



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA



**DISEÑO E IMPLEMENTACIÓN DE PROTOTIPO DE
UN SISTEMA BIOMÉTRICO PARA MEJORAR EL
CONTROL DE ASISTENCIA DEL PERSONAL
DOCENTE EN LA FACFYM.**

Tesis para optar el título profesional de Ingeniero Electrónico

PRESENTADA POR:

Bach. Díaz Collantes Juseph Andersson

Bach. Flores Soruluz Gyver Iván

Tesis presentada por:

Bach. Diaz Collantes Juseph Andersson

Bach. Flores Soraluz Gyver Iván

Tesis para optar por el título profesional de Ingeniero Electrónico

Aceptado por la Escuela Profesional de Ingeniería Electrónica:



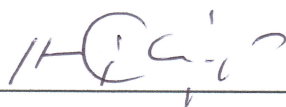
Ing. Manuel Javier Ramírez Castro

PRESIDENTE



Ing. Carlos Leonardo Oblitas Vera

VOCAL



Ing. Hugo Javier Chiclayo Padilla

SECRETARIO




Ing. Oscar Uchelly Romero Cortez

ASESOR



Bach. Juseph Andersson Diaz Collantes

AUTOR



Bach. Gyver Iván Flores Soraluz

AUTOR

DEDICATORIA

La presente investigación, la dedico a todas las personas que me han apoyado durante el desarrollo de la misma, en especial a mis padres cuya motivación por salir adelante, me han llevado a ser un profesional disciplinado, dedicado y creyente que todo se logra con esfuerzo y perseverancia.

JUSEPH DÍAZ

AGRADECIMIENTO

Agradezco a Dios, por darme salud, y fuerza de voluntad para lograr mis objetivos, agradezco a mi asesor de tesis, y los docentes de toda mi escuela profesional quienes, por sus conocimientos y guía, culminé el desarrollo de la tesis con éxito, consiguiendo así mi anhelado título profesional.

JUSEPH DÍAZ

DEDICATORIA

Dedico esta tesis a todas las personas que me apoyaron durante la realización de la misma, ya que todo esfuerzo y sacrificio que he realizado fue con la intención de tener un futuro digno convirtiéndome en un profesional exitoso y sobre todo dedicado con su vocación.

GYVER FLORES

AGRADECIMIENTO

En primer lugar, agradecer a Dios por darme salud y fuerzas para perseguir todos los objetivos que me he trazado, en segundo lugar, agradecer a mis padres por su infinita paciencia hacia mí, así como su incondicional apoyo a todas las decisiones que he tomado también agradecer a mis hermanos por siempre estar ahí para mí cuando los he necesitado dándonos consejos para salir adelante, también agradezco a mi asesor por siempre estar dispuesto a escucharnos y apoyarnos con el proyecto de tesis.

GYVER FLORES

RESUMEN

La Universidad Nacional Pedro Ruiz Gallo, cuenta entre sus facultades con la Facultad de Ciencias Físicas y Matemáticas, la cual usa fichas impresas para el registro de personal, tanto administrativo como docente desde sus inicios.

Este no cuenta con un sistema de control, que permita almacenar los datos de entrada y salida de personal en una base de datos para facilitar el proceso de computar los días de trabajo, inasistencias, faltas y tardanzas.

Es por ello que surge la necesidad de desarrollar dicho sistema para la Facultad de Ciencias Físicas y Matemáticas con el fin de automatizar el proceso de registro de asistencia.

El desarrollo del mencionado sistema se estimó como una investigación proyectiva, sostenida en un prototipo puesto a prueba en campo y documentada.

Así mismo para el modelado, se emplearon los conceptos de biometría, aplicados en microcontroladores, base de datos y desarrollo de aplicaciones basados en programación orientada a objetos.

Se sostuvo las pruebas en campo durante 30 días, durante este período, se corrigieron problemas de autenticación, de conectividad y diseño, no obstante, el objetivo principal se mantuvo, es decir la data de registro se almacenó y se pudo trabajar con ella elaborando cuadros de exportación listos para imprimir.

En cuanto a resultados, se demostró que el prototipo es en efecto una solución, que no genera colas, ni retardos, además que no cuenta con personal de supervisión, y no genera el cúmulo de papeles del día a día.

En conclusión, estos problemas de registro de personal son bien resueltos por sistemas de autenticación, y en esencia este, además evita la suplantación, en cuanto a temas de seguridad hablamos.

ABSTRACT

The state university Pedro Ruiz Gallo, counts among its faculties with the Physical and Mathematic Sciences School, which uses books for the staff registration, both administrative and teaching since its inception.

This does not have a control system, which allows storing the data of entry and exit workers in a database to facilitate the process of computing work days, absences, faults and delays.

That is why the need arises to develop such a system for the Physical and Mathematic Sciences School in order to improve the attendance registration process.

The development of the mentioned system was estimated as a projective investigation, sustained in a prototype put to test in the field and documented.

Also for modeling, the concepts of biometrics, applied in microcontrollers, database and application development based on object-oriented programming were used.

The tests were held in the field for thirty days, during this period, problems of authentication, connectivity and design were corrected, however, the main objective was maintained, that is, the registration data was stored and it was possible to work with it, generating tables of export ready to print.

In terms of results, it was demonstrated that the prototype is in effect a solution, which does not generate queues, or delays, besides that it does not have supervisory personnel, and does not generate the accumulation of day-to-day files.

In conclusion, these problems of personnel registration are well solved by authentication systems, and in essence this, also avoids the impersonation, if in terms of security issues we speak.

Tabla de contenido

| | |
|--|----------|
| INTRODUCCIÓN..... | 1 |
| CAPITULO I | 2 |
| 1. PLANTEAMIENTO DEL PROBLEMA CIENTÍFICO..... | 2 |
| 1.1 DESCRIPCIÓN DE LA REALIDAD PROBLEMÁTICA..... | 2 |
| 1.2 FORMULACIÓN DEL PROBLEMA CIENTÍFICO | 3 |
| 1.3 OBJETIVOS DE LA INVESTIGACIÓN..... | 3 |
| 1.3.1 <i>Objetivo General</i> | 3 |
| 1.3.2 <i>Objetivos Específicos</i> | 3 |
| 1.4 JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN..... | 4 |
| 1.5 LIMITACIONES DEL ESTUDIO | 4 |
| 1.6 VIABILIDAD..... | 5 |
| 1.6.1 <i>Factibilidad operativa</i> | 5 |
| 1.6.2 <i>Factibilidad económico-financiera</i> | 5 |
| 1.6.3 <i>Factibilidad técnica</i> | 5 |
| CAPITULO II | 6 |
| 2. MARCO TEORICO | 6 |
| 2.1 ANTECEDENTES DE LA INVESTIGACIÓN..... | 6 |
| 2.1.1 <i>Antecedente de la Universidad Nacional Pedro Ruiz Gallo</i> | 6 |
| 2.1.2 <i>Antecedentes internacionales</i> | 6 |
| 2.1.3 <i>Antecedentes nacionales</i> | 7 |
| 2.2 FUNDAMENTOS TEÓRICOS | 8 |
| 2.2.1 <i>Sistemas de Identificación Automática</i> | 8 |
| 2.2.1.1 Tarjetas de Identidad..... | 9 |
| 2.2.1.2 Bandas Magnéticas..... | 9 |
| 2.2.1.3 Códigos de Barras..... | 10 |
| 2.2.1.4 Identificación por Radiofrecuencia..... | 13 |
| 2.2.1.5 Sistemas Biométricos | 14 |
| 2.2.2 <i>Sistemas Arduino</i> | 15 |
| 2.2.2.1 Origen de Arduino | 15 |
| 2.2.2.2 Plataformas Arduino | 15 |
| 2.2.2.3 Ventajas de Arduino | 16 |
| 2.2.3 <i>Sistemas Biométricos</i> | 17 |
| 2.2.3.1 La Biometría | 17 |
| 2.2.3.2 Características para un Sistema Biométrico | 18 |
| 2.2.3.3 La huella dactilar | 19 |
| 2.2.4 <i>Sistemas de Gestión de Base de datos</i> | 20 |
| 2.2.4.1 Base de datos | 20 |
| 2.2.4.2 Tipos de bases de datos | 20 |
| 2.2.4.3 Modelo entidad-relación | 21 |
| 2.2.5 <i>Protocolo TCP-IP</i> | 22 |
| 2.2.5.1 Modelo cliente-servidor | 22 |
| 2.2.6 <i>Herramientas de Software</i> | 23 |

| | |
|--|-----------|
| 2.2.6.1 MySQL Workbench | 23 |
| 2.2.6.2 Microsoft Visual Studio | 24 |
| CAPITULO III | 26 |
| 3. DISEÑO DEL SISTEMA | 26 |
| 3.1. DISEÑO DEL HARDWARE | 26 |
| 3.1.1 Descripción de los elementos del sistema | 26 |
| 3.1.1.1 Arduino Mega | 26 |
| 3.1.1.2 ESP8266 | 27 |
| 3.1.1.3 LCD TFT/SD shield | 29 |
| 3.1.1.4 Modulo Pantalla TFT LCD | 29 |
| 3.1.1.5 Sensor de huella ZFM-20 | 31 |
| 3.1.1.6 Servidor | 33 |
| 3.1.1.7 Router | 33 |
| 3.1.2 Diagrama de conexión entre los elementos del sistema | 34 |
| 3.2 DISEÑO DEL SOFTWARE | 35 |
| 3.2.1 Programación Arduino | 35 |
| 3.2.1.1 Entorno de desarrollo | 35 |
| 3.2.1.2 Librerías usadas | 36 |
| 3.2.1.3 Diagrama de flujo programa principal arduino mega | 38 |
| 3.2.1.4 Diagrama de flujo registro en sistema de docente | 39 |
| 3.2.1.5 Diagrama de flujo registro de huella en sensor | 40 |
| 3.2.1.6 Diagrama de flujo registro de asistencia Arduino | 41 |
| 3.2.1.7 Diagrama de flujo comunicación ESP8266 | 42 |
| 3.2.2 Programación del software de aplicación | 43 |
| 3.2.2.1 Visual Basic | 43 |
| 3.2.2.2 Librerías del Visual Basic usadas en el sistema | 43 |
| 3.2.2.3 Diagrama de flujo de la aplicación del sistema | 44 |
| 3.2.2.4 Diagrama de flujo registro de asistencia en base de datos | 45 |
| 3.2.2.5 Diagrama de flujo registro de docentes en la base de datos | 46 |
| 3.2.2.6 Diagrama de flujo validación ingreso de administrador | 47 |
| 3.2.2.7 Diagrama de flujo consulta de hora actual | 48 |
| 3.2.2.8 Diagrama de flujo búsqueda de asistencias en base de datos | 49 |
| 3.3 COSTOS DEL SISTEMA | 50 |
| 3.3.1 Costos de equipamiento | 50 |
| 3.3.2 Costos de componentes electrónicos | 50 |
| 3.3.3 Costos del diseño e implementación | 52 |
| 3.3.4 Costo del sistema general | 52 |
| CAPITULO IV | 53 |
| 4.RESULTADOS | 53 |
| 4.1 IMPLEMENTACIÓN DEL PROTOTIPO | 53 |
| 4.1.1 Verificación del estado de elementos principales | 53 |
| 4.1.1.1 Verificación de conexión wifi con módulo esp8266 | 53 |
| 4.1.1.2 Verificación del estado del sensor de huella | 53 |
| 4.1.1.3 Calibración de la pantalla táctil | 55 |

| | |
|--|-----------|
| 4.1.1.4 Verificación de funcionamiento del buzzer..... | 56 |
| 4.1.1.5 Verificación de rtc clock..... | 57 |
| 4.1.2 <i>Ensamblaje del módulo electrónico</i> | 57 |
| 4.2 FUNCIONAMIENTO DEL PROTOTIPO | 58 |
| 4.2.1 <i>Interfaz gráfico de la pantalla lcd/tft</i> | 58 |
| 4.2.1.1 Verificación de la pantalla instrucciones | 59 |
| 4.2.1.2 Verificación del proceso de registro de docentes..... | 60 |
| 4.2.1.3 Verificación del proceso de identificación de docentes | 66 |
| 4.2.2 <i>Interfaz de la aplicación de escritorio</i> | 67 |
| 4.2.2.1 Verificación de login | 67 |
| 4.2.2.2 Verificación de función búsqueda..... | 68 |
| 4.2.2.3 Verificación de función exportar | 68 |
| 4.3 DATOS DE CONSUMO DEL MÓDULO..... | 69 |
| CAPITULO V | 70 |
| 5.CONCLUSIONES Y RECOMENDACIONES | 70 |
| 5.1 CONCLUSIONES | 70 |
| 5.2 RECOMENDACIONES..... | 71 |
| REFERENCIAS BIBLIOGRAFICAS..... | 72 |
| ANEXOS | 75 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1. Bandas magnéticas | 10 |
| Figura 2. Código de barras | 11 |
| Figura 3. Características de un código de barras | 12 |
| Figura 4. Plataformas Arduino | 16 |
| Figura 5. Huella dactilar | 19 |
| Figura 6. Huella dactilar 2 | 19 |
| Figura 7. Base de datos | 20 |
| Figura 8. Modelo entidad-relación | 21 |
| Figura 9. Modelo cliente-servidor..... | 22 |
| Figura 10. MySQL Workbench..... | 23 |
| Figura 11. Microsoft Visual Studio | 24 |
| Figura 12. Arduino mega..... | 27 |
| Figura 13. Esp8266..... | 28 |
| Figura 14. Lcd tft/sd shield | 29 |
| Figura 15. Diagrama de bloques sensor de huellas | 31 |
| Figura 16. Sensor de huella zfm-20..... | 32 |
| Figura 17. Diagrama de conexión entre los elementos del sistema | 34 |
| Figura 18. Diagrama de flujo programa principal Arduino Mega | 38 |
| Figura 19. Diagrama de flujo registro en sistema de docente | 39 |
| Figura 20. Diagrama de flujo registro de huella en sensor | 40 |
| Figura 21. Diagrama de flujo registro de asistencia Arduino | 41 |
| Figura 22. Diagrama de flujo comunicación esp8266..... | 42 |
| Figura 23. Diagrama de flujo de la aplicación del sistema | 44 |
| Figura 24. Diagrama de flujo registro asistencia en base de datos..... | 45 |
| Figura 25. Diagrama de flujo registro de docentes en base de datos | 46 |
| Figura 26. Diagrama de flujo validación ingreso de administrador | 47 |
| Figura 27. Diagrama de flujo consulta de hora actual | 48 |
| Figura 28. Diagrama de flujo búsqueda de asistencias en base de datos | 49 |
| Figura 29. Prueba funcionamiento sensor de huella | 53 |
| Figura 30. Test conexión Arduino – sensor de huella | 54 |
| Figura 31. Calibración pantalla táctil 1 | 55 |
| Figura 32. Calibración pantalla táctil 2..... | 55 |
| Figura 33. Calibración pantalla táctil 3..... | 56 |
| Figura 34. Prototipo 1..... | 57 |
| Figura 35. Versión final | 58 |
| Figura 36. Interfaz gráfica prototipo | 58 |
| Figura 37. Botón instrucciones pantalla principal | 59 |
| Figura 38. Pantalla instrucciones..... | 59 |

| | |
|---|----|
| Figura 39. Botón registro pantalla principal | 60 |
| Figura 40. Esperando huella administrador | 61 |
| Figura 41. Identificando huella | 61 |
| Figura 42. Lista de docentes a registrar | 62 |
| Figura 43. Formulario docente a registrar..... | 62 |
| Figura 44. Llenado de formulario | 63 |
| Figura 45. Botón registrar de formulario | 63 |
| Figura 46. Botón registrar huella..... | 64 |
| Figura 47. Esperando huella docente | 64 |
| Figura 48. Registrando huella en sensor | 65 |
| Figura 49. Botón enviar registro de docente | 65 |
| Figura 50. Botón presione aquí pantalla principal | 66 |
| Figura 51. Esperando huella dactilar | 66 |
| Figura 52. Reconociendo huella | 67 |
| Figura 53. Ventana login | 68 |
| Figura 54. Ventana principal | 68 |
| Figura 55. Pdf reporte de asistencias | 69 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1. Evolución de Visual Studio | 25 |
| Tabla 2. Características del Arduino mega..... | 26 |
| Tabla 3. Características del esp8266 | 28 |
| Tabla 4. Características pantalla TFT LCD..... | 30 |
| Tabla 5. Librerías Arduino Mega..... | 36 |
| Tabla 6. Costos de equipamiento | 50 |
| Tabla 7. Costos de componentes electrónicos | 50 |
| Tabla 8. Costos del diseño e implementación | 52 |
| Tabla 9. Costo del sistema general | 52 |
| Tabla 10. Datos de consumo del módulo | 69 |

INTRODUCCIÓN

La presente investigación se refiere al tema de registro de docentes, que se puede definir como control de asistencia de personal para la Facultad de Ciencias Físicas y Matemáticas, con la finalidad de asegurar la correcta información registrada por los mismos usuarios.

Una de las muchas soluciones frente a esta problemática es el uso de un sistema biométrico (referido a rasgos característicos únicos de un individuo) que en la actualidad es muy utilizado ya que evita la suplantación, además de almacenar los datos exactos de ingreso y salida de personal.

En base a esto se desarrolló nuestra tesis, cuyo propósito es el diseño e implementación de un sistema biométrico, que consta de un módulo físico conectado mediante red Wifi a una aplicación, y esta a una base de datos.

En el módulo físico contamos con el sensor de huella dactilar, un módulo Wifi, una pantalla, que interconectados con un microcontrolador podemos enlazarlos con la base de datos en un PC.

Esta tesis está dividida en cinco capítulos. En el capítulo I, se analiza la situación problemática en la que se encuentra la Facultad de Ciencias Físicas y Matemáticas en cuanto al escaso control sobre la fecha y hora de ingreso del personal docente. En el capítulo II se amplían los conceptos sobre la diversidad de sistemas de identificación hasta la actualidad. En el capítulo III se muestra el diseño del sistema describiendo sus elementos empleados, diagramas de flujo, detallando tanto el software como el hardware. Además, se evalúa el prototipo desde el punto de vista económico. En el capítulo IV, se da a conocer la implementación y el funcionamiento del prototipo y finalmente en el capítulo V se concluye con los resultados obtenidos y se sugieren recomendaciones para posibles mejoras de este proyecto.

CAPITULO I

1. PLANTEAMIENTO DEL PROBLEMA CIENTÍFICO

1.1 Descripción de la Realidad Problemática

Desde sus inicios el sistema de control de asistencia en la Facultad de Ciencias Físicas y Matemáticas de la Universidad Nacional Pedro Ruiz Gallo se realiza mediante fichas impresas en las que docentes deben firmar a puño y letra tanto su hora de entrada como hora de salida. Estas fichas deben ser almacenadas, durante un periodo de 5 años [art. 6, DS004-2006-TR]. Dichas fichas están dispuestas sobre un depósito al cual todo público en general tiene acceso, es decir nadie supervisa verdaderamente la hora en la que los docentes firman. Debido a este inconveniente el docente debe ser responsable a conciencia al registrar su ingreso y salida. Además de la no fiabilidad de este sistema, para poder realizar un reporte cada fin de ciclo académico, el personal administrativo debe contar las firmas de cada docente para justificar el salario de dicho; esto supone una pérdida de tiempo ya que la facultad cuenta con poco más de 130 docentes.

Con el avance de la tecnología se ha desarrollado diversos sistemas de identificación y autenticación tales como RFID, código de barras, firma electrónica, Números de identificación personal (PIN), y Sistemas de autenticación biométrica. Esta tecnología permite tener registro y almacenamiento de datos ya sean de objetos, de personal, entre otras aplicaciones, teniendo un mejor control sobre la información, entonces elaborar un reporte o buscar algún dato se vuelve cosa sencilla al acceder en la base.

El registro de personal actual en la región se realiza con la autenticación biométrica dactilar digital un ejemplo de ello son las universidades privadas tales como USAT, UCV, algunas entidades del estado como el Hospital Regional del Norte P.N.P Chiclayo y demás empresas ya han empezado a implementar este sistema para controlar la hora de entrada de su personal. Y es que el sistema biométrico dactilar digital se ha vuelto, por su simpleza, en la precisa herramienta para registrar

asistencia laboral. Consta generalmente de un sensor en cuyo encapsulado incluye un procesador para analizar la huella y obtener una matriz, también lleva una memoria flash para almacenar las plantillas, además cuenta con un puerto para conexión USB y así conectarse a un computador permitiendo almacenar los datos como hora y fecha; visualizando estos en una pantalla LCD.

Con el proyecto se pretende tener control exacto sobre la hora de entrada y salida del personal, además de facilitar la búsqueda de dichos, almacenando estos datos en un computador conectado vía Wifi al módulo. La aplicación de software desarrollada, permite filtrar búsquedas por fecha, apellido, DNI, exportar los datos en archivos PDF para poder ser impresas.

1.2 Formulación del Problema Científico

¿En qué forma el diseño e implementación del prototipo de un sistema biométrico podrá mejorar el control de asistencia del personal docente en la FACFyM?

1.3 Objetivos de la investigación

1.3.1 Objetivo General

Diseñar e implementar un prototipo de un sistema biométrico para mejorar el control de asistencia del personal docente en la FACFyM.

1.3.2 Objetivos Específicos

- Describir cómo se encuentra el sistema de asistencia en la actualidad.
- Almacenar el registro de entrada-salida en una base de datos.
- Desarrollo de aplicación para el reporte de asistencia.
- Diseño e implementación de módulo para la interfaz con el docente.
- Configuración de una red local para interactuar con el módulo.

1.4 Justificación e importancia de la investigación

Los sistemas que utilizan identificación biométrica presentan algunas ventajas sobre otros sistemas de identificación que utilizan contraseñas o tarjetas. Por ejemplo, la persona a ser identificada necesariamente tiene que estar presente en el punto de verificación, otra es que no es necesario recordar contraseñas o cargar algún artículo adicional el cual se puede extraviar de manera súbita.

Desde el punto de vista de eficiencia con respecto al registro manual de asistencia, tenemos algunas ventajas, por ejemplo, no existe influencia de las emociones humanas a la hora de tomar decisiones, a diferencia de los sistemas manuales los cuales son más laboriosos ya que requieren un seguimiento continuo para asegurarse de que cada registro se contabilice, el registro automático es rápido y eficaz. También es más fácil y mucho menos engorroso supervisar, monitorear y generar reportes de los registros hechos.

La implementación de este prototipo mejorará el control que se tiene sobre el registro de asistencia que actualmente se lleva a cabo en la Facultad de Ciencias Físicas y Matemáticas de la Universidad Nacional Pedro Ruiz Gallo, ya que este sistema biométrico ayudará a eliminar suplantaciones de terceros, a la vez nos reportará la hora exacta de entrada y salida de los docentes a diferencia del registro en papel (Kardex) que es propenso al error humano.

1.5 Limitaciones del estudio

En el desarrollo de esta investigación se presentó la siguiente limitación:

- Demora en registrar a los docentes de la facultad, ya que no se disponía en muchas ocasiones de su presencia.
- No se contaba con los estudios en programación de software con base de datos. Se recurrió a estudios autodidactas.

1.6 Viabilidad

1.6.1 Factibilidad operativa

Hubo apoyo y voluntad por parte del decano y personal de la Facultad de Ciencias Físicas y Matemáticas para el desarrollo y pruebas del sistema, ya que son conscientes de las deficiencias que presenta el sistema actual. Se logró recabar la información necesaria para el desarrollo de la base de datos.

1.6.2 Factibilidad económico-financiera

Todos los gastos realizados tanto en el desarrollo del software como del prototipo fueron asumidos por los autores de esta tesis.

1.6.3 Factibilidad técnica

Se hizo uso de Microsoft Visual Community Studio 2015 como herramienta de programación, MySQL Workbench para el diseño y soporte de la base de datos, Arduino IDE 1.6.11 como herramienta de programación, los cuales son software gratuito.

CAPITULO II

2. MARCO TEÓRICO

2.1 Antecedentes de la investigación

2.1.1 Antecedente de la Universidad Nacional Pedro Ruiz Gallo

Título: Sistema de control de acceso con tecnología biométrica e informática; caso: Control de asistencia y permanencia de personal utilizando huella dactilar.

Autor: Gil Seytuque, Segundo Francis, Luján Flores, César Augusto.

Año: 2005

Lugar: Lambayeque

En esta investigación el objetivo principal fue la comparación de las diferentes técnicas de control de acceso de personal.

Describir detalladamente las tecnologías biométricas, tanto las fisiológicas como conductuales.

Investigación basada en el análisis de métodos y algoritmos de captura, almacenamiento, acceso y recopilación de información de las tecnologías biométricas de huella dactilar, considerando la factibilidad económica, técnica y operativa del sistema desarrollado.

2.1.2 Antecedentes internacionales

Título: Sistema biométrico de control de asistencia laboral mediante el uso de huella dactilar.

Autor: Pérez Faria, Alberto José, El Safadi Fereira, Anthony

Año: 2014

Lugar: Venezuela

El objetivo de esta tesis fue desarrollar un sistema biométrico de control de asistencia laboral mediante el uso de huella dactilar.

Determinar la plataforma y la arquitectura que mejor se adapte al sistema de capta huellas para diseñar la distribución móvil y el modelo de negocio con la interfaz de la aplicación de escritorio y la aplicación web desarrollada, codificar el sistema de control de asistencia.

Realizar pruebas y diagnosticar el funcionamiento completo del sistema.

2.1.3 Antecedentes nacionales

Título: “Control de asistencia al personal administrativo de la Universidad Nacional de San Martín utilizando Biometría”

Autor: Carlos Augusto López Gonzales

Año: 2008

El objetivo de esta tesis fue comprobar la problemática de la Universidad Nacional de San Martín, el cuál es la suplantación de docentes administrativos, debido al uso de tarjetas con código de barra.

Esta investigación se centra en la implementación de un sistema biométrico para mejorar el control de asistencia del personal administrativo que labora en UNSM-T, la cual trata de demostrar la superioridad sobre los sistemas de código de barras, argumentándose en que dicho sistema no autentifica al personal, sino a tarjetas.

Título: Control de Asistencia de Docentes con Dispositivo Biométrico para la Universidad “San Luis Gonzaga” de Ica

Autor: Guerreros Huamaní, Luigi Leonardo ,Salguero Pizarro, Jerson Alonso

Año: 2013

El objetivo de esta tesis fue mejorar el sistema de control de asistencia de docentes de la “Universidad San Luis Gonzaga de Ica”.

Esta tesis evalúa de qué manera la incorporación de un sistema biométrico podrá menguar las deficiencias como: falta de compromiso de docentes, hojas de control manipulables, y uso extra del personal administrativo.

A la vez busca obtener reportes de asistencias de forma rápida y en tiempo real.

2.2 Fundamentos teóricos

2.2.1 Sistemas de Identificación Automática

La identificación es la acción de reconocer mediante características y particularidades físicas de una persona. [1]

Hoy en día es muy común ver casos de rotulado de productos (ya sea para su conteo, identificación) a través de distintas técnicas de entradas de datos como por ejemplo el reconocimiento óptico, la cinta magnética, código de barras, entre otros. En la mayoría de los casos se trata de una serie de técnicas las cuales decodifican datos de una imagen que es leída por equipos especiales (controladores) los cuales envían información a un computador. [2]

Gran cantidad de empresas en el mundo ya han adoptado algún sistema de estas características. Por ejemplo, en la Argentina, ya existen 15.000 empresas que utilizan los códigos de barras para identificar un total de más de 550.000 productos. [2]

Por otro lado, la identificación automática de individuos conocida como reconocimiento biométrico, se ha desarrollado mucho durante los últimos años, ya que establece la identidad mediante “algo que es” y no “algo que se posee”, utilizándose en muchas aplicaciones, entre ellas el control de asistencia. [3]

De esto concluimos que esta tecnología es más eficiente y segura, por el hecho de usar computadores; los cuales facilitan la tarea de controlar, cobrar servicios y mercaderías, disminuyendo la posibilidad de errores.

2.2.1.1 Tarjetas de Identidad

En algunos países, se utilizan tarjetas de identidad cotidianamente para diversos propósitos. Otros se han resistido a su uso cotidiano. La sensibilidad local hacia las tarjetas de identidad puede afectar la conveniencia de su uso en circunstancias particulares (como verificar la identidad de un ciudadano). [4]

Para propósitos electorales, pueden ser utilizadas para identificar a los electores cuando se registran para votar o cuando se presentan en los sitios de votación. También pueden ser utilizadas por el personal para acceder a los sitios de trabajo o algún otro centro. [4]

2.2.1.2 Bandas Magnéticas

Utiliza señales electromagnéticas de alta o baja energía para registrar y codificar información en una banda que puede ser leída por una máquina para identificación instantánea. Su aplicación quizás más difundida es en las tarjetas de crédito. Las instituciones financieras prefieren esta tecnología pues la reproducción es difícil de lograr sin el equipo adecuado, el cual es apreciablemente costoso. Cuando se realiza una comparación entre tecnologías, comúnmente se hace mención a que las bandas magnéticas se utilizan para la identificación de personas, mientras que los códigos de barras se aplican en la identificación de productos. [2]

Figura 1. Bandas magnéticas



Fuente: Propia de los autores

2.2.1.3 Códigos de Barras

El Código de Barras es un arreglo en paralelo que contiene información codificada en las barras y espacios del símbolo. Esta información puede ser leída por medio de dispositivos ópticos los cuales lo decodifican y envían hacia una computadora. El código de barras es la mejor tecnología para implementar un sistema de recolección de datos mediante identificación automática. Presenta muchos beneficios, alguno de los cuales son [2]:

- Se imprime a bajos costos.
- Permite porcentajes muy bajos de error.
- Los equipos de lectura e impresión de código de barras son flexibles y fáciles de conectar e instalar.
- Se mejora la exactitud de los datos.
- Se pueden contar con nuevas categorías de información.

Las aplicaciones del código de barras cubren prácticamente cualquier tipo de actividad humana, tanto en la industria como en el comercio, instituciones educativas, instituciones médicas, gobierno, etc. De hecho, se los utiliza para los más diversos usos como, por ejemplo: control de material en proceso, control de calidad, embarques y recibos, control de documentos,

facturación, control de tiempo y asistencia, bibliotecas, bancos de sangre, y muchos otros. [2]

Figura 2. Código de barras.



Fuente: Propia de los autores

Características de un código de barras:

Un código de barras es un símbolo que puede tener varias características, entre las cuales podemos nombrar:

Densidad: es el ancho del elemento (barra o espacio) más angosto dentro del símbolo de código de barras. Está dado en milésimas de pulgada. Un código de barras no se mide por su longitud física sino por su densidad.

WNR (por sus siglas en inglés de Wide to Narrow Ratio): es la razón del grosor del elemento más angosto contra el grosor del elemento más ancho. Usualmente, este valor es 1:3 o 1:2.

Quiet Zone: es el área blanca al principio y al final de un símbolo de código de barras. Esta área es necesaria para una lectura conveniente del símbolo por parte del dispositivo de captura de datos. [2]

Figura 3. Características de un código de barras.



Fuente: "Identificación Automática" [2]

Lectores de código de barra

entre los lectores de código de barra tenemos: Lápiz óptico, Láser de pistola, CCD, Láser omnidireccional, etc.

2.2.1.4 Identificación por Radiofrecuencia

La Identificación por Radio Frecuencia es una tecnología emergente que podría reemplazar potencialmente al código de barras en los próximos años. El RFID es un término genérico para tecnologías que usan ondas de radio para la identificación automática de personas u objetos. Hay varios métodos de identificación, pero el más común es el almacenamiento de un número seriado que identifica una persona u objeto sobre un microchip que está relacionado con una antena (el chip y la antena juntos son llamados transmisores RFID o etiqueta RFID). La antena permite al chip transmitir la información de identificación a un lector. El lector convierte las ondas de radio reflejadas desde la etiqueta RFID en información digital que puede luego ser pasada a una computadora que hará uso de ella. Los tags o etiquetas pueden confeccionarse en todos los tamaños y formatos, con un espesor tan fino que permite su aplicación en la superficie de cualquier producto. Tiene un componente electrónico con memoria que almacena los datos y un transmisor que contiene una antena de radio miniatura. [2]

Las etiquetas pasivas no necesitan alimentación eléctrica interna, mientras que las activas sí lo requieren. Una de las ventajas del uso de radiofrecuencia (en lugar, por ejemplo, de infrarrojos) es que no se requiere visión directa entre emisor y receptor.

2.2.1.5 Sistemas Biométricos

Un sistema biométrico en general consta de componentes tanto hardware como software necesario para el proceso de reconocimiento. Dentro del hardware se incluyen principalmente los sensores que son los dispositivos encargados de extraer la característica deseada. Una vez obtenida la información del sensor, será necesario realizar sobre ella las tareas de acondicionamiento necesarias, para ello se emplean diferentes métodos dependiendo del sistema biométrico utilizado. [5]

Por ello se han descrito los principales tipos de sistemas biométricos existentes:

- Reconocimiento de la huella dactilar
- Reconocimiento de la cara
- Reconocimiento de iris/retina
- Geometría de dedos/mano
- Autenticación de la voz
- Reconocimiento de la firma

Para cada uno de estos sistemas se ha descrito su funcionamiento y algunas de las técnicas que se utilizan para procesar los datos obtenidos a partir de los sensores. Los sistemas biométricos se han desarrollado como respuesta a la creciente demanda de seguridad existente en la actualidad y aunque algunos de ellos son altamente fiables, ningún sistema es efectivo al 100%, y estos sistemas también son susceptibles de ser engañados. [5]

2.2.2 Sistemas Arduino

2.2.2.1 Origen de Arduino

Arduino surgió alrededor del año 2005 en el instituto Ivrae, Italia como una solución económica para que los estudiantes de dicho instituto puedan hacer sus prototipos de hardware, para esto se constituyó como una empresa de hardware libre, los desarrolladores son Massimo Banzi, David Cuartiles, Tom Igoe, Gianluca Martino, David Mellis y Nicholas Zambetti, con la colaboración de Hernando Barragan. [6]

El hardware de Arduino está basado usualmente en un microcontrolador AVR Atmel, el cual incluye un bootloader pre-programado y un conversor serial usb-rs232 esto acelera el proceso de cargar y depurar el código ya que todo esto se hace desde el mismo IDE. [7]

El software consiste en un IDE (Entorno de Desarrollo) basado en el lenguaje de programación Wiring el cual permite compilar, depurar y cargar el código a la tarjeta Arduino. [7]

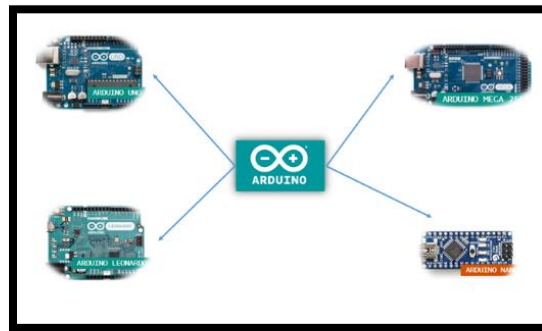
2.2.2.2 Plataformas Arduino

Arduino en la actualidad cuenta con una amplia variedad de tarjetas de diferentes tamaños, capacidad de memoria, conectividad y pines E/S por lo que uno puede elegir cuál de ellas se acomoda más a sus necesidades. [8]

En sus inicios Arduino contaba con tarjetas que usaban microcontroladores con características básicas como salidas y entradas digitales, pwm, analógicas pero con el avance de la tecnología los microprocesadores han evolucionado ahora podemos ver que cuentan con tarjetas con conectividad bluetooth, wifi, ethernet, gsm lo cual amplía mucho más tanto las aplicaciones en las que puedan ser usadas como el público al cual va orientado, ya no solo estudiantes y hobbistas sino también docentes, desarrolladores o profesionales en general.[8]

Algunas de las tarjetas electrónicas son: Arduino UNO, Arduino Nano, Arduino Leonardo, Arduino Mega, Arduino Yun, Arduino Due, etc. [8]

Figura 4. Plataformas Arduino.



Fuente: Arduino Products [8]

2.2.2.3 Ventajas de Arduino

Entre las ventajas más resaltantes que presenta Arduino en comparación con otras tarjetas embebidas tenemos [7]:

- Costo reducido: al ser open source diferentes fabricantes producen las tarjetas electrónicas por lo que el precio es accesible.
- Compatibilidad con otros componentes electrónicos: existen múltiples módulos con características diferentes que se pueden acoplar a una tarjeta Arduino.
- Open-source: actualmente es un modelo de desarrollo de software o hardware basado en la colaboración de las personas, va más allá de ser un software gratuito ya que se puede modificar la fuente del programa sin restricciones de licencia.
- Flexible: Arduino tiene aplicaciones en distintos campos de estudio, arte, arquitectura, ingeniería, etc.

2.2.3 Sistemas Biométricos

2.2.3.1 La Biometría

Se define a la biometría como el conjunto de caracteres, rasgos físicos, cualidades que diferencian de manera unívoca a cada ser humano y que pueden ser medidos. [3]

Desde la antigüedad los seres humanos han utilizado los rasgos biométricos tales como la cara, la voz, la manera de caminar para reconocerse unos a otros. [3]

El origen de la biometría se puede remontar hasta el siglo VIII en China al igual que en la antigua Babilonia y Persia se usaban las impresiones dactilares para autenticar registros en arcilla, pues ya se conocía su carácter único. Siglos más tarde luego que Europa también reconociera la unicidad se desarrollaron estudios como en 1868, Marcelo Malpigo hizo el primer estudio sistemático de huellas dactilares. [3]

En 1941, en los Laboratorios Bell de Murray Hill (Nueva Jersey), comenzó el estudio de la identificación por voz. En 1986, Sir Alec Jeffreys utilizó por primera vez el ADN para identificar al autor de unos asesinatos en Inglaterra. De esta manera nacen los sistemas biométricos, es decir por la necesidad de lidiar con el crimen. [3]

Actualmente en una sociedad interconectada como la nuestra, establecer la identidad de un individuo se ha convertido en un aspecto crítico, y a la vez cotidiano, en una gran variedad de escenarios tales como el uso de cajeros automáticos, la asistencia de personal, etc. El reconocimiento biométrico definido como la técnica que posibilita la identificación automática de individuos basándose en sus características físicas o de comportamiento, está ganando gran aceptación como método para determinar la identidad de cada persona y ya se está utilizando en multitud de aplicaciones tanto comerciales como público-gubernamentales, en ámbitos tanto civiles como forenses (es decir, relacionadas con lo policial y lo legal, "criminalística"). Mediante el uso del reconocimiento biométrico, es

posible establecer la identidad de una persona mediante “algo que se es”, a diferencia de los tradicionales sistemas basados en “algo que se posee” (como un DNI, una tarjeta de identificación o una llave), que puede perderse o robarse, o en “algo que se sabe” (como una clave), que puede ser olvidado [3].

2.2.3.2 Características para un Sistema Biométrico

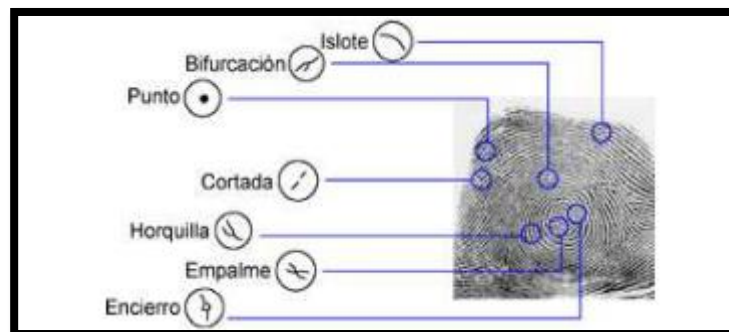
Para ser considerado un sistema biométrico, el indicador biométrico (la característica a ser medida) debe cumplir necesariamente las siguientes características [3]:

1. Universalidad: cualquier persona posee esa característica;
2. Unicidad: la existencia de dos personas con una característica idéntica tiene una probabilidad muy pequeña (casi nula).
3. Permanencia: la característica no cambia en el tiempo.
4. Cuantificación: la característica puede ser medida en forma cuantitativa.

2.2.3.3 La huella dactilar

La lectura de la huella dactilar es la más antigua y posee un alto grado de precisión ya que la misma es única e inalterable. Existen dos métodos de coincidencias de lectura de muestras, el primero es basado en minucias, que consiste en la determinación de formas fácilmente identificables y el segundo método es basado en correlación, el cual analiza el patrón global de la huella [9].

Figura 5. Huella dactilar.



Fuente: “Biometría” [9]

Figura 6. Huella dactilar 2.



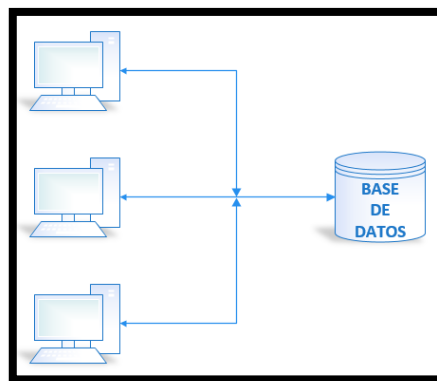
Fuente: “Biometría” [9]

2.2.4 Sistemas de Gestión de Base de datos

2.2.4.1 Base de datos

Una base de datos es un conjunto de información que comparte una misma estructura y está relacionada entre sí para que sea fácil de encontrar. Dicha información es compartida para varios usuarios. [10]

Figura 7. Base de datos.



Fuente: Propia de los autores

2.2.4.2 Tipos de bases de datos

Entre los tipos de base de datos más usadas tenemos [10]:

MySQL: es una base de datos con licencia GLP basada en un servidor. Se caracteriza por su rapidez.

PostgreSQL y Oracle: Sistemas de base de datos poderosos. Administra muy bien grandes cantidades de datos y suelen ser usadas en intranet y sistemas de gran calibre.

Microsoft Access: Desarrollada por Microsoft, se debe crear bajo el programa Access usada para bajas cantidades de información.

Microsoft SQL server: Base de datos más potente que Access desarrollada por Microsoft. Se utiliza para manejar grandes volúmenes de información.

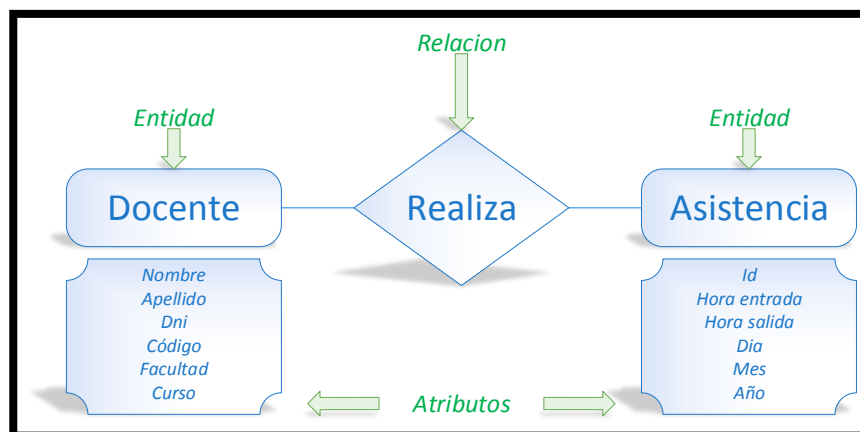
2.2.4.3 Modelo entidad-relación

Es un tipo de modelado de información, basado en una percepción del comportamiento de las personas o cosas a nuestro alrededor la cual se basa en elementos básico llamados entidades y las relaciones que existen entre ellas.

Entidad: representa una cosa, objeto o concepto del mundo real se diferencia del resto y persiste por sí misma.

Atributo: son cualidades de las entidades que permiten diferenciarlas entre sí. [11]

Figura 8. Modelo entidad-relación.



Fuente: Propia de los autores

2.2.5 Protocolo TCP-IP

El protocolo TCP fue creado en los años 70 como un proyecto de investigación basado en la conmutación de paquetes. [12]

Este protocolo es el encargado de asegurar que la información a transmitir llegue a su destino confirmando la recepción del paquete de datos, mediante el checksum. [13]

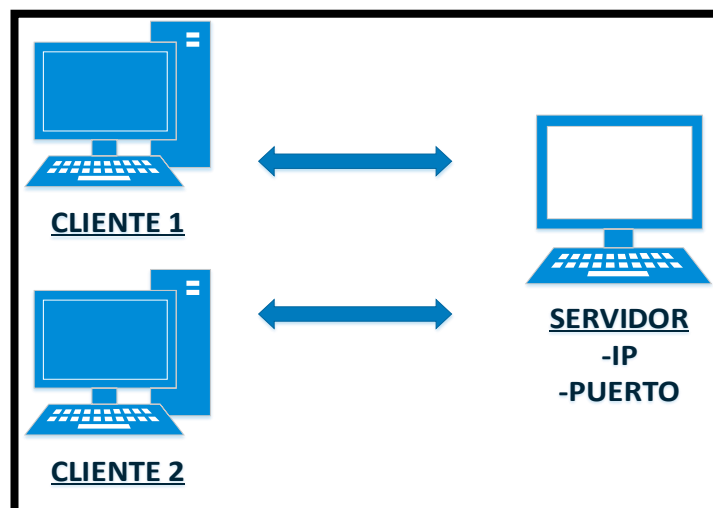
Esto es debido a que la información se divide en paquetes, los cuales tiene la información necesaria para llegar al destino y reconstruir el mensaje. Además, que cada paquete se maneja por separado por lo que, si algún paquete no lograra llegar porque algún nodo está siendo utilizado, puede ser enviado por otro camino. [13]

2.2.5.1 Modelo cliente-servidor

Es un modelo utilizado en aplicaciones que implementan el protocolo TCP-IP para comunicarse entre ellas. [14]

El servidor es un programa que recibe una solicitud, realiza el servicio requerido y devuelve los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo. [14]

Figura 9. Modelo cliente-servidor.



Fuente: Propia de los autores

2.2.6 Herramientas de Software

2.2.6.1 MySQL Workbench

Es una herramienta gráfica para trabajar con servidores y bases de datos MySQL. MySQL cuenta con las siguientes funciones [15]:

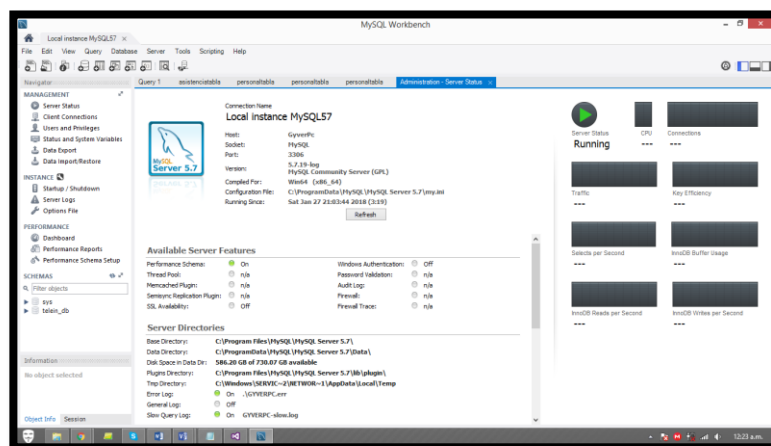
Desarrollo SQL: Permite crear y monitorear conexiones a los servidores de base de datos, así mismo configurar los parámetros de la conexión, permite ejecutar consultas SQL en las conexiones usando el editor.

Modelamiento de información: Permite crear modelos de bases de datos de manera gráfica, el editor de tablas permite editar columnas, índices, opciones, privilegios, etc.

Administración de servidor: Permite administrar instancias del servidor MySQL, crear respaldos y recuperar la información almacenada.

Migración de información: Permite la migración de base de datos creadas en SQL server, Microsoft Access, SQLite, PostgreSQL a MySQL.

Figura 10. Mysql workbench.



Fuente: Propia de los autores

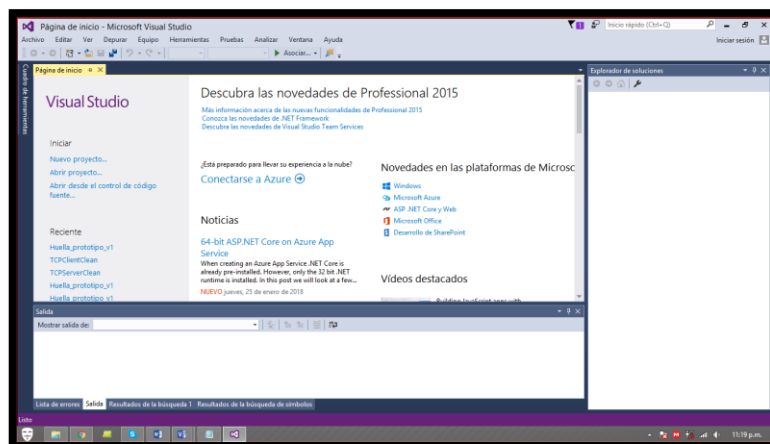
2.2.6.2 Microsoft Visual Studio

Es un entorno de desarrollo integrado el cual nos permite crear aplicaciones de escritorio, web, móviles, soporta diferentes lenguajes como C#, VB, C++, Javascript y Python. [16]

Este software cuenta con varias herramientas que permiten editar, diseñar, depurar las aplicaciones más fácil y rápido, por ejemplo: Peek to definition, GoTo, Code Lens, Light bulb icons, Error list, Refactoring. [16]

Esta versión community, nos proporciona todos los servicios de la versión profesional, pero está orientado a estudiantes y desarrolladores independientes. [16]

Figura 11. Microsoft visual studio.



Fuente: Propia de los autores

Tabla 1

Evolución de Visual Studio

| Versión | Mejora |
|--------------------------------|--|
| Visual studio 6.0 | Se lanzó en 1998, fue la última versión que se podía ejecutar en win9x. incluía visual J++, visual interdev, visual c++, visual foxpro y visual basic. |
| Visual studio .net 2002 | Fue un cambio sustancial de la versión anterior, ya que implementa la plataforma .NET la cual permite que el código de la aplicación sea independiente de la plataforma utilizada. |
| Visual studio .net 2003 | Incorporación de .NET framework 1.1. |
| Visual studio 2005 | Se implementaron mejoras sustanciales, y soporte para sistemas de 64bits. |
| Visual studio 2008 | Incorporación de .NET framework 2.0,3.0,3.5. |
| Visual studio 2010 | Incorporación de .NET Framework 2.0,3.0,3.5,4.0. |
| Visual studio 2012 | Incorporación de .NET Framework 2.0,3.0,3.5,4.0,4.5. |
| Visual studio 2013 | Incorporación de .NET Framework 2.0,3.0,3.5,4.0,4.5,4.5.1,4.5.2. |
| Visual studio 2015 | Incorporación de .NET Framework 2.0,3.0,3.5,4.0,4.5,4.5.1,4.5.2,4.6,4.6.1. |
| Visual studio 2017 | Incorporación .NET Framework 2.0,3.0,3.5,4.0,4.5,4.5.1,4.5.2,4.6,4.6.1,4.7. |
| Fuente: Visual Studio Web [17] | |

CAPITULO III

3. DISEÑO DEL SISTEMA

3.1. Diseño del hardware

3.1.1 Descripción de los elementos del sistema

3.1.1.1 Arduino Mega

Tarjeta de desarrollo en sistema libre, cuyas características superan a la tarjeta Arduino uno, pensado para proyectos de mayor robustez, sin mucha complejidad para su manejo.

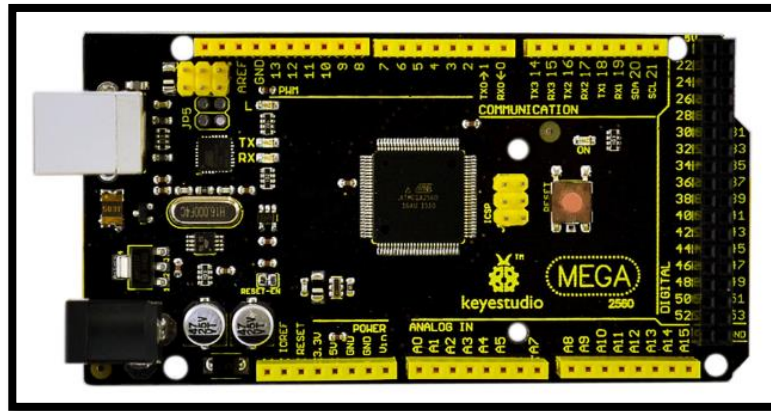
Tabla 2

Características del Arduino mega

| Características del Arduino Mega | | | |
|----------------------------------|------------|-------------------|---------|
| Microcontrolador | ATmega1280 | Salidas PWM | 4 pines |
| Voltaje | 5v | Comunicación SPI | 4 pines |
| Pines digitales i/o | 54 | Comunicación I2C | 2 pines |
| Pines analógicos | 16 | Comunicación UART | 8 pines |
| Corriente por pin | 40 mA | EEPROM | 4KB |
| Memoria flash | 128 KB | Velocidad | 16 MHz |
| SRAM | 8 KB | | |

Fuente: Arduino mega [18]

Figura 12. Arduino mega.



Fuente: Propia de los autores.

3.1.1.2 ESP8266

La serie ESP8266, o familia de chips Wi-Fi es un “sistema en chip” (SoC) creado para conectar todo objeto a internet, es decir fue creado para el desarrollo del “Internet de las cosas”.

El internet de las cosas (IoT) es la red de objetos cotidianos, elementos físicos integrados con componentes electrónicos, software, sensores que permiten la conectividad para el intercambio de información ya sean focos, electrodomésticos, sistema de riego, sistema de calefacción, todos estos conectados a Internet. [19]

Existen variedades de chip, cada uno de mayor capacidad que otro, sin embargo, en el desarrollo del módulo se usó el chip “Tensilica Xtensa LX3”, cuyas características son:

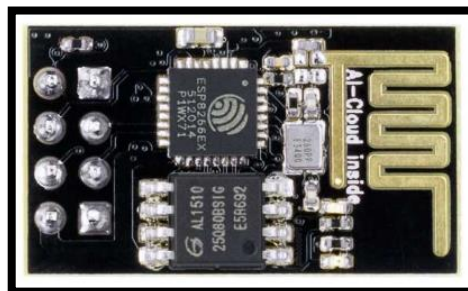
Tabla 3

Características del esp8266

| Características del esp8266 | | | |
|-----------------------------|----------------------|--------------|-------------|
| Microcontrolador | Tensilica Xtensa LX3 | Salidas UART | 2 pines |
| Voltaje de operación | 3.3v | RAM | 96KB |
| Pines digitales gpio | 4 | Wi-Fi | P2P,soft AP |
| Pines analógicos | 16 | Velocidad | 80MHz |
| Memoria flash | 4MB | RAM | 96KB |

Fuente: Naylampmechatronics esp8266 [20]

Figura 13. Esp8266.



Fuente: Propia de los autores.

3.1.1.3 LCD TFT/SD shield

Tarjeta electrónica que sirve como driver entre el Arduino Mega y la pantalla táctil, ya que ambos voltajes no son compatibles y se requiere acondicionamiento de las señales.

Esta shield utiliza 5 buffers Hc541 para comunicación con los 16 bits paralelos de la pantalla y el bus SPI del táctil.

Figura 14. Lcd tft/sd shield



Fuente: Propia de los autores

3.1.1.4 Módulo Pantalla TFT LCD

En el desarrollo del módulo se evaluaron diferentes pantallas para la visualización, entre las que fueron: LCD 16X2, OLED, TFT LCD (Thin Film Transistor-Liquid Crystal Display), se optó por esta última ya que, el uso de botones era imprescindible, y crearlos eventualmente es una facilidad que ofrecen las pantallas como estas.

Este módulo consta de un controlador SSD1289 para gestionar la pantalla TFT LCD de 7 pulgadas. Al ser TFT estos transistores permiten mejorar la calidad de imagen en comparación a las pantallas tradicionales LCD, lo cual la hace ideal para el sistema implementado, gracias a su panel táctil el usuario podrá manipular los menús desarrollados en la interfaz gráfica del sistema.

Una de las ventajas de esta pantalla TFT LCD es que se cuenta con una librería de apoyo realizada por Henning Karlsen (desarrollador de librerías de software libre para pantallas TFT LCD de cualquier tamaño [21]), para la administración de su hardware a través de Arduino, aunque se tuvo que modificar esta librería para que funcione con el modelo de controlador de la pantalla empleada y para que funcionase con la resolución de 800 x 480.

Además, cuenta con un lector SD, lo que le permite almacenar y leer imágenes y presentarlas como interfaz, para lo cual requiere que dichas imágenes estén en el formato RGB565 o RAW565 (. raw), este formato hace referencia al dato de los pixeles de un archivo de mapa de bits. En la programación de la tarjeta Arduino se debe especificar las dimensiones de las imágenes a emplear para que se desplieguen adecuadamente en la pantalla.

Tabla 4

Características pantalla TFT Lcd.

| CARACTERISTICA | DESCRIPCIÓN |
|--------------------------|-------------------------------------|
| Tamaño | 7 pulgadas |
| Resolución | 800px * 480px |
| Controlador | SSD1963 |
| Tecnología lcd | TFT transmisivo normal blanco |
| Corriente de consumo | 400 mA |
| Voltaje de operación | 3.3 ~ 5 V |
| Interfaz de comunicación | Interfaz de bus paralelo de 16 bits |
| Fuente: AliExpress [22] | |

3.1.1.5 Sensor de huella ZFM-20

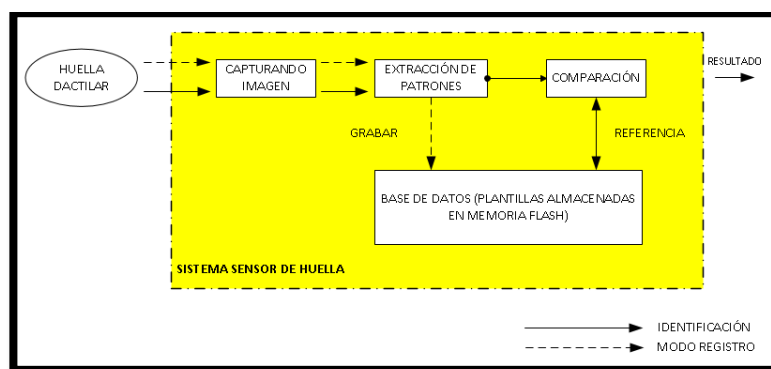
El dispositivo integra un sensor óptico, un procesador de alta velocidad (DSP), algoritmos de alto desempeño para la identificación/verificación de huellas y memoria flash para almacenamiento de huellas.

Su funcionamiento es muy sencillo, solo es necesario conectarlo a un microcontrolador mediante el puerto serial TTL, luego se puede solicitar la captura de huellas, búsqueda, detección, etc. Todo el procesamiento de captura/verificación/búsqueda es realizado por el DSP y responde con comandos sencillos de interpretar.

Se pueden almacenar hasta 162 huellas en la memoria flash del dispositivo (algunos módulos pueden almacenar hasta 1000, la diferencia radica en el tamaño de la memoria FLASH). El sensor óptico incluye un led rojo que se ilumina al momento de tomar una muestra, de esa forma podremos saber cuándo el sensor está trabajando.

El diagrama de bloques de este sensor es el siguiente:

Figura 15. Diagrama de bloques sensor de huella.



Fuente. Propia de los autores

Las especificaciones técnicas del sensor de huellas, son las siguientes [23]:

1. Voltaje de alimentación: 3.6 – 6.0 V dc
2. Corriente de operación: 120 mA
3. Corriente pico: 150 mA
4. Tiempo de captura de imagen: <1.0 segundo
5. Superficie de contacto: 14mm x 18mm
6. Archivo de firma: 256 bytes
7. Archivo de plantilla: 512 bytes
8. Capacidad de almacenamiento: 162 plantillas
9. Interface: TTL Serial
10. Velocidad de transmisión: 57600 baudios
11. Procesador: DSP ARM cortex M3 core

Figura 16. Sensor de huella zfm-20.



Fuente: Propia de los autores

3.1.1.6 Servidor

Para almacenar los datos obtenidos de la aplicación, se hace uso del modelo cliente-servidor. En el presente diseño se usa una laptop(servidor) para almacenar y enviar datos (base de datos), a la aplicación(cliente).

Las características de la portátil HP usada en pruebas son:

1. Intel Core i5-3230M 2.6 GHz (c/TB 3.2 GHz)
2. Pantalla LED 14" HD (1366x768)
3. 8GB RAM / Disco Duro 750GB
4. Intel HD Graphics 4000

3.1.1.7 Router

Para crear la red inalámbrica se configuró un router Inalámbrico TP-LINK TL-WR842ND. Un Router es un dispositivo que opera en capa tres de modelo OSI. Así, permite que varias redes u ordenadores se conecten entre sí y compartan una misma conexión de red. [24]

El router usado presenta las siguientes características [25]:

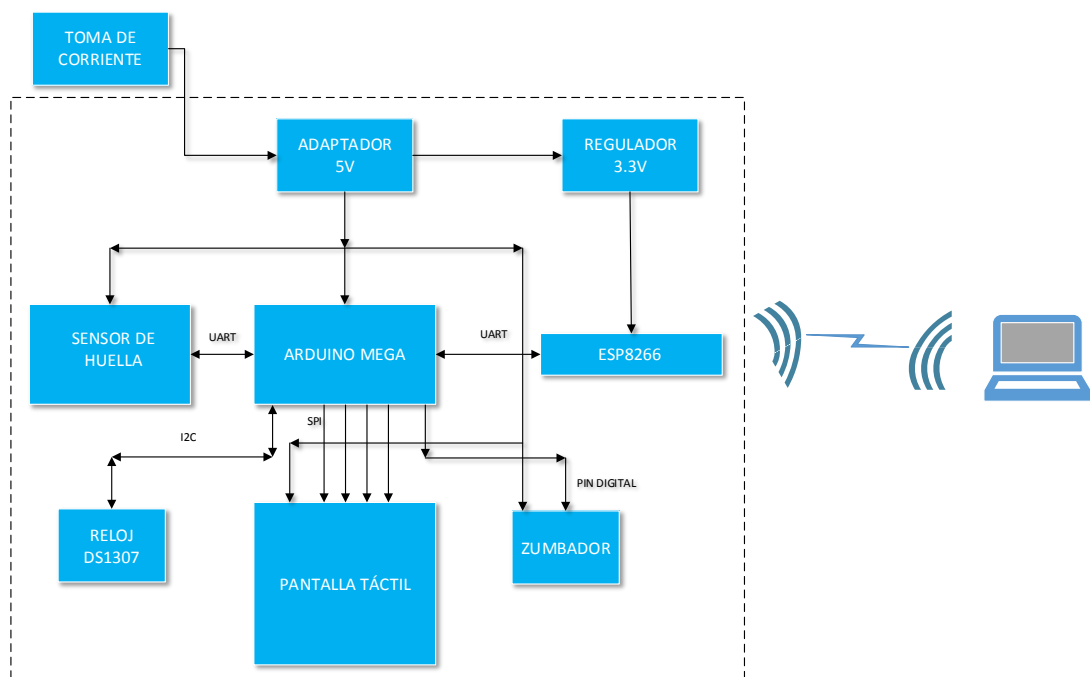
1. Potencia de transmisión es <30 dBm
2. Frecuencia: 2.4-2.4835GHz
3. Velocidad de transmisión: 300Mbps
4. Interfaz: LAN/WLAN

3.1.2 Diagrama de conexión entre los elementos del sistema

En el siguiente diagrama se puede apreciar a grandes rasgos los componentes principales que conforman el prototipo realizado.

Los cuales son: el Arduino Mega que es el controlador principal a él llegan las señales del sensor de huellas, el módulo wifi, el circuito reloj como entradas; y como salidas tenemos a la pantalla táctil y al zumbador.

Figura 17. Diagrama de conexión entre los elementos del sistema.



Fuente: Propia de los autores

3.2 Diseño del software

3.2.1 Programación Arduino

3.2.1.1 Entorno de desarrollo

El entorno de desarrollo de Arduino, nos permite editar, compilar y depurar el código. Además, nos permite establecer la conexión con las tarjetas electrónicas para poder cargar el programa a la memoria flash del mismo.

Asimismo, cuenta con librerías para manejar de manera más fácil y rápida distintos sensores y actuadores.

El código fuente del programa está dividido en 2 partes o funciones (Setup, Loop):

Setup: Aquí se ponen las tareas que se quiere realizar cuando inicializa el controlador.

Loop: Es un bucle infinito, aquí se ponen las tareas o eventos que se van a realizar o suceder constantemente.

3.2.1.2 Librerías usadas

Librerías Arduino Mega

Tabla 5

Librerías Arduino Mega

| LIBRERÍA | DESCRIPCIÓN |
|--------------------------|--|
| "Adafruit_Fingerprint.h" | Nos permite interactuar con el sensor dactilar, acceder a las funciones de registrar e identificar. |
| "tinyFAT.h" | Nos permite interactuar con la tarjeta SD, para leer y escribir en ella. |
| "UTFT.h" | Nos permite controlar la parte gráfica de la pantalla, podemos crear formas, texto, cambiar colores. |
| "UTFT_tinyFAT.h" | Nos permite importar y adaptar imágenes que se encuentren en la tarjeta SD a la pantalla lcd. |
| "URTouch.h" | Nos permite controlar la parte táctil de la pantalla, detectar las coordenadas donde fue presionada. |
| "UTFT_Buttons.h" | Nos permite crear botones y darles un aspecto personalizado, trabaja junto con la librería URTouch.h |
| "UTFT_DLB.h" | Nos permite agregar un nuevo estilo de letra al actual repertorio que trae UTFT.h |

Fuente: Propia de los autores

Librerías ESP8266

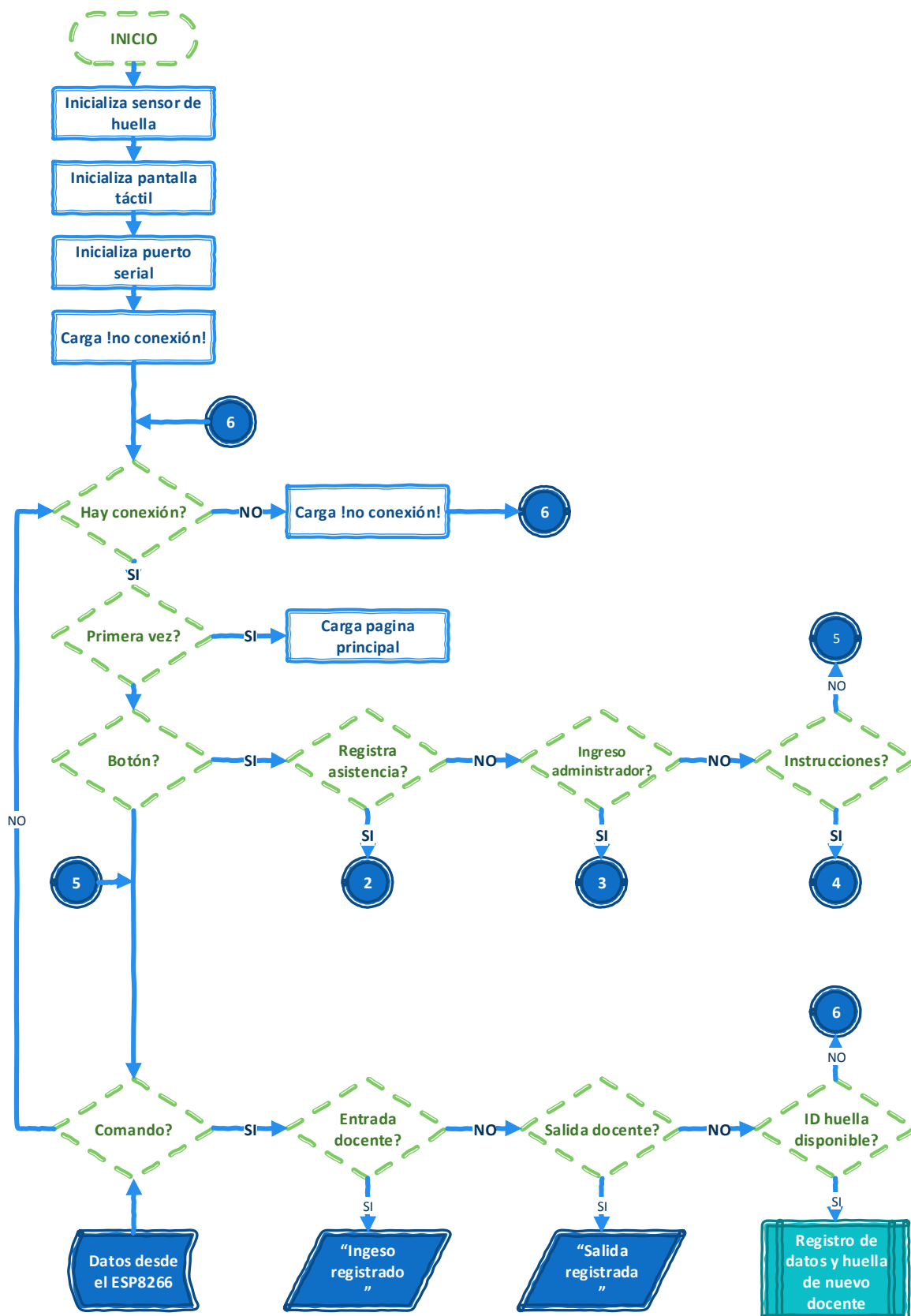
El IDE Arduino en su versión estándar no cuenta con el soporte para módulo esp8266, se tuvo que descargar el paquete del gestor de tarjetas.

```
#include <ESP8266WiFi.h>
```

Nos permite configurar el módulo para conectarlo a una red WiFi.

3.2.1.3 Diagrama de flujo programa principal Arduino Mega

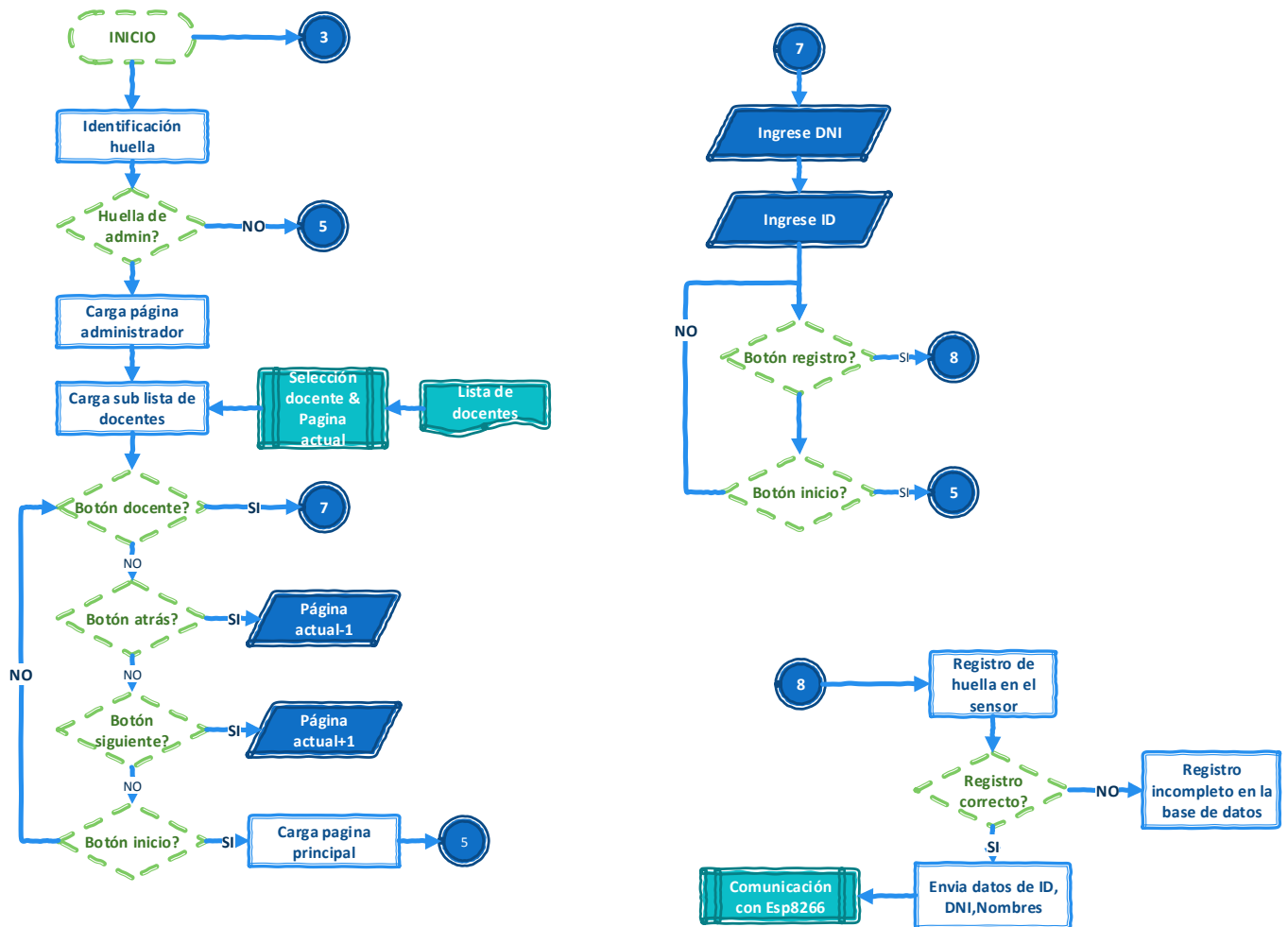
Figura 18. Diagrama de flujo programa principal Arduino Mega.



3.2.1.4 Diagrama de flujo registro en sistema de docente

El siguiente diagrama muestra los pasos a seguir para el registro de docentes tanto en el sensor como en la base de datos.

Figura 19. Diagrama de flujo registro en sistema de docente.

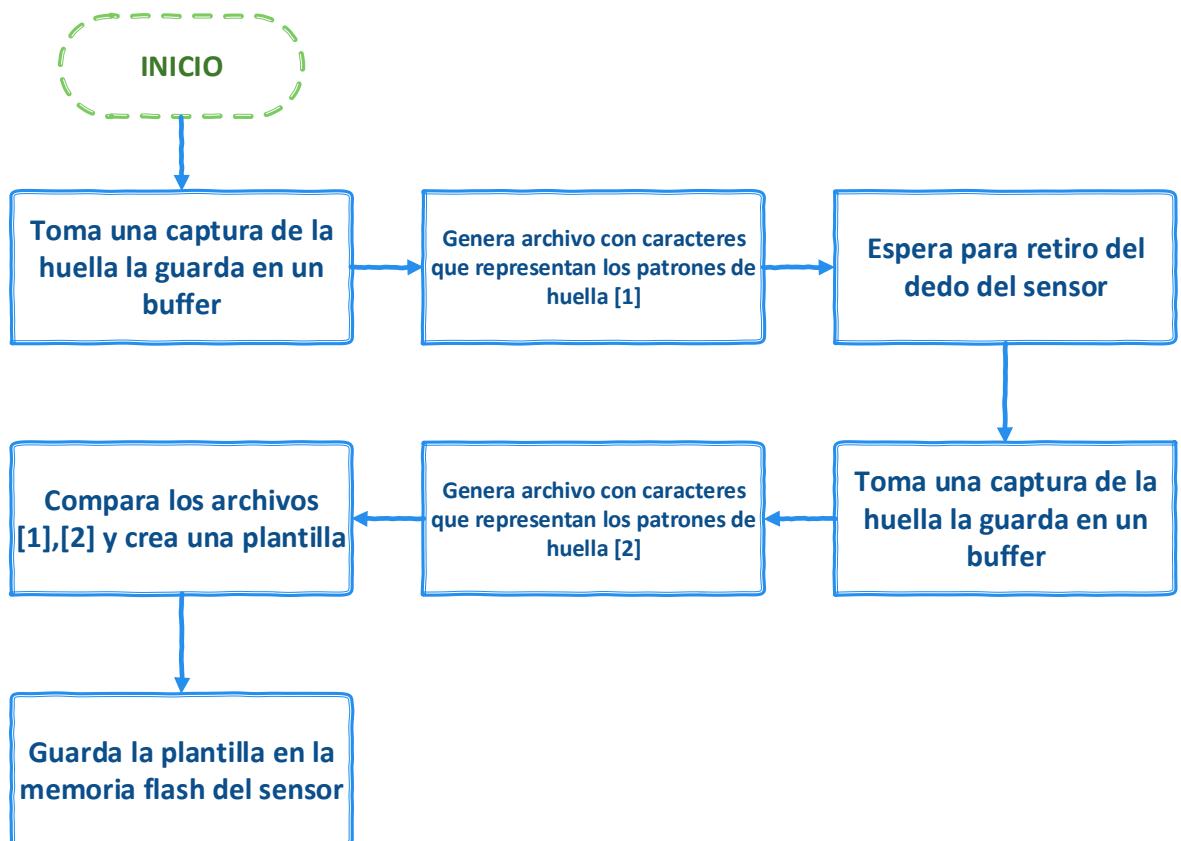


Fuente: Propia de los autores

3.2.1.5 Diagrama de flujo registro de huella en sensor

Diagrama de bloques sobre el funcionamiento interno del sensor de huella para el registro de un usuario nuevo.

Figura 20. Diagrama de flujo registro de huella en sensor.

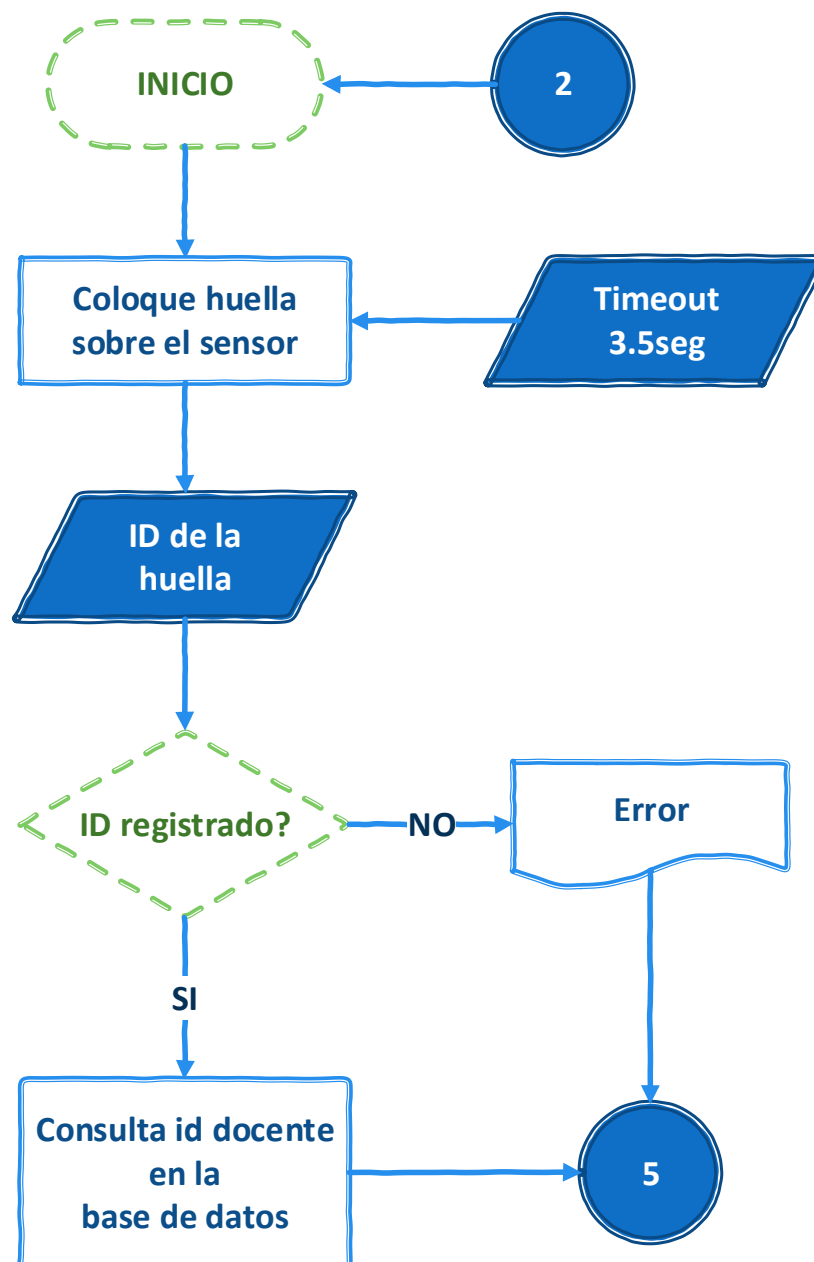


Fuente: Propia de los autores

3.2.1.6 Diagrama de flujo registro de asistencia Arduino

Diagrama de flujo para el registro de asistencia del personal docente en la base de datos, después de detectar una huella el sistema cuenta con un timeout de 3.5 segundos para que otro docente introduzca su huella esto con la intención de mitigar la formación de posibles colas.

Figura 21. Diagrama de flujo registro de asistencia Arduino.

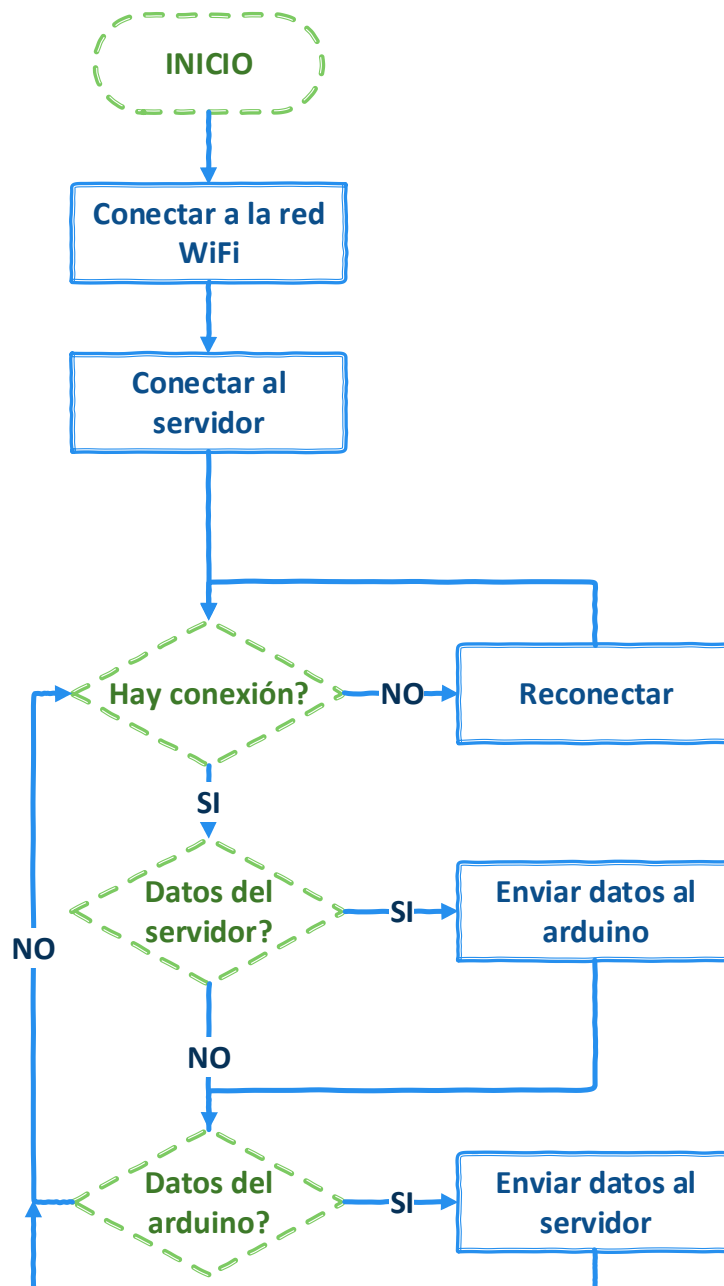


Fuente: Propia de los autores

3.2.1.7 Diagrama de flujo comunicación ESP8266

Diagrama que muestra el funcionamiento del dispositivo wifi, el cual primero intenta conectarse a la red y luego sirve de puente para la comunicación entre el módulo y la base de datos.

Figura 22. Diagrama de flujo comunicación ESP8266.



Fuente: Propia de los autores

3.2.2 Programación del software de aplicación

3.2.2.1 Visual Basic

Es un lenguaje de programación dirigido por eventos, está diseñado para aplicaciones orientadas a objetos.

El compilador genera un código el cual para funcionar necesita de una o varias librerías DLL, algunas son básicas y fundamentales para que funcione correctamente otras dan un valor agregado permitiendo acceder a rutinas del sistema operativo.

Elementos básicos:

Panel de control: Contiene diferentes elementos gráficos, Etiquetas, Cuadros de texto, Botones, Imágenes, etc.

Ventanas del proyecto: Explorador de proyectos, panel de propiedades.

Ventana de salida: Sirve para depurar el programa se pueden ver los errores, mensajes.

3.2.2.2 Librerías del Visual Basic usadas en el sistema

Mysql.data: Permite interactuar con bases de datos MySQL, crear, editar y borrar tablas.

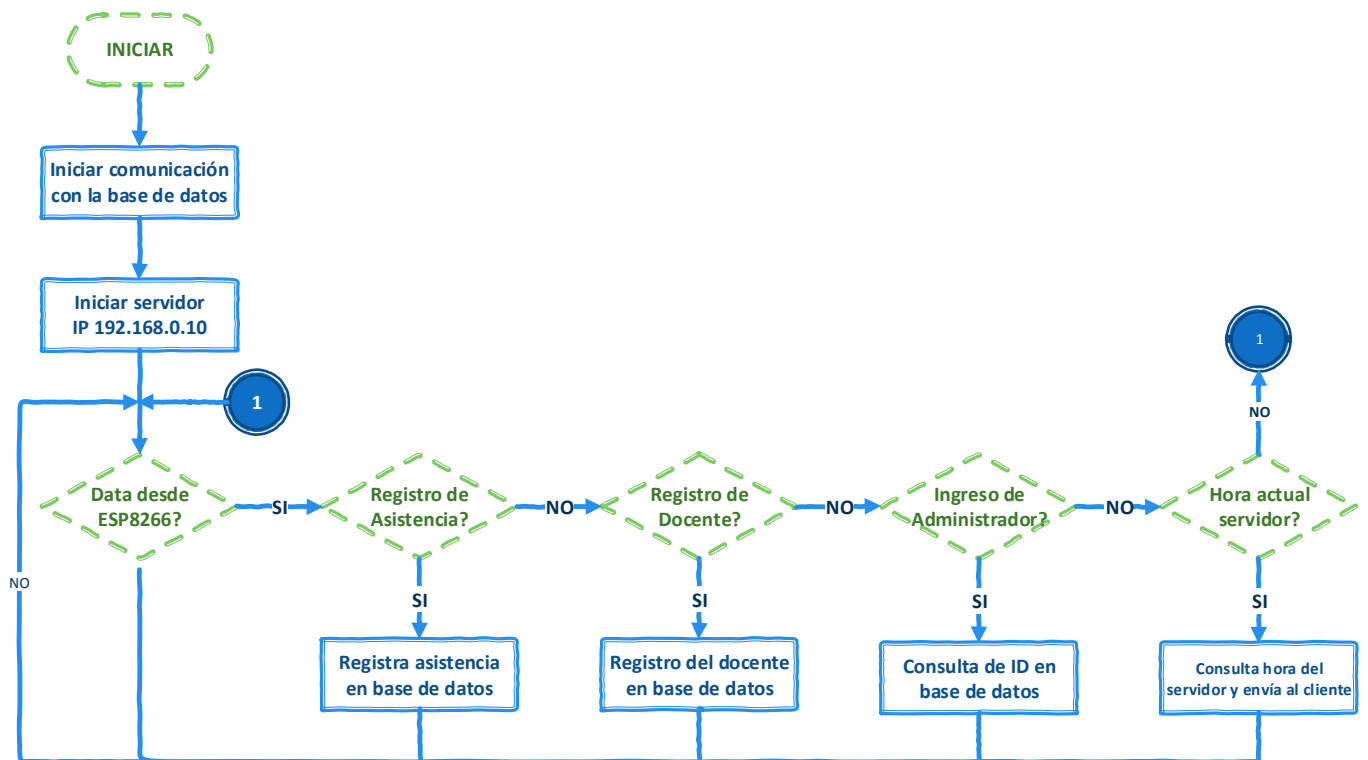
iTextSharp.text.pdf: Permite crear, editar, guardar y borrar documentos PDF.

System.net: Proporciona una interfaz fácil para los protocolos utilizados en telecomunicaciones actualmente.

3.2.2.3 Diagrama de flujo de la aplicación del sistema

El siguiente diagrama pertenece a la aplicación de escritorio que monitorea la base de datos, la comunicación con el módulo y genera los reportes.

Figura 23. Diagrama de flujo de la aplicación del sistema

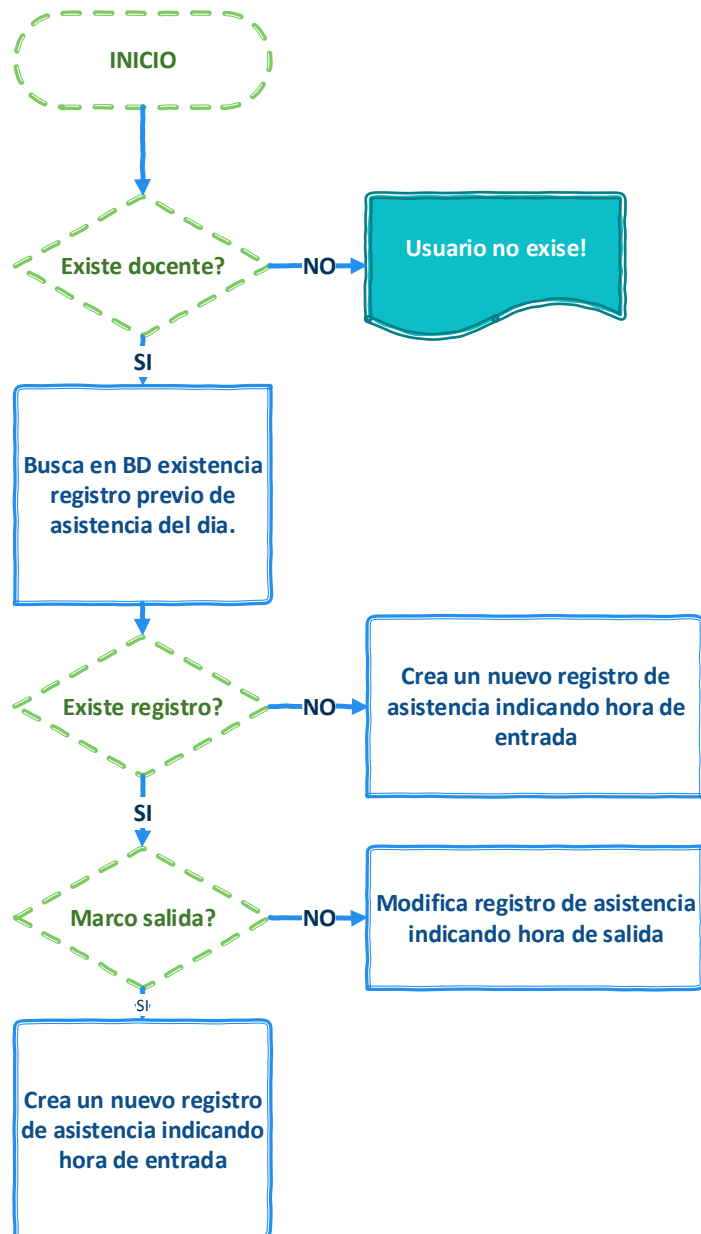


Fuente: Propia de los autores

3.2.2.4 Diagrama de flujo registro de asistencia en base de datos

Diagrama donde la aplicación registra la asistencia tanto de entrada como de salida en la base de datos siempre y cuando sea un usuario válido y cumpla con ciertos requisitos.

Figura 24. Diagrama de flujo registro de asistencia en base de datos.

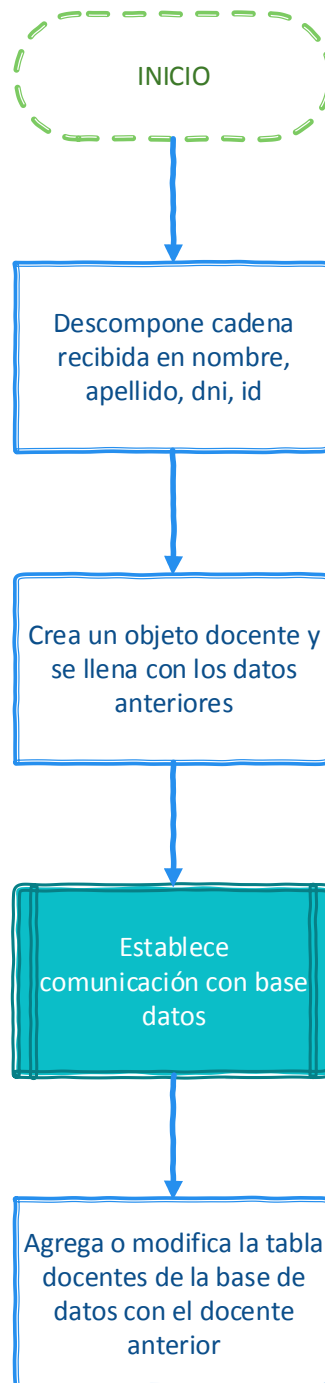


Fuente: Propia de los autores

3.2.2.5 Diagrama de flujo registro de docentes en la base de datos

Diagrama donde la aplicación reconoce a un nuevo usuario válido y lo registra en la base de datos.

Figura 25. Diagrama de flujo registro de docentes en la base de datos.

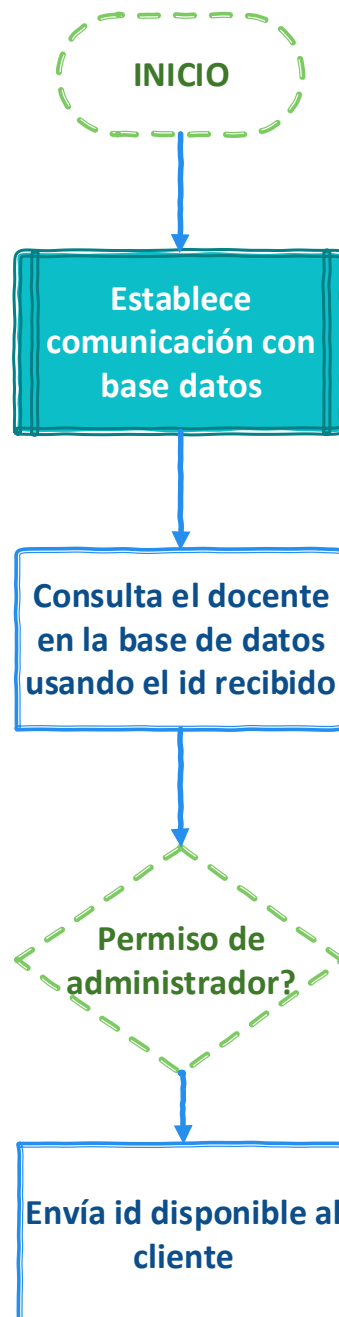


Fuente: Propia de los autores

3.2.2.6 Diagrama de flujo validación ingreso de administrador

Diagrama que realiza la consulta si el usuario tiene el permiso de administrador para así poder realizar el registro de docentes.

Figura 26. Diagrama de flujo validación ingreso de administrador.

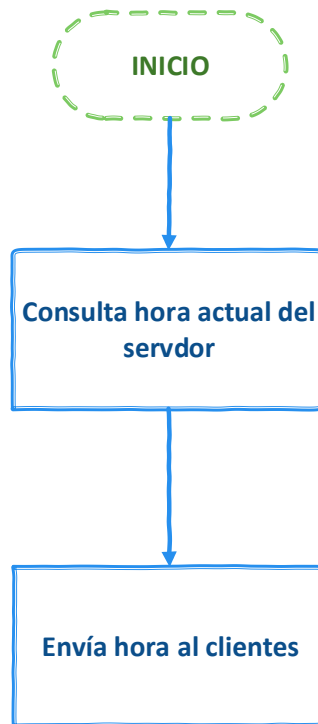


Fuente: Propia de los autores

3.2.2.7 Diagrama de flujo consulta de hora actual

Diagrama que consulta la hora actual del servidor para luego enviarla al módulo.

Figura 27. Diagrama de flujo consulta de hora actual.



Fuente: Propia de los autores

3.2.2.8 Diagrama de flujo búsqueda de asistencias en base de datos

Diagrama que busca los registros de asistencias en la base de datos para luego ser usados en la generación de reportes.

Figura 28. Diagrama de flujo búsqueda de asistencias en base de datos



Fuente: Propia de los autores.

3.3 COSTOS DEL SISTEMA

Teniendo en cuenta la descripción de los equipos que se utilizarán para la implementación del sistema, y considerando que durante el proceso se realizaron cambios para mejorar coste sin perder fiabilidad, es necesario desglosar ordenadamente los gastos de los recursos que se emplearán, es así que se tiene lo siguiente:

3.3.1 Costos de equipamiento

Tabla 6

Costo de equipamiento.

| Equipos | Cantidad | Costo unidad | Costo total |
|---------|----------|--------------|-------------|
| Laptop | 1 | S/ 2500 | S/ 2500 |
| Router | 1 | S/ 70 | S/ 70 |
| | | | S/ 2570 |

Fuente: Propia de los autores.

3.3.2 Costos de componentes electrónicos

Tabla 7

Costos de componentes electrónicos.

| Productos | Cantidad | Costo unidad | Costo total |
|-----------------------|----------|--------------|-------------|
| Pantalla táctil 7" | 1 | S/ 170.00 | S/ 170.00 |
| Esp8266 wifi | 1 | S/ 14.00 | S/ 14.00 |
| Tarjeta Arduino Mega | 1 | S/ 70.00 | S/ 70.00 |
| Sensor huella digital | 1 | S/ 170.00 | S/ 170.00 |

| | | | |
|------------------------------|---|----------|-----------|
| Fuente DC 5v-3A | 1 | S/ 20.00 | S/ 20.00 |
| Shield CTE LCD TFT | 1 | S/ 30.00 | S/ 30.00 |
| Resistencias | 8 | S/ 0.10 | S/ 0.80 |
| Capacitores | 6 | S/ 0.20 | S/ 1.20 |
| Cristal de cuarzo | 1 | S/ 1.50 | S/ 1.50 |
| Buzzer | 1 | S/ 2.00 | S/ 2.00 |
| Fusible(2A) | 1 | S/ 1.50 | S/ 1.50 |
| Portafusible | 1 | S/ 1.50 | S/ 1.50 |
| Interruptor neón | 1 | S/ 2.00 | S/ 2.00 |
| Transistor | 1 | S/ 0.30 | S/ 0.30 |
| Reloj ds1307ic | 1 | S/ 4.50 | S/ 4.50 |
| Regulador 3.3v | 1 | S/ 2.00 | S/ 2.00 |
| Portapila | 1 | S/ 1.00 | S/ 1.00 |
| Pila 3v | 1 | S/ 5.00 | S/ 5.00 |
| Conectores hembra(40 pines) | 3 | S/ 1.50 | S/ 4.50 |
| PCB fibra de vidrio(16x25cm) | 1 | S/ 16.00 | S/ 16.00 |
| Total | | | S/ 515.80 |

Fuente: Propia de los autores.

3.3.3 Costos del diseño e implementación

Tabla 8

Costos del diseño e implementación.

| Diseño | Elementos | Costo |
|---|-------------|------------|
| Diseño y desarrollo de aplicación de escritorio | Visual, SQL | S/ 2880.00 |
| Programación de microcontrolador de módulo | Arduino | S/ 1440.00 |
| Diseño de PCB para el módulo | Isis, Ares | S/ 720.00 |
| TOTAL | | S/ 5040.00 |

Fuente: Propia de los autores.

3.3.4 Costo del sistema general

Tabla 9

Costo del sistema general.

| Elementos | Costo total |
|------------------------------------|-------------|
| Costo de equipamiento | S/ 2570.00 |
| Costo de componentes electrónicos | S/ 515.80 |
| Costo del diseño de implementación | S/ 5040.00 |
| TOTAL | S/ 8125.80 |

Fuente: Propia de los autores.

CAPITULO IV

4.RESULTADOS

4.1 Implementación del prototipo

4.1.1 Verificación del estado de elementos principales

4.1.1.1 Verificación de conexión wifi con módulo esp8266

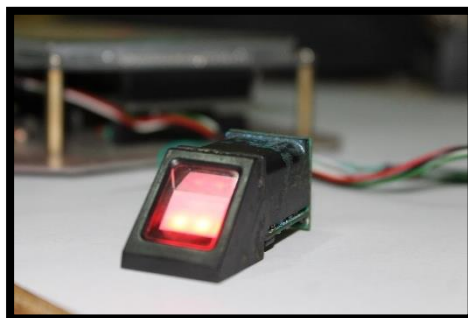
Para la verificación del módulo esp8266 se hizo uso de la librería “ESP8266WiFi.h” y del sketch de ejemplo “WiFiClient”, en este básicamente se le asigna una IP estática al módulo luego inicializamos la conexión con la red wifi.

Posteriormente configuramos e iniciamos la conexión con el servidor, si esto es correcto se comprueba que se puede realizar la lectura y escritura con el servidor también se comprobó que la conexión sea estable y no se perdiera con el paso del tiempo.

4.1.1.2 Verificación del estado del sensor de huella

Para la verificación del correcto funcionamiento del sensor de huella se descargó la librería opensource Adafruit fingerprint.h de la web Adafruit.com, esta librería contiene 6 sketch de ejemplo blank, delete, enroll, fingerprint, leo_passthru, show_finger_templates.

Figura 29. Prueba funcionamiento sensor de huella



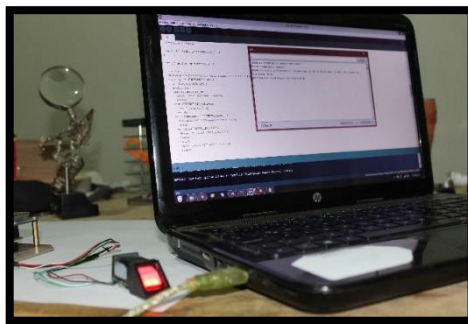
Fuente: Propia de los autores

Prueba 1: grabar huella en sensor

Para esto primero se modificó el ejemplo a nuestro diseño del hardware, originalmente el ejemplo hace uso de la librería `software_serial.h` la cual crea un puerto serial por software y ocupa los pines digitales 2, 3 sin embargo en nuestro diseño hacemos uso de un puerto serial propio del arduino mega (`Serial1`) .

Se cargó el ejemplo `enroll` al arduino mega, con esto verificamos la correcta conexión y comunicación del arduino con el sensor, conjuntamente probamos que el sensor sea capaz de guardar las huellas en la dirección especificada.

Figura 30. Test conexión Arduino – sensor de huella



Fuente: Propia de los autores

4.1.1.3 Calibración de la pantalla táctil

Para el correcto funcionamiento de la pantalla táctil tuvimos que calibrarla, para lo cual la librería URTouch nos proporciona el ejemplo “URTouch_Calibration”, procedimos a cargar dicho ejemplo y con ayuda de un objeto punta fina seguimos las indicaciones.

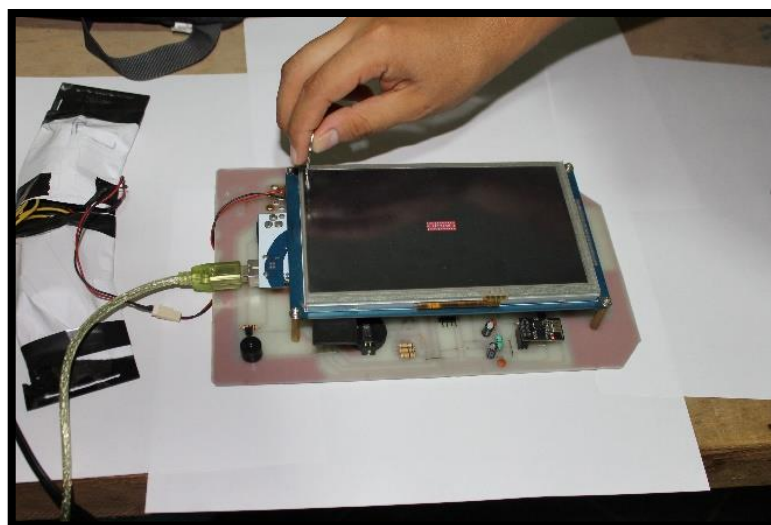
Figura 31. Calibración pantalla táctil 1



Fuente: Propia de los autores

Presionamos con el objeto en las marcas [x] que aparecen en la pantalla.

Figura 32. Calibración pantalla táctil 2



Fuente: Propia de los autores

Una vez terminado guardamos los nuevos valores x, y, z para escribirlos en el archivo de calibración (URTouChCD.h).

Figura 33. Calibración pantalla táctil 3



Fuente: Propia de los autores

4.1.1.4 Verificación de funcionamiento del buzzer

Para la verificación del correcto funcionamiento del buzzer, se hizo uso de la librería “Tone.h” la cual viene incluida a la hora de compilar el sketch.

Se hace uso de la función tone (pin, frecuencia, duración), para la selección de la frecuencia se hizo varias pruebas llegando a la conclusión que un sonido agudo, corto y fuerte sería la mejor alternativa para avisar al usuario.

Se optó por la función tone ya que hace uso del “Timer 1” para generar la frecuencia deseada lo cual permite que no se interrumpa la ejecución del loop principal.

4.1.1.5 Verificación de rtc clock

Primero usamos el sketch “i2c scanner” de la librería “wire.h”, para buscar qué dispositivos estaban conectados al bus i2c del Arduino, en este caso nos devolvió la dirección del chip ds1307.

Luego probamos el funcionamiento del RTC, usamos la función setTime para inicializar la fecha y hora del dispositivo, luego mediante el puerto Serial validamos que dichos valores sean correctos.

En la versión final del proyecto cada vez que ocurre una reconexión se hace una consulta sobre la hora del servidor y se le pasa como parámetro a la función setTime del chip ds1307 para evitar desfases de hora indeseados.

4.1.2 Ensamblaje del módulo electrónico

La primera versión que se realizó del prototipo tenía estructura de material mdf, una pantalla LCD de 16x2, un teclado numérico y una interfaz ethernet.

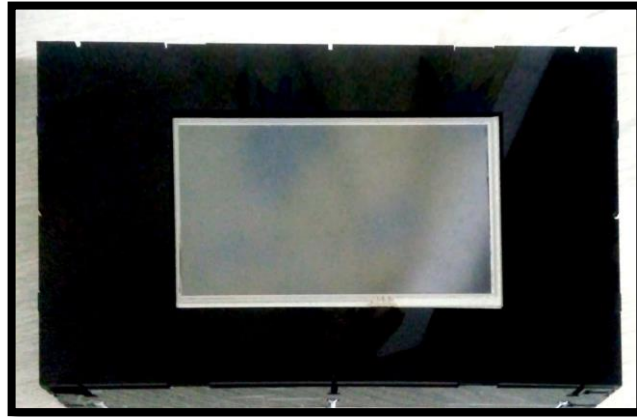
Figura 34. Prototipo 1



Fuente: Propia de los autores

La versión final tiene un diseño diferente cuenta con estructura hecha de acrílico, una pantalla lcd touch de 7" y una conexión wifi.

Figura 35. Versión final



Fuente: Propia de los autores

4.2 Funcionamiento del prototipo

4.2.1 Interfaz gráfico de la pantalla lcd/tft

Este módulo permite al usuario almacenar su registro de asistencia en la base de datos. Mediante la pantalla táctil realiza funciones de guardar nuevas huellas de docente y mostrar la fecha, hora.

A continuación, describiremos la página que muestra el interfaz en el momento que se conecta a la aplicación de escritorio.

Figura 36. Interfaz gráfica prototipo



Fuente: Propia de los autores

4.2.1.1 Verificación de la pantalla instrucciones

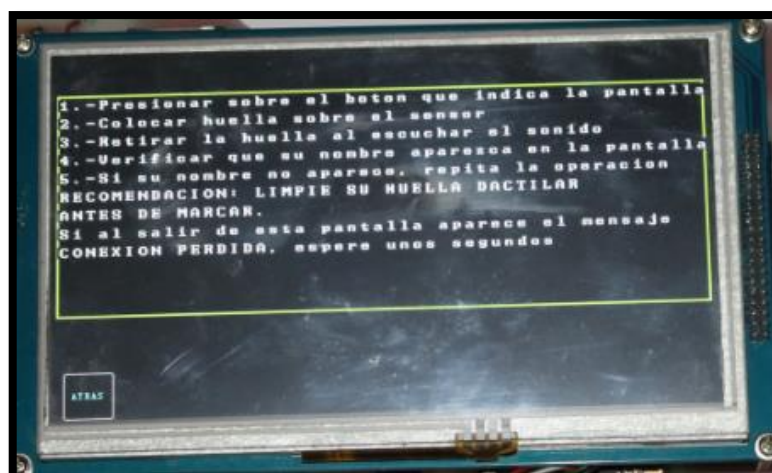
En la figura 41 podemos observar que al presionar sobre el botón táctil “Instrucciones”, nos lleva a la siguiente pantalla (figura 42), en la cual se muestran los pasos a seguir del usuario, para registrar su huella; indicaciones sobre cualquier evento que se muestre en pantalla, y lo que debe hacer.

Figura 37. Botón instrucciones pantalla principal



Fuente: Propia de los autores

Figura 38. Pantalla instrucciones

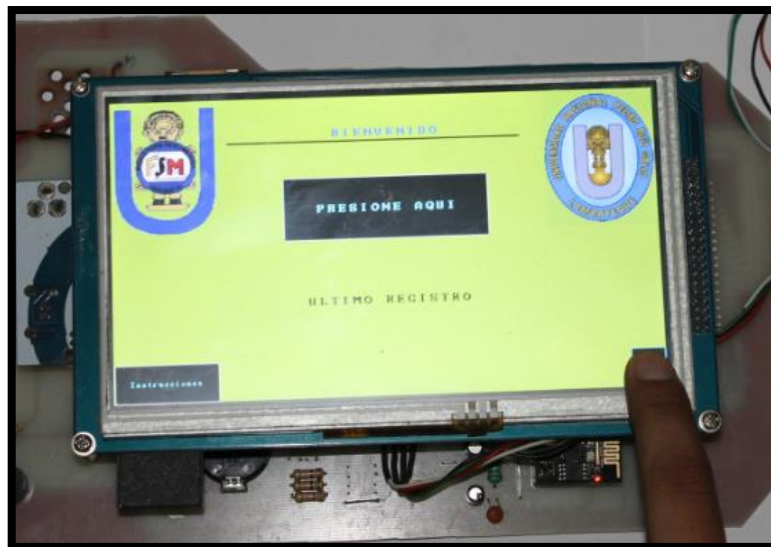


Fuente: Propia de los autores

4.2.1.2 Verificación del proceso de registro de docentes

El módulo cuenta con una tarjeta SD, en la que almacena los nombres de docentes de la facultad, esto facilita el registro de nuevo usuario, además permite actualizar la lista de los mismos, cuando sea necesario; en la figura 43 se muestra el primer paso para el registro, el cuál es ingresar la huella de administrador.

Figura 39. Botón registro pantalla principal



Fuente: Propia de los autores

Figura 40. Esperando huella de administrador



Fuente: Propia de los autores

Figura 41. Identificando huella



Fuente: Propia de los autores

Una vez validado el ingreso del administrador se mostrarán páginas con los nombres de los docentes almacenados en la SD (figura 46).

Figura 42. Lista de docentes a registrar



Fuente: Propia de los autores

Lo siguiente es seleccionar el docente a registrar, y luego presionar “OK”, ver figura 47.

Figura 43. Formulario docente a registrar



Fuente: Propia de los autores

Llenamos los campos faltantes como “DNI” o “ID”,

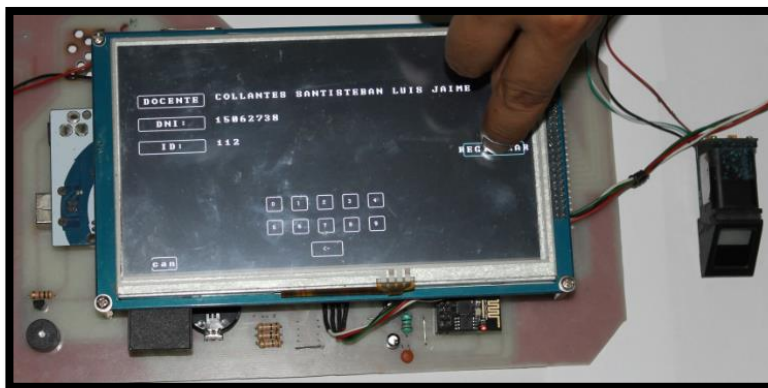
Figura 44. Llenado de formulario



Fuente: Propia de los autores

y procedemos presionando sobre “REGISTRAR”

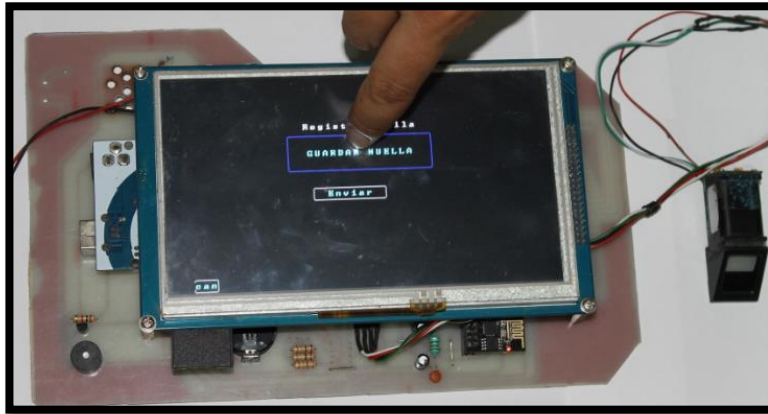
Figura 45. Botón registrar de formulario



Fuente: Propia de los autores

A continuación, nos aparecerá una nueva página, en la que podemos registrar nuestra huella, o cancelar el proceso

Figura 46. Botón registrar huella



Fuente: Propia de los autores

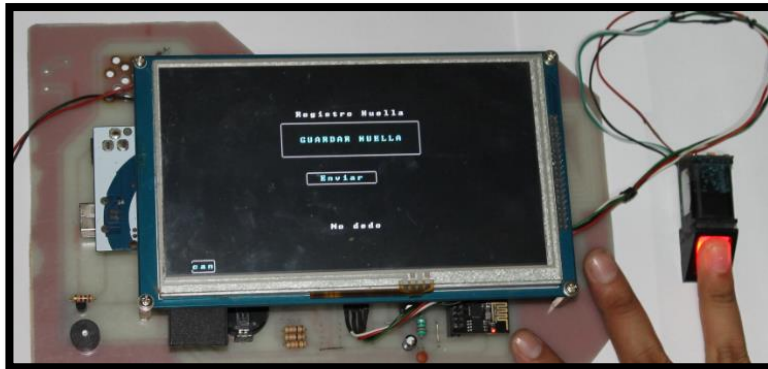
Ahora sólo queda colocar nuestra huella dactilar sobre el sensor que enciende con luz roja

Figura 47. Esperando huella docente



Fuente: Propia de los autores

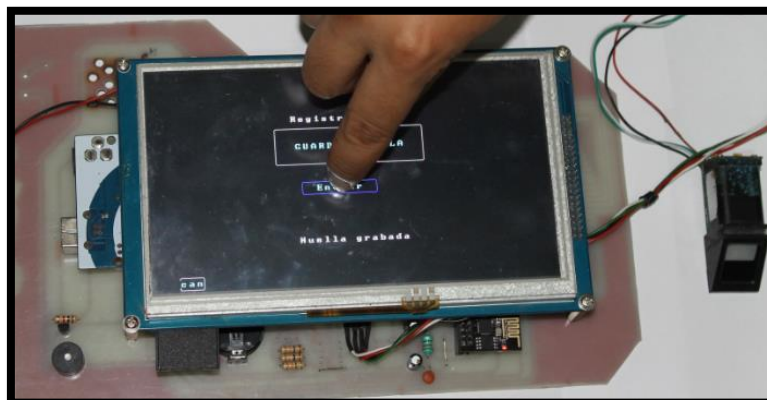
Figura 48. Registrando huella en sensor



Fuente: Propia de los autores

Finalmente presionamos sobre el botón “Enviar” para transmitir los datos mediante el ESP8266 y con eso concluye el proceso de grabación de un nuevo usuario.

Figura 49. Botón enviar registro de docente



Fuente: Propia de los autores

4.2.1.3 Verificación del proceso de identificación de docentes

En la pantalla principal del módulo, existen tres botones táctiles, de los cuales ya hemos hablado y explicado la función de los dos situados en la parte inferior izquierda y derecha, el tercer botón y el más resaltante “PRESIONE AQUI” (ver figura 54) nos permite realizar la función principal, la cuál es el registrar la asistencia laboral.

Al presionar sobre este, nos mostrará un mensaje en pantalla, indicando que el sensor está listo para ser usado.

Figura 50. Botón presione aquí pantalla principal



Fuente: Propia de los autores

A lo que prosigue colocar la huella dactilar registrada (ver figura 55), una vez hecho este paso, se escucharán dos sonidos provenientes de un “buzzer”, indicando que la captura fue exitosa, es decir sin errores.

Figura 51. Esperando huella dactilar



Fuente: Propia de los autores

Luego de esto sólo nos queda confirmar, ya que siempre queda registrada la última marcación, y podemos comprobar la hora en la parte superior.

Figura 52. Reconociendo huella



Fuente: Propia de los autores

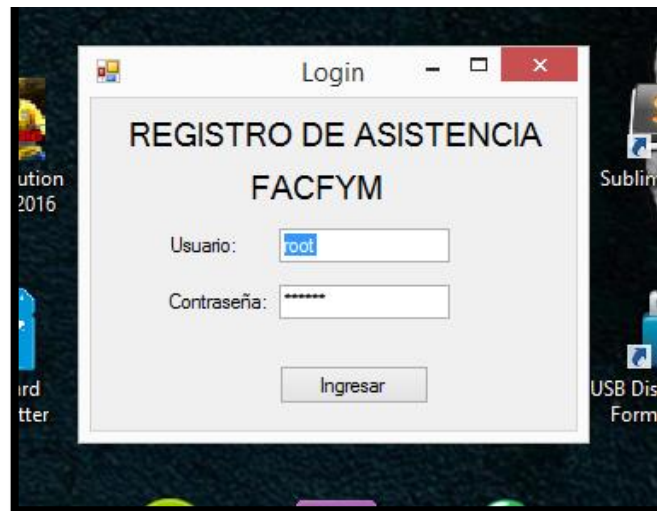
4.2.2 Interfaz de la aplicación de escritorio

La interfaz de la aplicación se realizó con un diseño amigable al usuario, está desarrollada usando Visual Studio específicamente el lenguaje Visual Basic, nos permite comunicarnos con el módulo electrónico a la vez cuenta con las funciones buscar y exportar.

4.2.2.1 Verificación de login

Al abrir la aplicación nos mostrará una ventana, donde se debe ingresar un usuario y una contraseña para tener acceso a la base de datos y ventana principal.

Figura 53. Ventana login

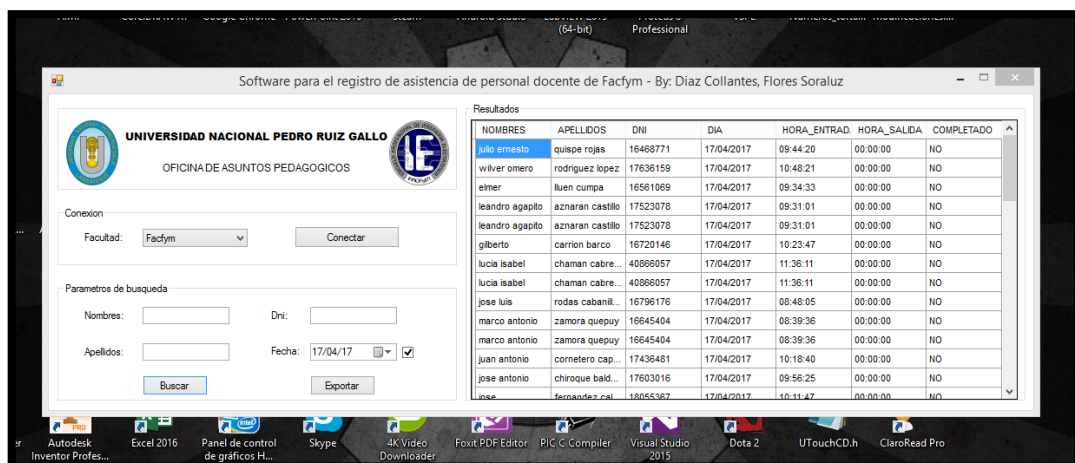


Fuente: Propia de los autores

4.2.2.2 Verificación de función búsqueda

Una vez en la ventana principal podemos realizar la búsqueda de registros de asistencia ya sea por nombres, apellidos, DNI o fecha.

Figura 54. Ventana principal




Fuente: Propia de los autores

4.2.2.3 Verificación de función exportar

La función exportar nos permite generar un archivo .pdf con el registro de asistencia solicitado para luego ser guardado o impreso según se requiera.

Figura 55. PDF reporte de asistencias



UNIVERSIDAD NACIONAL PEDRO RUIZ GALLO
OFICINA DE ASUNTOS PEDAGÓGICOS

Reporte de asistencia

Lista de asistencias martes, 23 de octubre de 2018 08:04:20 p.m.

| Nombres | Apellidos | Dni | Día | H.Entrada | H.Salida | Completado |
|-----------------|--------------------|----------|------------|-----------|----------|------------|
| julio ernesto | quispe rolas | 16468771 | 17/04/2017 | 09:44:20 | 00:00:00 | NO |
| wiliver omero | rodriguez lopez | 17636159 | 17/04/2017 | 10:48:21 | 00:00:00 | NO |
| eimer | luen cumpa | 16561069 | 17/04/2017 | 09:34:33 | 00:00:00 | NO |
| leandro agapito | aznaran castillo | 17523078 | 17/04/2017 | 09:31:01 | 00:00:00 | NO |
| leandro agapito | aznaran castillo | 17523078 | 17/04/2017 | 09:31:01 | 00:00:00 | NO |
| gilberto | carrion barco | 16720146 | 17/04/2017 | 10:23:47 | 00:00:00 | NO |
| lucia isabel | chaman cabrera | 40866057 | 17/04/2017 | 11:36:11 | 00:00:00 | NO |
| lucia isabel | chaman cabrera | 40866057 | 17/04/2017 | 11:36:11 | 00:00:00 | NO |
| jose luis | rodas cabanillas | 16796176 | 17/04/2017 | 08:48:05 | 00:00:00 | NO |
| marco antonio | zamora quepuy | 16645404 | 17/04/2017 | 08:39:36 | 00:00:00 | NO |
| marco antonio | zamora quepuy | 16645404 | 17/04/2017 | 08:39:36 | 00:00:00 | NO |
| juan antonio | cornetero capitán | 17436481 | 17/04/2017 | 10:18:40 | 00:00:00 | NO |
| jose antonio | chiroque baldera | 17603016 | 17/04/2017 | 09:56:25 | 00:00:00 | NO |
| jose | fernandez calderon | 18055367 | 17/04/2017 | 10:11:47 | 00:00:00 | NO |
| fredie | trujillo custodio | 16713673 | 17/04/2017 | 08:47:13 | 00:00:00 | NO |
| ruben esleban | burga barboza | 16761647 | 17/04/2017 | 08:47:28 | 00:00:00 | NO |
| carlos | obitas vera | 3701836 | 17/04/2017 | 09:09:50 | 00:00:00 | NO |
| eulatio | altamirano flores | 17623262 | 17/04/2017 | 08:43:21 | 00:00:00 | NO |
| jaime | abanto castillo | 16489936 | 17/04/2017 | 08:12:14 | 08:12:39 | SI |
| oscar | romero uochelly | 41812294 | 17/04/2017 | 08:16:04 | 12:19:14 | SI |
| darwin | diaz delgado | 16790017 | 17/04/2017 | 08:24:09 | 00:00:00 | NO |
| diana | castro carmen | 21520450 | 17/04/2017 | 08:40:45 | 00:00:00 | NO |
| amulto | cleza ramos | 16571279 | 17/04/2017 | 08:44:32 | 00:00:00 | NO |
| mardo | gonzales herrera | 16756765 | 17/04/2017 | 08:45:48 | 00:00:00 | NO |

Fuente: Propia de los autores

4.3 Datos de Consumo del módulo

Realizando las mediciones de corriente de consumo del módulo durante su operación, se registraron los siguientes datos:

Tabla 10. *Datos de consumo*

| Corriente | Voltaje | Potencia | Consumo |
|-----------|---------|------------|---------|
| 730 mA | 5 Vdc | 3.65 Watts | S/ 0.40 |

CAPITULO V

5.CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se construyó un módulo electrónico con una interfaz para el registro y almacenamiento de las huellas de los docentes, satisfaciendo el objetivo principal propuesto.
- El sistema propuesto cumplió con el objetivo de mejorar el control de asistencia de docentes, mitigando el retardo, las colas, asegurando la veracidad de los datos.
- Los registros de asistencia fueron almacenados en la laptop y se pudieron monitorear desde la aplicación de escritorio, así como ser exportados a un archivo PDF.
- Se creó una red inalámbrica basada en el protocolo TCP/IP, para la comunicación entre el módulo y la laptop.

5.2 Recomendaciones

En el diseño de este Sistema Biométrico que busca Mejorar el control de Asistencia del Personal Docente en la FACFyM, se usaron equipos que, sin perder fiabilidad redujeran costes adicionales a esto hacemos las siguientes recomendaciones, para un sistema más robusto:

- Este sistema genera archivos en PDF listos para su impresión, en el momento de elaborar un reporte, en tal caso se recomienda registrar cada vez que un administrador (persona que tiene acceso a la base de datos) realice algún cambio, para darle mayor transparencia.
- Los datos son almacenados directamente en la base de datos mediante conexión inalámbrica, se recomienda instalar sistema UPS, para evitar registros incompletos, ante un corte de energía.
- El sensor usado es susceptible a cambios en la huella, ya sea que se encuentre sucia, o con lesiones, se recomienda usar algún tipo de doble registro para garantizar el acceso por si alguna falla.
- El sistema se puede controlar únicamente desde el servidor por un tema de seguridad, si se requiere el acceso remoto, se recomienda la implementación de asignación de permisos.
- Añadir una foto para que se muestre en pantalla cada que un usuario lo use a manera de volver al sistema más amigable.

REFERENCIAS BIBLIOGRAFICAS

[1] Florencia Ucha, 15/11/2011, Identificar, Sitio: Definición ABC, visitado en Setiembre de 2018. Disponible en:

<https://www.definicionabc.com/general/identificar.php>

[2] Carro Paz, R. y Gonzáles Gómez, D., (2013), "Identificación Automática", Mar del Plata, Argentina. Disponible en:

http://nulan.mdp.edu.ar/1612/1/07_identificacion_automatica.pdf

[3] Ortega García, J. y Fernández, F. y Coomonte Belmont, R., (2008), "Biometría y Seguridad", Madrid, España. Disponible en:

<http://catedraisdefe.etsit.upm.es/wp-content/uploads/2010/09/CUADERNO-N%C2%BA-3.pdf>

[4] ACEPROJECT, "Elecciones y Tecnología: Sistemas de Identificación", México. Disponible en:

<http://aceproject.org/main/espanol/et/et73.htm>

[5] Tolosa Borja, C. y Giz Bueno, A., (SF), "Sistemas Biométricos". Disponible en:

https://www.dsi.uclm.es/personal/MiguelFGraciani/mikicurri/Docencia%20Bioinformatica/web_BIO/Documentacion/Trabajos/Biometria/Trabajo%20Biometria.pdf

[6] Arduino. (s.f). En Wikipedia. Recuperado el 02 de octubre de 2018 de <https://es.wikipedia.org/wiki/Arduino>

[7] Introduction Arduino (s.f). Disponible en:
#<https://www.arduino.cc/en/Guide/Introduction>

[8] Arduino Products (s.f). Disponible en:
<https://www.arduino.cc/en/Main/Products>

[9] Delgado Parra, G., "Biometría". Disponible en:

https://upload.wikimedia.org/wikipedia/commons/c/ce/Articulo_gerson_delgado_congsistel.pdf

[10] Damian Pérez Valdés, (26/10/18), "Qué son las bases de datos", Disponible en: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>

- [11]Modelo entidad-relación (s.f). Disponible en:
https://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n
- [12]Crespo Martínez, L. y Candelas Herías, (1998), "Introducción a TCP/IP", España. Disponible en:
https://rua.ua.es/dspace/bitstream/10045/4328/1/Crespo_Candelas_TCP_IP.pdf
- [13]Alsina Gonzales, G., (11/02/2009), "Definición de TCP/IP", Disponible en:
<https://www.definicionabc.com/tecnologia/tcpip.php>
- [14]El modelo cliente-servidor, Herramientas web para la enseñanza de protocolos de comunicación. Disponible en:
<http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>
- [15]MySQL Workbench, MySQL, Disponible en:
<https://www.mysql.com/products/workbench/>
- [16]Visual Studio features, Visual Studio, Disponible en:
<https://www.visualstudio.com/vs/ide/>
- [17] Microsoft Visual Studio, (s.f). Disponible en:
https://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- [18]Arduino mega 2560, (s.f), Store Arduino. Disponible en:
<https://www.arduino.cc/en/Main/ArduinoBoardMega>
- [19] What is the "Internet of things", The Internet of Things with ESP8266, (s.f). Disponible en: <http://esp8266.net/>
- [20] Módulo WiFi Serial ESP-01 ESP8266, Naylamp Mechatronics, (s.f). Disponible en:
<http://www.naylampmechatronics.com/inalambrico/48-modulo-wifi-serial-esp8266.html>
- [21] Henning Karlsen, (17/05/2015), Library: UTFT, Rinky-Dink Electronics. Disponible en: <http://www.rinkydinkelectronics.com/library.php?id=51>
- [22] 7 "pulgadas TFT LCD Módulo de 800x480 SSD1963, AliExpress (s.f) Disponible en: <https://es.aliexpress.com/item/New-7-inch-TFT-LCD-module-800x480-MD070SD-Touch-PWM-For-Arduino-AVR-STM32-ARM/2031328008.html>
- [23] Adafruit Optical Fingerprint Sensor, Adafruit learning system, (s.f). Disponible en: <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-optical-fingerprint-sensor.pdf>

[24]Modelo cliente servidor, (s.f), Herramientas web para la enseñanza de protocolos de comunicación. Disponible en:
<http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>

[25] Victoria Bembibre, Definición de Router, Definición ABC, (06/01/2009).
Disponible en: <https://www.definicionabc.com/tecnologia/router.php>

ANEXOS

ANEXO 1

Programación Arduino mega

```
#include <Adafruit_Fingerprint.h>
#include <UTFT.h>
#include <URTough.h>
#include <tinyFAT.h>
#include <UTFT_tinyFAT.h>
#include <UTFT_Buttons.h>
#include <UTFT_DLB.h>

//TFT01_70
//LCD
UTFT_DLB myGLCD(CTE70, 38, 39, 40, 41);
UTFT_tinyFAT myFiles(&myGLCD);
//Variables de pantalla
char* files800[]={ "IMAGE1.RAW", "IMAGE2.RAW", "", "" }; //
800x480
int picsize_x, picsize_y;
int vTimeMensaje_1=1500;
int vPantallaActual=-1;

//Fuente de letra externa
extern uint8_t DejaVuSans24[];
extern uint8_t BigFont[];
extern uint8_t SmallFont[];
extern uint8_t Dingbats1_XL[];

//BUZZER
int pinBuzzer=A0;

//WIFI
HardwareSerial &client = Serial2;
int pinCheckConexion=8;
String vWifiData="";
boolean vwifirdy=false;
boolean vdataWifiRdy=false;
boolean vPrimeraVez=true;

//TOUCHPAD
URTough myTouch(6,5,4,3,2);
UTFT_Buttons myButtons(&myGLCD, &myTouch);

//FINGERPRINT
```

```

Adafruit_Fingerprint finger = Adafruit_Fingerprint(&Serial1);
unsigned long mx=0;

//Variables de conexión
String vHoraActual="";

int vSel_DI=0;
String vDniValue="";
String vIdValue="";

uint8_t vRegistroCorrecto=1000;

String lcd_txt="";
char vector_lcd[60];

//BOTONES
int vlastBoton=-1;
int bPresioneAqui,bInstrucciones,bAdmin;//Pagina 0
int bAtras_i;//Pagina 4
intb Docentes[10],bAtras,bAdelante,bDocente,bCancelar_reg1;
//Pagina 5
int bCancelar_reg2,bFakeDocente,bDNI,bID;//Pagina 6
int bTecladoNumerico[10],bTecladoErase,bRegistrar;//Pagina 6
int bGuardaHuella,bCancelar_reg3,bEnviarHuella;//pagina 8

//DOCENTES - Page 2
String vDocentes[10];
int vNumeroDocentes=140;
int vPagDocenteActual=1;
String vDocenteSeleccionado="";
String vDocenteSeleccionado_show="";
int vCurrentPagina=1;
int vIdDisponible;
int vIdSeleccionada=-1;

unsigned long m_leeclave=0;

void setup() {
//SERIAL
Serial.begin(115200);

```

```

//WIFI
InitWifi();

//SDCARD
InitSD();
//LCD
InitLcd();

//TOUCH
InitTactil();
InitButtons();
//HUELLA
InitHuella();
//Buzzer
InitBuzzer();
//Load pagina 0
LoadConexionPerdida();
Serial.println("REINICIO!!");
//Establece Conexion
while(1){
    if(ConexionWifi()){
        break;
    }
}

void loop() {
    if(ConexionWifi()==true){
        if(vPrimeraVez==true){
            Btns_PresioneAqui(true);
            vPrimeraVez=false;
            Serial.println("Primera Vez");
        }
        if(myTouch.dataAvailable()!=0){
            BotonPresionado();
        }
        if(client.available()>0){
            vwifirdy=LeeWifi();
        }
        if(vwifirdy==true){
            vwifirdy=false;

```



```

AnalizaDatoRecibido(vWifiData);
}
}else{
LoadConexionPerdida();
vPrimeraVez=true;
}
}
void AnalizaDatoRecibido(String ptext){
int pos_LNM=ptext.indexOf("#LNM");
int pos_LST1=ptext.indexOf("#LST1");
int pos_LST2=ptext.indexOf("#LST2");
int pos_LHR=ptext.indexOf("#LHR");
if(pos_LNM>=0){
int pos_in=ptext.indexOf("IN");
int pos_out=ptext.indexOf("OT");
int pos_null=ptext.indexOf("Null");
int pos_sep=ptext.indexOf("-");
int pos_pong=ptext.indexOf("$");
if(pos_in>0){
String Nombres=ptext.substring(pos_LNM+6,pos_sep);
String Apellidos=ptext.substring(pos_sep+1,pos_pong);
String U_Nombres=ExtraelNombre(Nombres);
String U_Apellidos=ExtraelNombre(Apellidos);
String U_Hora=ptext.substring(pos_pong+1);
U_Hora=U_Hora.substring(0,U_Hora.length()-1);
LCD("INGRESO H:"+U_Hora,U_Nombres+" "+U_Apellidos);
}else if(pos_out>0){
String Nombres=ptext.substring(pos_LNM+6,pos_sep);
String Apellidos=ptext.substring(pos_sep+1,pos_pong);
String U_Nombres=ExtraelNombre(Nombres);
String U_Apellidos=ExtraelNombre(Apellidos);
String U_Hora=ptext.substring(pos_pong+1);
U_Hora=U_Hora.substring(0,U_Hora.length()-1);
LCD("SALIDA H:"+U_Hora,U_Nombres+" "+U_Apellidos);
}else if(pos_null>0){
LoadUsuarioNoRegistrado();
delay(1500);
LoadPresioneAqui(true);
}
}
}

```

```
}else if(pos_LST1>=0){  
}else if(pos_LST2>=0){  
int pos_id=ptext.indexOf("ID");  
String str_id=ptext.substring(pos_id+2);  
vIdDisponible=str_id.toInt();  
delay(5);  
LoadRegistro_1();  
}else if(pos_LHR>=0){  
vHoraActual=ptext.substring(pos_LHR+4,pos_LHR+9);  
}else{  
delay(1000);  
}  
}
```

```
void InitBuzzer() {  
pinMode(pinBuzzer,OUTPUT);  
}  
void beep2() {  
tone(pinBuzzer,3333,200);  
delay(350);  
tone(pinBuzzer,3333,200);  
}  
void beep1() {  
tone(pinBuzzer,3333,200);  
}
```

```

int LeeHuella(){
int ID_read=-2;
mx=millis();
do{
ID_read=getFingerprintIDez();
delay(50);
if(ID_read!=-2){
break;
}
}while((millis()-mx)<3500);
Serial.println("Huella Leida: "+String(ID_read));
return ID_read;
}
void InitHuella(){
finger.begin(57600);
}
int getFingerprintIDez() {
uint8_t p = finger.getImage();
if (p != FINGERPRINT_OK) return -2;
p = finger.image2Tz();
if (p != FINGERPRINT_OK) return -1;
p = finger.fingerFastSearch();
if (p != FINGERPRINT_OK) return -1;
//BEEP - TONE
beep2();
return finger.fingerID;
}
uint8_t getFingerprintEnroll(int pid) {
//LCD("COLOQUE SU DEDO","EN EL SENSOR");
int p = -1;
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
//LCD("Imagen tomada");

```

```

break;
case FINGERPRINT_NOFINGER:
break;
case FINGERPRINT_PACKETRECEIVEERR:
//LCD("Error comunicacion");
break;
case FINGERPRINT_IMAGEFAIL:
//LCD("Error imagen");
break;
default:
//LCD("Error desconocido");
break;
}
}
// OK success!
p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
//LCD("Imagen convertida");
break;
case FINGERPRINT_IMAGEMESS:
//LCD("Imagen confusa, limpie huella y sensor");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
//LCD("Error comunicacion");
return p;
case FINGERPRINT_FEATUREFAIL:
//LCD("No se encontro patrones en huella");
return p;
case FINGERPRINT_INVALIDIMAGE:
//LCD("No se encontro patrones en huella");
return p;
default:
//LCD("Error desconocido");
return p;
}
//LCD("Retire dedo");
//delay(2000);
//BEEP-TONE

```

```

beep2();
p = 0;
while (p != FINGERPRINT_NOFINGER) {
p = finger.getImage();
}
p = -1;
//LCD("Coloque dedo denuevo");
while (p != FINGERPRINT_OK) {
p = finger.getImage();
switch (p) {
case FINGERPRINT_OK:
LCD("Imagen tomada");
break;
case FINGERPRINT_NOFINGER:
LCD("No dedo");
break;
case FINGERPRINT_PACKETRECEIVEERR:
LCD("Error comunicacion");
break;
case FINGERPRINT_IMAGEFAIL:
LCD("Error imagen");
break;
default:
LCD("Error desconocido");
break;
}
}
// OK success!
p = finger.image2Tz(2);
switch (p) {
case FINGERPRINT_OK:
LCD("Iamgen convertida");
break;
case FINGERPRINT_IMAGEMESS:
LCD("Imagen confusa, limpie huella y sensor");
return p;
case FINGERPRINT_PACKETRECEIVEERR:
LCD("Error comunicacion");
return p;
}

```

```

case FINGERPRINT_FEATUREFAIL:
LCD("No se encontro patrones en huella");
return p;
case FINGERPRINT_INVALIDIMAGE:
LCD("No se encontro patrones en huella");
return p;
default:
LCD("Error desconocido");
return p;
}
// OK converted!
p = finger.createModel();
if (p == FINGERPRINT_OK) {
LCD("Huella coinciden");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
LCD("Error de comunicacion");
return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
LCD("Huella no coinciden");
return p;
} else {
LCD("Error desconocido");
return p;
}
p = finger.storeModel(pid);
if (p == FINGERPRINT_OK) {
LCD("Huella grabada");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
//LCD("Error de comunicacion");
return p;
} else if (p == FINGERPRINT_BADLOCATION) {
LCD("No se pudo grabar direccion huella");
return p;
} else if (p == FINGERPRINT_FLASHERR) {
LCD("Erro al grabar en memoria flash");
return p;
} else {
LCD("Error desconocido");
return p;
}
beep2();
}

```

```

void LCD(String t1,String t2){
  lcd_txt=t1+" "+t2;
  int string2=lcd_txt.length();
  lcd_txt.toCharArray(vector_lcd,string2+1);
  myGLCD.setColor(0,0,0);
  myGLCD.setBackColor(192,227,85);
  writing("
",16,'C',370);
  writing(vector_lcd,16,'C',370);
}
void LCD(String t1){
  int string2=t1.length();
  t1.toCharArray(vector_lcd,string2+1);
  myGLCD.setColor(255,255,255);
  myGLCD.setBackColor(0,0,0);
  writing("
",16,'C',370);
  writing(vector_lcd,16,'C',370);
  Serial.println("-->");
  Serial.println(t1);
  Serial.println("<--");
}
void LCD(int fila,String t1){
}
String VerHoraActual(){
  if(!(vHoraActual.equals(""))){
    String vHora=" HORA:"+vHoraActual;
    return vHora;
  }
  return " ";
}
void InitLcd(){
  myGLCD.InitLCD();
  myGLCD.clrScr();
  delay(250);
}

```



```

void CargaDocentesPag(int pPag){
file.openFile("TEXTO.TXT",FILEMODE_TEXT_READ);
myButtons.setTextFont(SmallFont);
int j=(pPag-1)*10;
int k=0;
for(int i=0;i<vNumeroDocentes;i++){
char vbuf[40];
file.readLn(vbuf,40);
if(i>=j && i<(pPag*10)){
String vdoc=Char2String(vbuf,40);
vDocentes[k]=vdoc;
vdoc.replace("@"," ");
vdoc.toCharArray(vbuf,40);
if((j+k)<vNumeroDocentes){
myButtons.relabelButton(bDocentes[k],vbuf,true);
}else{
myButtons.relabelButton(bDocentes[k]," ",true);
}
k++;
if(k>9){
break;
}
}
}
file.closeFile();
}
void InitSD(){
byte res;
res=file.initFAT();
if (res!=NO_ERROR)
{
while (true) {};
}
}
String Char2String(char vec[],int n){
String vAux="";
for(int i=0;i<n;i++){
vAux+=String(vec[i]);
}
return vAux;
}
}

```

```
String ExtraelNombre(String ptext){  
    int pos_nm=ptext.indexOf(" ");  
    //Primer nombre  
    String aux_text=ptext;  
    if(pos_nm>0){  
        aux_text=ptext.substring(0,pos_nm);  
    }  
    aux_text.toUpperCase();  
    return aux_text;  
}
```

```

//botones=36
void BotonPresionado() {
int vBut=myButtons.checkButtons();
Serial.println("Boton Presionado: "+String(vBut));
if(vBut!=-1){
beep1();
}
if(vBut==bPresioneAqui){
LoadColoqueHuella();
int vaID=LeeHuella();
if(vaID!=-2){
WriteWifi("#CID"+String(vaID));
}
LoadPresioneAqui(true);
}else if(vBut==bInstrucciones){
LoadInstrucciones();
}else if(vBut==bAtras_i){
LoadPresioneAqui(true);
}else if(vBut==bAdmin){
LoadColoqueHuella();
int vaID=LeeHuella();
if(vaID!=-2){
WriteWifi("#CPS"+String(vaID));
}
LoadPresioneAqui(true);
}else if(vBut==bAtras){
vPagDocenteActual--;
vPagDocenteActual=vPagDocenteActual<1?
14:vPagDocenteActual;
CargaDocentesPag(vPagDocenteActual);
}else if(vBut==bAdelante){
vPagDocenteActual++;
vPagDocenteActual=vPagDocenteActual>14?
1:vPagDocenteActual;
CargaDocentesPag(vPagDocenteActual);
}else if(vBut==bDocente){
if(CheckDocenteValue()==true){
LoadRegistro_2();
}
}else if(vBut==bTecladoErase){
if(vSel_DI==0){

```

```

vDniValue=vDniValue.substring(0,vDniValue.length()-1);
myGLCD.print(" ", 180, 150);
myGLCD.print(vDniValue, 180, 150);
}else if(vSel_DI==1){
myGLCD.print(" ", 180, 200);
vIdValue=vIdValue.substring(0,vIdValue.length()-1);
myGLCD.print(vIdValue, 180, 200);
}
}else if(vBut==bCancelar_reg1){
LoadPresioneAqui(true);
}else if(vBut==bCancelar_reg2){
LoadRegistro_1();
}else if(vBut==bCancelar_reg3){
LoadRegistro_2();
}else if(vBut==bFakeDocente){
//NADA
}else if(vBut==bDNI){
vSel_DI=0;
}else if(vBut==bID){
vSel_DI=1;
}else if(vBut==bRegistrar){
if(CheckDniValue()==true && CheckIdValue()==true){
vIdSeleccionada=vIdValue.toInt();
LoadRegistro_3();
}
}else if(vBut==bGuardaHuella){
vRegistroCorrecto=getFingerprintEnroll(vIdSeleccionada);
vIdSeleccionada=-1;
}else if(vBut==bEnviarHuella){
WriteWifi("#RUS"+vDocenteSeleccionado+"$"+vDniValue+"%"+vIdVa
l
ue);
delay(100);
LoadPresioneAqui(true);
vRegistroCorrecto=1000;
}else{
for(int i=0;i<10;i++){
if(vBut==bDocentes[i]){
String vStr;
myGLCD.setColor(255,255,255);
myGLCD.setBackColor(VGA_BLACK);

```

```

myGLCD.setFont(SmallFont);
for(int j=0;j<40;j++){vStr+=" ";}
myGLCD.print(vStr, CENTER, 450);
vDocenteSeleccionado=vDocentes[i];
vDocenteSeleccionado_show=vDocentes[i];
vDocenteSeleccionado_show.replace("@"," ");
myGLCD.print(vDocenteSeleccionado_show, CENTER, 450);
}else if(vBut==bTecladoNumerico[i]){
if(vSel_DI==0){
vDniValue+=String(i);
myGLCD.print(" ", 180, 150);
myGLCD.print(vDniValue, 180, 150);
}else if(vSel_DI==1){
vIdValue+=String(i);
myGLCD.print(" ", 180, 200);
myGLCD.print(vIdValue, 180, 200);
}
}
}
}
}
}
void InitButtons(){
myButtons.deleteAllButtons();
//PAGINA PRESIONE AQUI
bPresioneAqui=myButtons.addButton(250,130,300,100,"PRESIONE
AQUI");
bInstrucciones=myButtons.addButton(1,419,150,60,
"Instrucciones");
bAdmin=myButtons.addButton(749,419,50,60,"q");
//INSTRUCCIONES
bAtras_i=myButtons.addButton(10,419,60,60,"ATRAS");
//PAGINA DOCENTES
CreaDocentes_bt();
//PAGINA TECLADO
CreaTecladoNumerico_bt();
//PAGINA GUARDA HUELLA
bGuardaHuella=myButtons.addButton(250,130,300,80,"GUARDAR
HUELLA");
bCancelar_reg3 = myButtons.addButton(49,449,47,27,"can");
bEnviarHuella=myButtons.addButton(300,250,150,27,"Enviar");
}

```

```

void CreaDocentes_bt() {
for(int i=0;i<10;i++){
bDocentes[i]=myButtons.addButton(230,50+34*i,340,32," ");
}
bAtras = myButtons.addButton(229,389,47,27,"at");
bAdelante = myButtons.addButton(524,389,47,27,"ad");
bDocente = myButtons.addButton(670,230,42,27,"OK");
bCancelar_reg1 = myButtons.addButton(49,449,47,27,"can");
}
void CreaTecladoNumerico_bt() {
bTecladoNumerico[0]=myButtons.
addButton(275+(0*50),330,30,30,"0");
bTecladoNumerico[1]=myButtons.
addButton(275+(1*50),330,30,30,"1");
bTecladoNumerico[2]=myButtons.
addButton(275+(2*50),330,30,30,"2");
bTecladoNumerico[3]=myButtons.
addButton(275+(3*50),330,30,30,"3");
bTecladoNumerico[4]=myButtons.
addButton(275+(4*50),330,30,30,"4");
bTecladoNumerico[5]=myButtons.
addButton(275+(0*50),380,30,30,"5");
bTecladoNumerico[6]=myButtons.
addButton(275+(1*50),380,30,30,"6");
bTecladoNumerico[7]=myButtons.
addButton(275+(2*50),380,30,30,"7");
bTecladoNumerico[8]=myButtons.
addButton(275+(3*50),380,30,30,"8");
bTecladoNumerico[9]=myButtons.
addButton(275+(4*50),380,30,30,"9");
bTecladoErase=myButtons.addButton(260+((2)*50),430,60,30,
"<-");
bCancelar_reg2 = myButtons.addButton(49,449,47,27,"can");
bFakeDocente = myButtons.addButton(30,100,130,27,"DOCENTE");
bDNI=myButtons.addButton(30,150,130,27,"DNI:");
bID=myButtons.addButton(30,200,130,27,"ID:");
bRegistrar=myButtons.addButton(670,230,120,27,"REGISTRAR");
}

```

```

void Btns_PresioneAqui(boolean st){
myButtons.setTextFont(BigFont);
myButtons.setButtonColors(VGA_AQUA , VGA_GRAY, VGA_WHITE,
VGA_AQUA, VGA_BLACK);
DrawButton_en(bPresioneAqui,st);
myButtons.setTextFont(SmallFont);
DrawButton_en(bInstrucciones,st);
myButtons.setTextFont(Dingbats1_XL);
DrawButton_en(bAdmin,st);
if(st==true){
Btns_Instrucciones(false);
Btns_Docentes(false);
Btns_Teclado(false);
Btns_GuardaHuella(false);
vPantallaActual=0;
}
}

void Btns_Instrucciones(boolean st){
myButtons.setTextFont(SmallFont);
myButtons.setButtonColors(VGA_AQUA , VGA_GRAY, VGA_WHITE,
VGA_AQUA, VGA_BLACK);
DrawButton_en(bAtras_i,st);
if(st==true){
Btns_PresioneAqui(false);
Btns_Docentes(false);
Btns_Teclado(false);
Btns_GuardaHuella(false);
}
}

void Btns_Docentes(boolean st){
myButtons.setTextFont(SmallFont);
myButtons.setButtonColors(VGA_WHITE , VGA_GRAY, VGA_WHITE,
VGA_BLUE, VGA_BLACK);
for(int i=0;i<10;i++){
DrawButton_en(bDocentes[i],st);
}
DrawButton_en(bAtras,st);
DrawButton_en(bAdelante,st);
DrawButton_en(bDocente,st);
DrawButton_en(bCancelar_reg1,st);

```

```

if(st==true){
Btns_PresioneAqui(false);
Btns_Instrucciones(false);
Btns_Teclado(false);
Btns_GuardaHuella(false);
}
}
void Btns_Teclado(boolean st){
myButtons.setButtonColors(VGA_WHITE , VGA_GRAY, VGA_WHITE,
VGA_BLUE, VGA_BLACK);
myButtons.setTextFont(SmallFont);
for(int i=0;i<10;i++){
DrawButton_en(bTecladoNumerico[i],st);
}
DrawButton_en(bTecladoErase,st);
myButtons.setTextFont(BigFont);
DrawButton_en(bCancelar_reg2,st);
DrawButton_en(bFakeDocente,st);
DrawButton_en(bDNI,st);
DrawButton_en(bID,st);
DrawButton_en(bRegistrar,st);
if(st==true){
Btns_PresioneAqui(false);
Btns_Docentes(false);
Btns_Instrucciones(false);
Btns_GuardaHuella(false);
}
}
void Btns_GuardaHuella(boolean st){
myButtons.setTextFont(BigFont);
myButtons.setButtonColors(VGA_AQUA , VGA_GRAY, VGA_WHITE,
VGA_AQUA, VGA_BLACK);
DrawButton_en(bGuardaHuella,st);
DrawButton_en(bCancelar_reg3,st);
DrawButton_en(bEnviarHuella,st);
if(st==true){
Btns_PresioneAqui(false);
Btns_Docentes(false);
Btns_Instrucciones(false);
Btns_Teclado(false);
}
}

```



```

}
boolean CheckDniValue() {
if(vDniValue.equals(""))==false) {
return true;
}
return false;
}
boolean CheckIdValue() {
if(vIdValue.equals(""))==false) {
return true;
}
return false;
}
boolean CheckDocenteValue() {
if(vDocenteSeleccionado_show.equals(""))==false) {
return true;
}
return false;
}
void DrawButton_en(int but,boolean st){
if(st==true){
myButtons.enableButton(but,true);
}else{
myButtons.disableButton(but);
}
}
void InitTactil(){
myTouch.InitTouch();
delay(250);
myTouch.setPrecision(PREC_HI);
delay(250);
}

```

```
boolean LeeWifi(){
vWifiData=client.readStringUntil('\n');
Serial.print("-]Servidor_in: ");
Serial.println(vWifiData);
delay(5);
return true;
}
void WriteWifi(String str){
client.println(str);
delay(5);
Serial.print("-]Servidor_out: ");
Serial.println(str);
}
int ConexionWifi(){
if(digitalRead(pinCheckConexion)==0){
delay(250);
return 1;
}
delay(250);
return 0;
}
void InitWifi(){
client.begin(115200);//ESP8266
pinMode(pinCheckConexion,INPUT_PULLUP);
}
```

```

void writing(char* palabra, int fontsize, int x, int y){
if(fontsize=8)myGLCD.setFont(SmallFont);
if(fontsize=16)myGLCD.setFont(BigFont);
if(fontsize=24)myGLCD.setFont(DejaVuSans24);
myGLCD.print(palabra,x,y);
}
void writing(char* palabra, int fontsize, char x, int y){
if(fontsize==8)myGLCD.setFont(SmallFont);
if(fontsize==16)myGLCD.setFont(BigFont);
if(fontsize==24)myGLCD.setFont(DejaVuSans24);
int tam=myGLCD.getStringWidth(palabra);
tam=(800-tam)/2;
myGLCD.print(palabra,tam,y);
}
void writing(char* palabra, int fontsize, int x, char y){
if(fontsize=8)myGLCD.setFont(SmallFont);
if(fontsize=16)myGLCD.setFont(BigFont);
if(fontsize=24)myGLCD.setFont(DejaVuSans24);
myGLCD.print(palabra,x,CENTER);
}
void writing(char* palabra, int fontsize, char x, char y){
if(fontsize=8)myGLCD.setFont(SmallFont);
if(fontsize=16)myGLCD.setFont(BigFont);
if(fontsize=24)myGLCD.setFont(DejaVuSans24);
int tam=myGLCD.getStringWidth(palabra);
tam=(800-tam)/2;
myGLCD.print(palabra,tam,CENTER);
}
void line(int x, int x1, int y) // punto de inicio, punto
final, nivel en y
{
for(int i=0;i<3;i++){
myGLCD.drawLine(x,y+i,x1,y+i);
}
}
int button(int a,int b, int c, int d, char* e,char f, int
font, int id){

```

```

if(font==8)myButtons.setTextFont(SmallFont);
if(font==16)myButtons.setTextFont(BigFont);
if(font==24)myButtons.setTextFont(DejaVuSans24);
int but= myButtons.addButton( a, b, c, d, e);
myButtons.drawButton(id);
if(f=='b')myButtons.setButtonColors(VGA_BLACK , VGA_GRAY,
VGA_WHITE, VGA_BLUE, VGA_AQUA);
if(f=='w')myButtons.setButtonColors(VGA_WHITE , VGA_GRAY,
VGA_WHITE, VGA_BLUE, VGA_AQUA);
if(f=='a')myButtons.setButtonColors(VGA_AQUA , VGA_GRAY,
VGA_WHITE, VGA_AQUA, VGA_BLACK);
if(f=='g')myButtons.setButtonColors(VGA_GREEN , VGA_GRAY,
VGA_WHITE, VGA_GREEN, VGA_BLACK);
myButtons.drawButton(id);
return but;
}
int button(int a,int b, int c, int d, char* e,char f, int
font, int id, word symbol){
if(font==8)myButtons.setTextFont(SmallFont);
if(font==16)myButtons.setTextFont(BigFont);
if(font==24)myButtons.setTextFont(DejaVuSans24);
int but= myButtons.addButton( a, b, c, d, e, symbol);
myButtons.drawButton(id);
if(f=='b')myButtons.setButtonColors(VGA_BLACK , VGA_GRAY,
VGA_WHITE, VGA_BLUE, VGA_AQUA);
if(f=='w')myButtons.setButtonColors(VGA_WHITE , VGA_GRAY,
VGA_WHITE, VGA_BLUE, VGA_AQUA);
if(f=='a')myButtons.setButtonColors(VGA_AQUA , VGA_GRAY,
VGA_WHITE, VGA_AQUA, VGA_BLACK);
myButtons.drawButton(id);
return but;
}

```

```

void LoadPresioneAqui(boolean st){
LoadPlantilla();
Btns_PresioneAqui(st);
}
void LoadColoqueHuella(){// pagina 1
if(vPantallaActual!=1){
Btns_PresioneAqui(false);
myGLCD.setColor(VGA_BLACK);
myGLCD.fillRect(250,130,550,230);
myGLCD.setBackgroundColor(VGA_BLACK);
myGLCD.setColor(255,255,255);
writing("COLOQUE SU HUELLA",24,'C',160);
writing("SOBRE EL SENSOR",24,'C',200);
vPantallaActual=1;
}
}
void LoadUsuarioNoRegistrado(){//pagina 2
if(vPantallaActual!=2){
Btns_PresioneAqui(false);
myGLCD.setColor(VGA_BLACK);
myGLCD.fillRect(250,130,550,230);
myGLCD.setBackgroundColor(VGA_BLACK);
myGLCD.setColor(255,255,255);
writing("USUARIO NO",24,'C',160);
writing("REGISTRADO",24,'C',200);
vPantallaActual=2;
}
}
void LoadRegistroExitoso(){//pagina 3
if(vPantallaActual!=3){
myGLCD.setColor(VGA_BLACK);
myGLCD.fillRect(250,130,550,230);
myGLCD.setBackgroundColor(VGA_BLACK);
myGLCD.setColor(255,255,255);
writing("REGISTRO",24,'C',160);
writing("EXITOSO",24,'C',200);
vPantallaActual=3;
}
}

```

```

}
void LoadInstrucciones() { //pagina 4
if (vPantallaActual != 4) {
myGLCD.clrScr();
recuadro(1, 45, 3, 797, 300);
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_WHITE);
myGLCD.print("1.-Presionar sobre el boton que indica la
pantalla ", 2, 50);
myGLCD.print("2.-Colocar huella sobre el sensor ", 2, 75);
myGLCD.print("3.-Retirar la huella al escuchar el sonido
", 2, 100);
myGLCD.print("4.-Verificar que su nombre aparezca en la
pantalla ", 2, 125);
myGLCD.print("5.-Si su nombre no aparece, repita la
operacion ", 2, 150);
myGLCD.print("RECOMENDACION: LIMPIE SU HUELLA DACTILAR
", 2, 175);
myGLCD.print("ANTES DE MARCAR.", 2, 200);
myGLCD.print("Si al salir de esta pantalla aparece el
mensaje ", 2, 225);
myGLCD.print("CONEXION PERDIDA, espere unos segundos ", 2,
250);
Btns_Instrucciones(true);
vPantallaActual=4;
}
}
void LoadConexionPerdida() { //pagina 7
if (vPantallaActual != 7) {
if (vPantallaActual == 4 || vPantallaActual == 5 ||
vPantallaActual == 6 || vPantallaActual == -1) {
LoadPresioneAqui(false);
}
myGLCD.setColor(VGA_BLACK);
myGLCD.fillRect(250, 130, 550, 230);
myGLCD.setColor(255, 0, 0);
myGLCD.setBackgroundColor(VGA_BLACK);
writing("CONEXION PERDIDA", 24, 'C', 180);
vPantallaActual=7;
}
}
}

```

```

void LoadRegistro_1(){//pagina 5
if(vPantallaActual!=5){
vPagDocenteActual=1;
Btns_PresioneAqui(false);
myGLCD.clrScr();
vDocenteSeleccionado="";
vDocenteSeleccionado_show="";
recuadro(222,42,8,356,382);
Btns_Docentes(true);
CargaDocentesPag(1);
vPantallaActual=5;
}
}
void LoadRegistro_2(){//pagina 6
if(vPantallaActual!=6){
Btns_PresioneAqui(false);
myGLCD.clrScr();
vDniValue="";
vIdValue=String(vIdDisponible);
Btns_Teclado(true);
myGLCD.setFont(BigFont);
myGLCD.setColor(255,255,255);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.print(vDocenteSeleccionado_show, 180, 100);
myGLCD.print(" ", 180, 200);
myGLCD.print(String(vIdDisponible), 180, 200);
myGLCD.setBackgroundColor(VGA_BLACK);
vPantallaActual=6;
}
}
void LoadRegistro_3(){//pagina 8
if(vPantallaActual!=8){
Btns_PresioneAqui(false);
myGLCD.clrScr();
Btns_GuardaHuella(true);
myGLCD.setFont(BigFont);
myGLCD.setColor(255,255,255);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.print("Registro Huella", CENTER, 100);
myGLCD.setBackgroundColor(VGA_BLACK);
vPantallaActual=8;
}
}

```

```

}
}
void LoadPlantilla() {
if(vPantallaActual==4 || vPantallaActual==5 ||
vPantallaActual==6 || vPantallaActual==8 ||
vPantallaActual==-1) {
myGLCD.setColor(192,227,85);
myGLCD.fillRect(0,0,800,480);
myFiles.loadBitmap(0, 0, 165, 220, files800[0]);
myFiles.loadBitmap(800-165, 0, 165, 220, files800[1]);
}
myGLCD.setColor(3,130,193);
myGLCD.setBackColor(VGA_TRANSPARENT);
writing("BIENVENIDO",16,'C',50);
myGLCD.setColor(0,0,0);
writing("ULTIMO REGISTRO",16,'C',320);
line(170,600,70);
}

```



```

void recuadro(int x, int y,int thick, int width, int
lengthy){
myGLCD.setColor(VGA_BLACK);
myGLCD.fillRect(x,y,x+width,y+lengthy);
myGLCD.setColor(192,227,85);
for(int i=0;i<thick;i++){
myGLCD.drawRect(x+i,y+i,x+width-i,y+lengthy-i);
}
myGLCD.setColor(VGA_BLACK);
for(int esq=0;esq<4;esq++){
int a=0,b=0;
switch (esq){
case 0: a=x;
b=y;
break;
case 1: a=x+width-thick;
b=y;
break;
case 2: a=x+width-thick;
b=y+lengthy-thick;
break;
case 3: a=x;
b=y+lengthy-thick;
break;
}
for(int j=0;j<thick-5;j++){
myGLCD.drawRect(a+j,b+j,a+thick-j,b+thick-j);
}
}
}

```

ANEXO 2

Programación esp8266

```
#include <ESP8266WiFi.h>
const char* ssid = "HUELLA";
const char* password = "12345678";
//ETHERNET
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192, 168, 1, 13);
IPAddress gateway(192, 168, 1, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress server(192, 168, 1, 14);
WiFiClient client;
boolean alreadyConnected = false;
unsigned long m1=0;
void setup() {
  WiFi.mode(WIFI_STA);
  WiFi.config(ip, gateway, subnet);
  WiFi.begin(ssid, password);
  Serial.begin(115200);
  pinMode(2,OUTPUT);
  digitalWrite(2,HIGH);
  while(!Serial);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
  }
  //ETHERNET
  delay(1000);
  if (client.connect(server, 12)) {
    //Conectado
  } else {
    //Desconectado
  }
}
void loop() {
  if (client.connected()){
    digitalWrite(2,LOW);
    if (client.available()) {
      char c = client.read();
      Serial.print(c);
    }
    while(Serial.available()){
      String inChar1=Serial.readStringUntil('\n');
      client.println(inChar1);
    }
  }
}
```

```
}  
}  
if (!client.connected()) {  
  digitalWrite(2,HIGH);  
  client.stop();  
  delay(200);  
  tryReconnect();  
}  
}  
void tryReconnect(){  
  client.connect(server,12);  
}
```

ANEXO 3

Programación aplicación visual basic

```
Imports MySql.Data.MySqlClient
Imports System.Data.SqlClient
'#CID - Registra asistencia (entrada y salida)
'#RUS - Rregistra usuario
'#CPS - Consulta permiso (administrador)
Public Class Formulario_principal

    Dim vPrimerLogueo As Boolean = False
    Dim vCommandLen As Integer = 4
    Private Server As TCPControl

    Dim obj_busqueda As New BusquedaItem
    Private Sub bt_conectar_Click(sender As Object, e As EventArgs) Handles
bt_conectar.Click
        Server = New TCPControl
        AddHandler Server.MessageReceived, AddressOf OnLineReceived
        If (Server.ServerState = True) Then
            Debug.WriteLine("Conectado")
            bt_conectar.Enabled = False
        End If
    End Sub
    Private Sub OnLineReceived(sender As TCPControl, pMsjeLeido As String)
        Debug.WriteLine("[DB] Leyendo: " & pMsjeLeido)
        If vPrimerLogueo = False Then
            Server.Send("#SistemaHuella")
            vPrimerLogueo = True
        End If

        If pMsjeLeido <> "" Then
            Dim pos_Time As Integer = InStr(1, pMsjeLeido, "#STD",
CompareMethod.Text)
            Dim pos_Cid As Integer = InStr(1, pMsjeLeido, "#CID",
CompareMethod.Text)
            Dim pos_Rus As Integer = InStr(1, pMsjeLeido, "#RUS",
CompareMethod.Text)
            Dim pos_Cps As Integer = InStr(1, pMsjeLeido, "#CPS",
CompareMethod.Text)
            Dim pos_test As Integer = InStr(1, pMsjeLeido, "$$",
CompareMethod.Text)

            If pos_Time <> 0 Then
                Dim TimeYer As String = DateTime.Now.Year.ToString
                Dim TimeMon As String = DateTime.Now.Month.ToString
                Dim TimeDay As String = DateTime.Now.Day.ToString
                Dim TimeHor As String = DateTime.Now.Hour.ToString
                Dim TimeMin As String = DateTime.Now.Minute.ToString
                Dim TimeSec As String = DateTime.Now.Second.ToString
```

```

        Server.Send("#STD" + "SE" + TimeSec + "MI" + TimeMin + "HO" + TimeHor
+ "DA" + TimeDay + "MO" + TimeMon + "YE" + TimeYer + vbCrLf)
    ElseIf pos_Cid <> 0 Then
        Dim vID As Integer = CInt(Mid(pMsjeLeido, pos_Cid + vCommandLen))
        Dim vPersonal As New PersonalItem
        vPersonal = PersonalTabla.ObtenerUsuarioPorID(vID)
        If vPersonal.ID <> -1 Then
            'Registrando asistencia
            Dim in_or_out As Integer = RegistroAsistencia(vPersonal)
            Dim vHora_actual As TimeSpan = DateTime.Now.TimeOfDay
            Dim vHora_string As String =
vHora_actual.ToString("hh:mm:ss")
            If in_or_out = 1 Then
                Server.Send("#LNMIN" + vPersonal.NOMBRES + "-" +
vPersonal.APELLIDOS + "$" + vHora_string + vbCrLf)
            ElseIf in_or_out = 2 Then
                Server.Send("#LNMOT" + vPersonal.NOMBRES + "-" +
vPersonal.APELLIDOS + "$" + vHora_string + vbCrLf)
            End If
        Else
            Server.Send("#LNNull" + vbCrLf)
            'lb_msje.Text = "Usuario no registrado"
        End If
    ElseIf pos_Cps <> 0 Then
        Dim vID As Integer = CInt(Mid(pMsjeLeido, pos_Cps +
vCommandLen))

        Dim vPersonal As New PersonalItem
        vPersonal = PersonalTabla.ObtenerUsuarioPorID(vID)
        If vPersonal.ID <> -1 Then
            Debug.WriteLine("[DB] TRY ADMIN: " + vPersonal.NOMBRES +
Str(vPersonal.PERMISO))
            If vPersonal.PERMISO = 1 Then
                Dim in_or_out As Integer = RegistroAsistencia(vPersonal)
                Dim vHora_actual As TimeSpan = DateTime.Now.TimeOfDay
                Dim vHora_string As String =
vHora_actual.ToString("hh:mm:ss")
                Server.Send("#LST2ID" +
PersonalTabla.ObtenerIDDisponibles().ToString)
            End If
        End If
    ElseIf pos_Rus <> 0 Then
        Dim pos_sep1 As Integer = InStr(1, pMsjeLeido, "$",
CompareMethod.Text)
        Dim pos_sep2 As Integer = InStr(1, pMsjeLeido, "%",
CompareMethod.Text)
        Dim pos_sep3 As Integer = InStr(1, pMsjeLeido, "@",
CompareMethod.Text)
        If pos_sep1 <> 0 And pos_sep2 <> 0 Then
            Dim vApellidos As String = Mid(pMsjeLeido, pos_Rus + 4,
pos_sep3 - pos_Rus - 4)
            Dim vNombres As String = Mid(pMsjeLeido, pos_sep3 + 1,
pos_sep1 - pos_sep3 - 1)
            Dim vDni As String = Mid(pMsjeLeido, pos_sep1 + 1, pos_sep2
- pos_sep1 - 1)
            Dim vId As String = Mid(pMsjeLeido, pos_sep2 + 1)
            'MessageBox.Show(vNombres + " - " + vDni + " - " + vId,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            RegistraDocente(vNombres, vApellidos, vDni, vId)
        End If
    End If

```

```

        ElseIf pos_test <> 0 Then
            Server.Send("#")
            'UsoHuellaFunction(1, "#")
        End If
    End If
End Sub

Private Sub Formulario_principal_Load(sender As Object, e As EventArgs)
Handles MyBase.Load
    Timer1.Enabled = False
    Timer2.Enabled = False
    Timer3.Enabled = False
    ComboBox1.SelectedItem = ComboBox1.Items(0)
    InitDiseñoFormulario()
    'DateTimePicker1.Value = DateTime.Now.Date
End Sub

Public Function RegistroAsistencia(ByVal pPersonal As PersonalItem) As
Integer
    Dim retorno As Integer = 0
    Dim pAsistencia = AsistenciaTabla.ObtenerUsuarioPorDNI(pPersonal.DNI)
    If pAsistencia.INDICE_A > 0 And AsistenciaTabla.Modificar(pPersonal.DNI,
DateTime.Now.TimeOfDay) > 0 Then
        retorno = 2
    ElseIf pAsistencia.INDICE_A > 0 And AsistenciaTabla.Modificar(pPersonal.DNI,
DateTime.Now.TimeOfDay) = 0 Then
        pAsistencia = New AsistenciaItem()
        pAsistencia.llenar_datos(0, pPersonal.NOMBRES, pPersonal.APELLIDOS,
pPersonal.DNI, pPersonal.FACULTAD, DateTime.Now.Date, DateTime.Now.TimeOfDay, New
TimeSpan(), 0)
        If AsistenciaTabla.Agregar(pAsistencia) > 0 Then
            retorno = 1
        Else
            retorno = 2
            MsgBox("salida fake 1")
        End If
    Else
        pAsistencia = New AsistenciaItem()
        pAsistencia.llenar_datos(0, pPersonal.NOMBRES, pPersonal.APELLIDOS,
pPersonal.DNI, pPersonal.FACULTAD, DateTime.Now.Date, DateTime.Now.TimeOfDay, New
TimeSpan(), 0)
        If AsistenciaTabla.Agregar(pAsistencia) > 0 Then
            retorno = 1
        Else
            retorno = 2
            MsgBox("salida fake 2")
        End If
    End If
    Return retorno
End Function

```

```

Public Sub RegistraDocente(ByVal pNombres As String, ByVal pApellidos As String,
ByVal pDni As String, ByVal pId As String)
    Try
        ' Dim vdni As Integer = CInt(tb_dni.Text)
        Dim vPersonal As New PersonalItem
        Dim vID_disp As Integer = PersonalTabla.ObtenerIDDisponible()
        vPersonal.LlenarDatos(CInt(pId), CInt(pDni), LCase(pNombres),
LCase(pApellidos), "facfym", 0)
        Debug.WriteLine("[DB]: " + pNombres + " " + pApellidos + " " + pDni + "
" + pId)
        Using conexion As MySqlConnection = BaseDeDatosMySQL.ObtenerConexion()
            If BaseDeDatosMySQL.estado = True Then
                Dim Comando1 As New MySqlCommand(String.Format("Insert Into
PersonalTabla(ID,DNI,NOMBRES,APELLIDOS,FACULTAD,PERMISO) values
({0},{1},{2}','{3}','{4}','{5}')" , vPersonal.ID, vPersonal.DNI, vPersonal.NOMBRES,
vPersonal.APELLIDOS, vPersonal.FACULTAD, vPersonal.PERMISO), conexion)
                Dim Comando2 As New MySqlCommand(String.Format("Update
PersonalTabla Set
DNI=@DNI,NOMBRES=@NOMBRES,APELLIDOS=@APELLIDOS,FACULTAD=@FACULTAD,PERMISO=0 where
(ID='{0}')" , vPersonal.ID), conexion)
                Comando2.Parameters.AddWithValue("@DNI", vPersonal.DNI)
                Comando2.Parameters.AddWithValue("@NOMBRES", vPersonal.NOMBRES)
                Comando2.Parameters.AddWithValue("@APELLIDOS",
vPersonal.APELLIDOS)
                Comando2.Parameters.AddWithValue("@FACULTAD",
vPersonal.FACULTAD)
                Try
                    If vID_disp <> CInt(pId) Then
                        Comando2.ExecuteNonQuery()
                    Else
                        Comando1.ExecuteNonQuery()
                    End If
                Catch ex As Exception
                    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error)
                End Try
            End If
            conexion.Close()
        End Using
        Catch ex As Exception
            MessageBox.Show("Error registro docente:" + Chr(13) + ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub

```

```

Private Sub bt_buscar_Click(sender As Object, e As EventArgs) Handles
bt_buscar.Click
    Dim v_dni As Integer = 0
    If tb_dni.Text <> "" Then
        Try
            v_dni = CInt(tb_dni.Text)
        Catch ex As Exception
            MessageBox.Show("Dato ingresado inválido" & Chr(13) & ex.Message,
"Error", MessageBoxButtons.OK, MessageBoxIcon.Error)
            Return
        End Try
    End If

    obj_busqueda.llenar_datos(tb_nombres.Text, tb_apellidos.Text, v_dni,
DateTimePicker1.Value, cb_fecha.CheckState)
    If obj_busqueda.BIT_APELLIDOS = True Or obj_busqueda.BIT_DNI = True Or
obj_busqueda.BIT_FECHA = True Or obj_busqueda.BIT_NOMBRES = True Then
        DataGridView1.DataSource = AsistenciaTabla.Reporte_del_dia(obj_busqueda)
        DataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill
    End If

End Sub

Private Sub cb_fecha_CheckedChanged(sender As Object, e As EventArgs) Handles
cb_fecha.CheckedChanged
    If cb_fecha.CheckState = 1 Then
        DateTimePicker1.Enabled = True
    Else
        DateTimePicker1.Enabled = False
    End If
End Sub

Private Sub bt_exportar_Click(sender As Object, e As EventArgs) Handles
bt_exportar.Click
    If obj_busqueda.BIT_APELLIDOS = True Or obj_busqueda.BIT_DNI = True Or
obj_busqueda.BIT_FECHA = True Or obj_busqueda.BIT_NOMBRES = True Then
        'ExportarExcel.Asistencia(AsistenciaTabla.Reporte_del_dia(obj_busqueda))
        ExportarPDF.GeneraDoc(DataGridView1)
    End If
End Sub

Private Sub Form1_FormClosing(sender As Object, e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing
    Debug.WriteLine("CERRANDO")
    If bt_conectar.Enabled = False Then
        Server.IsListening = False
        Server.ClientData.Close()
    End If
End Sub

```



```

Private Sub InitDiseñoFormulario()

    gb_titles.Top = 10
    gb_titles.Left = 10
    gb_titles.Width = 460
    gb_titles.Height = 100

    PictureBox1.SizeMode = PictureBoxSizeMode.StretchImage
    PictureBox1.Scale(New SizeF(0.6, 1.5))
    PictureBox1.Left = 10
    PictureBox1.Top = 20

    PictureBox2.SizeMode = PictureBoxSizeMode.StretchImage
    PictureBox2.Scale(New SizeF(0.7, 1.5))
    PictureBox2.Left = gb_titles.Width - PictureBox2.Width - 8
    PictureBox2.Top = 20

    lb_title1.Font = New Font("Arial Black", 9, FontStyle.Bold)
    lb_title1.Left = (gb_titles.Width - lb_title1.Width) / 2
    lb_title1.Top = 30
    lb_title2.Font = New Font("Arial", 9, FontStyle.Regular)
    lb_title2.Top = lb_title1.Bottom + 20
    lb_title2.Left = (gb_titles.Width - lb_title2.Width) / 2

    gb_conexion.Width = gb_titles.Width '244
    gb_conexion.Left = 10 '(Me.Width - gb_conexion.Width) / 2
    gb_conexion.Top = gb_titles.Bottom + 20

    gb_parametros.Width = gb_titles.Width '406
    gb_parametros.Left = 10 'Me.Width - gb_parametros.Width - 30 '(Me.Width -
gb_parametros.Width) / 2
    gb_parametros.Top = gb_conexion.Bottom + 20

    gb_resultados.Width = 645
    gb_resultados.Height = gb_parametros.Bottom - 10
    DataGridView1.Width = gb_resultados.Width - 15
    DataGridView1.Height = gb_resultados.Height - 20
    DataGridView1.Font = New Font("Arial", 8, FontStyle.Regular)
    DataGridView1.RowHeadersWidth = 5

    gb_resultados.Left = gb_titles.Right + 10 '((Me.Width - gb_resultados.Width)
/ 2) - 5
    gb_resultados.Top = gb_titles.Top

    Me.Width = gb_resultados.Right + 25
    Me.Height = gb_resultados.Bottom + 50
    Me.FormBorderStyle = FormBorderStyle.FixedSingle

End Sub
End Class

```

