. Solicitud de huella.	48
Figura 57. Lectura de huella exitosa.	49
Figura 58. Archivo de registros.	49
Figura 59. Datos registrados en el archivo.	50
Figura 60. Envío de SMS	50
Figura 61. SMS enviado para pruebas.	51

LISTA DE TABLAS

Tabla 1. Tipos de placas Arduino	10
Tabla 2. Especificaciones técnicas del sensor biométrico.....	18
Tabla 3. Especificaciones técnicas RTC.	19
Tabla 4. Especificaciones técnica módulo GSM-GPRS-SMS-SIM900.	20
Tabla 5. Especificaciones técnicas LCD.	21
Tabla 6. Especificaciones técnicas módulo SD CARD.....	22
Tabla 7. Especificaciones técnicas teclado.	23
Tabla 8. Tabla de materiales y precios.	33
Tabla 9. Tabla pruebas de arranque.	51
Tabla 10. Tabla pruebas de funcionamiento.	52

CAPITULO I. FUNDAMENTACION TEORICA

1.1. Historia de la biometría.

El método biométrico ha sido utilizado desde hace cientos de años atrás cuando por los egipcios de esas épocas utilizaban el método de la antropometría con el fin de identificar a los individuos, para lo cual realizaban la toma de medidas de varias partes del cuerpo.

Joao de Barros conocido como explorador y escritor, después de un arduo proceso de exploración escribió que en la antigua china utilizaban la identificación con huellas dactilares al menos desde siglo XIV, esta técnica la utilizaban los comerciantes de este país, para reconocer a los niños uno de otro. El Italiano Marcello Malpighi descubrió que las marcas de identificación que se posee en los dedos eran diferentes, este suceso se dio en el año de 1686.

En 1823 el médico y científico Checo Jan Evangelista Purkine, identificó que cada individuo contaba con huellas digitales únicas, Jan también dio a conocer los espirales, elipses y triángulos que posee cada individuo en sus dedos.

En 1883 Alphonse Bertillon encargado del área de fotografía de la policía de París desarrollo un proceso antropométrico conocido también como Bertillonage, este sistema lo utilizaban para reconocimiento de criminales y dio paso a que la biometría sea estudiada, el sistema Bertillonage dejo de ser garantizado a finales de la década de los 90 se utilizó la técnica de huellas dactilares. EE UU y Oeste de Europa vienen utilizando la identificación con huellas dactilares por más de un siglo.

En los años 60 se dio a conocer un sistema biométrico conocido como Identimat el mismo que media el perfil de la mano y el largo de cada dedo aparentando ser un componente del reloj, estos dispositivos fueron utilizados para poder ingresar a lugares que tenían acceso solo personal autorizado como Western Electric, en la Naval de EE UU en el área inteligencia, energía etc. y otras empresas más.

Identimat cimentó bases para los trabajos en sistemas biométricos a realizarse en el futuro basado en las medidas de la mano.

La identificación por huellas dactilares por medio de sistemas biométricos se desarrolló entre los años 1960 y 1970. En esa época varias empresas se dedicaron al desarrollo de productos que ayuden a automatizar la identificación de las huellas

dactilares con el propósito de temas legales. Al finalizar la década de los sesenta el FBI fue el ente que utilizó el sistema automático de huellas dactilares y fue instalando un gran número de estos sistemas en todo EE UU, el sistema como AFIS (*Automated Fingerprint Identification System*). AFIS actualmente es utilizado en el mundo entero por los departamentos de policía.

En los 80 el Dr. Jon Daughman miembro de Cambridge desarrolla la primera tecnología para la medición de retina. Tomar en consideración que la identificación basada en firmas y rostro es relativamente nueva.

La biometría es un campo de investigación de varias universidades desde hace dos o tres décadas, cabe recalcar que la mayoría de productos que han salido al mercado, se debe al intenso estudio realizado en instituciones de educación superior.

Caltech y MIT (*Modern University for technology & information*) son líderes en el campo de la biometría y debido a que se encuentran ligados a una larga historia en la biometría, las personas de estas universidades se encuentran muy relacionadas con las innovaciones de productos que involucran a la biometría.

En el siglo XX la mayoría del mundo hace uso de las huellas digitales ya que es un sistema práctico y seguro para la identificación de una persona u otra, en este siglo se empezó a utilizar nuevos métodos de identificación con otros rasgos morfológicos tales como: el iris de los ojos, la voz, el calor facial, etc.

En la actualidad la biometría ofrece un portafolio amplio de aplicaciones para la identificación humana, demostrando así ser el mejor método.

1.2. Modelos de biométricos.

1.2.1. Biométricos de huellas dactilares.

El sistema de identificación de individuos por medio de las huellas dactilares es un método usado desde hace siglos atrás, cabe recalcar que este método es uno de los más confiable debido a que cada persona posee huellas únicas e inalterables, generalmente las huellas digitales se encuentran constituidas por un sin número de líneas oscuras que forman las crestas y los espacios blancos que se pueden apreciar en la figura 1, Son conocidos como valles, existen dos métodos de reconocimiento de huellas: el primero conocido como minucias, en este método se puede observar dirección y ubicación de las crestas, bifurcaciones, deltas y valles.

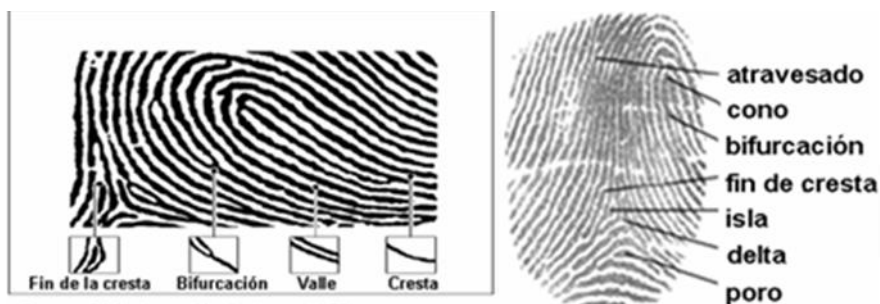


Figura 1. Método minucias.

Fuente: GAO adaptación de datos del FBI

Y el método de correlación el mismo que fue presentado en la tesis doctoral de Purkinje.



Figura 2 . Método correlación.

Fuente: Colegio Minimalista (2012). Recuperado de <http://estudiarlicenciaturaencriminalistica.blogspot.com>

En la figura 3, se observa el proceso básico de registro y verificación de huellas dactilares.

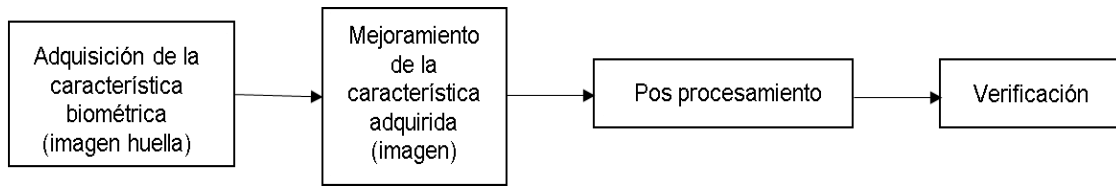


Figura 3. Etapas de registro y verificación dactilar.

**Fuente: Madrigal. C, Madrigal. J, Hoyos. J, & Fernández. D (2007).
Recuperado de <http://www.scielo.org.co>**

En la figura 4, se detalla el proceso de reconocimiento del biométrico de huellas digitales.

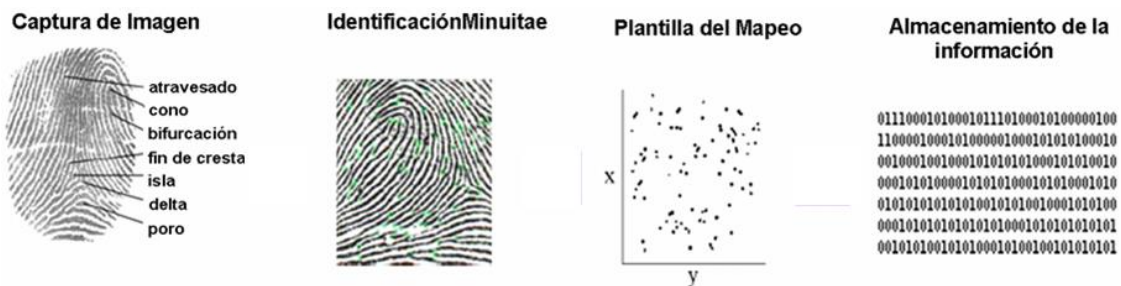


Figura 4. Reconocimiento dactilar.

Fuente: Galvis. C (s.f.). Recuperado de <https://www.monografias.com>

En la figura 5, se evidencia el ejemplo de un reloj biométrico de huella dactilar.



Figura 5. Reloj biométrico huella digital.

Fuente: Reloj biométrico anviz (s.f.). Recuperado de <http://ventaredonda.com.ve>

1.2.2. Biométrico facial

El sistema de reconocimiento facial se lo desarrolló en los años sesenta, hay que tomar en consideración que este sistema es menos confiable que el de huellas digitales, el reconocimiento facial fue diseñado para identificar automáticamente la imagen digital del usuario que se encuentra registrada en una base de datos facial.

La etapa de validación facial se lo realiza con la toma de una imagen real o una bidimensional, el sistema toma puntos nodales de la cara como referencia para el reconocimiento de las personas, estos puntos pueden ser la distancia entre los ojos, distancia ojos vs nariz, el largo de la línea mandibular, la profundidad de los ojos, la figura de los pómulos o la anchura de la nariz, a paso seguido el sistema de la alineación, ajuste de dimensiones y orientación de la imagen ingresada, crea una plantilla facial para ser comparada con la base de datos. Tolosa. C, Giz. A (s.f)



Figura 6. Puntos claves de la estructura de tejidos duros del rostro – inamovibles

Fuente: Tolosa. C, Giz. A (s.f.). Recuperado de <https://www.dsi.uclm.es>

En la figura 7, se evidencia el proceso básico de registro y verificación de imágenes para el reconocimiento facial.

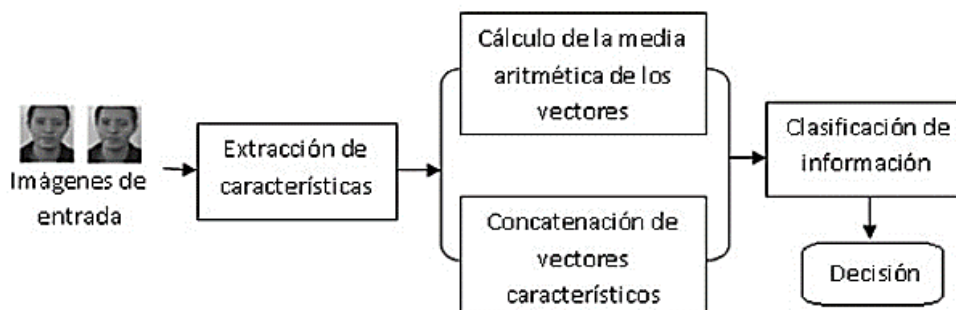


Figura 7. Etapas de la identificación por reconocimiento facial.

Fuente: Garcia. E, Escamilla. E, Nakano. N & Perez. H (2014). Recuperado de <https://scielo.conicyt.cl>

En la figura 8, se detalla el proceso que realiza el biométrico de reconocimiento facial.



Figura 8. Sistema de reconocimiento facial.

Fuente: Galvis. C (s.f.). Recuperado de <https://www.monografias.com>

En la figura 9, se puede observar un ejemplo de biométrico facial.



Figura 9. Biométrico Facial.

Fuente: Reconocimiento facial (2017). Recuperado de <http://www.biometricos.net>

1.2.3. Biométrico de iris

Es un método que reconoce a las personas por medio del iris de sus ojos es más confiable de los sistemas de identificación, debido a que cuenta con una sola característica que son los 266 puntos únicos de referencia a comparación del resto de alternativas biométricas que cuentan 13 a 60 características diferentes.

El escaneo del iris del ojo se lo hace con una cámara de video, que registra una imagen del ojo, creando un modelo de registro único para almacenar en la base, este registro se lo hace comúnmente a una distancia que va desde los 30cm hasta 50 cm aproximadamente.

En la figura 10, se puede observar el proceso básico de registro y verificación de iris.

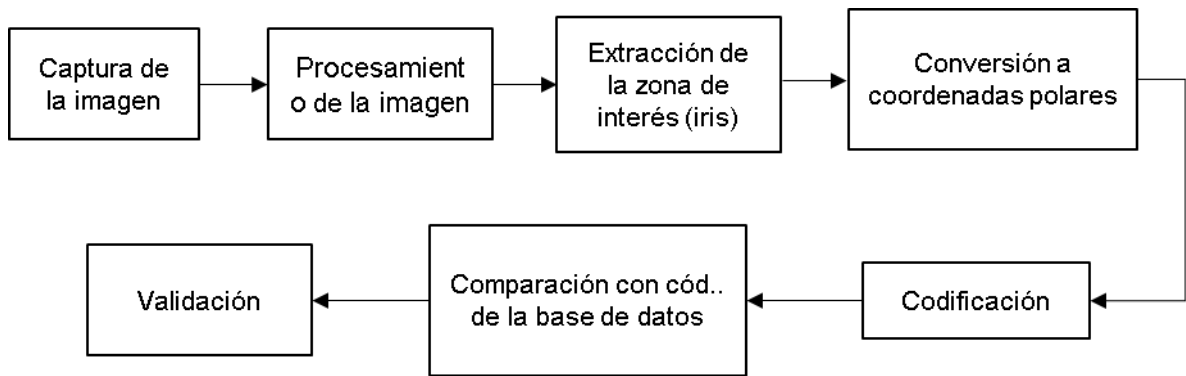


Figura 10. Etapas de la identificación por reconocimiento de iris.

Fuente: Vaca. P (2010)

En la figura 11, se detalla el proceso que realiza el biométrico de reconocimiento de iris.

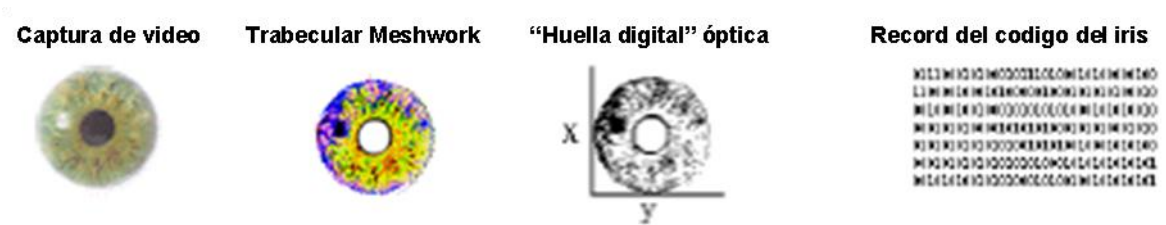


Figura 11. Mapeo del iris del ojo para los sistemas de reconocimiento del iris.

Fuente: Galvis. C (s.f.). Recuperado de <https://www.monografias.com>

En la figura 12, se observa un ejemplo de biométrico de reconocimiento de iris.



Figura 12. Lector biométrico de iris.

Fuente: Alcalde. A (2017). Recuperado de <https://elbauldelprogramador.com>

1.2.4. Biométrico de voz.

El sistema biométrico que reconoce por medio de la voz es utilizado para la identificación de individuos, debido a los factores biométricos físicos y de comportamiento forman una huella vocal única, este tipo de sistemas cuenta con tres formas de reconocer la voz:

- 1.- Dependencia de texto: este método de reconocimiento está basado a que se tiene un texto específico a ser repetido para su identificación.
- 2.- Texto aleatorio: El sistema proporciona un texto aleatorio que debe ser repetido por la persona para su reconocimiento.
- 3.- independencia de texto: En esta ocasión el individuo puede decir cualquier frase para la validación de voz.

Para realizar el ingreso de la voz al sistema se lo hace por medio de un micrófono y aplicaciones de validación de voz; en este registro de voz se toman características tales como: tracto vocal, tamaño de la boca, fosas nasales, pronunciación, acento y velocidad de habla, en este sistema se puede presentar errores por factores como ruido de fondo.

En la figura 13, se puede observar el proceso básico de registro y verificación de un sistema de reconocimiento de voz.



Figura 13. Etapas en el procesamiento de señal de voz.

Fuente: Alezones. Z, Baquero. Y, Borrero. H & Backer. M (2012). Recuperado de <http://orinoquia.unillanos.edu.co>

En la figura 14, se detalla el proceso que realiza el biométrico de reconocimiento de voz.

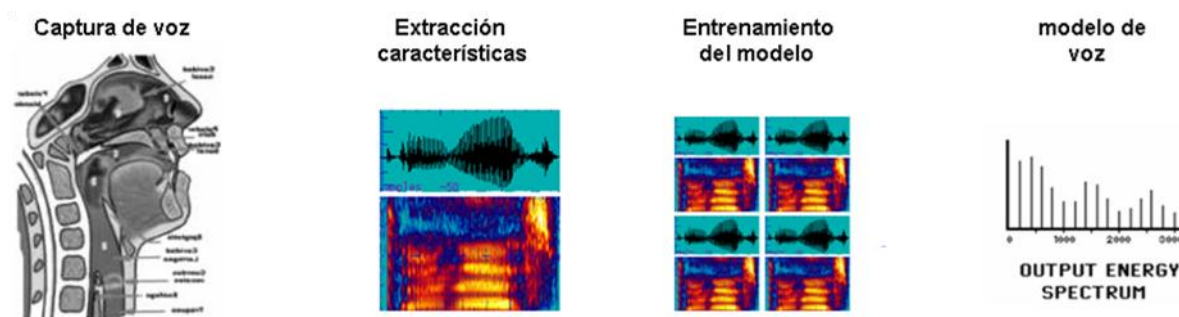


Figura 14. Sistema de reconocimiento de voz.

Fuente: Galvis. C (s.f.). Recuperado de <https://www.monografias.com>

En la figura 15, se puede observar un ejemplo de biométrico de reconocimiento de voz.



Figura 15. Sistema BioCloser de reconocimiento de voz.

Fuente: seguridadinformatica353 (2016). Recuperado de <https://seguridadinformatica353.wordpress.com>

1.3. Componentes Electrónicos

1.3.1. Arduino

1.3.1.1. ¿Qué es Arduino?

Arduino es una placa diseñada para la elaboración de proyectos de estudiantes, diseñadores o simplemente por hobby, esta placa interactiva también es utilizada para el campo de la educación, debido a que cuenta con software y hardware fácil de usar,

Arduino cuenta con una plataforma de código abierto, y lo puede usar con varios sistemas operativos tales como: Linux, Windows, Mac, entre otras. (Ojeda L. s.f.)


Arduino tiene en su placa un micro Atmel AVR, el mismo que se lo programa utilizando *Arduino Programming Language* basado en la plataforma de programación Wiring y *Arduino Development Environment* que se basa en la plataforma de Processing).






Arduino en el año 2006 por ser una gran ayuda en el campo de la electrónica recibió una mención honorífica por parte de *Digital Communities del Ars Electronica Prix*. (Hacedores. 2014).







1.3.1.2. Tipos de placas Arduino







Arduino se ha visto obligado a desarrollar nuevas versiones, debido a las necesidades y complejidades de los proyectos a ser ejecutados, la tabla 1 muestra las diferentes placas Arduino que puede encontrar en el mercado.


Tabla 1. Tipos de placas Arduino

TIPOS DE ARDUINO	
MODELO Y GRAFICO	CARACTERÍSTICAS
Arduino UNO 	Microcontrolador: ATmega328 Voltaje de funcionamiento: 5 V Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) Pines de entradas analógicas: 6 Corriente DC por cada pin I/O: 40 mA Corriente DC en el pin de 3.3 V: 50 mA Memoria Flash: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados por el bootloader SRAM: 2 KB (ATmega328) EEPROM: 1 KB (ATmega328) Velocidad de reloj: 16 MHz

<p>Arduino Leonardo:</p> 	<p>Microcontrolador: ATmega32u4 Voltaje de funcionamiento: 5 V Pines I/O digitales: 20 Canales PWM: 7 Pines de entradas análogas: 12 Corriente DC por cada pin I/O: 40 mA Corriente DC en el pin de 3.3 V: 50 mA Memoria Flash: 32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader SRAM: 2 KB (ATmega32u4) EEPROM: 1 KB (ATmega32u4) Velocidad de reloj: 16 MHz</p>
<p>Arduino Due</p> 	<p>Microcontrolador: AT91SAM3X8E Voltaje de funcionamiento: 3.3 V Pines I/O digitales: 54 (de los cuales 12 proveen salida PWM) Pines de entradas análogas: 12 Corriente DC total en todos los pines I/O: 130 mA Corriente DC en el pin de 5 V: 800 mA Corriente DC en el pin de 3.3 V: 800 mA Memoria Flash: 512 KB disponibles para las aplicaciones de usuario. SRAM: 96 KB (dos bancos: 64KB Y 32 KB) Velocidad de reloj: 84 MHz</p>
<p>Arduino Yún</p> 	<p>Microcontrolador AVR Arduino: ATmega32u4 Voltaje de funcionamiento: 5 V Pines I/O digitales: 20 Canales PWM: 7 Pines de entradas análogas: 12 Corriente DC por cada pin I/O: 40 mA Corriente DC en el pin de 3.3 V: 50 mA Memoria Flash: 32 KB (de los cuales 4 KB son utilizados por el bootloader) SRAM: 2.5 KB EEPROM: 1 KB Velocidad de reloj: 16 MHz Procesador Linux: Atheros AR9331 Arquitectura: MIPS @400MHz Ethernet: IEEE 802.3 10/100Mbit/s WiFi: IEEE 802.11b/g/n USB Tipo A: 2.0 Lector de tarjeta: sólo Micro-SD RAM: 64 MB DDR2 Memoria Flash: 16 MB</p>
<p>Arduino Robot</p> 	<p>Microcontrolador: ATmega32u4 Voltaje de funcionamiento: 5 V Pines I/O digitales: 5 Canales PWM: 6 Canales de entradas análogas: 4 (de los pines digitales I/O) Canales (multiplexados) de entradas análogas: 8 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader SRAM: 2 KB (ATmega32u4) EEPROM (interno): 1 KB (ATmega32u4) EEPROM (externo): 512 KB (I2C) Velocidad de reloj: 16 MHz Teclado: 5 teclas Perilla: Potenciómetro conectado a un pin análogo LCD a color: Comunicación SPI Lector de tarjetas SD: Para tarjetas formateadas FAT16 Altavoz: 8 Ohms Compás digital: Proporciona la desviación desde el norte geográfico en grados Áreas de prototipado: 4</p>
<p>Arduino Esplora</p> 	<p>Microcontrolador: ATmega32u4 Voltaje de funcionamiento: 5 V Memoria Flash: 32 KB de los cuales 4 KB son utilizados por el bootloader SRAM: 2.5 KB EEPROM: 1 KB Velocidad de reloj: 16 MHz 4 Push buttons Joystick análogo con un push button central Potenciómetro lineal Micrófono Fotorresistor Sensor de temperatura Acelerómetro de 3 ejes Buzzer Led RGB Conector para LCD</p>

<p>Arduino Mega ADK</p> 	<p>Microcontrolador: ATmega2560 Voltaje de funcionamiento: 5 V Pines I/O digitales: 54 (de los cuales 15 proveen salida PWM) Pines de entradas analógicas: 16 Corriente DC por cada pin I/O: 40 mA Corriente DC en el pin de 3.3 V: 50 mA Memoria Flash: 256 KB de los cuales 8 KB son utilizados por el bootloader SRAM: 8 KB EEPROM: 4 KB Velocidad de reloj: 16 MHz</p>
<p>Arduino Ethernet</p> 	<p>Microcontrolador: ATmega328 Voltaje de funcionamiento: 5 V Pines I/O digitales: 14 (de los cuales 4 proveen salida PWM) Pines de entradas analógicas: 6 Corriente DC por cada pin I/O: 40 mA Corriente DC en el pin de 3.3 V: 50 mA Memoria Flash: 32 KB (ATmega328) de los cuales 0.5 KB son utilizados por el bootloader SRAM: 2 KB (ATmega328) EEPROM: 1 KB (ATmega328) Velocidad de reloj: 16 MHz Controlador embebido Ethernet W5100 TCP/IP Tarjeta MicroSD, con adaptadores activos de voltaje</p>
<p>Arduino Mega 2560</p> 	<p>Microcontrolador: ATmega2560 Voltaje de funcionamiento: 5 V Pines I/O digitales: 54 (de los cuales 15 proveen salida PWM) Pines de entradas analógicas: 16 Corriente DC por cada pin I/O: 40 mA Corriente DC en el pin de 3.3 V: 50 mA Memoria Flash: 256 KB de los cuales 8 KB son utilizados por el bootloader SRAM: 8 KB (ATmega328) EEPROM: 4 KB (ATmega328) Velocidad del reloj: 16 MHz</p>
<p>Arduino Mini</p> 	<p>Microcontrolador: ATmega328 Voltaje de funcionamiento: 5 V Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) Pines de entradas analógicas: 8 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 32 KB de los cuales 2 KB son utilizados por el bootloader SRAM: 2 KB EEPROM: 1 KB Velocidad de reloj: 16 MHz</p>
<p>Arduino Nano</p> 	<p>Microcontrolador: ATmega168 Voltaje de funcionamiento: 5 V Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) Pines de entradas analógicas: 8 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 16 KB de los cuales 2 KB son utilizados por el bootloader SRAM: 1 KB EEPROM: 512 bytes Velocidad de reloj: 16 MHz</p>
<p>Arduino Pro Mini</p> 	<p>Microcontrolador: ATmega168 Voltaje de funcionamiento: 3.3 V Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) Pines de entradas analógicas: 8 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 16 KB de los cuales 2 KB son utilizados por el bootloader SRAM: 1 KB EEPROM: 512 bytes Velocidad de reloj: 8 MHz</p>

<p>Arduino Pro</p> 	<p>Microcontrolador: ATmega168 Voltaje de funcionamiento: 3.3 V Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) Pines de entradas analógicas: 6 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 16 KB de los cuales 2 KB son utilizados por el bootloader SRAM: 1 KB EEPROM: 512 bytes Velocidad de reloj: 8 MHz</p>
<p>Arduino Micro</p> 	<p>Microcontrolador: ATmega32u4 Voltaje de funcionamiento: 5 V Pines I/O digitales: 20 Canales PWM: 7 Pines de entradas analógicas: 12 Corriente DC por cada pin I/O: 40 mA Corriente DC en el pin de 3.3 V: 50 mA Memoria Flash: 32 KB (ATmega32u4) de los cuales 4 KB son utilizados por el bootloader SRAM: 2.5 KB (ATmega32u4) EEPROM: 1 KB (ATmega32u4) Velocidad de reloj: 16 MHz</p>
<p>Arduino Fio</p> 	<p>Microcontrolador: ATmega328P Voltaje de funcionamiento: 3.3 V Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) Pines de entradas analógicas: 8 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 32 KB de los cuales 2 KB son utilizados por el bootloader SRAM: 2 KB EEPROM: 1 KB Velocidad de reloj: 8 MHz</p>
<p>LilyPad Arduino USB</p> 	<p>Microcontrolador: ATmega32u4 Voltaje de funcionamiento: 3.3 V Pines I/O digitales: 9 Canales PWM: 4 Pines de entradas analógicas: 4 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 32 KB de los cuales 4 KB son utilizados por el bootloader SRAM: 2.5 KB EEPROM: 1 KB Velocidad de reloj: 8 MHz</p>
<p>LilyPad Arduino Simple</p> 	<p>Microcontrolador: ATmega328 Voltaje de funcionamiento: 2.7-5.5 V Pines I/O digitales: 9 (de los cuales 5 proveen salida PWM) Pines de entradas analógicas: 4 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 32 KB de los cuales 2 KB son utilizados por el bootloader SRAM: 2 KB EEPROM: 1 KB Velocidad de reloj: 8 MHz</p>
<p>LilyPad Arduino SimpleSnap</p> 	<p>Microcontrolador: ATmega328 Voltaje de funcionamiento: 2.7-5.5 V Pines I/O digitales: 9 (de los cuales 5 proveen salida PWM) Pines de entradas analógicas: 4 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 32 KB de los cuales 2 KB son utilizados por el bootloader SRAM: 2 KB EEPROM: 1 KB Velocidad de reloj: 8 MHz</p>

<p>LilyPad Arduino</p> 	<p>Microcontrolador: ATmega168V Voltaje de funcionamiento: 2.7-5.5 V Pines I/O digitales: 14 (de los cuales 6 proveen salida PWM) Pines de entradas análogas: 6 Corriente DC por cada pin I/O: 40 mA Memoria Flash: 16 KB de los cuales 2 KB son utilizados por el bootloader SRAM: 1 KB EEPROM: 512 bytes Velocidad de reloj: 8 MHz</p>
--	--

Fuente: hacedores (2014). Recuperado de <https://hacedores.com>

1.3.1.3. Arduino Mega 2560

Arduino Mega fue seleccionado para utilizar en el proyecto por las características que brinda y se encuentran descritas continuación, las características principales por las que fue elegida esta placa es por la capacidad de memoria disponible para la programación que es de 256 KB, la cantidad y distribución de pines que tiene disponible para la conexión de los elementos electrónicos que forman parte del proyecto, a diferencia de Arduino UNO que posee una memoria de 32 KB para programar y un número de pines reducido. Esta versión de Arduino es una placa que posee un microcontrolador Atmega 2560 y razón por la cual adquiere su nombre comercial.

Arduino Mega cuenta de 54 pines para entradas y salidas digitales, de estos pines se puede utilizar 14 como salidas con PWM (*Pulse-width modulation* "modulación por ancho de pulso"), 16 pines de entradas analógicas, un cristal oscilador de 16 MHz, 4 UART (Puertos Series), conexión a USB, conector de alimentación, conector ICSP (*In Chip Serial Programmer* "Programación serial en el circuito") y un botón reset, esta placa para empezar a utilizar simplemente se la debe conectar a la PC con un cable USB.

Esta placa realiza su conversión de USB a Serie con el uso del microcontrolador Atmega8U2. Arduino Mega2560 por sus características y facilidades que presta es utilizado en proyectos de mucha importancia, entre los más relevantes se encuentra Las Impresoras 3D y Domótica.



Figura 16. Placa Arduino Mega2560.

Fuente: Garcia. A (2013). Recuperado de <http://panamahitek.com>

1.3.1.4. Microcontrolador Atmega2560

Este microcontrolador es la parte esencial de Arduino Mega, Atmega 2560 cuenta con las siguientes características: arquitectura RISC, Memoria flash para programación de 256KB y que de los cuales 8 KB son usados al momento de arrancar el sistema (bootloader), Empaquetado TQFP-100, bajo consumo de potencia, cuenta con Memoria SRAM de 8 KB, Memoria EEPROM de 4 KB, Interfaz 2-wire, Interfaz SPI, Interfaz USART, 86 pines programables de salida / entrada, 6 temporizadores, 16 canales ADC que trabajan a 10 bits, trabaja con 8 bits y una velocidad de 16 MHz.



Figura 17. Microcontrolador ATMEGA2560

Fuente: ebay (2018). Recuperado de <https://www.ebay.es>

1.3.1.5. Plataforma de Arduino.

Para elaborar los algoritmos que controlaran la programación del dispositivo Biométrico se va a utilizar la plataforma libre de Arduino, la misma que cuenta con un idioma de programación C++ simplificado.



Figura 18. Logo de arranque de plataforma Arduino

Fuente: Elaborado por el Autor.

Antes de empezar con la programación en Arduino hay que descargar las librerías faltantes, las mismas que se encuentran en el internet. Una vez descargadas las librerías se las incluye en el programa.

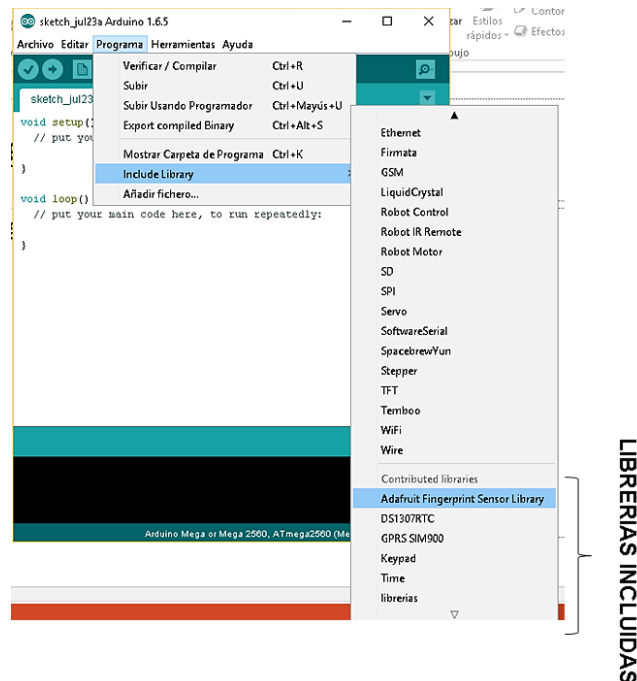


Figura 19. Inclusión de librerías en Arduino.

Fuente: Elaborado por el Autor.

1.3.2. Sensor finger print FPM10A

El lector de huella digital es un sensor biométrico que es ideal para realizar un sistema capaz de proteger lo que se requiera por medio del análisis de huellas dactilares en Arduino.

El sistema realiza procesamiento digital de imágenes de manera interna con un DSP (Procesador de Señales Digitales). Este sensor es idóneo para realizar comparaciones y actualizaciones en la base de datos.

Finger print funciona con protocolo serial, y puede trabajar con tarjetas de desarrollo como con microcontroladores. Cuenta con un espacio de almacenamiento de hasta 162 huellas dactilares en su memoria FLASH interna.

El LED del sensor se enciende cada vez que realiza tomas de imágenes en busca de huellas digitales. Sensor *Finger Print* FPM10A fue seleccionado para ser usado en el proyecto por sus características y la fácil adquisición en el mercado local.

FPM10A se encarga de realizar la recepción y almacenamiento de las huellas dactilares para realizar los registros de entrada y salida.



Figura 20. Lector de huella digital

Fuente: Aliexpress (2018). Recuperado de www.aliexpress.com

Tabla 2. Especificaciones

ESPECIFICACIONES TÉCNICAS DEL DISPOSITIVO
Voltaje de alimentación: 3.3V
Corriente de operación: 100mA – 150Ma
Tiempo de entrada de imágenes de huellas dactilares: <1 segundo
Tamaño de ventana: 14 x 18 mm
Interfaz: UART TTL
Modo de paridad de huella: 1:1 1:N
Baud Rate: 9600*N
N = 1 a 12 (Por defecto es 6)
Tiempo de adquisición menor a 1 segundo
5 Niveles de seguridad
Dimensión de la ventana: 14x18mm
Entorno de trabajo: -10°C a 40°C (Humedad Relativa 40% a 85%)
Dimensiones: 5.5 x 2.1 x 2.0 cm
Peso: 22g

Fuente: Electronilab (2018). Recuperado de <https://electronilab.co>

1.3.3. Módulo Tiny RTC DS 1307

El módulo RTC DS 1307 es un reloj en tiempo real, esta placa electrónica funciona con un bajo consumo de energía. Este dispositivo es de gran utilidad en el proyecto ya que facilita hora y fecha en actualizadas lo cual facilita el registro de entrada y salida de los colaboradores.

También se lo utiliza para el envío de SMS a la hora previamente programada (10 min. antes de la hora de ingreso)

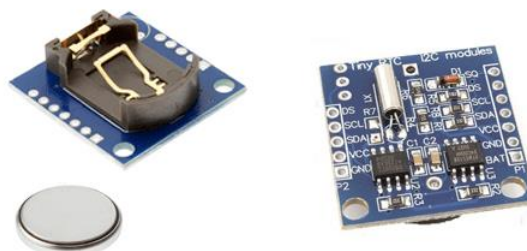


Figura 21. Módulo RTC DS 1307.

Fuente: Llamas. L (2016). Recuperado de <https://www.luisllamas.es>

Tabla 3. Especificaciones técnicas RTC.

ESPECIFICACIONES TÉCNICAS
Reloj en tiempo real (RTC) Cuenta segundos, Minutos, horas, fecha del mes, mes, día de la semana, y año con año bisiesto Compensación Válido hasta 2100.
56-Byte, con respaldo de batería, no volátil (NV) de RAM para almacenamiento de datos
Interface Serie I2C.
Onda-Cuadrada programable de la señal de salida.
Detector Automático Fallo-Energía y Circuito Conmutación.
Consume menos de 500nA en la batería -- Modo de copia de seguridad con el oscilador funcionando.
Rango de temperatura Industrial Opcional: -40 ° C a +85 ° C
Disponible en 8-Pin Plástico DIP o SO
Reconocido Underwriters Laboratory (UL)

Fuente: hispavila (s.f.). Recuperado de <http://hispavila.com>

1.3.4. Módulo GSM-GPRS-SMS-SIM900

El GPRS SIMCOM SIM900 GSM Quad band GSM shield es una tarjeta ultra compacta de comunicación inalámbrica. Este módulo es compatible con todos los modelos de Arduino con el formato UNO, es decir, que la puede controlar también con otros microcontroladores.

La tarjeta del módulo está basada en el módulo SIM900 GSM 4, el GPRS está configurado y controlada por vía UART usando comandos AT. Por lo tanto sólo conecta la tarjeta al microcontrolador, Arduino, etc, y comienza a comunicarte a través de comandos AT. (LARA, 2015)

Ideal para sistemas remotos, comunicación recursiva, puntos de control, mandar mensajes de texto a celulares, etc. El módulo GSM-GPRS-SMS-SIM900 se utiliza en el proyecto para enviar los mensajes cortos (SMS)



Figura 22. Módulo GSM-GPRS-SMS-SIM900

Fuente: ebay (2018). Recuperado de <https://www.ebay.in>

Tabla 4. Especificaciones técnica módulo GSM-GPRS-SMS-SIM900.

ESPECIFICACIONES TÉCNICAS
Quad-Band 850 / 900/ 1800 / 1900 MHz - funciona en todas las redes celulares del planeta.
GPRS multi-slot class 10/8
Estación móvil GPRS clase B
Cumple con GSM phase 2/2+
Clase 4 (2 W @ 850 / 900 MHz)
Clase 1 (1 W @ 1800 / 1900MHz)
Controlable via comandos AT estándar: GSM 07.07 & 07.05 Comandos mejorados: SIMCOM AT Commands.
Servicio Short Message Service(SMS) - para poder enviar pequeños paquetes de datos a través de la red celular.
Pila TCP/UDP incorporada - permite enviar datos a un servidor web. Ejem: Pachube, Fusion Tables
Incorpora un Real Time Clock - RTC. (Requiere pila)
Puerto serial configurable para comunicación con el shield.
Soporte para comunicación por softwareSerial (pines 6 y 7).
Jack para audífonos y micrófono (handsfree)
Bajo consumo (en modo sleep) - 1.5Ma
Jack para alimentación externa.
Soporta rangos temperatura - -40°C to +85 °C

Fuente: botscience (2013). Recuperado de <http://botscience.net> <http://botscience.net>

1.3.5. Pantalla LCD 20X4:

Pantalla LCD que cuenta de 20x4 caracteres, esta LCD trabaja con tecnología STN, las dimensiones de la pantalla es de 9.8 cm x 6 cm de ancho, este módulo es utilizado en el proyecto para presentar la información que imprime el dispositivo. Caldas (s.f.)

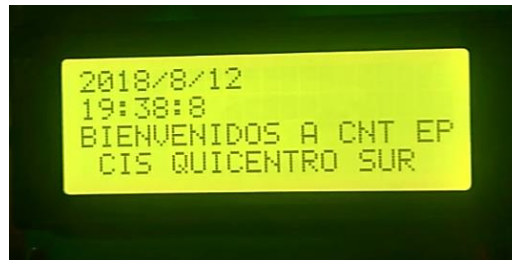


Figura 23. Pantalla LCD 20x4.

Fuente: Elaborado por el Autor.

Tabla 5. Especificaciones técnicas LCD.

ESPECIFICACIONES TÉCNICAS
20 caracteres x 4 líneas
Caracteres de 5x8 puntos
Tamaño de caracteres: 4.74 mm x 2.94 mm
Puede mostrar letras, números, caracteres especiales, y hasta 8 caracteres creados por el usuario
Interface paralela. Puede operar en modo de 8 bits, o de 4 bits para ahorrar pines del microcontrolador
Posee controlador KS0066 o equivalente on-board (compatible Hitachi HD44780)
Voltaje de alimentación: 5 V
Tamaño total: 98 mm x 60 mm x 14 mm

Fuente: Caldas (s.f.) Recuperado de <https://www.electronicoscaldas.com>

1.3.6. Módulo SD CARD

Módulo diseñado para el almacenamiento de información por medio de una tarjeta SD y que después va a ser transferido a la PC para la consolidación de la información recogida por el biométrico.



Figura 24. Módulo SD CARD.

Fuente: naylamp (s.f.). Recuperado de <https://naylampmechatronics.com>

Tabla 6. Especificaciones técnicas módulo SD CARD.

ESPECIFICACIONES TÉCNICAS
Voltaje de Operación: 3.3V-5V
Interfaz: SPI
Cuenta con todos los pines SPI de la tarjeta SD: MOSI, MISO, SCK, CS

Fuente: naylamp (s.f.). Recuperado de <https://naylampmechatronics.com>

1.3.7. Teclado 4X4.

Teclado cuenta con 4 filas distribuidas en (A, B, C, D) y 4 columnas (1, 2, 3,4), en total cuenta con 16 teclas independientes pero por estar conectado en una matriz de pulsadores en la placa de Arduino solo se ocupa 8 pines.

Este teclado va ser utilizado en el proyecto para el ingreso de código al momento de realizar el registro de asistencias.

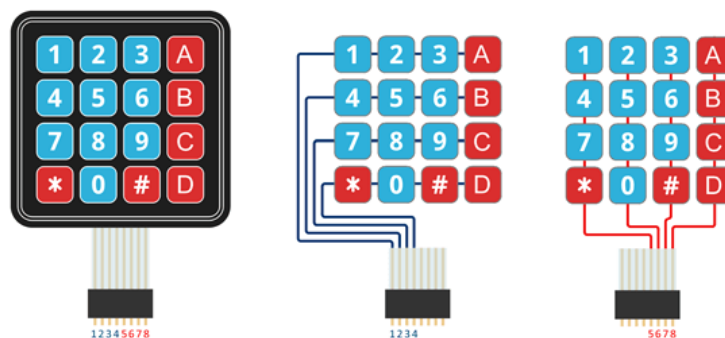


Figura 25. Teclado 4X4.

Fuente: Llamas. L (2016). Recuperado de <https://www.luisllamas.es>

Tabla 7. Especificaciones técnicas teclado.

ESPECIFICACIONES TÉCNICAS
16 botones con organización matricial (4 filas x 4 columnas)
Teclado tipo membrana
Mayor resistencia al agua y al polvo
Auto adhesivo en la parte de atrás
Tiempo de rebote (Bounce time): ≤5 ms
Máximo voltaje operativo: 24 V DC
Máxima corriente operativa: 30 Ma
Resistencia de aislamiento: 100 MΩ (@ 100 V)
Voltaje que soporta el dieléctrico: 250 VRMS (@ 60Hz, por 1 min)
Expectativa de vida: 1.000.000 de operaciones
Dimensiones del pad: 6.9 x 7.6 cm aprox.
Cable de cinta plana de 8.5 cm de largo aprox. (incluido el conector)
Conector tipo DuPont hembra de una fila y 8 contactos con separación estándar 0.1" (2.54mm)
Temperatura de operación: 0 a 50 °C

Fuente: Marker electrónico (2018). Recuperado de <https://www.makerelectronico.com>

CAPITULO II. MARCO METODOLÓGICO

Este capítulo hace mención al tipo de metodología que se utilizara en el diseño e implementación del dispositivo propuesto, las actividades y recursos utilizados en su desarrollo, también se describe los diferentes tipos de estudios realizados para cumplir con los objetivos planteados en la presente investigación.

2.1. Modelo de investigación.

En toda investigación es importante alinearse a un modelo que regule su desarrollo, en la ejecución del estudio del tema propuesto se utiliza métodos que han sido comprobados anteriormente por otros investigadores.

El uso de los métodos de estudio contribuye para tener una exitosa conclusión de la investigación, dar veracidad, confiabilidad y practicidad a los resultados conseguidos en el desarrollo.

Tomando en consideración los objetivos planteados de la investigación, el autor del tema debe definir el mejor método que debe utilizar para solucionar el problema planteado, para lo cual es importante desarrollar un diseño de investigación específico, tomando en consideración las siguientes particularidades: los recursos con los que se cuenta para el desarrollo, problemas a solventar, método idóneo para la obtención de datos, entre otros.

El presente tema de estudio es considerado como una investigación proyectiva en función al objetivo que se debe cumplir, descriptiva por la manera de exponer los conocimientos de adquiridos durante la investigación, también se usa estrategia aplicada y no experimental.

Este modelo de investigación se lo conoce como proyecto factible, porque consiste en la realización de un plan, un programa y una propuesta para solucionar el problema planteado, intenta responder interrogantes sobre eventos hipotéticos del futuro, o partir de un diagnóstico preciso de las necesidades encontradas en el momento. En esta categoría se utilizan investigaciones para inventos, programas y diseños (Hurtado de Barrera, 2008).

Se utiliza para hallar una solución práctica a un problema, se describe como se debe alcanzar los objetivos propuestos. Cuando las situaciones no marchan como deberían, es importante realizar modificaciones porque hay características potenciales que no se están aprovechando y se las desperdicia en procesos ineficientes.

El investigador determina el problema y da a conocer sus teorías para la elaboración de una propuesta en base a la información recolectada. Este método de estudio es el adecuado para la ejecución de este proyecto ya que se basa en el desarrollo de un prototipo electrónico que ayudara a solucionar el problema existente, porque en la actualidad se está utilizando un método de registro obsoleto, motivo por el cual se desaprovecha el uso adecuado de recursos, energías y esfuerzos.

En la manera que son expuestos los conocimientos, es de una forma descriptiva. Por medio de uso de la investigación descriptiva se implementan procedimientos de análisis para caracterizar el método de estudio. Usualmente se utiliza la investigación descriptiva cuando el estudio necesita grades niveles de profundidad (Tamayo y Tamayo, 2005).

Usualmente la finalidad del investigador es explicar situaciones y eventos, razón por la cual describe, como es, como se manifiesta un fenómeno. El método de estudio descriptivo busca las propiedades específicas e importantes del fenómeno que es objeto de investigación.

Haciendo referencia el punto de vista científico nos especifica que describir es medir, razón por la cual en el estudio descriptivo del tema, se escoge características a ser medidas independientemente cada una ellas y explicar los resultados de la investigación.

El método de estudio descriptivo en comparación al poco estructurado de los estudios exploratorios, es indispensable tener conocimientos del área que se investiga para poder formular preguntas específicas que se busaca responder. La descripción del tema puede ser profunda en cualquier caso y se basa en la medición de uno o más propiedades del fenómeno descrito, es por tal razón que la investigación se considera descriptiva, porque se describen los eventos observados.

La estrategia para implementar, se considera de campo no experimental ya que los datos recolectados son obtenidos directamente de realidad de donde se presentan y que son conocidos como de primera mano o primarios.

Usualmente se utiliza este método de estudio para comprender y solventar dudas de alguna situación, necesidad o problema del contexto determinado. El investigador trabaja en el ambiente real en que conviven las personas y las fuentes consultadas que obtendrán los datos más importantes a ser analizados (Hernández, 2009)

El investigador se limita a observar situaciones ya existentes, sin provocar cambios intencionales, las variables independientes están predefinidas y no pueden ser manipuladas.

2.2. Técnica de recolección de datos.

Las técnicas de recolección de datos seleccionada resulta fundamental para un estudio, su propósito es el de obtener información en profundidad sobre el tema a tratar, con sus propias palabras, definiciones o términos de los sujetos en su contexto y utilizando palabras reflexivas y se trata de minimizar sus creencias, fundamentos o experiencias de vida asociados con el tema de estudio para así no interferir con la recolección de datos, y obtener datos más relevantes (Hernández 2009)

CAPITULO III. PROPUESTA

Para realizar la implementación del biométrico inteligente en la agencia de CNT EP del C.C. Quicentro Sur, es indispensable realizar el diseño de hardware y software que va a ser usado en el dispositivo biométrico.

El control del dispositivo está regido por medio de Arduino Mega 2560, se utilizó esta placa de Arduino por las características que cuenta entre las más importantes esta su capacidad de memoria y número de pines, estos son necesarios para la conexión de los módulos y componentes con los que cuenta el proyecto, y por la capacidad de memoria que permite realizar toda la programación necesaria para que trabaje el dispositivo.

Descripción de componentes:

- Sensor FINGERPRINT FPM10A: sensor encargado de receptar las huellas dactilares de los colaboradores para realizar los registros de entrada y salida.
- Módulo RTC (Reloj de tiempo real): este módulo es vital dentro del sistema ya que nos ayuda a mantener hora y fecha actualizada para el control de registros y envío de SMS a la hora programada.
- Módulo GSM-GPRS-SMS-SIM900: módulo utilizado en el dispositivo para el envío de alertas por medio SMS cortos.
- Pantalla LCD 20 x 4: la LCD nos muestra la información del registro ingresado ya sea nombre, fecha, hora y por las características que presta el display permite ver el texto de una forma mejor distribuida.
- Módulo SD CARD: módulo que alberga una tarjeta SD para la recolección de la información recolectada por el biométrico.
- Teclado 4x4: permite el ingreso del código del empleado al momento del registro y también se lo para eliminar usuarios del sistema.

En el diagrama de la figura 26, se ve la estructuración del *hardware* que está conformado de sus respectivas etapas (Entrada, Proceso & Control y Salida) y *software* (Algoritmos en Arduino), componentes que conforman el dispositivo biométrico.

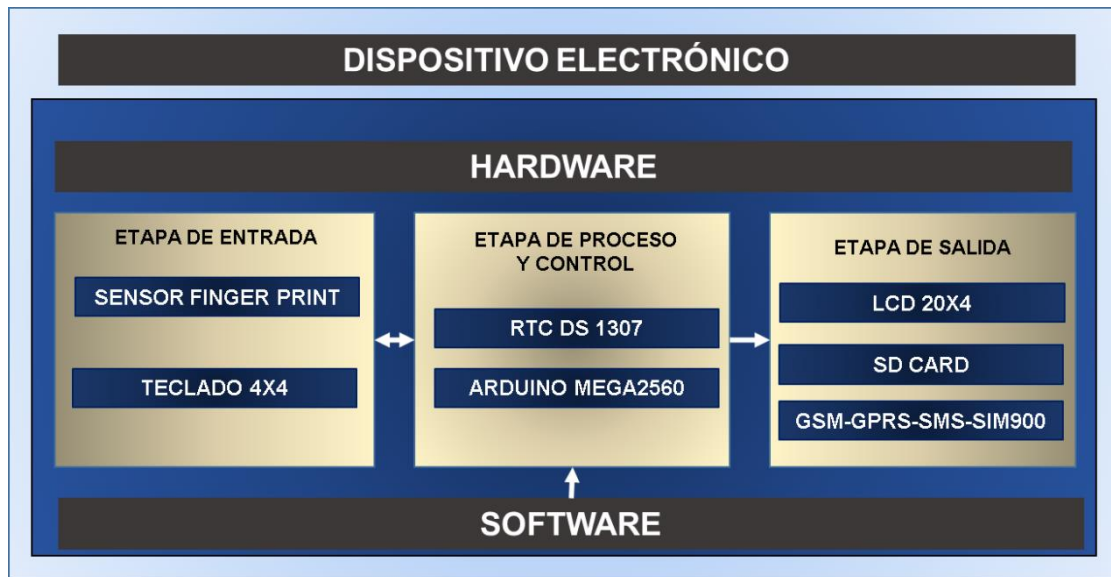


Figura 26. Diagrama estructural de bloques con *software* y *hardware* del dispositivo.

Fuente: Elaborado por el Autor.

3.1. Diseño de hardware.

3.1.1. Etapa de entrada

El ingreso de la información está a cargo del sensor *FINGER PRINT FMP 10 A* y del teclado matricial 4x4.

El sensor *FINGER PRINT* es el encargado de recibir la información de las huellas dactilares para el ingreso de información a la base de datos y también para el ingreso de registros diarios a la base de asistencia.

En la fase de ingreso de información para la base de datos, la información es recibida por medio de una imagen que es capturada por el sensor, luego la captura es procesada para crear un patrón único que es almacenado en la memoria del biométrico en binario, para lo cual 1 es blanco y 0 es negro.

El sensor en la fase de registros diarios recibe la información y la compara con la que ya tiene en la base de datos, luego de esto procede a dar procedente o rechazar el registro.

El teclado 4x4 permite el ingreso de información para el almacenamiento de registros y la eliminación de usuarios.

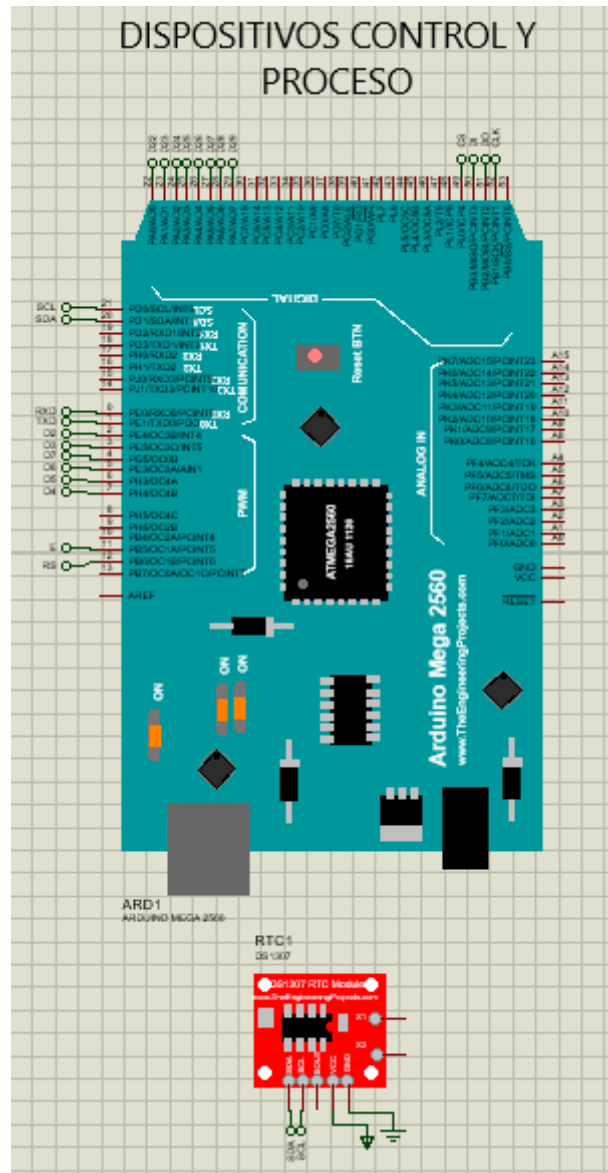


Figura 27. Diseño etapa de control proceso.

Fuente: Elaborado por el Autor.

3.1.2. Etapa de salida

Los datos procesados se los puede obtener y visualizar en los siguientes dispositivos: módulo GSM-GPRS-SMS-SIM900, Pantalla LCD 20 x 4, módulo SD CARD.

El módulo GSM-GPRS-SMS-SIM900 que enviará el siguiente texto “RECUERDA QUE TU HORA DE ENTRADA ES 9H50”.

El módulo SD CARD almacena información en formato .txt, datos que serán desplegados en un bloc de notas para luego ser transferido a un libro de Excel.

En imagen de la figura 28, se observa el diseño de los dispositivos que conforman la etapa de salida.

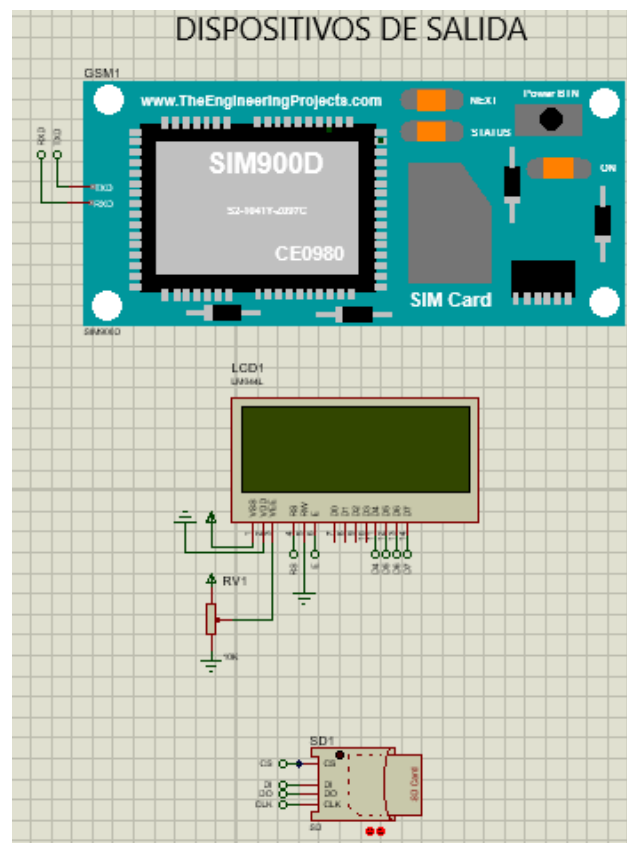


Figura 28. Dispositivos de etapa de salida.

Fuente: Elaborado por el Autor.

En la figura 29. Presenta el diseño total del dispositivo biométrico, se puede observar el diseño del hardware que será implementado en el proyecto.

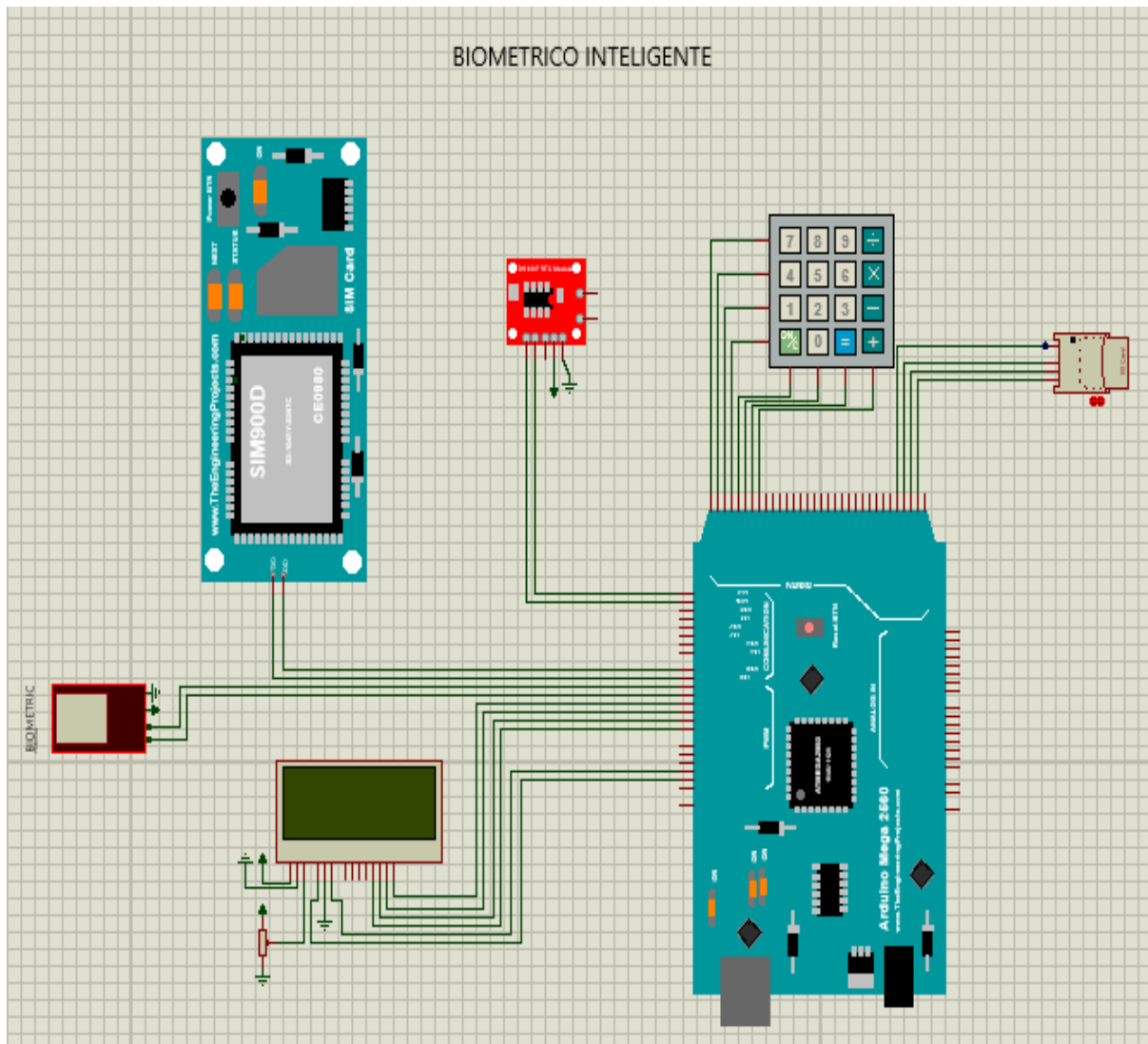


Figura 29. Diseño hardware

Fuente: Elaborado por el Autor.

3.2. Diseño del software

Para poder entender las configuraciones del programa a diseñar se lo representara en un flujograma que se puede visualizar en la imagen de la figura 30.

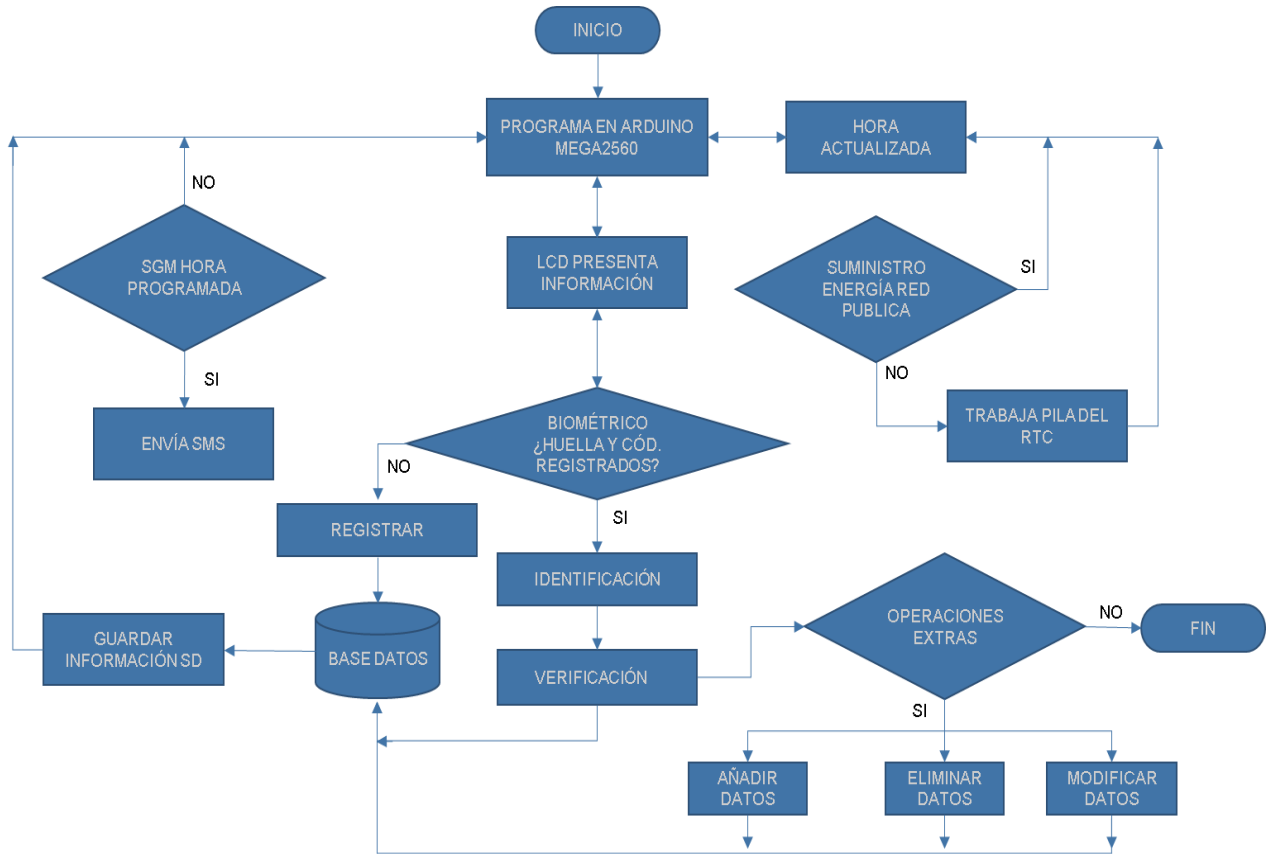


Figura 30. Flujograma de funcionamiento del programa.

Fuente: Elaborado por el Autor.

3.3. Presupuesto y materiales.

A continuación se describirá los elementos que son necesarios para la implementación de dispositivo propuesto.

Tabla 8. Tabla de materiales y precios.

N°	DESCRIPCION	CANT.	V.UNIT	V. TOTAL
1	ARDUINO MEGA 2560	1	\$ 24,50	\$ 24,50
2	TECLADO 4X4	1	\$ 3,90	\$ 3,90
3	RTC	1	\$ 4,50	\$ 4,50
4	PILA PARA RTC CP2032	1	\$ 1,50	\$ 1,50
5	SENSOR FINGER PRINT	1	\$ 49,00	\$ 49,00
6	MODULO SD GR	1	\$ 6,50	\$ 6,50
7	SIM900 GPRS SGM	1	\$ 45,00	\$ 45,00
8	FUENTE DE PODER 5V 2A	1	\$ 8,90	\$ 8,90
9	PLACA PCB PARA INTERCONEXION	1	\$ 7,90	\$ 7,90
10	REGLETA HEMBRA	2	\$ 0,50	\$ 1,00
11	REGLETA MACHO	1	\$ 0,49	\$ 0,49
12	POTENCIOMETRO	1	\$ 0,49	\$ 0,49
13	CONDENSADOR DE 100uf	1	\$ 0,29	\$ 0,29
14	LED	1	\$ 0,25	\$ 0,25
15	CABLES	30	\$ 0,10	\$ 3,00
16	LCD 20X4	1	\$ 16,50	\$ 16,50
17	CHIP CNT	1	\$ 4,48	\$ 4,48
18	SALDO	2	\$ 5,00	\$ 10,00
19	MICRO SD CON ADAPTADOR	1	\$ 6,00	\$ 6,00
20	ADAP. USB PARA SD	1	\$ 5,00	\$ 5,00
21	HORA HOMBRE	120	\$ 3,50	\$ 420,00
22	CAJA CONTENEDOR DISPOSITIVO	1	\$ 20,00	\$ 20,00
23	IMPREVISTOS Y VARIOS	1	\$ 50,00	\$ 50,00
TOTAL				\$ 689,20

Fuente: Elaborado por el Autor.

3.4. Ventajas que presta dispositivo biométrico.

- No necesita de una red de internet para la transferencia y almacenamiento de registros de entrada y salida de los empleados ya que cuenta de una SD que recolecta la información.
- Ayudará a reducir el índice de atrasos por el recordatorio de horario por medio de SMS que llegaran a los colaboradores.

- Reducir uso de papel.
- Veracidad en la información de ingresos y salidas de los empleados.

CAPITULO IV. IMPLEMENTACIÓN

4.1. Desarrollo

Para la elaboración del dispositivo biométrico fue necesario realizar las siguientes comprobaciones para asegurar que los elementos se encuentran funcionando, adicional se citará el nombre de los programas que se utilizaron para el desarrollo de hardware y software del dispositivo.

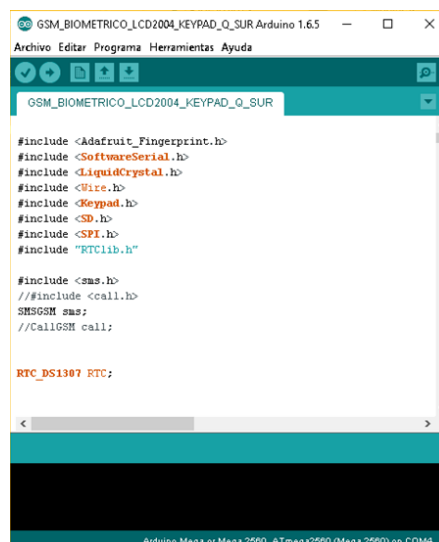
4.1.1. Desarrollo de software.

4.1.1.1. Arduino versión 1.6.5

Arduino es un programa de plataforma libre y que posee un lenguaje de programación C++ simple. En Arduino se elaboró la programación de logaritmos que controlara la secuencia de actividades que ejecutara el dispositivo biométrico.

Previo a iniciar la elaboración de la programación, fue necesario bajar las librerías necesarias del internet. Una vez que se obtuvo todas las librerías, se procedió adjuntar en la carpeta LIBRARY de Arduino.

Cuando la plataforma de Arduino se encuentre con las librerías completas se procede a realizar las líneas de programación necesarias para el control del biométrico.



```
GSM_BIOMETRICO_LCD2004_KEYPAD_Q_SUR Arduino 1.6.5
Archivo Editar Programa Herramientas Ayuda
GSM_BIOMETRICO_LCD2004_KEYPAD_Q_SUR

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include <Keypad.h>
#include <SD.h>
#include <SPI.h>
#include "RTClib.h"

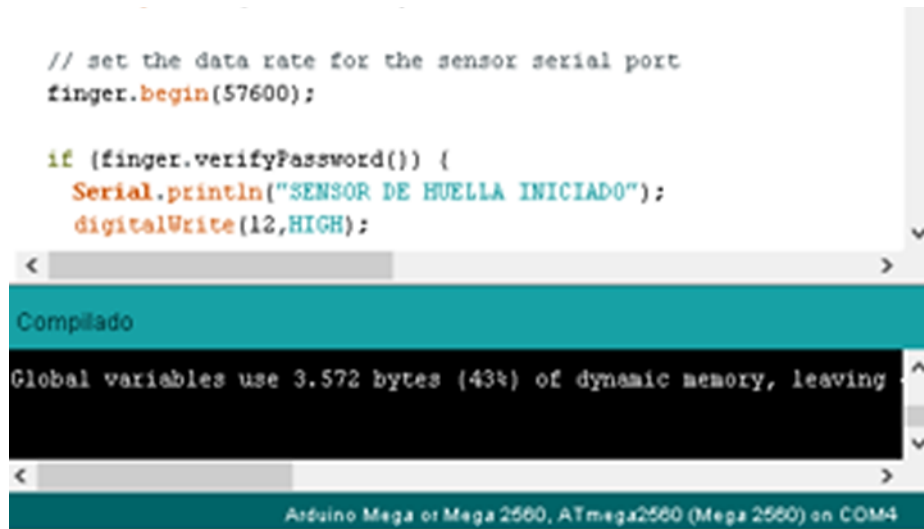
#include <sms.h>
// #include <call.h>
MSGSM sms;
// Call GSM call;

RTC_DS1307 RTC;
```

Figura 31. Programación Arduino versión 1.6.5

Fuente: Elaborado por el Autor.

Después de tener la programación lista se procede a compilar, para verificar que se encuentre correcta.



```
// set the data rate for the sensor serial port
finger.begin(57600);

if (finger.verifyPassword()) {
  Serial.println("SENSOR DE HUELLA INICIADO");
  digitalWrite(12,HIGH);
}
```

Compilado

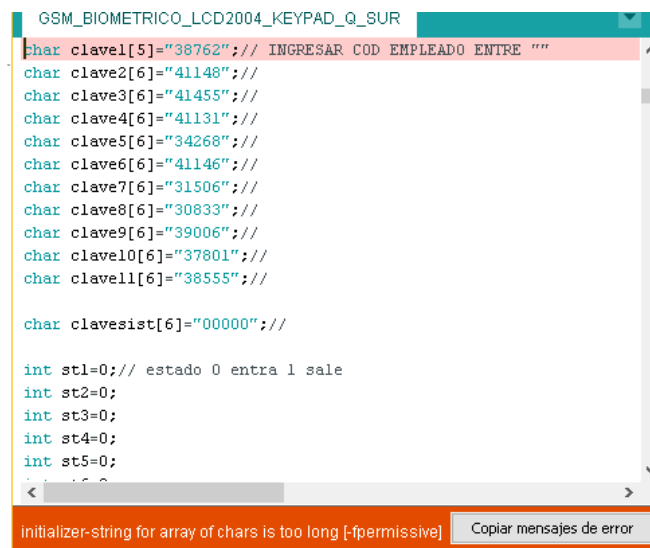
Global variables use 3.572 bytes (43% of dynamic memory, leaving

Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM4

Figura 32. Compilación de programación.

Fuente: Elaborado por el Autor.

En el caso que en la compilación haya un error de programación se mostrará un mensaje de advertencia, el programa se detendrá y mostrará la línea o líneas que presentan errores, los mismos que se los debe corregir.



```
GSM_BIOMETRICO_LCD2004_KEYPAD_Q_SUR
char clave1[5]="38762";// INGRESAR COD EMPLEADO ENTRE ""
char clave2[6]="41148";//
char clave3[6]="41455";//
char clave4[6]="41131";//
char clave5[6]="34268";//
char clave6[6]="41146";//
char clave7[6]="31506";//
char clave8[6]="30833";//
char clave9[6]="39006";//
char clave10[6]="37801";//
char clave11[6]="38555";//

char clavesist[6]="00000";//

int st1=0;// estado 0 entra 1 sale
int st2=0;
int st3=0;
int st4=0;
int st5=0;
int st6=0;
```

initializer-string for array of chars is too long [-fpermissive] Copiar mensajes de error

Figura 33. Error en la compilación.

Fuente: Elaborado por el Autor.

Después de realizar las respectivas correcciones en la programación, se compila nuevamente y debe salir un mensaje en la parte inferior que diga compilado.

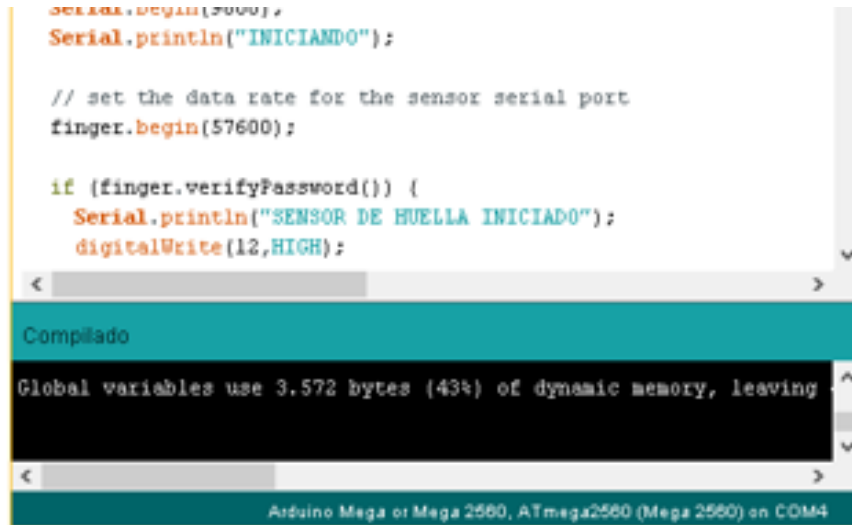


Figura 34. Compilación exitosa.

Fuente: Elaborado por el Autor.

Cuando la programación este compilada, se procede con la subida de los logaritmos a la tarjeta de Arduino MEGA 2560 para realizar las respectivas pruebas con todos los elementos.

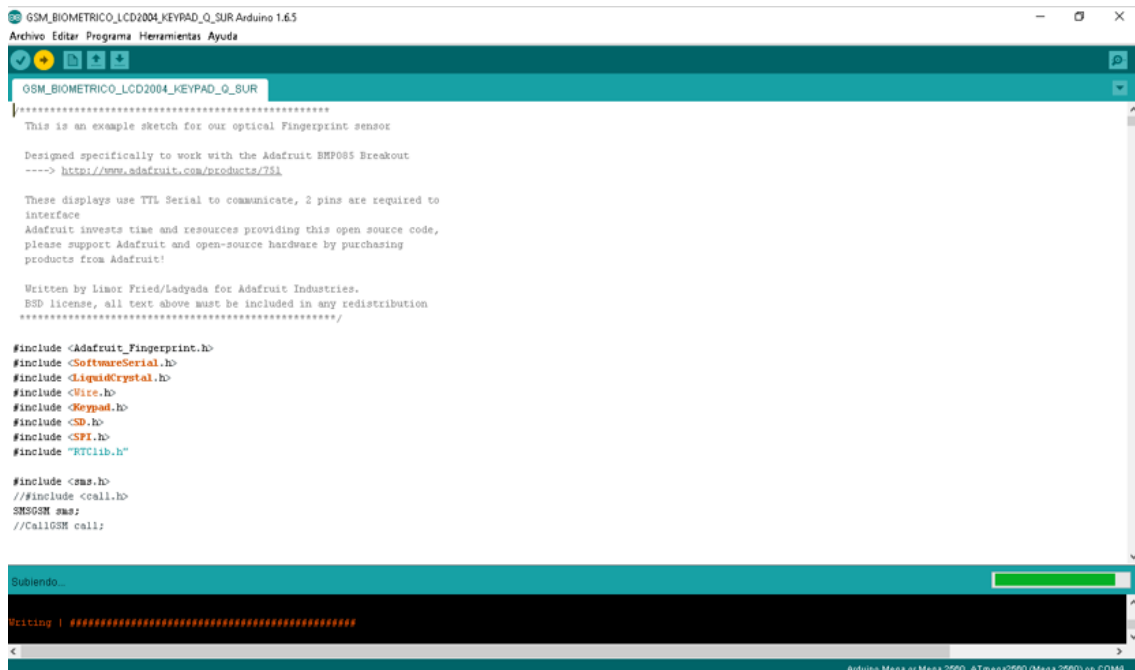


Figura 35. Programa subiendo a tarjeta Arduino.

Fuente: Elaborado por el Autor.

```

#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include <Keypad.h>
#include <SD.h>
#include <SPI.h>
#include "RTClib.h"

#include <sms.h>
//#include <call.h>
MSGSM sms;
//CallGSM call;

```

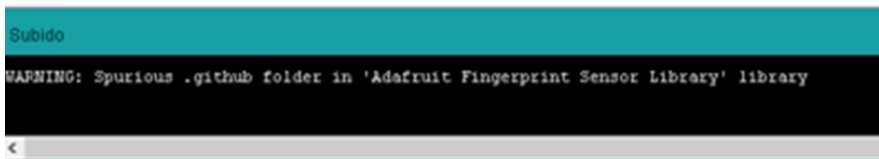


Figura 36. Programa subido con éxito.

Fuente: Elaborado por el Autor.

4.1.2. Desarrollo de hardware.

4.1.2.1. Desarrollo de montaje del dispositivo.

Previo a la implementación del dispositivo se realizó pruebas de funcionamiento a los elementos que conformar el dispositivo biométrico.

Arduino MEGA 2560 fue el primer elemento en realizarse la prueba de funcionamiento previo a su montaje, para lo cual se conectó a la PC y encendió sin novedad alguna.



Figura 37. Verificación de funcionamiento Arduino MEGA 2466

Fuente: Elaborado por el Autor.

La siguiente prueba de funcionamiento fue realizada entre Arduino MEGA 2560 y módulo GSM-GPRS-SMS-SIM900, dando un resultado positivo, lo que se evidencia en la imagen de la figura 38, las placas se encuentran encendidas.

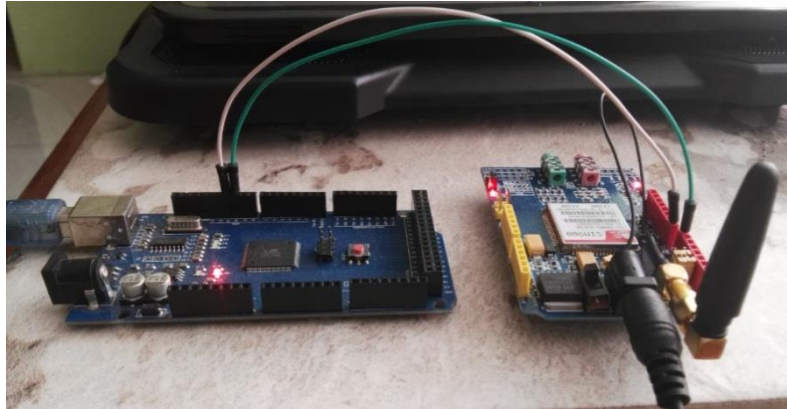


Figura 38. Comprobación Arduino MEGA 2560 con SIM 900.

Fuente: Elaborado por el Autor.

En la figura 39, se puede evidenciar la prueba de funcionamiento realizada con Arduino MEGA 2560 y el sensor finger print FPM10A, con un resultado positivo.

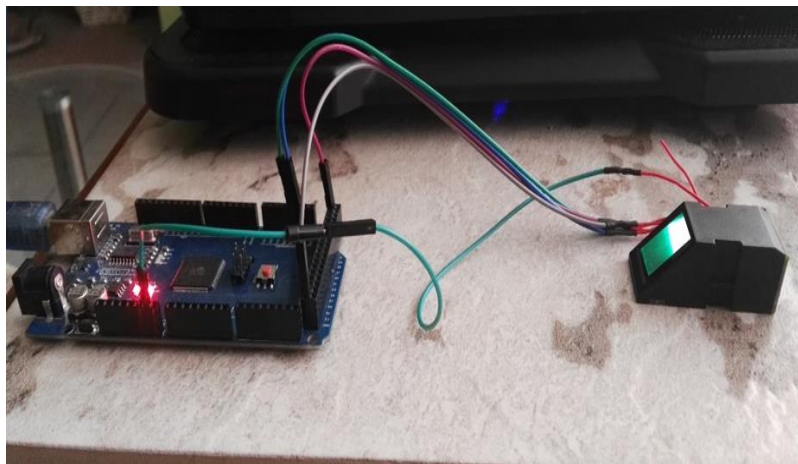


Figura 39. Comprobación Arduino MEGA 2560 con sensor FPM10A.

Fuente: Elaborado por el Autor.

4.1.2.2. Proteus 8.5

Proteus es un programa diseñado para la elaboración del diseño de circuitos electrónicos y diseños PCB para la impresión de tarjetas, también permite realizar simulaciones previas a las pruebas a realizar en los elementos electrónicos.

Proteus fue utilizado para realizar el diseño de la placa PCB que es utilizada para realizar las conexiones de los elementos del dispositivo y la tarjeta Arduino. Este programa también fue utilizado para la elaboración de las conexiones de los elementos que se puede evidenciar en la figura 40.

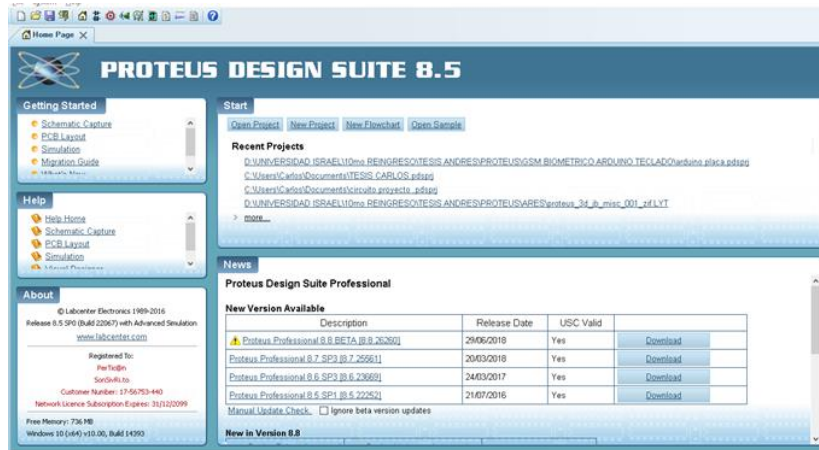


Figura 40. Proteus 8.5
Fuente: Elaborado por el Autor.

Realizadas las respectivas pruebas de los elementos principales por separado, el diseño de la PCB para las conexiones entre los componentes del depósito y la programación que controlara el biométrico realizaremos el montaje e implantación.

4.2. Implementación

La implementación del dispositivo empieza con la impresión y montaje de los componentes necesarios de la placa PCB para conexión de los elemento.

En la figura 41 y 42 se observa el montaje de las regletas hembra que servirán para colocar la lcd 20x4, módulo RTC y módulo SD CARD y cables del sistema.

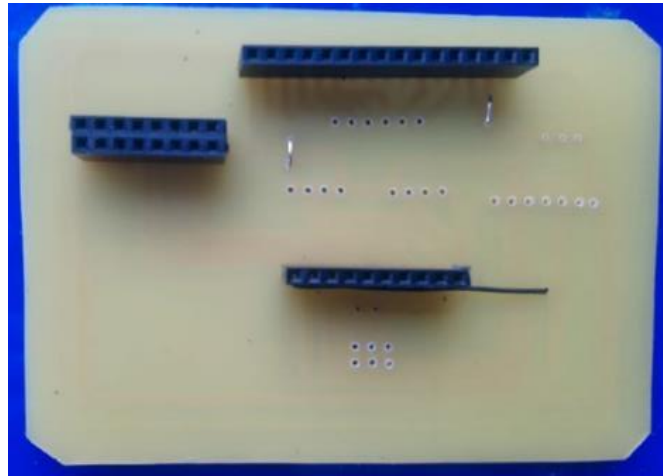


Figura 41. Montaje de regletas hembras.

Fuente: Elaborado por el Autor.

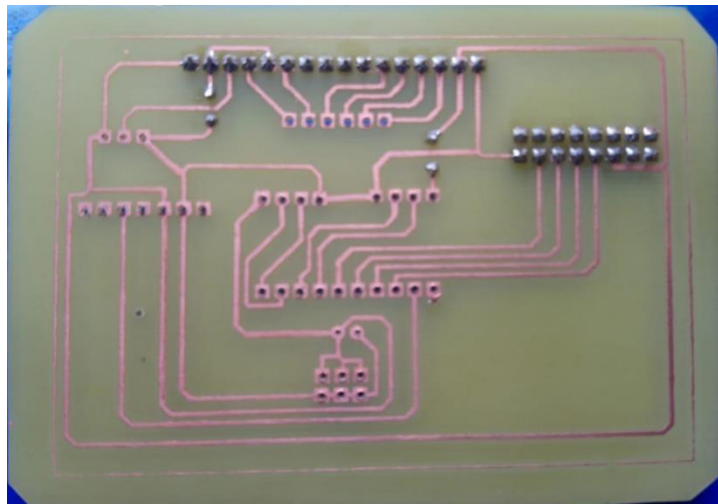


Figura 42. Soldadura de montaje de regletas hembras.

Fuente: Elaborado por el Autor.

El figura 43 y 44 podemos evidenciar las regletas completas y adicional se puede ver el montaje del potenciómetro que trabaja con la LCD, condensador 100uf y regleta macho para la conexión de 5V y GND del circuito.

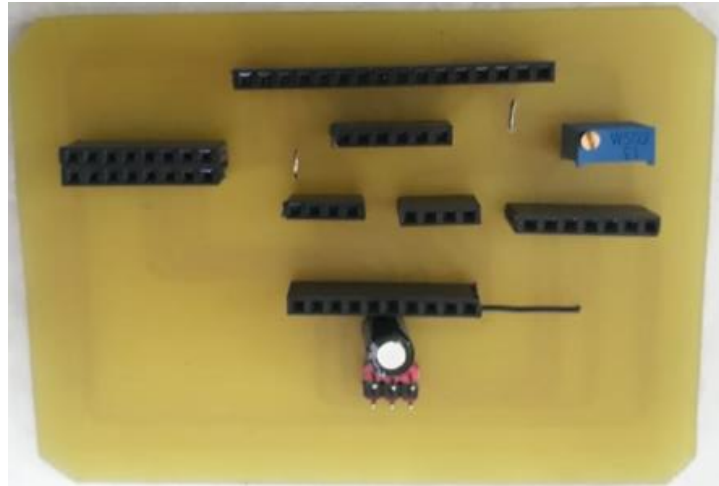


Figura 43. Montaje completo PCB.

Fuente: Elaborado por el Autor.

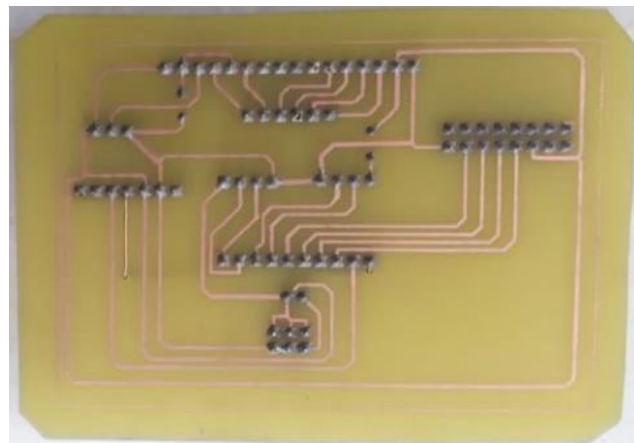


Figura 44. Soldadura de montaje completo.

Fuente: Elaborado por el Autor.

Una vez que se tiene montado todos los elementos pertenecientes a la caja se procedió a realizar el montaje de los componentes que conforman el dispositivo, tales como son el módulo RTC y el módulo SD CARD, los mismos que se pueden evidenciar en la figura 45.



Figura 45. Montaje de RTC y módulo SD

Fuente: Elaborado por el Autor.

Para que el módulo GSM-GPRS-SMS-SIM900 realice su función de enviar SMS, se procedió a colocar una SIM CARD en la ranura que se encuentra en el módulo, el chip pertenece a la operadora de CNT EP, y tiene asignado el número celular 0961213623.



Figura 46. Módulo GSM-GPRS-SMS-SIM900 con SIM CARD CNT EP.

Fuente: Elaborado por el Autor.

Se realizó en montaje y conexión de todos los elementos que conforman el dispositivo.

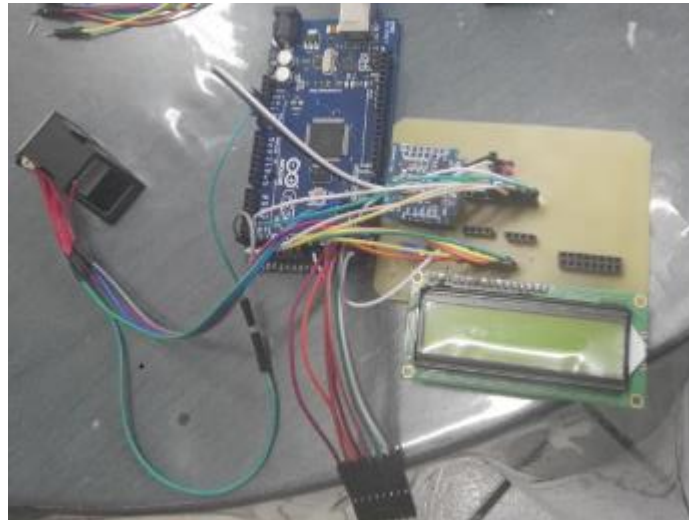


Figura 47. Montaje y conexión.

Fuente: Elaborado por el Autor.

Previo al montaje definitivo del dispositivo se realizó uno provisional en una tabla MDF, que aloja a todos los elementos que conforma el biométrico.



Figura 48. Montaje provisional

Fuente: Elaborado por el Autor

Para realizar la ejecución del montaje definitivo e implementación del dispositivo se diseñó una caja que servirá para colocar los componentes del dispositivo y realizar la entrega del biométrico en la Agencia de Quicentro Sur, perteneciente a CNT EP.

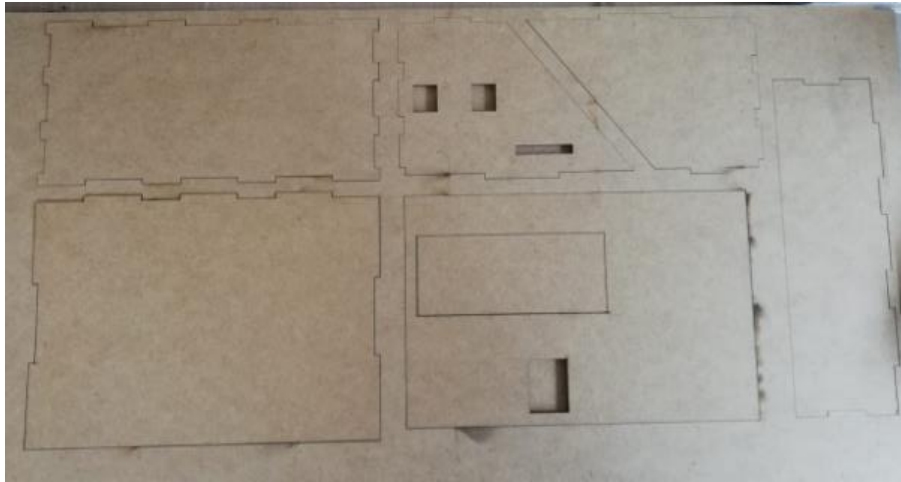


Figura 49. Carcasa para dispositivo

Fuente: Elaborado por el Autor.

Para la implementación se fijó un lugar que sea accesible para los empleados y que cuente con una toma corriente de 110V cerca.

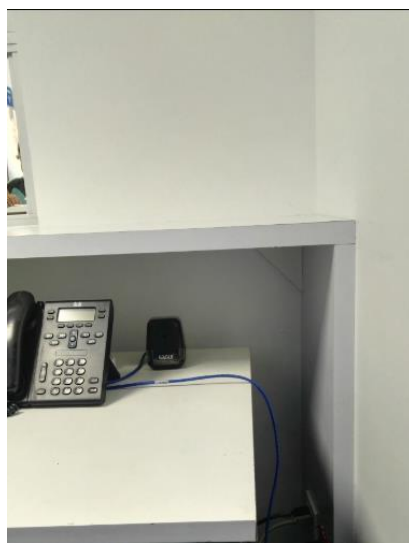


Figura 50. Lugar asignado para dispositivo

Fuente: Elaborado por el Autor.

Implementación de biométrico en lugar asignado por parte del supervisor de la agencia previa verificación de acceso al dispositivo para el personal.



Figura 51. Implementación dispositivo.

Fuente: Elaborado por el Autor.

4.2. Pruebas de funcionamiento.

Se realizó pruebas de arranque del dispositivo, para lo cual se utilizó el monitor serial de Arduino, donde indica que todos los equipos arrancaron correctamente.

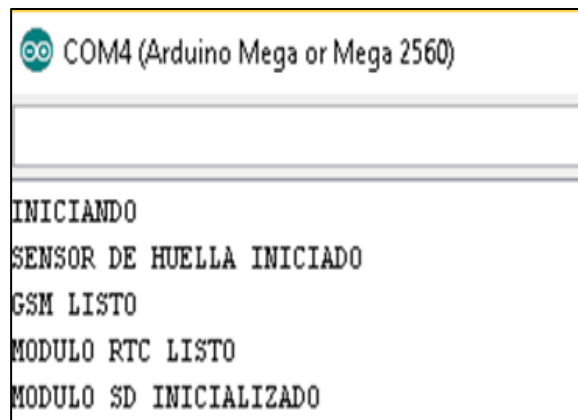


Figura 52. Comprobación de arranque.

Fuente: Elaborado por el Autor.

La imagen de la figura 53, muestra el arranque del dispositivo previo a la ejecución de las respectivas pruebas de funciones que deben cumplir los diferentes componentes del biométrico.



Figura 53. Arranque del dispositivo.

Fuente: Elaborado por el Autor.

Después de arranque exitoso del dispositivo se procedió hacer pruebas de ingreso de claves y huellas dactilares.



Figura 54. Digitación de clave.

Fuente: Elaborado por el Autor.

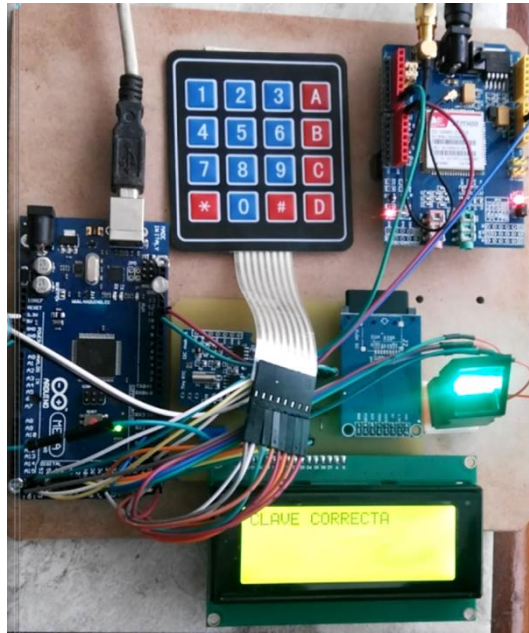


Figura 55. Validación de clave.

Fuente: Elaborado por el Autor.

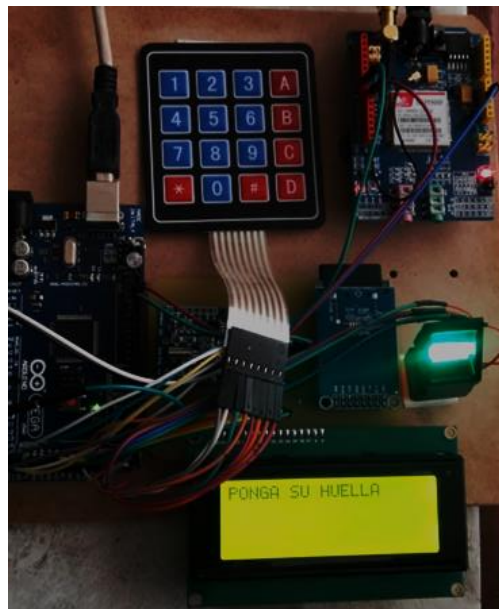


Figura 56. Solicitud de huella.

Fuente: Elaborado por el Autor.



Figura 57. Lectura de huella exitosa.

Fuente: Elaborado por el Autor.

Una vez que se realizaron las pruebas de registro de huellas y claves del usuario se sustrajo la tarjeta SD CARD del dispositivo para verificar el registro de los ingresos en la PC.

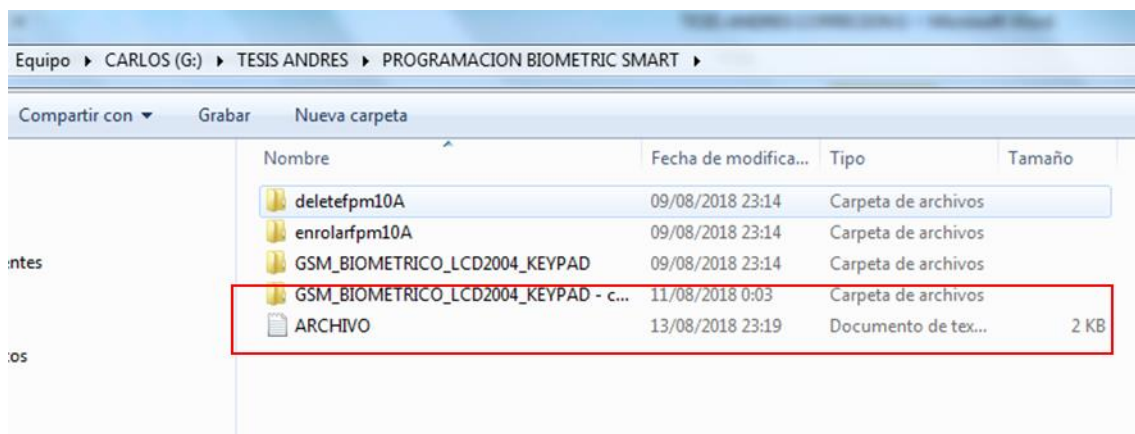


Figura 58. Archivo de registros.

Fuente: Elaborado por el Autor.

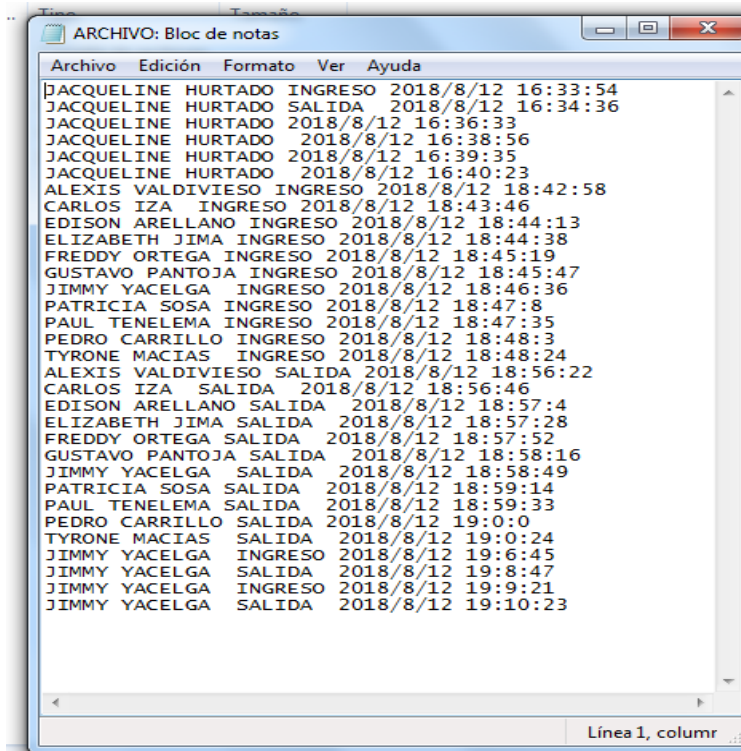


Figura. 59. Datos registrados en el archivo.

Fuente: Elaborado por el Autor.

La siguiente prueba que se realizó es el envío de SMS a los empleados de la Agencia Quicentro Sur.



Figura 60. Envío de SMS

Fuente: Elaborado por el Autor.

A continuación se ve en las imágenes de la figura 61, los SMS de prueba que se enviaron a los integrantes de la Agencia de Quicentro Sur.

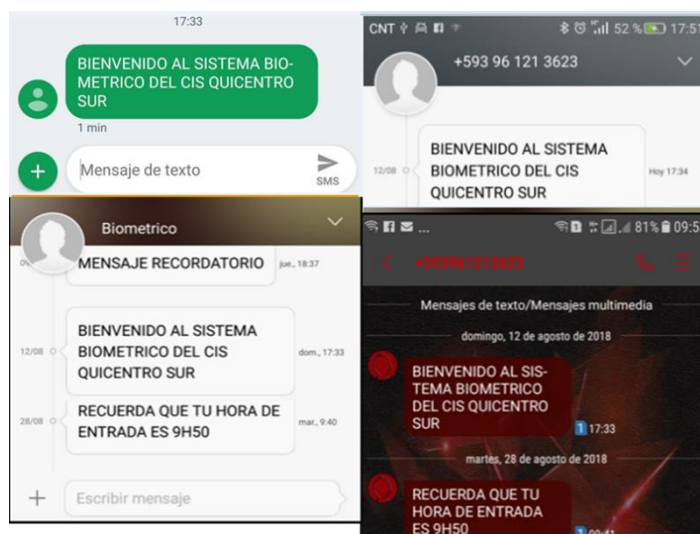


Figura 61. SMS enviado para pruebas.

Fuente: Elaborado por el Autor.

4.3. Análisis de resultados

En la tabla 9, se describe los resultados de las pruebas de arranque que se realizó en el dispositivo, esta información se la puede evidenciar en la figura 52 y 53.

Tabla 9. Tabla pruebas de arranque.

PRUEBA DE ARRANQUE DE DISPOSITIVO		
ELEMENTOS	ARRANCA	
	SI	NO
Arduino Mega 2560	✓	
Modulo SD CARD	✓	
Modulo GSM-GPRS-SMS-SIM900	✓	
Sensor Finger Print FPM10A	✓	
LCD 20X4	✓	
Modulo RTC	✓	

Fuente: Elaborado por el Autor.

En la tabla 10, se describe las actividades de pruebas realizadas en el dispositivo, las mismas que votaron un resultado positivo, las evidencias de las pruebas realizadas se puede evidenciar en las imágenes de las figuras 54,55,56,57, 58, 59,60 y 61.

Tabla 10. Tabla pruebas de funcionamiento.

RESULTADO DE PRUEBAS DE FUNCIONAMIENTO		
ACTIVIDADES	SE EJECUTO	
	SI	NO
REGISTRO DE CLAVE	✓	
REGISTRO DE HUELLA	✓	
ENVIO DE SMS	✓	
REGISTRO DE INFORMACION EN SD CARD	✓	

Fuente: Elaborado por el Autor.

CONCLUSIONES

La elaboración del dispositivo biométrico no presentó mucha complejidad, gracias a la información relevante que se encuentra en fuentes bibliográficas y que es de gran utilidad para la ejecución del proyecto.

Este proyecto se ejecutó con elementos electrónico que se puede encontrar en el mercado local, y que fácilmente se puede conseguir información para hacer el uso correcto de los componentes.

El dispositivo biométrico fue creado por la necesidad de la agencia de Quicentro Sur, debido a que el personal frecuentemente llega atrasado, y con la implementación del dispositivo está obligado a llegar temprano para realizar su registro de entrada, y de la misma manera sirvió para tener un mejor control del tiempo que se toma el personal para el almuerzo.

Los algoritmos de programación que controlan las funciones del dispositivo se las realizó con la plataforma de Arduino, esta plataforma cuenta con ejemplos en cada una sus librerías, ejemplos que dan una pauta para poder realizar la trama de programación que se necesita para el proyecto.

Usar el módulo GSM-GPRS-SMS-SIM 900 fue la mejor opción para el envío de SMS ya que se lo puede encontrar dentro del mercado local, tiene las características necesaria para poder cumplir con las funciones encomendadas, otra de las causas por lo que fue elegido este módulo es por las buenas referencias que tenían compañeros que lo habían utilizado en sus proyectos.

RECOMENDACIONES

Es de vital importancia el uso de una fuente de poder de 5V /2A, para evitar que el dispositivo se queme por una sobrecarga en sus elementos.

Para que el sensor biométrico del dispositivo funcione de correctamente hay que limpiarlo con frecuencia, porque puede dar una lectura errónea al momento de realizar los registros, esto se puede dar debido a que la grasa que tenemos en las huellas dactilares de los dedos quedan impresas en el lector, de la persona que realizaron los registros antes.

El dispositivo debe ser instalado en un lugar seco y ventilado, debido a que la humedad puede dañar sus componentes electrónicos.

La pila del módulo RTC, tiene un tiempo de vida aproximado de 2 años por lo cual es recomendado cambiar antes, en el caso que pase por alto esta recomendación el biométrico va generar novedades al momento de realizar registros y envío de SMS.

Para el envío de SMS se recomienda mantener activado un plan de mensajes escritos, ya que en el caso de no contar con este requisito, el dispositivo no enviara los SMS a los usuarios.

BIBLIOGRAFIA

WORDPRESS.(2009). Historia de la biometría.obtenido de INBIOSYS E.U:
<https://inbiosys.wordpress.com>

Tolosa. C y Giz. A (s.f.): Sistemas Biometricos
<https://www.dsi.uclm.es>

ES.SCRIBD.(s.f.) Biometria capacidades de ID, soluciones tecnológicas y empresas UPM.obtenido de Universidad Politecnica de Madrid
<https://es.scribd.com>

Tapia.X (2017) "Revisión Sistemática de Literatura: Vulnerabilidad de sistemas biométricos basados en huellas dactilares
<http://dspace.unl.edu.ec>

Delgado.G(s.f.) Biometria
<http://dspace.unl.edu.ec>

TEC-MEX.COM.MX.(2003) Como Funcionan los Lectores de Huella Digital obtenido de TEC Electronica, S.A. de C.V.
<https://www.tec-mex.com.mx>

Gonzales.J Y Sanchez.R(2013) Sistema de Identificacion Biometrica Basado en huella dactilar mediante binarizacion sobre plataformas Android
<https://earchivo.uc3m.es>

MONOGRAFÍAS.COM(s.f.) Revelado de huellas lofoscópicas en papel obtenido de monografias .com
<https://www.monografias.com>

Serna.R y Sanchez.R(2013) soluciones match-on-device y processingon-device en sistemas móviles
<https://core.ac.uk>

Garcia.E, Escamilla.E, Nakano.M y Perez.H(2014) Sistema de Reconocimiento de Rostros usando Visión Estéreo
<https://scielo.conicyt.cl>

Alcalde.A(2017) Biometria aplicada a la seguridad
<https://elbauldprogramador.com>

UNILLANOS.EDU.CO(2012) Reconocimiento de palabras aisladas para control de navegación de robot móvil obtenido de la revista Orinoquia
<http://orinoquia.unillanos.edu.co>

Perez.E(s.f.) Sistemas Biometricos
<https://sites.google.com>

Madrigal.C, Ramirez.J, Hoyo.J y Fernandez.D(2007) Diseño de un sistema biométrico de identificación usando sensores capacitivos para huellas dactilares

<http://www.scielo.org.co>

Seru.A(2014) Protección Industrial de Hardware Libre para Arduino y Derivados
<http://librearduino.blogspot.com>

GEEKFACTORY.MX(2015) Tutorial DS1307 Tiny RTC con Arduino obtenido de Geek Factory
<https://www.geekfactory.mx>

Llamas.L(2016) Reloj y calendario en Arduino con los RTC DS1307 y DS3231
<https://www.luisllamas.es>

PROGRAMARFACIL.COM(s.f.) cómo controlar los tiempos con un RTC
<https://programarfacil.com>

Chacon.R(2015) RTC DS1307 I2C+ Arduino + Display LCD 16x2
<https://hetpro-store.com>

Fernandez.O(2016) Arduino rtc ds1307
<http://codigoelectronica.com>

ELECTRONICOSCALDAS(2004) QC2004A
<http://www.electronicoscaldas.com>

NAYLAMPMECHATRONICS.(2016) Tutorial Arduino y memoria SD y micro SD.
<https://naylampmechatronics.com>

NAYLAMPMECHATRONICS.(2016) Módulo SD Card
<https://naylampmechatronics.com>

DINASTIATECNOLOGIA(s.f.) Modulo SD Card Lectura Escritura Arduino
<http://dinastiatecnologica.com>

PROMETEC.(s.f.) Teclados Matriarcales
<https://www.prometec.net>

Llamas.L(2016) Usar un teclado matricial con Arduino
<https://www.luisllamas.es>

ELECTRONICOSCALDAS.(s.f.) 4x4 Matrix Membrane Keypad (#27899)
<http://www.electronicoscaldas.com>

ELECTRONICOSCALDAS.(s.f.) switches / teclados
<http://www.electronicoscaldas.com>

INGENIERIAELECTRONICA(2016) Utilizar el teclado matricial 4x4 con Arduino
<https://ingenieriaelectronica.org>

MARKERELECTRONICO(s.f.) Teclado 16 botones Matricial 4x4
<https://www.makeerelectronico.com>

MAXPHI(2017) 4x4 membrane keypad Interfacing with Arduino
<https://www.maxphi.com>

González A. (2013). Arduino Mega: Características, Capacidades y donde conseguirlo en Panamá. Obtenido de PANAMAHITEK: <http://panamahitek.com>

ARDUINODHTICS. (s.f.). ¿Qué es Arduino?. Obtenido de ARDUINO: TECNOLOGÍA PARATODOS :<https://arduinodhtics.weebly.com>

ADMIN_GAREMAR. (2015). Historia de Arduino. Obtenido de Industrino: <http://industrino.es>

Benitez R. (2012). Historia de Arduino y su nacimiento. Obtenido de botscience: <https://botscience.wordpress.com>

Ojeda L. (s.f.). ¿QUÉ ES ARDUINO? Obtenido de arduino.cl: <http://arduino.cl/que-es-arduino/>

Hacedores. (2014). ¿Cuántos tipos diferentes de Arduino hay? Obtenido de hacedores: <https://hacedores.com>

ARDUGEEK. (2014). Tipos de Arduinos. Obtenido de wordpress: <https://ardugeek.files.wordpress.com>

Tinchorton. (2014). Arduino Mega 2560. Obtenido de patagoniatec: <https://saber.patagoniatec.com/>

ICSP. (2016). Aprendiendo Arduino. Obtenido de Aprendiendo Arduino: <https://aprendiendoarduino.wordpress.com>

Torres H. (2014). Lector de huella digital con Arduino. Obtenido de HETPRO: <https://hetpro-store.com>

Layda A. (2018). Optical Fingerprint Sensor. Obtenido por Adafruit: <https://cdn-learn.adafruit.com>

prometec. (s.f.). MÓDULO GSM/GPRS: LLAMAR Y ENVIAR SMS. Obtenido de PROMETEC: <https://www.prometec.net>

Lara E.(2015). SIM900 GSM GPRS Shield con Arduino UNO. HETPRO: <https://hetpro-store.com>

Plaza G y Road Ch. (2011). SIM 900 SIM 900 –RS232 GSM/GPRSModem User Manual. Obtenido de rhydolabz: <http://www.rhydolabz.com>

HetPro. (2016). SIM900 GSM GPRS SHIELD CON ARDUINO UNO. Obtenido de instructables: <http://www.instructables.com>

Santos S. (2017). Guide to SIM900 GSM GPRS Shield with Arduino. Obtenido de randomnerdtutorials: <https://randomnerdtutorials.com>

GitHub. (2018). GPRS Shield and Cheap SIM800L PCB BOARDS. Obtenido de github: <https://github.com>

Modtronix. (2018). LCD 4x20, Blue (Negative), White Backlight. Obtenido de Modtronix: <http://modtronix.com>

GLOSARIO

ADK: (Accessory Development Kit) Kit de desarrollo de accesorios.

AFIS: (Automated Fingerprint Identification System) Sistema Automático de Identificación de Huellas Dactilares.

Bootloader: Gestor de arranque.

Caltech: (California Institute of Technology) Instituto de Tecnología de California

DIP: (Dual In-line Package) Paquete doble en línea.

DYI: (Do It Yourself) Hágalo usted mismo.

EEPROM: (Electrically Erasable Programmable Read-Only Memory) Memoria de solo lectura programable y borrable eléctricamente.

FBI: (Federal Bureau of Investigation) Buró Federal de Investigaciones.

GAO: (U.S. Government Accountability Office) Oficina de Responsabilidad Gubernamental de EE. UU.

GPRS: (General Packet Radio Service) Servicio General de Paquetes vía radio.

GPS: (Global Positioning System) Sistema de Posicionamiento Global.

GSM: (Global System for Mobile communications) Sistema Global para Comunicaciones Móviles.

ICSP: (In Chip Serial Programmer) Programación serial en el circuito.

MTI: (Modern University for technology & information) Universidad Moderna para Tecnología e Información.

PWM: (Pulse-Width Modulation) Modulación por Ancho de Pulsos.

RISC: (Reduced Instruction Set Computer) Computadoras con un conjunto de instrucciones reducido.

RTC: (Real-Time Clock) Reloj en Tiempo Real.

SIM: (Subscriber Identity Module), Módulo De Identificación De Suscripción.

SMS: (Short Message Service) Servicio de Mensajes Cortos o Servicio de Mensajes Simples.

SPI: (Serial Peripheral Interface) Interfaz Periférica Serial.

SRAM: (Static Random Access Memory) Memoria estática de acceso aleatorio O RAM Estática.

TCP: (Transmission Control Protocol) Protocolo de Control de Transmisión.

UART: (Universal Asynchronous Receiver-Transmitter) Transmisor-Receptor Asíncrono Universal.

UDP: (User Datagram Protocol) Protocolo de Datagramas de Usuario.

USB: (Universal Serial Bus) Bus Universal en Serie.

Wiring: Plataforma para prototipos electrónicos.

ANEXOS

PROGRAMACIÓN DE DISPOSITIVO BIOMETRICO.

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include <Keypad.h>
#include <SD.h>
#include <SPI.h>
#include "RTClib.h"

#include <sms.h>
// #include <call.h>
MSGSMS sms;
// CallGSM call;

RTC_DS1307 RTC;

LiquidCrystal lcd(36, 37, 38, 39, 40, 41);
int numdata;
boolean started=false;
char smsbuffer[160];
char n[20];

char sms_position;
char phone_number[20];
char sms_text[100];
String txt3;

const byte filas = 4; //Numero de filas del teclado
const byte columnas = 4; //Numero de columnas del teclado

//Defino una matriz 4x4 con la posicion de las filas y columnas
char matriz[filas][columnas] =
{
  { '1', '2', '3', 'A'},
  { '4', '5', '6', 'B'},
  { '7', '8', '9', 'C'},
  { '*', '0', '#', 'D'},
};

byte pinesFilas[filas] = {22, 23, 24, 25}; //Pines donde van conectadas las filas del teclado
byte pinesColumnas[columnas] = {26, 27, 28, 29}; //Pines donde van conectadas las columnas del teclado

//Inicializo el teclado con el numero de filas, columnas, los pines del Arduino utilizados y la matriz
Keypad teclado = Keypad( makeKeymap(matriz), pinesFilas, pinesColumnas, filas, columnas);

//Variables para el control de la posicion del cursor
int posX = 0;
int posY = 0;
int stk=0;
//Variales para el control de las teclas
char presionando;
```

```

int veces_presionada = 0;
int mayuscula_activada = 0;
int mostrar_cursorPantalla = 1;
int contador = 0;

//Variables para el control del contador, ya que no se usa ningun delay() en este codigo
unsigned long tiempo_anterior = 0;
int periodo = 500; //500 milisegundos

int getFingerprintIDez();
int bio=0;

/////////////////////////////////////////hora para envio de sms
int hr=2;
int mn=30;

// pin #2 is IN from sensor (GREEN wire)
// pin #3 is OUT from arduino (WHITE wire)
//SoftwareSerial mySerial(2, 3);
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);

// On Leonardo/Micro or others with hardware serial, use those! #0 is green wire, #1 is white
//Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);

uint8_t id;
uint8_t getFingerprintEnroll();

String string1;
char car;

String nombre;

String nombre1="NN AA";//INGRESAR NOMBRE ENTRE ""
String nombre2=" NN AA ";
String nombre3=" NN AA ";
String nombre4=" NN AA ";
String nombre5=" NN AA ";
String nombre6=" NN AA ";
String nombre7=" NN AA ";
String nombre8=" NN AA ";
String nombre9=" NN AA ";
String nombre10=" NN AA ";
String nombre11=" NN AA ";

char msg1[]="RECUERDA QUE TU HORA DE ENTRADA ES 9H50"; //mensaje que se envia a todos los usuarios//

char num1[13]="0000000000";// INGRESAR NUMERO ENTRE ""
char num2[13]=" 0000000000";//
char num3[13]=" 0000000000";//
char num4[13]=" 0000000000";//
char num5[13]=" 0000000000";//
char num6[13]=" 0000000000";//
char num7[13]=" 0000000000";//
char num8[13]=" 0000000000";//
char num9[13]=" 0000000000";//
char num10[13]=" 0000000000";//
char num11[13]=" 0000000000";//

char clave1[6]="00000";// INGRESAR COD EMPLEADO ENTRE ""
char clave2[6]="00000";//
char clave3[6]="00000";//
char clave4[6]="00000";//
char clave5[6]="00000";//
char clave6[6]="00000";//
char clave7[6]="00000";//

```

```

char clave8[6]=" 00000";//
char clave9[6]=" 00000";//
char clave10[6]=" 000001";//
char clave11[6]=" 00000";//

char clavesist[6]="0000";//

int st1=0;// estado 0 entra 1 sale
int st2=0;
int st3=0;
int st4=0;
int st5=0;
int st6=0;
int st7=0;
int st8=0;
int st9=0;
int st10=0;
int st11=0;
int stsd=0;

int en1=1;// habilitacion
int en2=1;
//int en3=1;
int en4=1;
int en5=1;
int en6=1;
int en7=1;
int en8=1;
int en9=1;
int en10=1;
int en11=1;

const int chipSelect=53; // AD t4

int st=0;
int t1=0;
int t2=0;
int t3=0;
int t4=0;
int c=0;

int ledPin=13;
void setup()
{
  lcd.begin(20, 4);
  lcd.clear();
  lcd.setCursor(0, 0);
  pinMode(ledPin, OUTPUT);
  lcd.print("INICIANDO...");
  while (!Serial); // For Yun/Leo/Micro/Zero/...
  pinMode(12,OUTPUT);
  Serial.begin(9600);
  Serial.println("INICIANDO");

  // set the data rate for the sensor serial port
  finger.begin(57600);

  if (finger.verifyPassword()) {
    Serial.println("SENSOR DE HUELLA INICIADO");
    digitalWrite(12,HIGH);
  } else {
    Serial.println("SENSOR DE HUELLA NOOOO INICIADO :(");
    digitalWrite(12,LOW);
    while (1);
  }
}

if (gsm.begin(9600))

```

```

    {
        Serial.println("GSM LISTO");
        started=true;
    }
    else {
        Serial.println("GSM ERROR");
    }
}

Wire.begin();
RTC.begin();
RTC.adjust(DateTime(__DATE__, __TIME__)); // DESHABILITO // PARA ACTUALIZAR HORA

if (! RTC.isrunning()) {
    Serial.println("RTC ERROR");
    // following line sets the RTC to the date & time this sketch was compiled
    RTC.adjust(DateTime(__DATE__, __TIME__));
}
Serial.println("MODULO RTC LISTO");

if (!SD.begin(chipSelect)) {
    Serial.println("MODULO SD ERROR");
    // don't do anything more:
    return;
}
Serial.println("MODULO SD INICIALIZADO");
delay(1000);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print(" DIGITE CLAVE");
}

uint8_t readnumber(void) {
    uint8_t num = 0;
    boolean validnum = false;
    while (1) {
        while (! Serial.available());
        char c = Serial.read();
        if (isdigit(c)) {
            num *= 10; // ANTES 10
            num += c - '0';
            validnum = true;
        } else if (validnum) {
            return num;
        }
    }
}

void loop() // run over and over again
{
    st=0;
    ingreso();// recibe las tecla presionadas
    clave();// compara claves
    delay(2000);
    if (st>0){
        leerfinger();
    }
    delay(50);
}

void leerfinger(){

```

```

    lcd.clear();

    lcd.setCursor(0, 0);
    lcd.print("PONGA SU HUELLA");
    delay(2000);
    int contime=0;
    bio=90;
    getFingerprintIDez();
    while(bio==90){
        getFingerprintIDez();
        digitalWrite(ledPin, HIGH);
        if(bio==1){
            nombre=nombre1;
            lcd.clear();
            lcd.print(nombre);
            stsd=st1;
            grabarsd();
            if(st1==0){st1=1;}else{st1=0;}
            delay(2000);
        }

        if(bio==2){
            nombre=nombre2;
            lcd.clear();
            lcd.print(nombre);
            stsd=st2;
            grabarsd();
            if(st2==0){st2=1;}else{st2=0;}
            delay(2000);
        }

        if(bio==3){
            nombre=nombre3;
            lcd.clear();
            lcd.print(nombre);
            stsd=st3;
            grabarsd();
            if(st3==0){st3=1;}else{st3=0;}
            delay(2000);
        }

        if(bio==4){
            nombre=nombre4;
            lcd.clear();
            lcd.print(nombre);
            stsd=st4;
            grabarsd();
            if(st4==0){st4=1;}else{st4=0;}
            delay(2000);
        }

        if(bio==5){
            nombre=nombre5;
            lcd.clear();
            lcd.print(nombre);
            stsd=st5;
            grabarsd();
            if(st5==0){st5=1;}else{st5=0;}
            delay(2000);
        }

        if(bio==6){
            nombre=nombre6;
            lcd.clear();
            lcd.print(nombre);
            stsd=st6;
            grabarsd();
        }
    }

```



```
if(st6==0){st6=1;}else{st6=0;}
delay(2000);
}
```

```
if(bio==7){
nombre=nombre7;
lcd.clear();
lcd.print(nombre);
stsd=st7;
grabarsd();
if(st7==0){st7=1;}else{st7=0;}
delay(2000);
}
```

```
if(bio==8){
nombre=nombre8;
lcd.clear();
lcd.print(nombre);
stsd=st8;
grabarsd();
if(st8==0){st8=1;}else{st8=0;}
delay(2000);
}
```

```
if(bio==9){
nombre=nombre9;
lcd.clear();
lcd.print(nombre);
stsd=st9;
grabarsd();
if(st9==0){st9=1;}else{st9=0;}
delay(2000);
}
```

```
if(bio==10){
nombre=nombre10;
lcd.clear();
lcd.print(nombre);
stsd=st1;
grabarsd();
if(st10==0){st1=10;}else{st10=0;}
delay(2000);
}
```

```
if(bio==11){
nombre=nombre11;
lcd.clear();
lcd.print(nombre);
stsd=st11;
grabarsd();
if(st11==0){st11=1;}else{st11=0;}
delay(2000);
}
```

```
if(bio==99){
lcd.clear();
lcd.print("ERROR DE");
lcd.setCursor(0,1);
lcd.print("AUTENTICACION");
delay(4000);
st=0;
}
delay(50);
```

```

Serial.print("bio");Serial.println(bio);
} //while
digitalWrite(ledPin, LOW);

}

```

```

void ingreso(){
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("DIGITE CLAVE");
  delay(2000);

  char customKey='A';
  c=0;
  digitalWrite(ledPin, HIGH);
  delay(100);
  digitalWrite(ledPin, LOW);

  while(customKey!='#'){
    customKey = teclado.getKey();

  while(c<5){

    customKey = teclado.getKey();
    checkhora();
    //Serial.println(customKey);
    DateTime now = RTC.now();
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(now.year(), DEC);
    lcd.print('/');
    lcd.print(now.month(), DEC);
    lcd.print('/');
    lcd.print(now.day(), DEC);

    lcd.setCursor(0, 1);
    lcd.print(now.hour(), DEC);
    lcd.print(':');
    lcd.print(now.minute(), DEC);
    lcd.print(':');
    lcd.print(now.second(), DEC);

    lcd.setCursor(0, 2);
    lcd.print("BIENVENIDOS A CNT EP");
    lcd.setCursor(0, 3);
    lcd.print(" CIS QUICENTRO SUR");

  delay(100);
  if (customKey!=NO_KEY){
    if(customKey=='A'){menuborrar();}else{c++;}
    if(c==1){
      clavesist[0]=customKey;digitalWrite(ledPin, HIGH);delay(100);digitalWrite(ledPin, LOW);delay(100);
      lcd.setCursor(0, 1);
      lcd.print(" *      ");
    }
    if(c==2){
      clavesist[1]=customKey;digitalWrite(ledPin, HIGH);delay(100);digitalWrite(ledPin, LOW);delay(100);
      lcd.setCursor(2, 1);
      lcd.print("**");
    }
    if(c==3){
      clavesist[2]=customKey;digitalWrite(ledPin, HIGH);delay(100);digitalWrite(ledPin, LOW);delay(100);

```



```

lcd.setCursor(0, 1);
lcd.print("ELIMINADO");
}

if(customKey2=='4'){
  en4=0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(nombre4);
  lcd.setCursor(0, 1);
  lcd.print("ELIMINADO");
}

if(customKey2=='5'){
  en5=0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(nombre5);
  lcd.setCursor(0, 1);
  lcd.print("ELIMINADO");
}

if(customKey2=='6'){
  en6=0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(nombre6);
  lcd.setCursor(0, 1);
  lcd.print("ELIMINADO");
}

if(customKey2=='7'){
  en7=0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(nombre7);
  lcd.setCursor(0, 1);
  lcd.print("ELIMINADO");
}

if(customKey2=='8'){
  en8=0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(nombre8);
  lcd.setCursor(0, 1);
  lcd.print("ELIMINADO");
}

if(customKey2=='9'){
  en9=0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(nombre9);
  lcd.setCursor(0, 1);
  lcd.print("ELIMINADO");
}

if(customKey2=='10'){
  en10=0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(nombre10);
  lcd.setCursor(0, 1);
  lcd.print("ELIMINADO");
}

if(customKey2=='11'){
  en10=0;
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print(nombre11);
  lcd.setCursor(0, 1);

```

```

    lcd.print("ELIMINADO");
}

}

}

void checkhora(){
    DateTime now = RTC.now();
    int mact=now.minute();
    int hact=now.hour();
    //Serial.println(hact);
    //Serial.println(mact);

    if((hr==now.hour())&&(mn==now.minute())){
        Serial.println("ENVIANDO MSG");
        lcd.clear();
        lcd.setCursor(0, 0);
        lcd.print("ENVIANDO MSJ");
        lcd.setCursor(0, 1);
        lcd.print("---> GSM");
        //sendm1(1); // PARA NO ENVIAR SMS A N REGISTRO COLOCAR //sendm
        sendm2();
        //sendm3();
        //sendm4();
        //sendm5();
        //sendm6();
        //sendm7();
        //sendm8();
        //sendm9();
        //sendm10();
        //sendm11();
        delay(55000);
    }

}

void clave(){
    lcd.clear();
    lcd.setCursor(0, 0);
    Serial.println(clavesist);
    Serial.println(clave1);

    if((clavesist[0]==clave1[0])&&(clavesist[1]==clave1[1])&&(clavesist[2]==clave1[2])&&(clavesist[3]==clave1[3])&&(clavesist[4]==clave1[4])&&(en1==1)){
        lcd.print("CLAVE CORRECTA");
        st=1;
    }

    if((clavesist[0]==clave2[0])&&(clavesist[1]==clave2[1])&&(clavesist[2]==clave2[2])&&(clavesist[3]==clave2[3])&&(clavesist[4]==clave2[4])&&(en2==1)){
        lcd.print("CLAVE CORRECTA");
        st=1;
    }

    if((clavesist[0]==clave3[0])&&(clavesist[1]==clave3[1])&&(clavesist[2]==clave3[2])&&(clavesist[3]==clave3[3])&&(clavesist[4]==clave3[4])&&(en3==1)){
        lcd.print("CLAVE CORRECTA");
        st=1;
    }

    if((clavesist[0]==clave4[0])&&(clavesist[1]==clave4[1])&&(clavesist[2]==clave4[2])&&(clavesist[3]==clave4[3])&&(clavesist[4]==clave4[4])&&(en4==1)){
        lcd.print("CLAVE CORRECTA");
    }
}

```

```

    st=1;
}

if((clavesist[0]==clave5[0])&&(clavesist[1]==clave5[1])&&(clavesist[2]==clave5[2])&&(clavesist[3]==clave5[3])
&&(clavesist[4]==clave5[4])&&(en5==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
}

if((clavesist[0]==clave6[0])&&(clavesist[1]==clave6[1])&&(clavesist[2]==clave6[2])&&(clavesist[3]==clave6[3])
&&(clavesist[4]==clave6[4])&&(en6==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
}

if((clavesist[0]==clave7[0])&&(clavesist[1]==clave7[1])&&(clavesist[2]==clave7[2])&&(clavesist[3]==clave7[3])
&&(clavesist[4]==clave7[4])&&(en7==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
}

if((clavesist[0]==clave8[0])&&(clavesist[1]==clave8[1])&&(clavesist[2]==clave8[2])&&(clavesist[3]==clave8[3])
&&(clavesist[4]==clave8[4])&&(en8==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
}

if((clavesist[0]==clave9[0])&&(clavesist[1]==clave9[1])&&(clavesist[2]==clave9[2])&&(clavesist[3]==clave9[3])
&&(clavesist[4]==clave9[4])&&(en9==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
}

if((clavesist[0]==clave10[0])&&(clavesist[1]==clave10[1])&&(clavesist[2]==clave10[2])&&(clavesist[3]==clave10[3])
&&(clavesist[4]==clave10[4])&&(en10==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
}

if((clavesist[0]==clave11[0])&&(clavesist[1]==clave11[1])&&(clavesist[2]==clave11[2])&&(clavesist[3]==clave11[3])
&&(clavesist[4]==clave11[4])&&(en11==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
}
delay(200);
if(st==0){
    //digitalWrite(beep, HIGH);delay(100);
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("CLAVE INCORRECTA");
    delay(2000);
    //digitalWrite(beep, LOW);
}
}
}

```

```

void grabarsd(){
    DateTime now = RTC.now();
    delay(10);
    // see if the card is present and can be initialized:
    SD.begin(chipSelect);
    delay(10);
    String datastring = "";
    datastring+=nombre;
    datastring+=String(' ');
}

```

```

if(stsd==0){datastring+="INGRESO ";}
if(stsd==1){datastring+="SALIDA ";}
datastring+=String(now.year(), DEC);
datastring+=String("/");
datastring+=String(now.month(), DEC);
datastring+=String("/");
datastring+=String(now.day(), DEC);
datastring+=String(" ");
datastring+=String(now.hour(), DEC);
datastring+=String(":");
datastring+=String(now.minute(), DEC);
datastring+=String(":");
datastring+=String(now.second(), DEC);

Serial.println(datastring);
delay(10);
File dataFile = SD.open("archivo.txt", FILE_WRITE);
delay(10);

// if the file is available, write to it:
if (dataFile) {
  dataFile.println(datastring);
  delay(10);

  dataFile.close();
  lcd.setCursor(0,1);
  lcd.print("EVENTO REGISTRADO");
  Serial.println("GRABACION EXITOSA");
  delay(1000);
}
// if the file isn't open, pop up an error:
else {
  lcd.setCursor(0,1);
  lcd.print("EVENTO NO REGISTRADO");
  Serial.println("error opening ARCHIVO");
}
}

```

```

void enrollar(){
  Serial.println("Ready to enroll a fingerprint! Please Type in the ID # you want to save this finger as...");
  id = readnumber();
  Serial.print("Enrolling ID #");
  Serial.println(id);

  while (! getFingerprintEnroll() );
}

```

```

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECEIVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      Serial.println("Unknown error");
  }
}

```

```

    return p;
}

// OK success!

p = finger.image2Tz();
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK converted!
p = finger.fingerFastSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);
bio=finger.fingerID;
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
    uint8_t p = finger.getImage();
    if (p != FINGERPRINT_OK) {return -1;}

    p = finger.image2Tz();
    if (p != FINGERPRINT_OK) {return -1;}

    p = finger.fingerFastSearch();
    if (p != FINGERPRINT_OK) {bio=99; return -1;}

    // found a match!
    Serial.print("Found ID #"); Serial.print(finger.fingerID);
    Serial.print(" with confidence of "); Serial.println(finger.confidence);
    bio=finger.fingerID;
    return finger.fingerID;
}

```



```

uint8_t getFingerprintEnroll() {

    int p = -1;
    Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.println(".");
                break;
            case FINGERPRINT_PACKETRECEIVEERR:
                Serial.println("Communication error");
                break;
            case FINGERPRINT_IMAGEFAIL:
                Serial.println("Imaging error");
                break;
            default:
                Serial.println("Unknown error");
                break;
        }
    }

    // OK success!

    p = finger.image2Tz(1);
    switch (p) {
        case FINGERPRINT_OK:
            Serial.println("Image converted");
            break;
        case FINGERPRINT_IMAGEMESS:
            Serial.println("Image too messy");
            return p;
        case FINGERPRINT_PACKETRECEIVEERR:
            Serial.println("Communication error");
            return p;
        case FINGERPRINT_FEATUREFAIL:
            Serial.println("Could not find fingerprint features");
            return p;
        case FINGERPRINT_INVALIDIMAGE:
            Serial.println("Could not find fingerprint features");
            return p;
        default:
            Serial.println("Unknown error");
            return p;
    }

    Serial.println("Remove finger");
    delay(2000);
    p = 0;
    while (p != FINGERPRINT_NOFINGER) {
        p = finger.getImage();
    }
    Serial.print("ID "); Serial.println(id);
    p = -1;
    Serial.println("Place same finger again");
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;
            case FINGERPRINT_NOFINGER:
                Serial.print(".");
                break;
        }
    }
}

```

```

case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    break;
case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
default:
    Serial.println("Unknown error");
    break;
}
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Image converted");
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    return p;
case FINGERPRINT_PACKETRECIEVEERR:
    Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:
    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();
if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
} else {

```

```

        Serial.println("Unknown error");
        return p;
    }
}

void sendm1(){
if (sms.SendSMS(num1, msg1))//
{
    Serial.println(msg1);
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm2(){
if (sms.SendSMS(num2, msg1))//
{
    Serial.println(msg1);
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm3(){
if (sms.SendSMS(num3, msg1))//
{
    Serial.println(msg1);
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm4(){
if (sms.SendSMS(num4, msg1))//
{
    Serial.println(msg1);
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm5(){
if (sms.SendSMS(num5, msg1))//
{
    Serial.println(msg1);
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm6(){
if (sms.SendSMS(num6, msg1))//
{
    Serial.println(msg1);
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm7(){
if (sms.SendSMS(num7, msg1))//
{
    Serial.println(msg1);
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm8(){
if (sms.SendSMS(num8, msg1))//
{
    Serial.println(msg1);
}else{

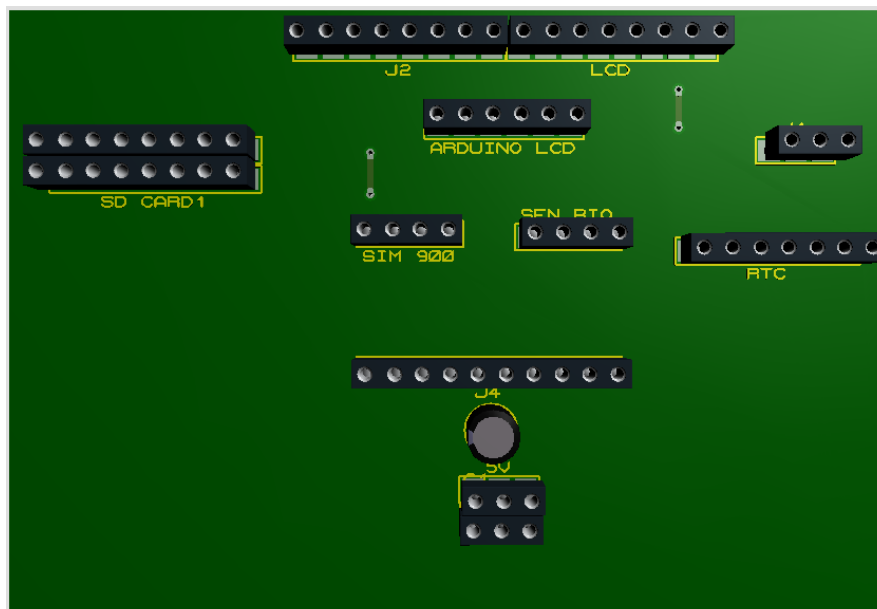
```

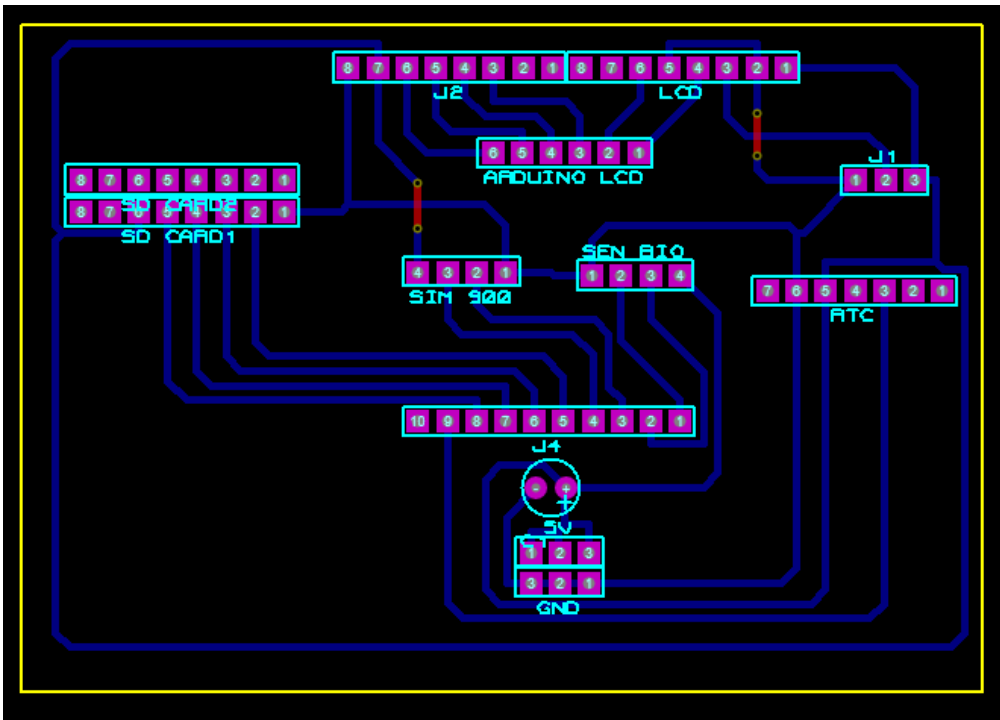
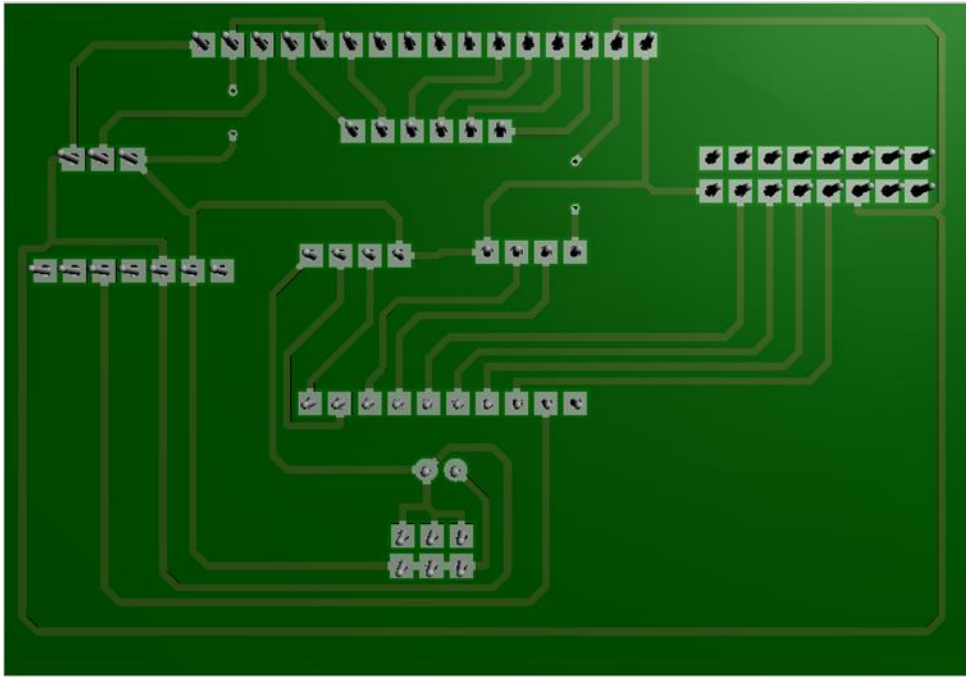
```

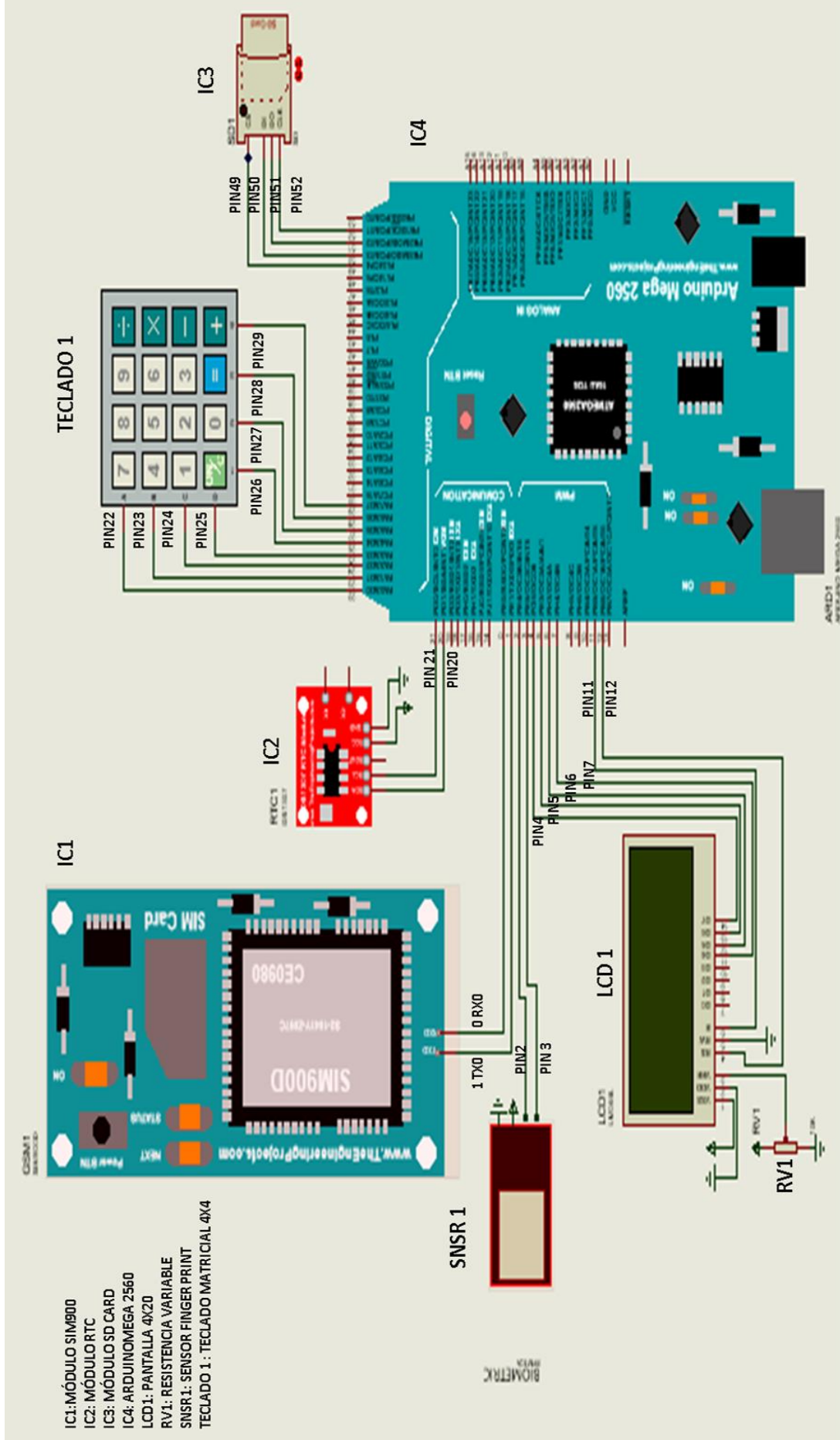
        //Serial.println("\nError sending SMS.");
    }
}
void sendm9(){
if (sms.SendSMS(num9, msg1)//
{
    Serial.println(msg1);
}
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm10(){
if (sms.SendSMS(num10, msg1)//
{
    Serial.println(msg1);
}
}else{
    //Serial.println("\nError sending SMS.");
}
}
void sendm11(){
if (sms.SendSMS(num11, msg1)//
{
    Serial.println(msg1);
}
}else{
    //Serial.println("\nError sending SMS.");
}
}
}
}

```

DISEÑO DE PLACA PCB.







- IC1: MÓDULO SIM900
- IC2: MÓDULO RTC
- IC3: MÓDULO SD CARD
- IC4: ARDUINOMEGA 2560
- LCD1: PANTALLA 4x20
- RV1: RESISTENCIA VARIABLE
- SNSR1: SENSOR FINGER PRINT
- TECLADO 1: TECLADO MATRICIAL 4x4

Diagrama electrónico biométrico inteligente.

Fuente: Elaborado por el Autor.

MANUAL DEL USUARIO

BIOMÉTRICO INTELIGENTE



Contenido

Instrucciones de seguridad	1
1.- Riesgos eléctricos:	1
2.- Riesgos personales:	1
3. Uso y trato cuidadoso del dispositivo electrónico:	1
Introducción:.....	2
Arranque de dispositivo:.....	4
Proceso de registro de ingresos y salidas.....	5
1.- Digital clave del empleado	5
2.- Registro y validación de huella dactilar.	7
3.- Transferencia de registros de dispositivo hacia PC.....	9

Instrucciones de seguridad

Advertencias de peligro generales para el uso del dispositivo biométrico.



Lea íntegramente estas advertencias de peligro e instrucciones.

1.- Riesgos eléctricos:

- Mantenga los cables de alimentación de energía en buenas condiciones, ya que un cable eléctrico sin su recubrimiento puede provocar descargas eléctricas, corto circuito e incluso incendios de las instalaciones.
- Garantice que la toma corriente sea de 110V, para evitar una sobrecarga eléctrica al dispositivo.
- Por ningún motivo permita que el dispositivo tenga contacto con líquidos, ya que esto puede causar daños al dispositivo, personas e instalaciones por causa de un incendio.

2.- Riesgos personales:

- Para poder realizar mantenimiento preventivo o correctivo del dispositivo es recomendable desconectar del suministro de energía eléctrica.

3. Uso y trato cuidadoso del dispositivo electrónico:

- Para tener un funcionamiento óptimo del dispositivo es de vital importancia realizar una limpieza periódica del sensor de huellas.
- No permita que los niños manipulen el dispositivo ya que su mal uso puede averiar lo.
- Es importante que lea las instrucciones de este manual para el uso adecuado de su dispositivo.

Introducción:

En este manual usted encontrara las indicaciones de uso de su dispositivo y la descripción de los elementos con los que cuenta su biométrico.

Su dispositivo

A continuación se describirá los componentes con los que cuenta su dispositivo.



Figura 1. Su Dispositivo vista de frontal.

Fuente: Elaborado por el Autor.

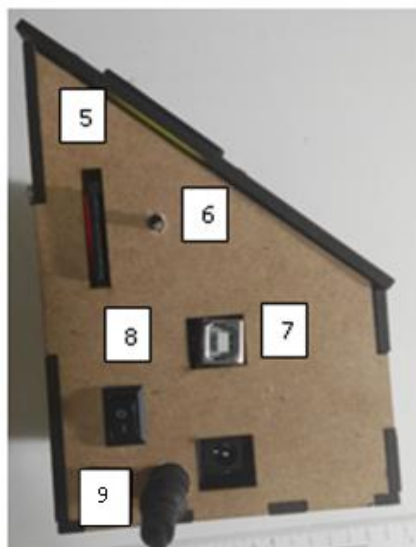


Figura 2. Su Dispositivo vista lado izquierdo.

Fuente: Elaborado por el Autor.

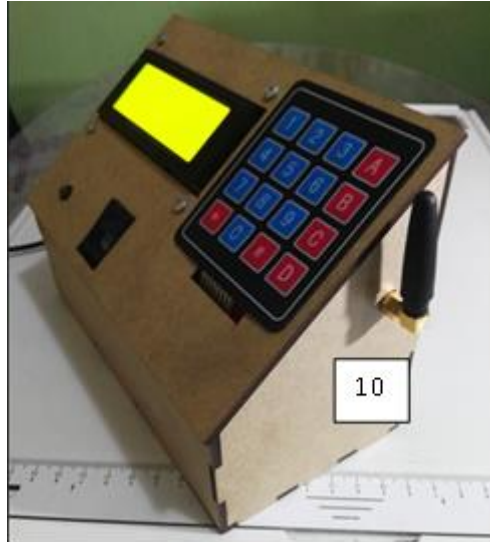


Figura 3. Su Dispositivo vista lado derecho.

Fuente: Elaborado por el Autor.

Tabla 1. Lista de elementos del dispositivo.

N°	ELEMENTOS
1	LCD 20X4
2	TECLADO 4X4
3	SENSOR BIOMETRICO
4	LED TESTIGO
5	RANURA SD CARD
6	BOTON INICIO SIM900
7	CONECTOR PROGRAMADOR
8	INTERUPTOR
9	PIN PATA FUENTE DE PODE
10	ANTENA SIM 900
11	FUENTE DE PODER

Arranque de dispositivo:

Para realizar el arranque del dispositivo es importante conectar a una red eléctrica de 110V.



Figura 4. Toma corriente de 110V

Fuente: Elaborado por el Autor.

Una vez que se encuentre conectado el dispositivo se debe encender, presionando su botón interruptor para que pase de estado 0 a estado 1.



Figura 5. Interruptor de dispositivo I/O.

Fuente: Elaborado por el Autor

A continuación se debe mantener presionado por 5 segundos el botón de arranque de la SIM 900 para tener señal para el envío de SMS.

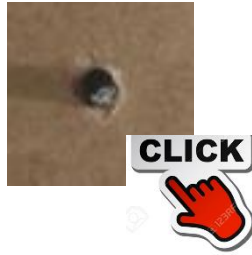


Figura 6. Botón de arranque SIM 900.

Fuente: Elaborado por el Autor.

Proceso de registro de ingresos y salidas.

Para realizar ingresos y salidas en el dispositivo se debe seguir los siguientes pasos.

1.- Digitar clave del empleado.- cada vez que se presione una tecla para ingresar el código de su clave, como se puede ver en la fotografía de la figura 7.



Figura 7. Ingreso de clave.

Fuente: Elaborado por el Autor.

Una vez que se haya ingresado la clave el sistema solicita presionar la tecla # para realizar la validación de su clave.

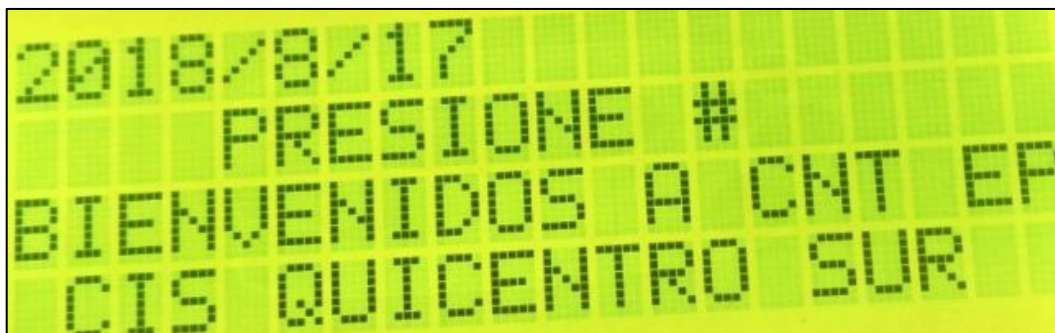


Figura 8. Presione # para validar huella.

Fuente: Elaborado por el Autor.

Si la clave es correcta le solicitara continuar con el siguiente paso, que es el registro de la huella dactilar.



Figura 9. Solicitud de ingreso de huella.

Fuente: Elaborado por el Autor.

Caso contrario en la pantalla le mostrara un mensaje de "CLAVE INCORRECTA" y le solicitara que ingrese su clave una vez más.

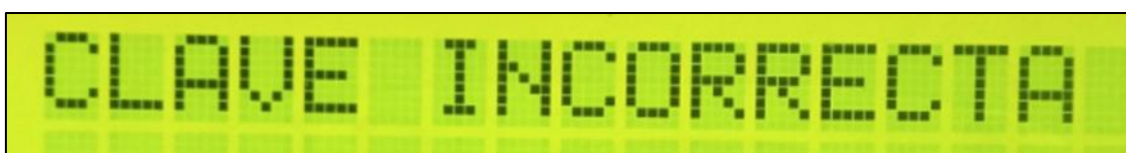


Figura 10. Clave incorrecta.

Fuente: Elaborado por el Autor.

2.- Registro y validación de huella dactilar.- luego de haber ingresado la clave satisfactoria mente, se debe procede al registro y validacion de la huella dactilar para crear el registro de entrada o salida del usuario.

Para realizar el registro hay que esperar que en la palla se muestre el mensaje que se ve en la imagen de la figura 9, y que se encienda el LED testigo para poder colocar el dedo de la huella dactilar que se encuentre registrada y se realizara la validación en el sistema.



Figura 11. Registro de huella dactilar.

Fuente: Elaborado por el Autor.

Luego de haber registrado clave y huella el sistema procederá a validar la información del usuario y mostrará un mensaje de evento registrado o caso contrario declara “ERROR DE AUTENTICACIÓN” y solicitará que ingrese la clave nuevamente.



Figura 12. Error de autenticación.

Fuente: Elaborado por el Autor.

Si la validación de clave y huella del usuario es correcta, registrara el evento y mostrará en la pantalla el nombre del usuario más la palabra “EVENTO REGISTRADO”



Figura13. Registró exitoso.

Fuente: Elaborado por el Autor.

Después de haber realizado el registro exitoso de la entrada o salida el biométrico está listo para realizar otro registro siguiendo los pasos antes descritos.



El dispositivo está diseñado para registrar entradas y salidas

por lo que es importante registrarse al momento de llegada y salida del almuerzo y jornada de trabajo, ya que si no realizan el sistema registrara según la secuencia que está programada.

3.- Transferencia de registros de dispositivo hacia PC.

Cuando se requiera pasar los registros guardados en el dispositivo se debe retirar la SD CARD de su ranura e introducir en la PC para pasar la información almacenada en formato txt. a un Excel.



Figura 14. Presionar y sustraer SD CARD.

Fuente: Elaborado por el Autor.

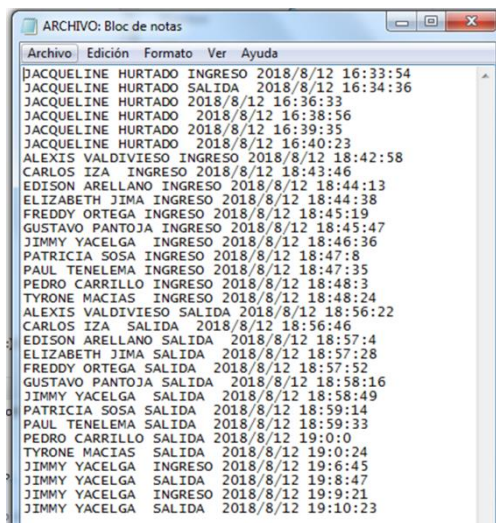


Figura 14. Registro TXT.

Fuente: Elaborado por el Autor.

Seleccione esta información y pegue en Excel, la información se desplegara en una sola fila.

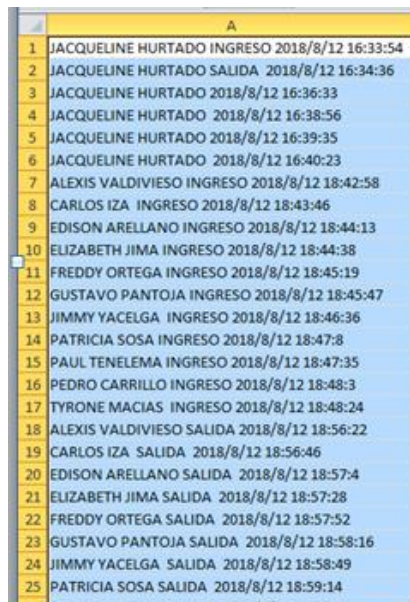


Figura 14. Información pasada a Excel.

Fuente: Elaborado por el Autor.

Para distribuir la información de mejor manera y generar reporte siga los siguientes pasos.

En la barra de herramientas seleccione la pestaña de DATOS, luego seleccione la pestaña de TEXTO EN COLUMNAS, como se indica en la imagen de la figura 15.

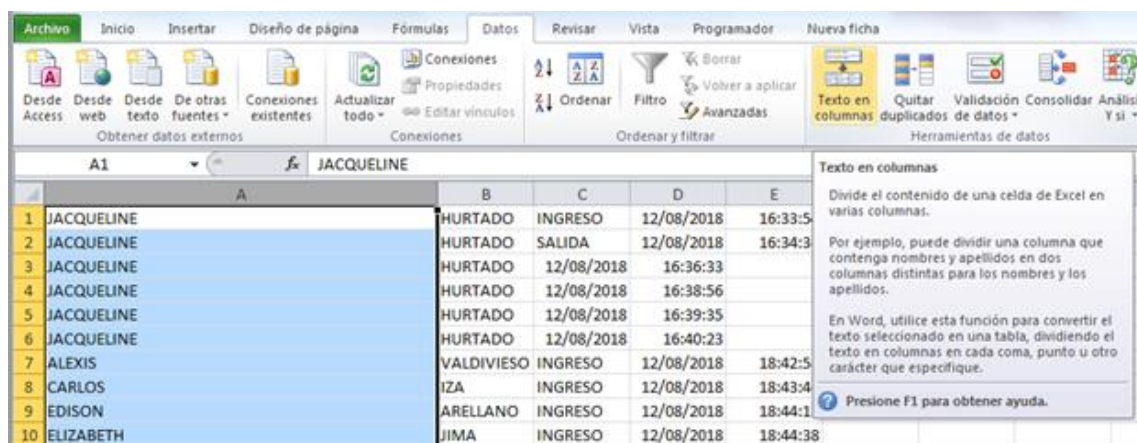


Figura 15. Pasos para distribuir de mejor manera la información en Excel.

Fuente: Elaborado por el Autor

Luego se desplegara una pantalla, donde se debe seguir los pasos siguientes descritos en las imágenes que se muestran a continuación.

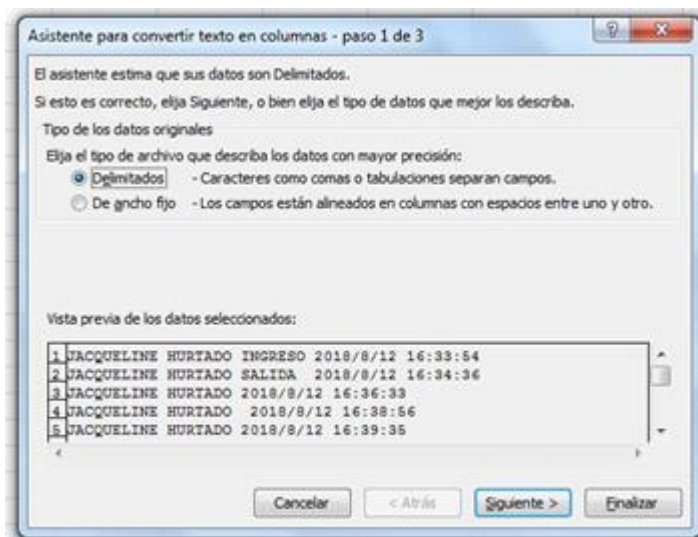


Figura 16. Pasos para distribuir de mejor manera la información en Excel.

Fuente: Elaborado por el Autor

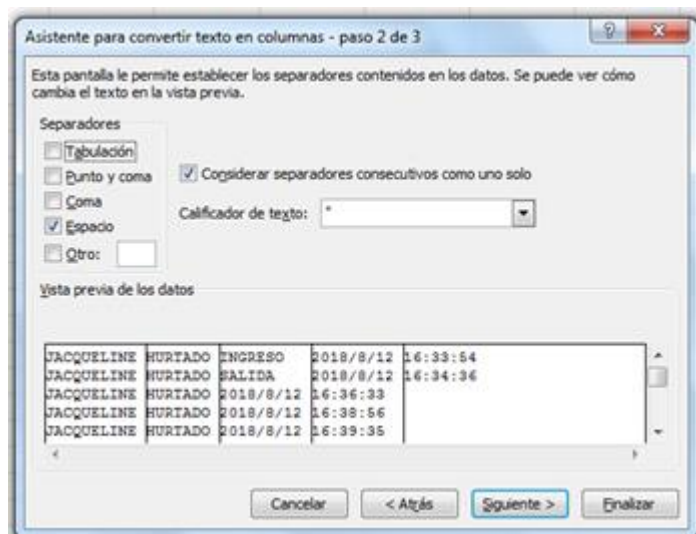


Figura 17. Pasos para distribuir de mejor manera la información en Excel.

Fuente: Elaborado por el Autor

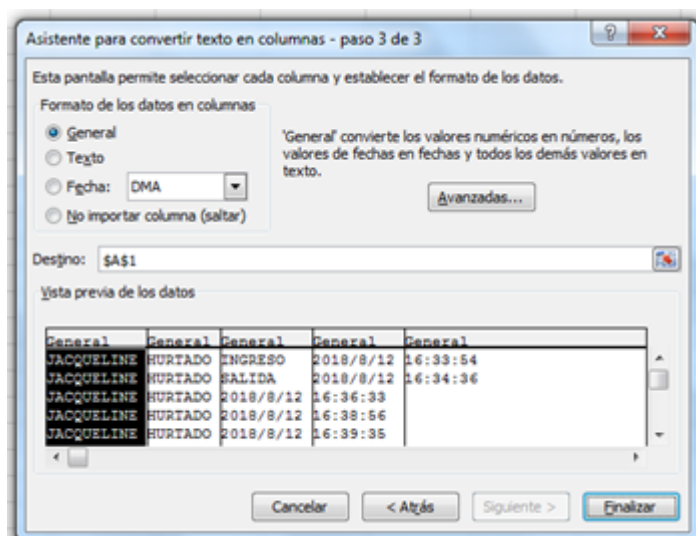


Figura 18. Pasos para distribuir de mejor manera la información en Excel.

Fuente: Elaborado por el Autor

Después de seguir todos los pasos descritos la información se presentara distribuida de la siguiente manera y lista para generar sus reportes e informes.

JACQUELINE	HURTADO	INGRESO	12/08/2018	16:33:54
JACQUELINE	HURTADO	SALIDA	12/08/2018	16:34:36
JACQUELINE	HURTADO	INGRESO	12/08/2018	16:36:33
JACQUELINE	HURTADO	SALIDA	12/08/2018	16:38:56
JACQUELINE	HURTADO	INGRESO	12/08/2018	16:39:35
JACQUELINE	HURTADO	SALIDA	12/08/2018	16:40:23
ALEXIS	VALDIVIESO	INGRESO	12/08/2018	18:42:58
CARLOS	IZA	INGRESO	12/08/2018	18:43:46
EDISON	ARELLANO	INGRESO	12/08/2018	18:44:13
ELIZABETH	JIMA	INGRESO	12/08/2018	18:44:38
FREDDY	ORTEGA	INGRESO	12/08/2018	18:45:19
GUSTAVO	PANTOJA	INGRESO	12/08/2018	18:45:47
JIMMY	YACELGA	INGRESO	12/08/2018	18:46:36
PATRICIA	SOSA	INGRESO	12/08/2018	18:47:08
PAUL	TENELEMA	INGRESO	12/08/2018	18:47:35
PEDRO	CARRILLO	INGRESO	12/08/2018	18:48:03
TYRONE	MACIAS	INGRESO	12/08/2018	18:48:24
ALEXIS	VALDIVIESO	SALIDA	12/08/2018	18:56:22
CARLOS	IZA	SALIDA	12/08/2018	18:56:46
EDISON	ARELLANO	SALIDA	12/08/2018	18:57:04

Figura 18. Distribución de información en Excel.

Fuente: Elaborado por el Autor

El dispositivo enviara mensajes a los usuarios que se encuentran registrados a la hora programada.



Figura 18. Mensaje que muestra a la hora programada del envío de SMS.

Fuente: Elaborado por el Autor

MANUAL TÉCNICO

BIOMÉTRICO INTELIGENTE



Contenido

Objetivo general:	1
Objetivos específicos:	1
Componentes del dispositivo	1
Arduino Mega 2560	1
Sensor finger print FPM10A	3
Modulo Tiny RTC DS 1307	4
Módulo GSM-GPRS-SMS-SIM900	5
Pantalla LCD 20X4:	6
Modulo SD CARD	7
Teclado 4X4.	8
Diseño de PCB para conexiones del dispositivo:	9
Conexiones del dispositivo.	11
Grabar y borrar usuarios y huellas:	15
Cambio de hora y mensaje:	19

Objetivo general:

Brindar la información necesaria al técnico encargado del mantenimiento y modificaciones en la programación del dispositivo.

Objetivos específicos:

- Proporcionar las características de cada uno de los componentes del dispositivo.
- Brindar información clara sobre las conexiones del dispositivo
- Proporcionar diseño de PCB del circuito.

Componentes del dispositivo.

El dispositivo biométrico está compuesto por la siguiente lista de materiales.

TABLA 1. Materiales.

N°	LISTA DE MATERIALES
1	Placa Arduino Mega2560.
2	Sensor Finger Print FPM10A
3	Módulo Tiny RTC DS 1307
4	Módulo GSM-GPRS-SMS-SIM900
5	Pantalla LCD 20X4:
6	Modulo SD CARD
7	Fuente de poder de 5V/2A

Fuente: Elaborado por el Autor.

A continuación se describirá información sobre cada uno de los componentes y sus características.

Arduino Mega 2560

Este modelo de Arduino es una placa que posee un microcontrolador Atmega 2560 y razón por la cual adquiere su nombre comercial.

Esta placa realiza su conversión de USB a Serial con el uso del microcontrolador Atmega8U2.

Arduino Mega2560 por sus características y facilidades que presta es utilizado en proyectos de mucha importancia, entre los más relevantes se encuentra Las Impresoras 3D y Domótica.



Figura 1. Placa Arduino Mega2560.

Fuente: Garcia. A (2013). Recuperado de <http://panamahitek.com>

TABLA 2. Especificaciones técnicas Arduino Mega 2560

ESPECIFICACIONES ARDUINO MEGA 2560
Microcontrolador: ATmega2560
Voltaje Operativo: 5V
Tensión de Entrada: 7-12V
Voltaje de Entrada(límites): 6-20V
Pines digitales de Entrada/Salida: 54 (de los cuales 14 proveen salida PWM)
Pines análogos de entrada: 16
Corriente DC por cada Pin Entrada/Salida: 40 mA
Corriente DC entregada en el Pin 3.3V: 50 mA
Memoria Flash: 256 KB (8KB usados por el bootloader)
SRAM: 8KB
EEPROM: 4KB
Clock Speed: 16 MHz

Fuente: Garcia. A (2013). Recuperado de <http://panamahitek.com>

Sensor Finger Print FPM10A

El lector de huella digital es un sensor biométrico que es ideal para realizar un sistema capaz de proteger lo que se requiera por medio del análisis de huellas dactilares en Arduino.

El sistema realiza procesamiento digital de imágenes de manera interna con un DSP (Procesador de Señales Digitales).



Figura 2. Lector de huella digital.

Fuente: Aliexpress (2018). Recuperado de www.aliexpress.com

Tabla 3. Especificaciones técnicas del dispositivo biométrico.

ESPECIFICACIONES TÉCNICAS DEL DISPOSITIVO
Voltaje de alimentación: 3.3V
Corriente de operación: 100mA – 150Ma
Tiempo de entrada de imágenes de huellas dactilares: <1 segundo
Tamaño de ventana: 14 x 18 mm
Interfaz: UART TTL
Modo de paridad de huella: 1:1 1:N
Baud Rate: 9600*N
N = 1 a 12 (Por defecto es 6)
Tiempo de adquisición menor a 1 segundo
5 Niveles de seguridad
Dimensión de la ventana: 14x18mm
Entorno de trabajo: -10°C a 40°C (Humedad relativa 40% a 85%)

Dimensiones: 5.5 x 2.1 x 2.0 cm
Peso: 22g

Fuente: Electronilab (2018). Recuperado de <https://electronilab.co>

Módulo Tiny RTC DS 1307

El módulo RTC DS 1307 es un reloj en tiempo real, esta placa electrónica funciona con un bajo consumo de energía.

Este dispositivo es de gran utilidad en el proyecto ya que facilita hora, fecha en tiempo real y facilita el registro de entrada y salida de los colaboradores.

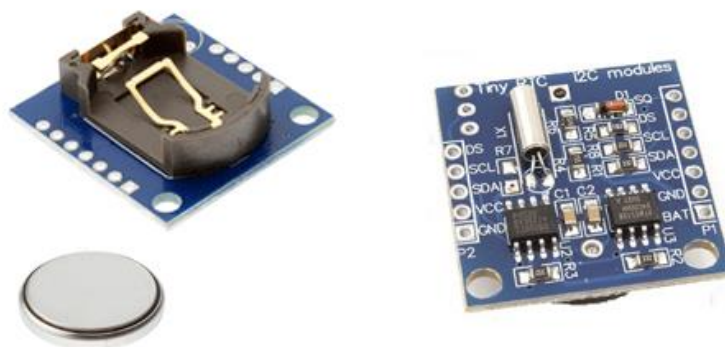


Figura 31. Módulo RTC DS 1307.

Fuente: Llamas. L (2016). Recuperado de <https://www.luisllamas.es>

Tabla 4. Especificaciones Técnicas RTC.

ESPECIFICACIONES TÉCNICAS
Reloj en tiempo real (RTC) Cuenta segundos, Minutos, horas, fecha del mes, mes, día de la semana, y año con año bisiesto Compensación Válido hasta 2100.
56-Byte, con respaldo de batería, no volátil (NV) de RAM para almacenamiento de datos
Interface Serie I2C.

Onda-Cuadrada programable de la señal de salida.
Detector Automático Fallo-Energía y Circuito Conmutación.
Consume menos de 500nA en la batería -- Modo de copia de seguridad con el oscilador funcionando.
Rango de temperatura Industrial Opcional: -40 ° C a +85 ° C
Disponible en 8-Pin Plástico DIP o SO
Reconocido Underwriters Laboratory (UL)

Fuente: hispavila (s.f.). Recuperado de <http://hispavila.com>

Módulo GSM-GPRS-SMS-SIM900

El GPRS SIMCOM SIM900 GSM Quad band GSM shield es una tarjeta ultra compacta de comunicación inalámbrica.

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

La tarjeta del módulo está basada en el módulo SIM900 GSM 4, el GPRS está configurado y controlada por vía UART usando comandos AT.

Por lo tanto sólo conecta la tarjeta al microcontrolador, Arduino, etc, y comienza a comunicarte a través de comandos AT.



Figura 4. Módulo GSM-GPRS-SMS-SIM900.

Fuente: ebay (2018). Recuperado de <https://www.ebay.in>

Tabla 5. Especificaciones técnica módulo GSM-GPRS-SMS-SIM900.

ESPECIFICACIONES TÉCNICAS
Quad-Band 850 / 900/ 1800 / 1900 MHz - funciona en todas las redes celulares del planeta.

GPRS multi-slot class 10/8
Estación móvil GPRS clase B
Cumple con GSM phase 2/2+
Clase 4 (2 W @ 850 / 900 MHz)
Clase 1 (1 W @ 1800 / 1900MHz)
Controlable via comandos AT estándar: GSM 07.07 & 07.05 Comandos mejorados: SIMCOM AT Commands.
Servicio Short Message Service(SMS) - para poder enviar pequeños paquetes de datos a través de la red celular.
Pila TCP/UDP incorporada - permite enviar datos a un servidor web. Ejem: Pachube, Fusion Tables
Incorpora un Real Time Clock - RTC. (Requiere pila)
Puerto serial configurable para comunicación con el shield.
Soporte para comunicación por softwareSerial (pines 6 y 7).
Jack para audífonos y micrófono (handsfree)
Bajo consumo (en modo sleep) - 1.5Ma
Jack para alimentación externa.
Soporta rangos temperatura - -40°C to +85 °C

**Fuente: botscience (2013). Recuperado de <http://botscience.net>
<http://botscience.net>**

Pantalla LCD 20X4:

Pantalla LCD que cuenta de 20x4 caracteres, esta LCD trabaja con tecnología STN, las dimensiones de la pantalla es de 9.8 cm x 6 cm de ancho, este módulo es utilizado en el proyecto para presentar la información que imprime el dispositivo.

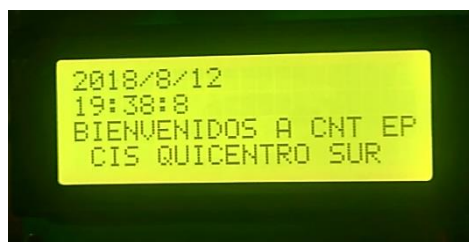


Figura 5. Pantalla LCD 20x4.

Fuente: Elaborado por el Autor

Tabla 6. Especificaciones técnicas LCD.

ESPECIFICACIONES TÉCNICAS
20 caracteres x 4 líneas

Caracteres de 5x8 puntos
Tamaño de caracteres: 4.74 mm x 2.94 mm
Puede mostrar letras, números, caracteres especiales, y hasta 8 caracteres creados por el usuario
Interface paralela. Puede operar en modo de 8 bits, o de 4 bits para ahorrar pines del microcontrolador
Posee controlador KS0066 o equivalente on-board (compatible Hitachi HD44780)
Voltaje de alimentación: 5 V
Tamaño total: 98 mm x 60 mm x 14 mm

Fuente: Caldas (s.f.) Recuperado de <https://www.electronicoscaldas.com>

Módulo SD CARD

Módulo diseñado para el almacenamiento de información por medio de una tarjeta SD y que después va a ser transferido a la PC para la consolidación de la información recogida por el biométrico.



Figura 6. Módulo SD CARD.

Fuente: naylamp (s.f.). Recuperado de <https://naylampmechatronics.com>

Tabla 7. Especificaciones técnicas módulo SD CARD.

ESPECIFICACIONES TÉCNICAS
Voltaje de Operación: 3.3V-5V
Interfaz: SPI
Cuenta con todos los pines SPI de la tarjeta SD: MOSI, MISO, SCK, CS

Fuente: <https://naylampmechatronics.com/modulos/35-modulo-sd-card.html>

Teclado 4X4.

Teclado cuenta con 4 filas distribuidas en (A, B, C, D) y 4 columnas (1, 2, 3,4), en total cuenta con 16 teclas independientes pero por estar conectado en una matriz de pulsadores solo utiliza 8 pines de la placa de Arduino.

Este teclado va ser utilizado en el proyecto para el ingreso de código al momento de realizar el registro de asistencias.

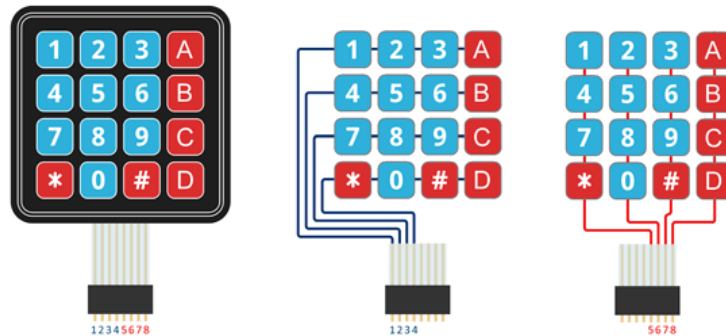


Figura 72. Teclado 4X4.

Fuente: Llamas. L (2016). Recuperado de <https://www.luisllamas.es>

Tabla 8. Especificaciones técnicas teclado.

ESPECIFICACIONES TÉCNICAS
16 botones con organización matricial (4 filas x 4 columnas)
Teclado tipo membrana
Mayor resistencia al agua y al polvo
Auto adhesivo en la parte de atrás
Tiempo de rebote (Bounce time): ≤ 5 ms
Máximo voltaje operativo: 24 V DC
Máxima corriente operativa: 30 Ma
Resistencia de aislamiento: 100 M Ω (@ 100 V)
Voltaje que soporta el dieléctrico: 250 VRMS (@ 60Hz, por 1 min)
Expectativa de vida: 1.000.000 de operaciones
Dimensiones del pad: 6.9 x 7.6 cm aprox.
Cable de cinta plana de 8.5 cm de largo aprox. (incluido el conector)
Conector tipo DuPont hembra de una fila y 8 contactos con separación estándar 0.1" (2.54mm)
Temperatura de operación: 0 a 50 °C

Fuente: Marker electrónico (2018). Recuperado de <https://www.makeelectronico.com>

Diseño de PCB para conexiones del dispositivo:

El diseño de la PCB fue creado en Proteus, continuación se evidenciara estos diseños en las siguientes figuras.

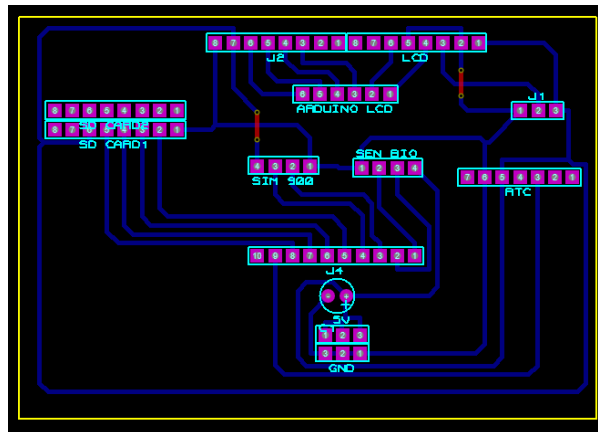


Figura 8. Diseño de conexiones en la PCB

Fuente: Elaborado por el Autor.

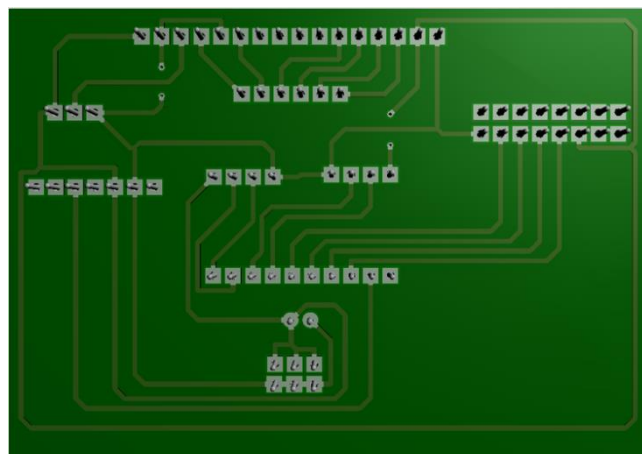


Figura 9. PCB 3D parte posterior.

Fuente: Elaborado por el Autor.

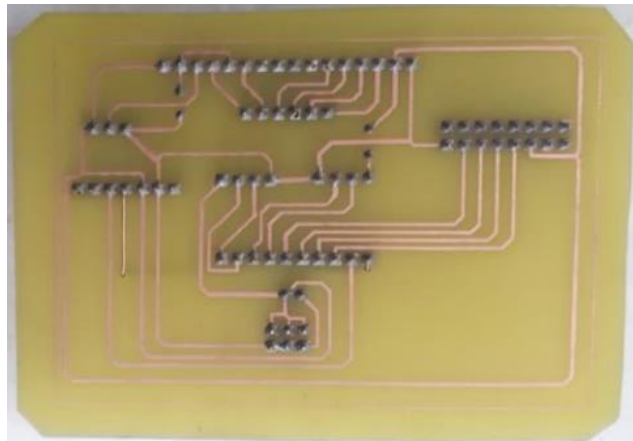


Figura 10. Circuito impreso en baquelita.

Fuente: Elaborado por el Autor.

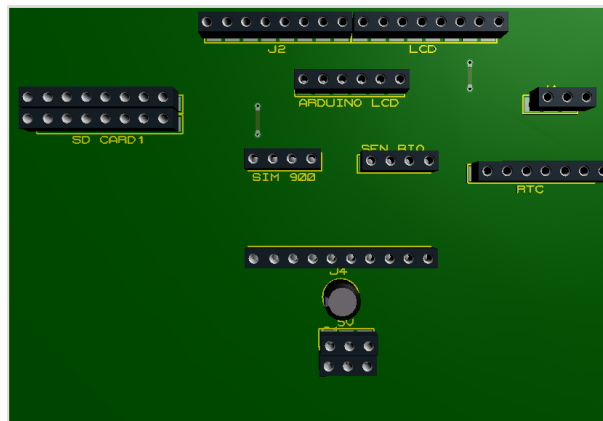


Figura 11. Simulación de regletas en 3D.

Fuente: Elaborado por el Autor.

Tabla 9. Detalle de conexión teclado 4x4 y Arduino.


TECLADO 4X4 DE LH A RH		
1	PLOMO	PIN 22 ARDUINO
2	BLANCO	PIN 23 ARDUINO
3	VERDE	PIN 24 ARDUINO
4	CAFÉ I	PIN 25 ARDUINO
5	VIOLETA I	PIN 26 ARDUINO
6	NARANJA	PIN 27 ARDUINO
7	CAFÉ II	PIN 28 ARDUINO
8	VIOLETA II	PIN 29 ARDUINO



Fuente: Elaborado por el Autor.

Tabla 10. Detalle de conexión SIM 900, Arduino y placa PCB.

SIM900 GMS GPRS				
1A	NA			
2A	NA			
3A	NA			
4A	AZUL	PIN4	PLACA	SIM900
5A	NA			
6A	NA			
1R	NA			
2R	NA			
3R	NA			
4R	NA			
5R	NA			
6R	NA			
7R	NA			
8R	VIOLETA	PIN11	ARDUINO	
9R	VERDE	PIN10	ARDUINO	
10R	NA			
11R	NA			
12R	NA			
13R	NA			
14R	NA			
15R	NA			
16R	NA			



Fuente: Elaborado por el Autor

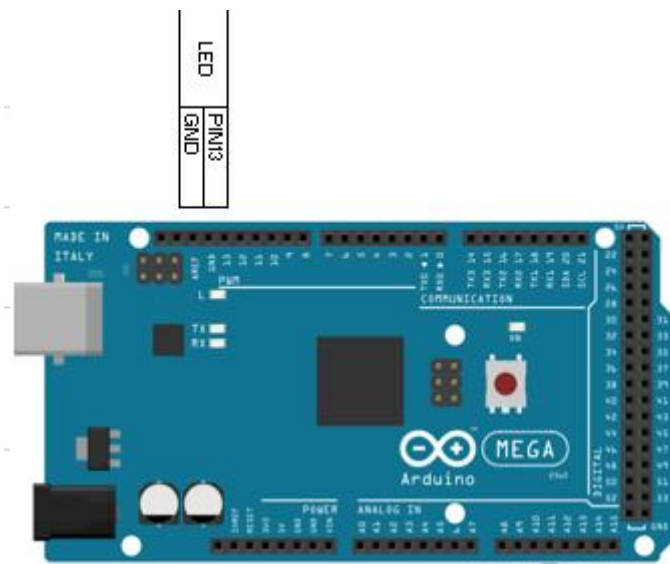

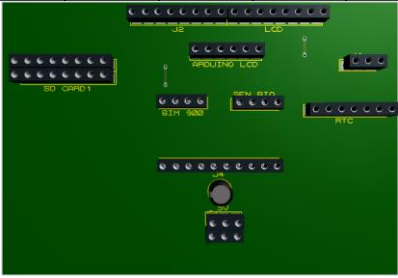



Figura 14. Conexión Arduino vs LED.

Fuente: Elaborado por el Autor

Tabla 11. Detalle de conexión LCD 20X4, Arduino, placa PCB, módulo RTC y módulo SD CARD.

PLACA PARA CONEXIONES BIOSMART																															
																															
PLACA PARA CONEXIONES BIOSMART																															
<table border="1"> <thead> <tr> <th colspan="7">PINES LCD</th> </tr> </thead> <tbody> <tr> <td>ARD LCD</td> <td>PIN1</td> <td>PIN2</td> <td>PIN3</td> <td>PIN4</td> <td>PIN5</td> <td>PIN6</td> </tr> <tr> <td>ARDUINO</td> <td>PIN41</td> <td>PIN40</td> <td>PIN39</td> <td>PIN38</td> <td>PIN37</td> <td>PIN36</td> </tr> </tbody> </table>											PINES LCD							ARD LCD	PIN1	PIN2	PIN3	PIN4	PIN5	PIN6	ARDUINO	PIN41	PIN40	PIN39	PIN38	PIN37	PIN36
PINES LCD																															
ARD LCD	PIN1	PIN2	PIN3	PIN4	PIN5	PIN6																									
ARDUINO	PIN41	PIN40	PIN39	PIN38	PIN37	PIN36																									
SD CARD	PIN1	PIN2	PIN3	PIN4	PIN5	PIN6	PIN7	PIN8																							
MODULO	NA	NA	5V	SDCS	MOSI	SCK	MISI	GND																							
																															
SIM900	PIN1	PIN2	PIN3	PIN4																											
MODULO	NA	NA	NA	4A																											
RTC	PIN1	PIN2	PIN3	PIN4	PIN5	PIN6	PIN7																								
MODULO	NA	GND	VCC	SDA	SCL	DS	NA																								
J4	PIN1	PIN2	PIN3	PIN4	PIN5	PIN6	PIN7	PIN8	PIN9	PIN10																					
ARDUINO	PIN20 SDA	PIN21 SCL	PIN53	PIN51	PIN52	PIN50	NA	NA	NA	NA																					
5V	PIN1	PIN2	PIN3																												
GND	PIN1	PIN2	PIN3																												
																															

Fuente: Elaborado por el Autor.

Tabla 12. Detalle de conexión de sensor finger print y Arduino.

SENSOR FINGER PRINT			
1	VERDE I	GND	
2	LIBRE	NA	
3	VIOLETA	5V	
4	VERDEII	RX1	PIN 19 ARDUINO
5	AZUL	TX1	PIN18 ARDUINO
6	PLOMO	GND	



Fuente: Elaborado por el Autor.

Grabar y borrar usuarios y huellas:

Para grabar y borrar usuarios se lo debe hacer por medio de software, y para lo cual se utiliza la programación creada para el control del dispositivo.

Hay que aumentar o reducir líneas de los siguientes tramos de programación.

```
String nombre;

String nombre1="naa";//INGRESAR NOMBRE ENTRE ""
String nombre2="naa ";
String nombre3="naa";
////////////////////////////////////
char num1[13]="n° celular";// INGRESAR NUMERO ENTRE ""
char num2[13]=" n° celular ";//
char num3[13]=" n° celular ";//
////////////////////////////////////
char clave1[6]="cod.");// INGRESAR COD EMPLEADO ENTRE ""
char clave2[6]="cod";//
char clave3[6]="cod";//
////////////////////////////////////

char clavesist[6]="00000";//

int st1=0;// estado 0 entra 1 sale
int st2=0;
int st3=0;

int en1=1;// habilitacion
```

```

int en2=1;
////////////////////////////////////
int en3=1;
  delay(2000);
  int contime=0;
  bio=90;
getFingerprintIDez();
while(bio==90){
  getFingerprintIDez();
  digitalWrite(ledPin, HIGH);
  if(bio==1){
    nombre=nombre1;
    lcd.clear();
    lcd.print(nombre);
    stsd=st1;
    grabarsd();
    if(st1==0){st1=1;}else{st1=0;}
    delay(2000);
  }
  //////////////////////////////////////

  if(customKey2=='1'){
    en1=0;
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print(nombre1);
    lcd.setCursor(0, 1);
    lcd.print("ELIMINADO");
  }

  //////////////////////////////////////

  sendm1(); // PARA NO ENVIAR SMS A N REGISTRO COLOCAR //sendm
  sendm2();
  sendm3();
  //////////////////////////////////////

  void clave(){
  lcd.clear();
  lcd.setCursor(0, 0);
  Serial.println(clavesist);
  Serial.println(clave1);

  if((clavesist[0]==clave1[0])&&(clavesist[1]==clave1[1])&&(clavesist[2]==clave1[2])&&(clavesist[3]==clave1[
3])&&(clavesist[4]==clave1[4])&&(en1==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
  }

  if((clavesist[0]==clave2[0])&&(clavesist[1]==clave2[1])&&(clavesist[2]==clave2[2])&&(clavesist[3]==clave2[3
]))&&(clavesist[4]==clave2[4])&&(en2==1)){
    lcd.print("CLAVE CORRECTA");
    st=1;
  }
  //////////////////////////////////////

}
}

void sendm1(){
if (sms.SendSMS(num1, msg1))//
{
  Serial.println(msg1);
}
}

```


Para grabar las huellas se debe compilar la programación enrolada, y a continuación con la ayuda del monitor serial grabar las huellas de los usuarios, en la pantalla se observa las indicaciones de enrolamiento

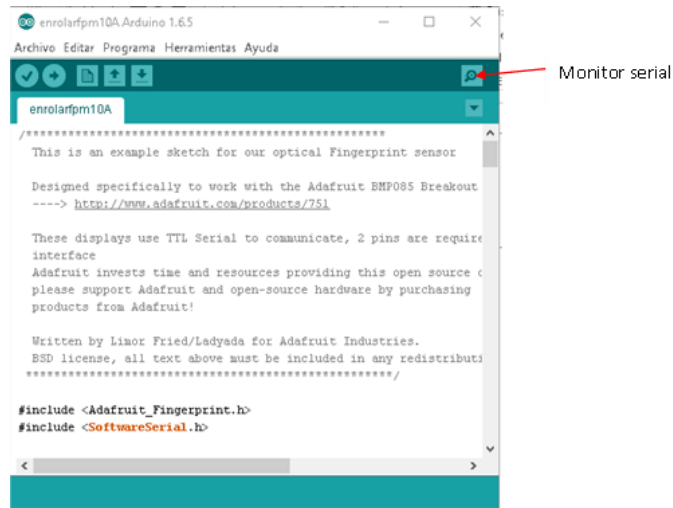



Figura 14. Programa para compilar

Fuente: Elaborado por el Autor.

En la pantalla de se debe digitar un del 1 al 99 más las tecla # y  enter, a paso seguido se debe colocar el dedo en el sensor de huellas por 1 segundo retirarlo y volver a colocar el dedo en el sensor por 2 segundos, se grabará la huella del usuario.

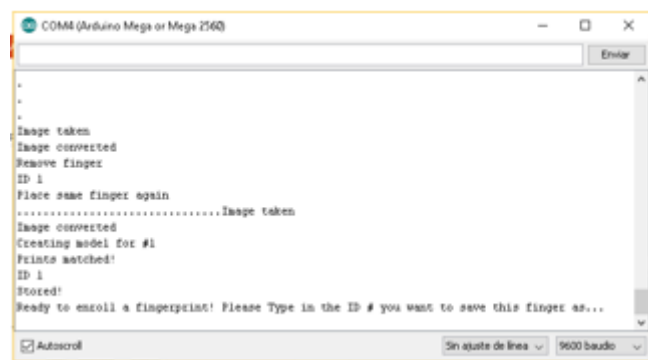


Figura 15. Enrolamiento con pantalla serial

Fuente: Elaborado por el Autor.

Para borrar se realiza las mismas actividades, pero utilizando el programa para borrar, se coloca el número de usuario a borrar más # y enter, para que la huella ya no se encuentre en la memoria del biométrico.

Cambio de hora y mensaje:

El cambio de hora de envío de SMS y mensaje que se envía se realiza su cambio con software en las siguientes líneas.

```

////////////////////////////////////hora para envio de sms
int hr=h;
int mn=m

////////////////////////////////////

char msg1[]="RECUERDA QUE TU HORA DE ENTRADA ES 9H50"; //mensaje que se envia a todos los
usuarios//

```

Tabla 13. Daños y soluciones.

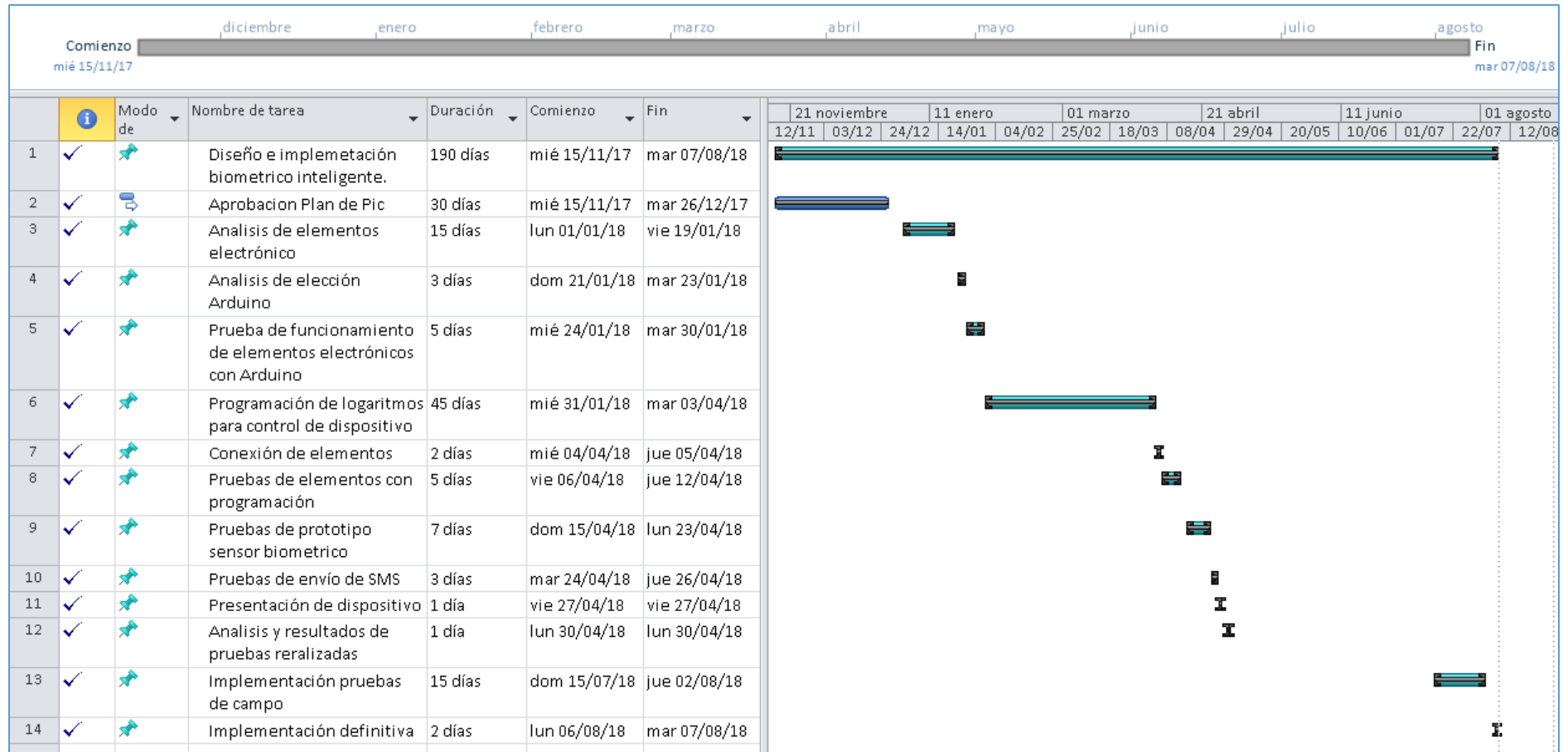
NOVEDAD	SOLUCION
Equipo no enciende	Verificar alimentación eléctrica
led no enciende	Verificar conexiones en tablas antes descritas del circuito.
Huella no se registra	Limpia sensor biométrico
Hora desactualizada	Cambie pila de RTC, cada 2 años

Fuente: Elaborado por el Autor.

En el caso de no poder solucionar el problema, llamar a su fabricante, Tnlgo. Carlos Iza
 Celular: 0991929877.

CRONOGRAMA DE A ACTIVIDADES.

PROYECTO: DISEÑO E IMPLEMENTACIÓN DE UN BIOMÉTRICO INTELIGENTE QUE ENVÍE SMS DE ALERTA A LOS EMPLEADOS ANTES DE SU HORA DE ENTRADA.



DATASHEET