UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE ELECTRÓNICA Y TELECOMUNICACIONES

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TITULO EN:

"INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES"

Título:

"DISEÑO E IMPLEMENTACION DE UN CONTROL DE ASISTENCIA INALAMBRICO POR HUELLA DACTILAR"

Autor: (es):

LIZET MARÍA RAMOS DILLON

JHONNY PAÚL PILCO NINABANDA

Director:

ING. Fabián Gunsha

RIOBAMBA – ECUADOR

2013

Los miembros del Tribunal de Graduación del proyecto de investigación de título: DISEÑO E IMPLEMENTACION DE UN CONTROL DE ASISTENCIA INALAMBRICO POR HUELLA DACTILAR, presentado por: Lizet María Ramos Dillon, Jhonny Paul Pilco Ninabanda y dirigida por: Fabián Celso Gunsha Maji.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Yesenia Cevallos

---------- **----------**

Presidente del Tribunal **Firma**

Ing. Fabián Gunsha

---------- **----------**

Director del Proyecto **Firma**

Ing. Aníbal Llanga

---------- **----------**

Miembro del Tribunal **Firma**

AUTORÍA DE LA INVESTIGACIÓN

"La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Lizet María Ramos Dillon, Jhonny Paúl Pilco Ninabanda y Ing. Fabián Gunsha Maji; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

AGRADECIMIENTO

Queremos hacer llegar un sentido agradecimiento a la Universidad Nacional de Chimborazo; autoridades, docentes y personal administrativo, quienes además de plantar en nosotros los conocimientos necesarios para enfrentarnos a un ámbito profesional, nos han dejado importantes lecciones de responsabilidad, respeto y humanismo.

En especial al Ing. Fabián Gunsha, tutor de este trabajo de grado y quien ha puesto todo su compromiso en el desarrollo del mismo.

Lizet Ramos D.

Jhonny Pilco N.

DEDICATORIA

A Dios, por permitirme llegar a este momento tan especial en mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valorarlo cada día más, A mis padres por su apoyo y dedicación en el transcurso de mi vida, de manera especial a mi madre por ser la persona que ha sabido compartir junto a mí los triunfos, alegrías, tristezas, lágrimas y ha sabido expresar las palabras justas en aquellos momentos difíciles, a mis tías quienes han sido testigos de mi trabajo en este arduo camino, en especial a mi tía Mariana, quien el llamado de Dios se adelantó y no pudo estar físicamente presente en estos momentos, pero que de igual manera sé que estará compartiendo esta alegría. A Jhonny Pilco el amigo que se ha convertido en el mejor equipo de trabajo a lo largo de esta vida estudiantil y con quien ahora concluimos esta etapa A mis amigos, quienes siempre han sabido brindarme una palabra de aliento y apoyo a lo largo de esta etapa.

Lizet Ramos D.

DEDICATORIA

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme el haber llegado hasta este momento tan importante de mi formación profesional. A mis padres, mi hermana por ser el pilar más importante y por demostrarme siempre su cariño y apoyo incondicional. A mis tíos, primos y demás familiares. A mis amigos quienes fueron un gran apoyo a lo largo de esta vida estudiantil. A Lizet Ramos por trabajar conjuntamente conmigo y brindarme su apoyo en toda esta vida estudiantil.

Jhonny Pilco N.

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

RESUMEN

El presente trabajo tiene como objetivo la implementación de un sistema de control de asistencia dirigido al alumnado, el control de asistencia emplea un sistema biométrico y la transmisión de los datos es inalámbrica.

Los sistemas biométricos ofrecen varias opciones, seleccionando de entre ellas la identificación por huella dactilar, por ser un método seguro, accesible, fácil de utilizar y económico en comparación con sus similares.

La comunicación inalámbrica utiliza la tecnología WIFI que permite comunicar transmitir los datos correspondientes a la huella dactilar de cada usuario de manera inalámbrica hacia un computador en donde se almacenará en una base de datos, esta información será manejada por un software que fue programado en Visual Basic.Net

Las pruebas realizadas con datos reales, presentan un eficiente trabajo del sistema completo.

SUMARY

The present work has the objective of implementing an assistance control system for students, that uses a biometric system and data transmission which is wireless.

Biometric systems offer many options, including selection of the fingerprint identification, that is safe, accessible, easy to use and inexpensive compared to other expensive system.

Wireless communication using WIFI technology allows data communication to transmit the fingerprint of each user wirelessly to a computer on which it is stored in a database. This information will be handled by software that is programmed in Visual Basic. Net

Tests with real data, present a complete efficient working system.

INTRODUCCIÓN

Existen varias maneras de asignar códigos de autenticación a cada persona, de aquí que nace la Biometría como el estudio de métodos automáticos para el reconocimiento único de humanos basados en uno o más rasgos físicos intrínsecos.

El sistema biométrico por huella dactilar es el de más aceptación en sistemas de seguridad por su facilidad de uso, costo y fiabilidad, pese a sus ventajas este sistema no posee la característica de portabilidad, esta característica puede ser mejorada añadiendo un sistema de comunicación inalámbrica.

Los sistemas de comunicación inalámbrica permiten dar movilidad a variedad de aplicaciones en las que se las involucra, el sistema wireless realiza la transmisión de información por medio de ondas electromagnéticas, la transmisión Wi-Fi permite dar movilidad al equipo de control de asistencia de alumnado.

Lo datos transmitidos vía Wi-Fi son recibidos por un computador con tarjeta inalámbrica y concentrada en una base de datos en MySQL, y procesados por un programa diseñado en Visual Basic.Net, el programa considera dos modos de uso: registro e identificación de usuario.

Las pruebas arrojaron datos satisfactorios, en las dos fases en las que fue diseñado, por lo que el equipo es considerado como efectivo para su uso.

CAPITULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. BIOMETRÍA

La biometría es una rama de la biología que estudia y mide los datos de los seres vivientes. El término "biometría" deriva de la palabra griega "bios" (vida) y "metros" (medida) y en su significado corriente en informática, sirve para indicar la identificación automática o la verificación de la identidad de un sujeto, sobre la base de sus características físicas o de su comportamiento.

La biometría abarca el estudio de todos aquellos métodos que permiten reconocer de forma única a una persona, estos métodos son denominados técnicas biométricas. Para que este reconocimiento sea efectivo el método usado emplea alguna característica reconocible que sea totalmente diferente entre dos personas, como por ejemplo el reconocimiento de la huella dactilar.

La biometría permite agilizar muchos procesos que son ejecutados miles de veces cada día y también garantizar ciertos niveles de seguridad, como por ejemplo fichar al entrar y salir del trabajo o controlar el acceso a zonas restringidas.

Sin embargo estas técnicas no son socialmente aceptadas; gran cantidad de veces los usuarios de estos métodos no ven con muy buenos ojos la introducción de estos sistemas. Por un lado estos métodos permiten tenerlos más controlados y evitar que ocurran las picarescas maneras de evadirlos y por tanto se pierdan horas de trabajo (como el caso en el cual se ficha mediante tarjetas y los compañeros se las dejan unos a otros cuando saben que van a faltar al trabajo o llegar tarde).

Fig. 1 Sistemas Biométricos
Fuente: Biometría aplicada a la seguridad.pdf

Por otro lado hay ciertos métodos biométricos mediante los cuales un usuario puede ver comprometida cierta información de la cual no quisiera que nadie tuviera constancia; en el caso concreto del análisis de ADN que además de reconocer al usuario podríamos conocer también si ha consumido ciertos tipos de estupefacientes. Este tipo de información podría provocar el despido del mismo de la empresa aunque las sustancias que el empleado consuma fuera del horario laboral no estuvieran interfiriendo en el trabajo.

En el caso concreto del reconocimiento de iris, algunos usuarios pueden ser un tanto escépticos ante la (inocua) luz que se proyecta en el ojo en la fase de reconocimiento, esto provoca un recelo a la hora de usar el mismo.

1.1.1. Sistemas Biométricos

Entenderemos por sistema biométrico a un sistema que fundamenta sus decisiones de reconocimiento mediante una característica personal que puede ser reconocida o verificada de manera automatizada. En los sistemas informáticos, la autentificación es el proceso de verificar la identidad de un usuario o de un dispositivo que opera en un ambiente de trabajo en red, para lo cual la autentificación biométrica ofrece un alto nivel de seguridad.
El desarrollo de sistemas para la autenticación biométrica es muy complejo, pero su uso protege de muchas vulnerabilidades de mecanismos tradicionales como

credenciales compartidas, contraseñas olvidadas, robadas o usadas inadecuadamente, para obtener una fuerte seguridad en los sistemas es recomendable a menudo combinar dos o más métodos de autentificación.

1.1.2. Características de un indicador biométrico.

Un indicador biométrico es alguna característica con la cual se puede realizar biometría. Cualquiera sea el indicador, debe cumplir los siguientes requerimientos:

- Universalidad
- Unicidad
- Permanencia
- Cuantificación

Los requerimientos anteriores sirven como criterio para descartar o aprobar a alguna característica como indicador biométrico. Luego de seleccionar algún indicador que satisfaga los requerimientos antes señalados, es necesario imponer restricciones prácticas sobre el sistema que tendrá como misión recibir y procesar a estos indicadores.

1.1.3. Características de un sistema biométrico

Las características básicas que un sistema biométrico para identificación personal debe cumplir pueden expresarse mediante las restricciones que deben ser satisfechas. Las restricciones antes señaladas apuntan a que el sistema considere:

- El desempeño, que se refiere a la exactitud, la rapidez y la robustez alcanzada en la identificación, además de los recursos invertidos y el efecto de factores ambientales y/u operacionales
- La aceptabilidad, que indica el grado en que la gente está dispuesta a aceptar un sistema biométrico en su vida diaria. Es claro que el sistema no debe representar peligro alguno para los usuarios y debe inspirar "confianza" a los mismos.

- La fiabilidad, que refleja cuán difícil es burlar al sistema. El sistema biométrico debe reconocer características de una persona viva, pues es posible crear dedos de látex, grabaciones digitales de voz prótesis de ojos, etc. Algunos sistemas incorporan métodos para determinar si la característica bajo estudio corresponde o no a la de una persona viva.

1.1.4. Arquitectura de un sistema biométrico

Los dispositivos biométricos poseen tres componentes básicos. El primero se encarga de la adquisición análoga o digital de algún indicador biométrico de una persona, como por ejemplo, la adquisición de la imagen de una huella dactilar mediante un escáner. El segundo maneja la compresión, procesamiento, almacenamiento y comparación de los datos adquiridos (en el ejemplo una imagen) con los datos almacenados. El tercer componente establece una interfaz con aplicaciones ubicadas en el mismo u otro sistema. La arquitectura típica de un sistema biométrico se presenta en la Fig. 2 Esta puede entenderse conceptualmente como dos módulos:

- El módulo de inscripción (enrolment module), se encarga de adquirir y almacenar la información proveniente del indicador biométrico con el objeto de poder contrastar a ésta con la proporcionada en ingresos posteriores al sistema. Las labores ejecutadas por el módulo de inscripción son posibles gracias a la acción del lector biométrico y del extractor de características. El primero se encarga de adquirir datos relativos al indicador biométrico elegido y entregar una representación en formato digital de éste. El segundo extrae, a partir de la salida del lector, características representativas del indicador. El conjunto de características anterior, que será almacenado en una base de datos central u otro medio como una tarjeta magnética, recibirá el nombre de template, ver Fig. 3 En otras palabras un template es la información representativa del indicador biométrico que se encuentra almacenada y que será utilizada en las labores de identificación al ser comparada con la información proveniente del indicador biométrico en el punto de acceso.

- El módulo de identificación (identification module), es el responsable del reconocimiento de individuos, por ejemplo en una aplicación de control de acceso. El proceso de identificación comienza cuando el lector biométrico captura la característica del individuo a ser identificado y la convierte a formato digital, para que a continuación el extractor de características produzca una representación compacta con el mismo formato de los templates. La representación resultante se denomina query y es enviada al comparador de características que confronta a éste con uno o varios templates para establecer la identidad.

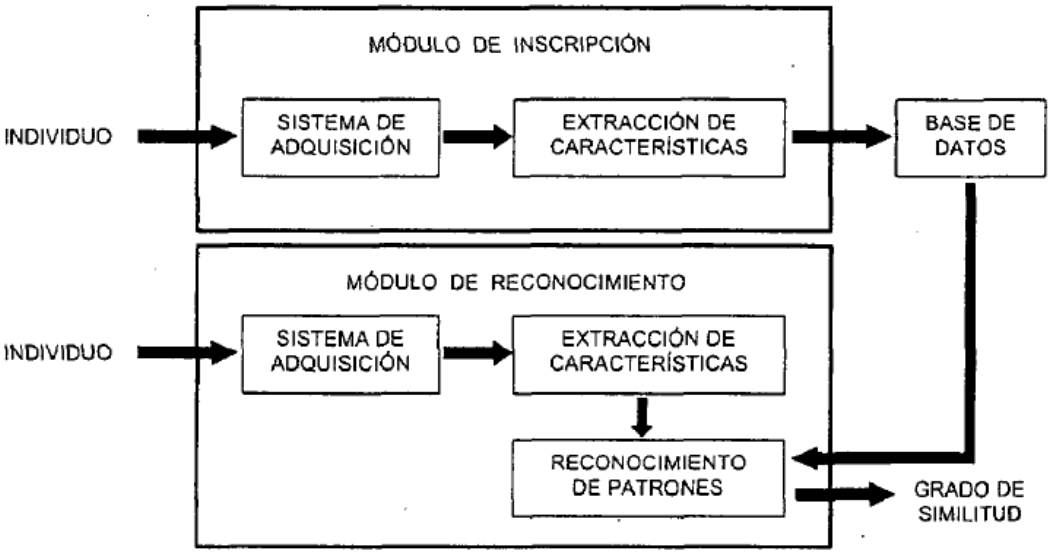

Fig. 2 Arquitectura de un sistema biométrico para identificación de personal.

Fuente: Reconocimiento de patrones biométricos.pdf

Fig. 3 Adquisición del template

Fuente: Autores

El conjunto de procesos realizados por el módulo de inscripción recibe el nombre de fase de inscripción, mientras que los procesos realizados por el módulo de identificación reciben la denominación de fase operacional.

1.1.5. Tipos de biometría

Biometría Estática**:** Es la que mide la anatomía del usuario, comprende entre otras:

- Huellas Dactilares.
- Análisis del iris.
- Análisis de retina.
- Venas del dorso de la mano.
- Reconocimiento Facial.

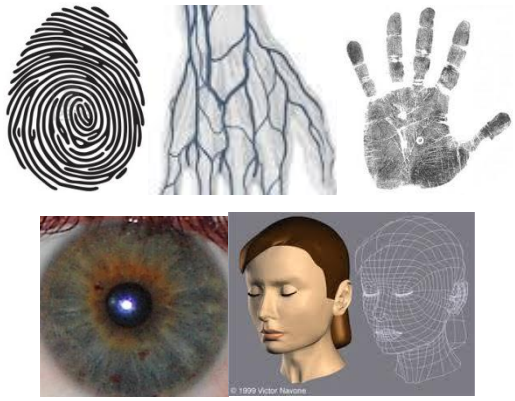

Fig. 4 Ejemplos de Biometría Estática

Fuente: Autores

Biometría Dinámica: Es la que mide el comportamiento del usuario, entre otras:

- Patrón de Voz.
- Firma manuscrita.
- Dinámica de tecleo.
- Cadencia del paso.
- Análisis gestual.

Fig. 5 Ejemplos de Biometría Dinámica

Fuente: Autores

Se realiza además una comparación entre los sistemas biométricos más utilizados, esta se encuentra en la Tabla 1, de donde observamos que la técnica de huella dactilar es la más aceptada por los usuarios debido a su alta fiabilidad y reducido costo frente a los demás sistemas de biometría, es por ello que se ha escogido al sistema biométrico por huella dactilar para la realización del presente proyecto.

Tecnología	Como trabaja	Tamaño Plantilla (Bytes)	Fiabilidad	Facilidad de uso	Posibles incidencias	Costo	Aceptación usuario
Huella Digital	Captura y compara patrones de huella	250-1000	Muy Alta	Alta	Ausencia de miembro	Bajo	Alta
Geometría de la mano	Mide y compara dimensiones de la mano	9	Baja	Alta	Edad, Ausencia de miembro	Bajo	Alta
Retina	Captura y compara patrones de retina	96	Baja	Baja	Gafas	Alto	Baja
Iris	Captura y compara patrones de iris	512	Baja	Baja	Luz	Muy Alta	Baja
Geometría Facial	Captura y compara patrones faciales	84 - 1300	Baja	Baja	Edad, Cabello, luz	Medio	Baja
Voz	Captura y compara, cadencia y tono de la voz	10000 - 20000	Alta	Media	Ruido, Temperatura y meteorología	Alto	Media
Firma	Captura y compara ritmo, aceleración y presión de la firma	1000 - 3000	Alta	Media	Edad, cambios, analfabetismo	Alto	Media

Tabla 1 Comparación de los Sistemas Biométricos

Fuente:

1.2. HUELLA DACTILAR

Las huellas dactilares son una característica propia de las personas, de tal forma que es posible identificar a cada una por sus huellas dactilares. Sin llegar a tal especificidad que requiere métodos sofisticados, es posible identificar el tipo de huella que tenemos cada uno de nosotros, ya que las huellas dactilares están constituidas por rugosidades en forma de salientes llamadas crestas papilares y

depresiones llamadas surcos inter papilares, se pueden clasificar en cuatro tipos: lazo, compuesta, arco y espiral, que se pueden observar en la Fig. 6.

Fig. 6 Tipos de huellas dactilares

Fuente: Autores

La huella dactilar presenta crestas y valles.

Existen 2 características que presentan las crestas que nos sirven para realizar la identificación de huellas dactilares y toman el nombre de minucias.

- Final de Cresta.- Característica definida como el punto en el que la cresta se acaba de forma abrupta.

- Bifurcación de la Cresta.- Característica definida como el punto en que la cresta se bifurca en dos o más crestas.

1.2.1. Extracción de Características.

Dentro de la imagen de una huella dactilar se encuentran las minucias descritas en la Fig. 7.

Características	
	Terminación
	Bifucarción
	Laguna
	Borde independiente
	Punto o isla
	Aguijón
	Cruce

Fig. 7 Descripción de minucias.

1.2.2. Reconocimiento de Huellas Dactilares

La Fig. 8, muestra un esquema del proceso a seguir para el reconocimiento de huellas dactilares, estas etapas inician con la captura de la huella dactilar, pasando por un pre proceso para el tratamiento de la imagen, de esta manera las características principales de la huella pueden ser identificadas y extraídas, sobre un patrón se aplica un algoritmo de reconocimiento el cual va concatenado a una base de datos, de modo que la huella entrante pueda ser comparada con el historial de huellas existente en la base de datos e identificar si el individuo corresponde a un usuario ya registrado en la misma.

Fig. 8 Proceso de Reconocimiento por Huella Dactilar

Fuente: Biometría aplicada a la Seguridad.pdf

1.3. SENSOR DE HUELLA DACTILAR

1.3.1. Ópticos Reflexivos.

Se basan en la técnica más antigua, consiste en colocar el dedo sobre una superficie de cristal o un prisma que está iluminado por un diodo LED. Cuando las crestas de las huellas del dedo tocan la superficie, la luz es absorbida, mientras que entre dichas crestas se produce una reflexión total. La luz resultante y las zonas de oscuridad son registradas en un sensor de imagen.

En la práctica existen algunas dificultades con esta técnica: las imágenes obtenidas con dedos húmedos y secos son muy diferentes y, además, el sistema es sensible al polvo y a la suciedad de la superficie. La unidad tiene un tamaño considerable, poco práctico y caro. Este sistema es fácil de engañar y si la piel está deteriorada o dañada, la huella no se reconoce correctamente. El reconocimiento de la huella dactilar de las personas mayores también es difícil de hacer ya que la piel no es lo suficientemente elástica. En algunas circunstancias esto puede producir un reconocimiento falso. Si la huella almacenada fue tomada con menos presión, se pueden producir aceptaciones falsas.

1.3.2. Ópticos Transmisivos

Esta técnica funciona sin contacto directo entre el dedo y la superficie del sensor. La luz pasa a través del dedo desde la cara de la uña, y al otro lado, mientras que una cámara toma una imagen directa de la huella dactilar.

La humedad no produce ninguna dificultad. El sensor ve a través de la superficie de la piel sobre una superficie más profunda y produce una imagen multi espectral. El uso de diferentes longitudes de onda para generar imágenes nos proporciona información de diferentes estructuras subcutáneas, indicación de que el objeto en cuestión es un dedo genuino. El uso de filtros polarizados ortogonales asegura que solamente la luz que tiene importancia a su paso bajo la piel es la que pasa, y bloquea la luz que se reflejaría directamente de la superficie. Solamente

unos dedos artificiales muy precisos podrían tener la posibilidad de engañar a este sensor.

1.3.3. Capacitivos

El sensor es un circuito integrado de silicio cuya superficie está cubierta por un gran número de elementos transductores (o píxeles), con una resolución típica de 500 dpi. Cada elemento contiene dos electrodos metálicos adyacentes. La capacidad entre los electrodos, que forma un camino de realimentación para un amplificador inversor, se reduce cuando el dedo se aplica sobre dicha superficie: se reduce más cuando detecta crestas y menos cuando detecta el espacio entre ellas, en la Fig. 9 se muestra el funcionamiento de un sensor capacitivo.

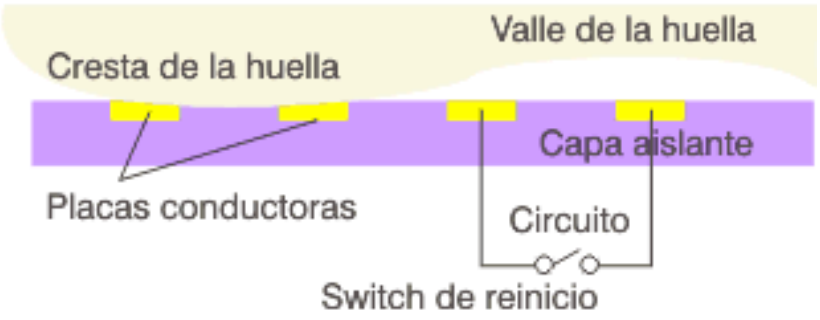

Fig. 9 Muestra del sensor capacitivo

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

1.3.4. Mecánicos

Se trata de decenas de miles de diminutos transductores de presión que se montan sobre la superficie del sensor. Un diseño alternativo utiliza conmutadores que están cerrados cuando son presionados por una cresta, pero permanecen abiertos

cuando están bajo un valle. Esto sólo proporciona un bit de información por píxel, en lugar de trabajar con una escala de grises.

1.3.5. Térmicos

En este caso se detecta el calor conducido por el dedo, el cual es mayor cuando hay una cresta que cuando hay un valle. Se ha desarrollado un componente de silicio con una matriz de píxeles denominado "Finger Chip", es decir, "circuito integrado dedo", cada uno de los cuales está cubierto con una capa de material piro eléctrico en el que un cambio de temperatura se traduce en un cambio en la distribución de carga de su superficie. La imagen está en la escala de grises que tiene la calidad adecuada incluso con el dedo desgastado, con suciedad, con grasa o con humedad. El sensor dispone de una capa protectora robusta y puede proporcionar una salida dinámica.

1.3.6. Salida dinámica

La mayoría de los sensores descritos han sido alterados en el pasado. Para evitar esto, se ha añadido un nuevo modo de funcionamiento. En lugar de colocar sencillamente el dedo de forma estática sobre el sensor, el dedo se desplaza lentamente a lo largo del mismo. El sensor sólo dispone de una estrecha zona sensible, y genera una secuencia completa de imágenes, las cuales pueden ser re-ensambladas, mediante un procesador, en una imagen completa. Las prestaciones se mejoran de modo apreciable y se garantiza la eliminación de cualquier grasa residual.

1.4. TARJETA BIOMÉTRICA

Una de las metas que este proyecto busca alcanzar es la inviolabilidad en el control de la asistencia de los estudiantes, así pues estudios anteriores demuestran que la biometría es un sistema de reconocimiento humano basado en las características físicas de las personas, se fija en "quién" es la persona, basándose en una única e inalterable característica humana que no puede ser perdida,

olvidada, sustraída o duplicada, por lo tanto, proporciona el máximo nivel de seguridad y facilidad de uso.

Por facilidad, confiabilidad, costos, el método más efectivo es el reconocimiento por huella dactilar, es por eso que se ha elegido el módulo de huella dactilar FIM5360 de NITGEN.

Mediante la incorporación de una CPU de gran velocidad y un algoritmo de reconocimiento de huella optimizado, los terminales FIM ofrecen una alta capacidad de reconocimiento y una gran velocidad para operaciones de identificación 1:N, y para la carga y descarga de datos, proporcionando las condiciones óptimas para su aplicación en sistemas de control de acceso. Todos los dispositivos FM disponen de entradas digitales para registro de huellas, identificación, borrado parcial o completo y reset.

En la Fig. 10 se muestra la imagen del sensor de huella dactilar FIM5360 es un módulo de reconocimiento de huella digital autónomo compuesto por un sensor óptico y una placa de procesado, además ofrece un entorno de desarrollo cómodo y seguro para aplicaciones on-line y off-line.

Fig. 10 Sensor de huella Dactilar FIM 5360

Fuente: www.Nitgen.com

Enseguida se detalla sus principales características.

- Funcionalidad de identificación de huella dactilar on-line y off-line.
- Diseño optimizado para aplicaciones de control de acceso: tiempo de identificación reducido mediante algoritmo de reconocimiento 1:1 y 1:N.
- Algoritmo y sensor óptico de elevada dureza (7 Moh).
- Rápida adquisición de todo tipo de huellas bajo cualquier condición.
- Tasa de identificaciones muy elevada: FAR: 0.001% y FRR: 0.1%.
- Memoria con capacidad para 1.000 dedos (cada dedo registra 2).
- Métodos de autentificación: verificación 1:1 e identificación 1:N. El acceso al dispositivo desde el host puede protegerse por huella o password.
- Memorización de eventos: hasta 30.000 autenticaciones.
- Interfaz de comunicación RS-232.
- Ofrece un entorno de desarrollo cómodo sin necesidad de conexión a PC
- Protocolo de comunicaciones ASCII.
- Tensión de alimentación de 5V.
- Compatible con normativa RoHS.

Las características que ofrece esta tarjeta ha sido la razón que ha justificado su elección dentro de la gama de sensores de huella dactilar que se encuentran en el mercado, además esta tarjeta por ser autónoma es precisa para el desarrollo de sistemas embebidos, otras de sus aplicaciones pueden ser:

- Sistemas de control de acceso
- Sistemas de control de presencia
- Sistemas de gestión de asistencia laboral
- Cajeros automáticos
- Terminales de punto de venta
- Otras aplicaciones en las que se requiera identificación cómoda y segura.

1.4.1. Especificaciones de Hardware

La Tabla. 2 presenta la especificaciones de hardware del módulo FIM 5360, el módulo trabaja con una tensión de alimentación de 3.3 Vcd, lo que garantiza un ahorro en el consumo de energía prolongando la vida útil de la batería de alimentación, el módulo como tal posee una memoria RAM con una capacidad de almacenamiento de 1000 usuarios, con el registro de dos huella por cada uno es decir que teóricamente es capaz de almacenar 2000 huellas dactilares.

ITEM		FIM5360
Board Spec.	CPU	S3C2410 (ARM9 266Mhz)
	DRAM	16MByte SDRAM
	Flash ROM	8MByte
Dimension		43 x 60 [mm²]
Sensor		NITGEN OPP06
Supply Voltage		5 / 3.3 [V]
Current Consumption	Normal	70 [mA]
	Max	220 [mA]
Operating Temperature		-20 ~ 60 [℃]
Humidity		~ 90 [% RH]
ESD Tolerance		±8 [KV] (indirect)
Communication Channel		RS-232 level UART Speed: 9600 ~ 115200 [bps] (1 start bit, 8 data bit, no parity, 1 stop bit)
Maximum Template Storage		Up to 2,000 templates
Maximum Log Storage		Up to 30,705 Logs

Tabla 2 Especificaciones de hardware

Fuente: DataSheet módulo FIM5360

1.4.2. Diagrama de bloques del sensor de huella Dactilar

En la Fig. 11 se muestra el diagrama de bloques del módulo de huella dactilar FIM 5360 y las interfaces de comunicación de la misma.

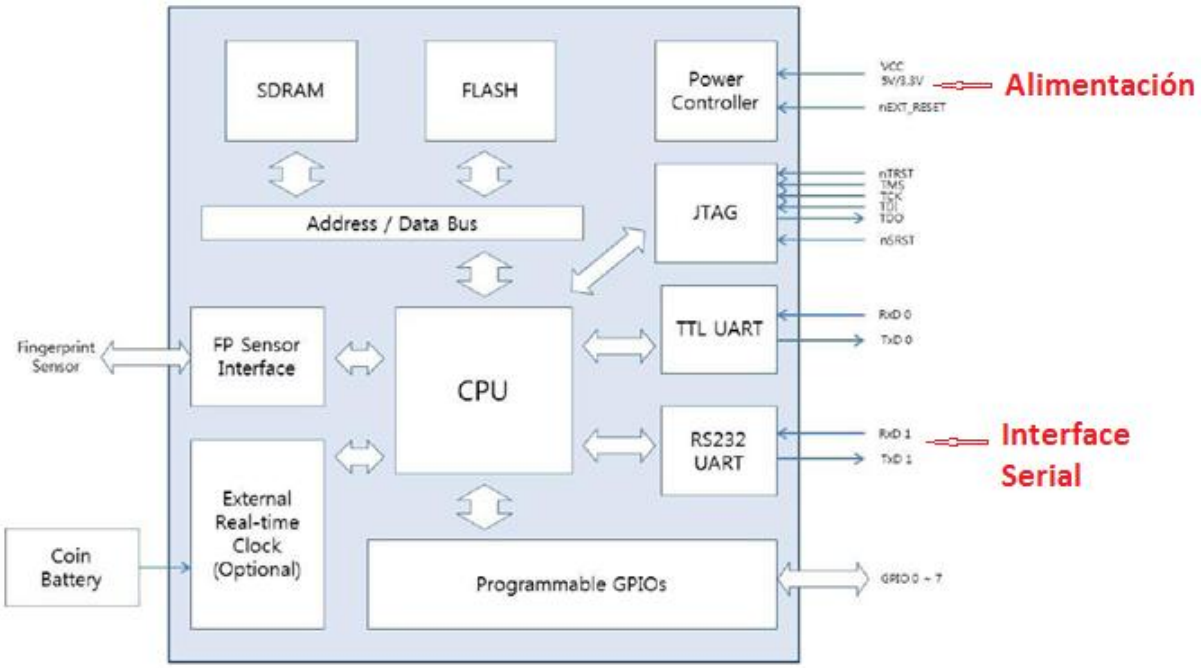

Fig. 11 Diagrama de bloques del módulo FIM 5360

Fuente: Autores

La tarjeta esta compuesta por un sensor capacitivo quien a traves de un bus de datos se comunica con la tarjeta quien internamente posee un regulador de voltaje (Power Controller), que estabiliza el voltaje de alimentación a 3.3 Vcd, una interfaz de comunicación RS232 UART, TTL UART, JTAG para diferenctes aplicaciones, un CPU que se encarga del procesamiento de las diferentes tareas del módulo, una memoria Flash para el almacenamiento de la huella en aplicaciones on-line, Entradas y salidas de propósito general GPIOS para el envio de señales de control para aplicaiones off-line.

La interface para comunicación RS232, se encuentra configurada por defecto a 8 bits de datos, sin paridad, 1 bit de inicio y 1 bit de parada, esta configuración puede ser cambiada por el usuario, ya qe soporta 6 modos de baudrate: 9600, 14400, 19200, 38400, 57600, y 115200 bps, el cambio de estos parámetros se la realizará con el uso de sus comandos de programación que se los detallará en los Anexo 3.

El dispositivo cuenta con un socket de 9 pines (JP2) para la comunicación y operación de la tarjeta, el detalle de sus pines se muestra en la Tabla 3.

Pin	Pin Name	Description
1	VCC	Supply Voltage (HV – 5V, LV – 3.3V)
2	EXT_RXD	UART Channel 1 port receiving signal from host (RS232 Level)
3	EXT_TXD	UART Channel 1 port transmitting signal to host (RS232 Level)
4	GPIO0	General Purpose Input / Output 0
5	GPIO1	General Purpose Input / Output 1
6	GPIO2	General Purpose Input / Output 2
7	GPIO3	General Purpose Input / Output 3
8	GPIO4	General Purpose Input / Output 4
9	GND	Ground

Tabla 3 Detalle de pines de socket JP2
Fuente: Datasheet módulo FIM5360

Alimentación de 3.3 Vcd se suministra por pin 1 y 9, la comunicación serial RS232 a traves de los pines 2 y 3 para aplicaciones en modo on-line, ademas los pines 4,5,6,7,8 son entradas y salidas GPIO que me permiten realizar el registro, identificación, verificación y borrado de usuarios solo con el uso de pulsos lógicos para el caso de aplicaciones off-line en las que el módulo deba trabajar de forma autónoma.

Dado a las caracteristicas del proyecto la tarjeta FIM5360 sera utilizada en modo on-line, las primeras pruebas se las realizaron a traves de una PC con la finalidad de conocer de mejor manera las caracteristicas y las funciones que contiene la misma, para ello contamos con el software EVTools cortecía de NITGEN.

1.5. ARDUINO

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. Se creó para artistas,

diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida.

1.5.1. Microcontrolador

Un microcontrolador es un circuito integrado, que incorpora en su interior los bloques básicos para formar un sistema embebido o una PC en menor escala, es decir el microcontrolador es un chip en cuyo interior encontramos una CPU, Memoria, Reloj, Puertos de Comunicación y Módulos Periféricos de E/S. Cada uno de estos bloques internos, cumple una función específica y permite al diseñador un mejor control de los procesos del sistema, el CPU se dice que es un microprocesador en pequeño y de menor potencia, la Memoria que sirve para almacenar el Firmware o programa a ejecutar, el Reloj provee una señal de sincronización para todos las tareas del sistema, los Puertos de comunicación le permiten al microcontrolador tener comunicación bi-direccional con otros microcontroladores o un PC, por ejemplo puerto RS232, USB, ISP, I2C, y los Módulos Periféricos de E/S que permiten el intercambio de información de tipo digital o análoga con el exterior del sistema, es dentro de estos periféricos que se pueden encontrar: Puertos Digitales E/S, Conversores Análogo digital, Contadores, Temporizadores, Módulos PWM, entre otros.

Además de su estructura un microcontrolador posee ciertas características de desarrollo, como son: su lenguaje de programación, el IDE para la escritura de programas, la forma en que es programa la memoria interna, el hardware externo necesario para realizar esta grabación. Son estas características las que hacen la diferencia al momento de la elección correcta de un tipo de microcontrolador.

Existen decenas de empresas fabricantes de microcontroladores, entre las que podemos nombrar: Intel, Motorola, Texas Instrument, Microchip, Cypress, Atmel, entre otras. Pero dentro de toda esta gama se destacan dos familias de microcontroladores: la familia AVR y la familia PIC, cuya popularidad es alta entre diseñadores de sistemas embebidos que requieren un rendimiento alto y bajo costo, y eligen uno u otro ya sea por su nivel de integración, por su arquitectura, la disponibilidad de recursos o su lenguaje de programación.

1.5.2. Microcontrolador Atmel AVR

Los AVR son una familia de microcontroladores fabricada por la compañía noruega ATMEL, estos microcontroladores de 8 bits cuentan con una CPU RISC y su memoria de programa viene implementada en FLASH, cuentan con periféricos como puertos Digitales, ADC, PWM, entre otros.

El microcontrolador por ser un sistema digital programable, necesita de un código de programa o firmware que incluya las instrucciones necesarias para realizar el control del sistema embebido. El lenguaje de programación de un microcontrolador, es el Lenguaje Ensamblador (.asm), lenguaje de bajo nivel.

Pero dentro del mundo de los microcontroladores podemos encontrar Compiladores de un lenguaje de alto nivel a ensamblador o mejor aún a Lenguaje de Maquina (.hex). Para los microcontroladores AVR podemos encontrar compiladores de lenguaje C, C++, Basic, cada uno de ellos brinda distintas ventajas, una de ellas es el hecho de no tener que aprender Ensamblador y trabajar en un lenguaje que el usuario domine, además cada uno cuenta con IDE (Ambiente Integrado de Desarrollo) para un mejor diseño de los programas.

Como ventaja principal se debe citar que todos estos IDE se pueden descargar gratis o en versiones Demo desde la web de sus respectivos fabricantes. Existen algunas plataformas educacionales de desarrollo basadas en micros AVR, como el

ARDUINO, que básicamente es un chip AVR con un bootloader, lo cual hace que sea aún más sencilla la programación.

Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo costo que permiten el desarrollo de múltiples diseños, como lo son los microcontroladores Arduino Diecimila, Arduino Duemilanove y Arduino Mega, en la Tabla. 4, se presenta una comparación entre los microcontroladores mencionados.

	Atmega168	**Atmega328**	**Atmega1280**
Voltaje operativo	5 V	5 V	5 V
Voltaje de entrada recomendado	7 - 12 V	7 - 12 V	7 - 12 V
Voltaje de entrada límite	6 - 20 V	6 - 20 V	6 - 20 V
Pines de entrada y salida digital	14 (6 PWM)	14 (6 PWM)	54 (14 PWM)
Pines de entrada analógica	6	6	16
Intensidad de corriente	40 mA	40 mA	40 mA
Memoria Flash	16KB (2KB bootloader)	32KB (2KB bootloader)	128KB (4KB bootloader)
SRAM	1 KB	2 KB	8 KB
EEPROM	512 bytes	1 KB	4 KB
Frecuencia de reloj	16 MHz	16 MHz	16 MHz

Tabla 4 Características de los Microcontroladores Atmega

Fuente: www.Arduino.com.cc

Para esta aplicación será utilizada la Placa Arduino Uno. El Arduino Uno es una placa electrónica basada en el microprocesador Atmega328. Tiene 14 entradas/salidas digitales y 6 de estas pueden utilizarse para salidas PWM (Pulse Width Modulation). Además tiene 6 entradas analógicas, un oscilador de 16MHz,

una conexión USB, un conector de alimentación, un header ICSP y un pulsador de reseteo. La placa lleva todo lo necesario para soportar el microprocesador.

Para empezar a utilizar la placa sólo es necesario conectarla al ordenador a través de un cable USB, o alimentarla con un adaptador de corriente AC/DC. También, puede alimentarse sencillamente con una batería.

Una de las características principales de la UNO es que no utiliza el convertidor USB-serial FTDI. Por lo contrario, ofrece el microprocesador Atmega8U2 programado como convertidor USB-serial. La Fig.12, presenta la placa Arduino Uno.

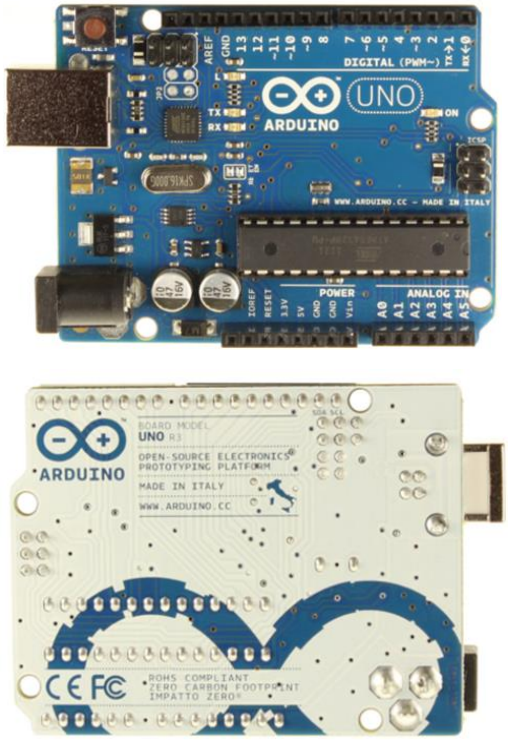

Fig. 12 Placa Arduino Uno

Fuente: www.Arduino.com.cc

1.5.3. Características

- Microprocesador ATmega328
- Tensión operativa 5V

- Tensión de alimentación (recomendado) 7-12V
- Tensión de alimentación (limites) 6-20V 14 Entradas / Salidas Digitales (6 de estas se pueden utilizar para salidas PWM)
- 6 Entradas Analógicas
- Máxima corriente continua para las entradas: 40 mA
- Máxima corriente continua para los pines 3.3V: 50 mA
- Flash Memory 32 KB (el bootloader usa 0.5 KB).
- SRAM 2 KB
- EEPROM 1 KB
- Velocidad del Clock 16 MHz

La Fig.13 muestra una descripción clara de las partes que conforman la placa Arduino Uno.

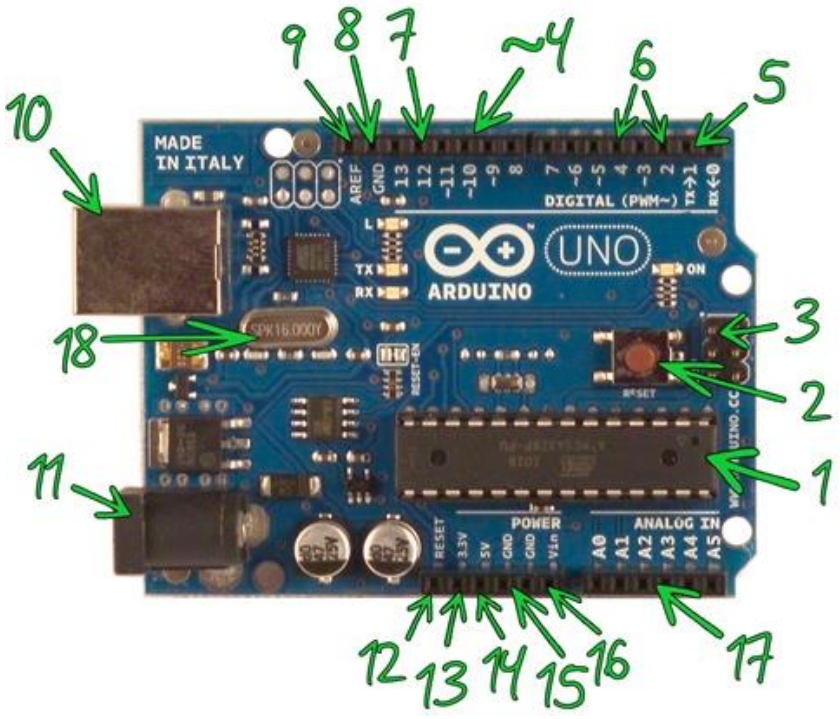

Fig. 13 Vista de componentes de la placa Arduino Uno

Fuente: http://www.elboby.com

El Uno y la versión 1.0 serán las versiones de referencia para los siguientes Arduinos. El Uno es el último de una serie de placas Arduino USB, y el modelo de referencia para la plataforma Arduino.

1.5.4. Componentes placa Arduino

1.- Microcontrolador ATmega328.- es un microcontrolador de la compañía Atmel que cuenta con 32KB de memoria flash, 2KB de memoria RAM y 1KB de memoria EEPROM. El microcontrolador puede ser utilizado como reemplazo del microcontrolador de las Freeduino o las Arduino Duemilanove o Diecimila o también puede utilizarse para realizar el montaje de una Arduino desde protoboard.

- Voltaje de Operación: 5V
- Memoria Flash: 32 KB de los cuales 512 bytes son utilizados por el bootloader
- SRAM 2 KB
- EEPROM 1 KB
- Velocidad del Reloj 16 MHz
- Bootloader preinstalado

2.- Botón Reset.- Suministrar un valor LOW (0V) para reiniciar el microcontrolador. Típicamente usado para añadir un botón de reset a los shields que no dejan acceso a este botón en la placa.

3.- ICSP.- Conector para la programación ICSP (In Circuit Serial Programming, o Programación Serial en circuito). El ICSP es el sistema utilizado en los dispositivos PIC para programarlos sin necesidad de tener que retirar el chip del circuito del que forma parte.

4.- PWM.- pines 3, 5, 6, 9, 10 y 11 provee de 8 bits de salida PWM con la función analog Write (). La modulación por ancho de pulsos (también conocida como PWM, siglas en inglés de pulse-width modulation) de una señal o fuente de energía es una técnica en la que se modifica el ciclo de trabajo de una señal periódica, ya sea para transmitir información a través de un canal de comunicaciones o para controlar la cantidad de energía que se envía a una carga.

5.- Serie: 0 (RX) y 1 (TX).- Se utiliza para recibir (RX) y transmisión (TX) datos serie TTL. Estos pines están conectados a los pines correspondientes de la ATmega8U2 USB-to-TTL de chips de serie.

6.- Interrupciones externas.- Pines 2 y 3 Estos pines pueden ser configurados para activar una interrupción en un valor bajo, un flanco ascendente o descendente, o un cambio en el valor.

7.- SPI.- 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK); Estos pines sirven de apoyo a la comunicación SPI con la biblioteca de SPI. El Bus SPI (del inglés Serial Peripheral Interface) es un estándar de comunicaciones, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos. El bus de interfaz de periféricos serie o bus SPI es un estándar para controlar casi cualquier dispositivo electrónico digital que acepte un flujo de bits serie regulado por un reloj.

8.- GND.- Pines de tierra. Abreviación de Ground que traducido al español es Tierra y en el contexto de la electrónica significa el común del circuito adonde se supone que existe 0 voltios.

9.- AREF.- Tensión de referencia para las entradas analógicas. Se utiliza con analog Reference ().

10.- USB.- El Arduino Uno tiene una serie de facilidades para comunicarse con una computadora, Usando los canales de comunicación de esta serie a través de USB y aparece como un puerto COM virtual en el ordenador. Utiliza el estándar de los controladores USB COM, y no necesita ningún controlador externo. Sin embargo, en Windows es necesario un archivo .inf. El RX y TX LED de la placa parpadean cuando se transmiten datos a través del USB al chip serie y viceversa.

11.- Conector de alimentación.- Plug hembra de 2.1mm para la conexión de alimentación en la placa.

12.- Reset.- Suministrar un valor LOW(0V) para reiniciar el microcontrolador. Típicamente usado para añadir un botón de reset a los shields que no dejan acceso a este botón en la placa.

13.- 3.3 V.- Una fuente de voltaje a 3.3 voltios generada en el chip FTDI integrado en la placa. La corriente máxima soportada 50mA.

14.- 5V.- La fuente de voltaje estabilizado usado para alimentar el microcontrolador y otros componentes de la placa. Esta puede provenir de VIN a través de un regulador integrado en la placa, o proporcionada directamente por el USB u otra fuente estabilizada de 5V.

15.- GND.- Pines de toma de tierra.

16.- VIN.- La entrada de voltaje a la placa Arduino usa una fuente externa de alimentación (en opuesto a los 5 voltios de la conexión USB). Se puede proporcionar voltaje a través de este pin, o, si se está alimentado a través de la conexión de 2.1mm, acceder a ella a través de este pin.

17.- Analog In.- El Uno tiene 6 entradas analógicas, y cada una de ellas proporciona una resolución de 10bits (1024 valores). Por defecto se mide de tierra a 5 voltios, aunque es posible cambiar la cota superior de este rango usando el pin AREF y la función analog Reference.

18.- Cristal.- Un cristal oscilador a 16Mhz, El oscilador de cristal tiene estabilidad de frecuencia y pureza de fase, dada por el resonador. La frecuencia es estable frente a variaciones de la tensión de alimentación.

Software ampliable y de código abierto- El software Arduino está publicado bajo una licencia libre y preparada para ser ampliad por programadores experimentados. El lenguaje puede ampliarse a través de librerías de C++, y si se está interesado en profundizar en los detalles técnicos, se puede dar el salto a la programación en el lenguaje AVR C en el que está basado. De igual modo se puede añadir directamente código en AVR C en tus programas si así lo deseas.

Arduino hace fácil escribir código y cargarlo a la placa E/S, funciona en Windows, Mac OS X y Linux. El entorno está escrito en Java y basado en Processing, avr-gcc y otros programas también de código abierto, la descarga de su Software se lo puede hacer desde la página principal de Arduino www.arduino.cc, la Fig. 14, vemos la presentación de software.

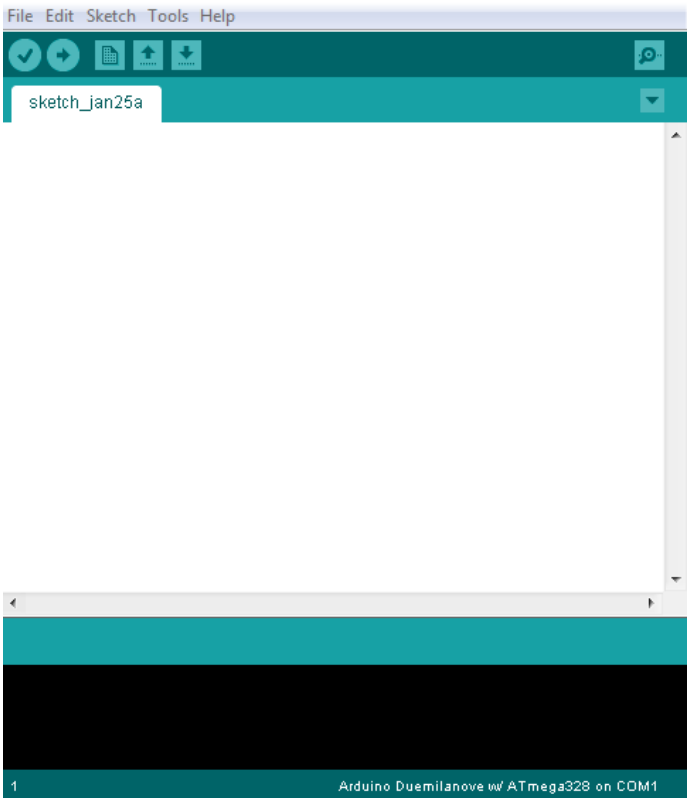

Fig. 14 Software de Programación Arduino

Fuente: www.Arduino.com.cc

El entorno de Desarrollo Arduino está constituido por un editor de texto para escribir el código, un área de mensajes, una consola de texto, una barra de herramientas con botones para las funciones comunes, y una serie de menús.

Permite la conexión con el hardware de Arduino para cargar los programas y comunicarse con ellos.

Arduino utiliza para escribir el software lo que denomina "sketch" (programa). Estos programas son escritos en el editor de texto. Existe la posibilidad de cortar/pegar y buscar/remplazar texto. En el área de mensajes se muestra información mientras se cargan los programas y también muestra errores.

La consola muestra el texto de salida para el entorno de Arduino incluyendo los mensajes de error completos y otras informaciones. La barra de herramientas permite verificar el proceso de carga, creación, apertura y guardado de programas, y la monitorización serie, en la Tabla. 5 se muestra los botones principales de Arduino.

	Verify/Compile Chequea el código en busca de errores y carga el programa en la placa Arduino.
	Stop Finaliza la monitorización serie y oculta otros botones
	New Crea un nuevo sketch.
	Open Presenta un menú de todos los programas sketch de su "sketchbook", (librería de sketch). Un click sobre uno de ellos lo abrirá en la ventana actual.
	Save Salva el programa sketch.
	Upload to I/O Board Compila el código y lo vuelca en la placa E/S de Arduino.
	Serial Monitor Inicia la monitorización serie Monitorización Serie.

Tabla 5 Botones de Software Arduino

Fuente: www.Arduino.com.cc

Encontrará otros comandos en los cinco menús: File, Edit, Sketch, Tools, Help. Los menús son sensibles al contexto, lo que significa que estarán disponibles sólo los elementos relevantes para la tarea que esté realizando en ese momento.

1.6. WIRELESS

Wireless (inalámbrico o sin cables) es un término usado para describir las telecomunicaciones en las cuales las ondas electromagnéticas (en vez de cables) llevan la señal sobre parte o toda la trayectoria de la comunicación. Algunos dispositivos de monitorización, tales como alarmas, emplean ondas acústicas a frecuencias superiores a la gama de audiencia humana; éstos también se clasifican a veces como wireless.

Los primeros transmisores sin cables vieron la luz a principios del siglo XX usando la radiotelegrafía (código Morse). Más adelante, como la modulación permitió transmitir voces y música a través de la radio, el medio se llamó radio. Con la aparición de la televisión, el fax, la comunicación de datos, y el uso más eficaz de una porción más grande del espectro, se ha resucitado el término wireless.

1.6.1. WI-FI

El router inalámbrico más popular es el Wi-Fi por ser el más utilizado para acceder a Internet desde cualquier lugar donde haya un punto de acceso (Access Point o AP), sobre todo en portátiles y PDAs con tarjeta Wi-Fi. También conocido como 802.11, es el dispositivo que reúne el conjunto de estándares para la WLAN (Wireless Local Area Network - red de área local inalámbrica). El estándar IEEE 802.11 es una frecuencia de radio desarrollado por el IEEE (Institute of Electrical and Electronics Engineers - Instituto de Ingenieros Eléctricos y Electrónicos), y la mayoría de los sistemas operativos lo soportan, así como muchos de los portátiles, celulares/móviles de última generación, consolas, impresoras y otros periféricos.

Los tipos de comunicación WIFI se basan en las diferentes clases de estándares IEEE, siendo la mayoría de los productos de la especificación b y de la g:

1.6.2. Estándares IEEE

El Comité 802, o proyecto 802, del Instituto de Ingenieros en Eléctrica y Electrónica (IEEE) definió los estándares de redes de área local (LAN). La mayoría de los estándares fueron establecidos por el Comité en los 80´s cuando apenas comenzaban a surgir las redes entre computadoras personales.
El estándar IEEE 802.11 define el uso de los dos niveles inferiores de la arquitectura OSI (capas física y de enlace de datos), especificando sus normas de funcionamiento en una WLAN, así mismo este estándar tiene una sub clasificación:

- 802.11a
- 802.11b
- 802.11g
- 802.11n

1.6.2.1. Estándar 802.11a

Emite a una velocidad de 54 Mb/seg (megabytes por segundo), volumen de información (Throughput) de 27 Mb/seg, Banda de frecuencia de 5 GHz.

El IEEE creaba en 1997 el estándar 802.11 con velocidades de transmisión de 2Mb/seg, hasta que en 1999 desarrollaron el estándar 802.11a que era una revisión del estándar original y que utiliza el mismo juego de protocolos de base que este. También llamado Wi-Fi 5, el estándar 802.11a opera en la banda de 5 Ghz que está menos congestionada y utiliza la modulación OFDM (orthogonal frequency-division multiplexing) con 52 sub portadoras, lo que le infiere dos notables ventajas respecto al 802.11b: incrementa la velocidad máxima de transferencia de datos por canal (de 11 Mbps a 54 Mbps) y aumenta el número de canales sin solapamiento.

Pero el uso de esta banda también tiene sus desventajas, puesto que restringe el uso de los equipos 802.11a sólo a puntos en línea de vista, siendo necesario la instalación de un mayor número de puntos de acceso 802.11a para cubrir la misma zona; debido a esto las ondas no pueden penetrar tan lejos como los del estándar 802.11b, ya que estas son más fácilmente absorbidas por las paredes y otros objetos sólidos en su camino pues su longitud de onda es menor.

1.6.2.2. Estándar 802.11b

Emite a una velocidad de 11 Mb/seg, volumen de información (Throughput) de 5 Mb/seg, banda de frecuencia de 2,4 GHz.

Uno de los más usados, desarrollado en 1999, es una extensión directa de la técnica de modulación definida en el estándar original 802.11. Su espectacular incremento en Throughput (volumen de información que fluye a través de las redes de datos) comparado con el estándar original junto con sustanciales reducciones de precios ha llevado a la rápida aceptación de 802.11b como la tecnología inalámbrica LAN definitiva.

Como desventaja los dispositivos 802.11b sufren interferencias de otros productos operando en la banda 2.4 GHz, como pueden ser hornos microondas, dispositivos Bluetooth, monitores de bebés y teléfonos inalámbricos. Por otro lado, los productos de estándar 802.11b no son compatibles con los productos de estándar 802.11a por operar en bandas de frecuencia distintas.

1.6.2.3. Estándar 802.11g

Emite a una velocidad de 54 Mb/seg, volumen de información (Throughput) de 22 Mb/seg, banda de frecuencia de 2.4 GHz.
Desarrollado en 2003, el 802.11g es el tercer estándar de modulación y la evolución del802.11b, es además el más usado en la actualidad. Los productos IEEE 802.11g poseen un alto grado de compatibilidad con versiones anteriores

pues trabaja en la banda de 2.4 GHz como 802.11b, pero usa el mismo esquema de transmisión basado en OFDM como802.11a, utilizando 48 sub portadoras.

802.11g fue rápidamente adoptado por los consumidores en Enero de 2003, antes de su ratificación en Junio, debido al deseo de velocidades de transmisión superiores y reducciones en los costes de fabricación. Para el verano de 2003, la mayoría de los productos de doble banda 802.11a/b pasaron a ser dual-band/tri-mode (doble banda/tres modos), esto quiere decir que pueden funcionar en la banda de 2.4 GHz o de 5 GHz y en cualquiera de los tres modos aceptados por la IEEE: el a, b y g.

Como el estándar 802.11b, los dispositivos de estándar 802.11g les afectan las interferencias de otros productos operando en la banda de 2.4 GHz.

1.6.2.4. Estándar 802.11n

Emite a una velocidad de 600 Mb/seg, volumen de información (Throughput) de 144 Mb/seg, bandas de frecuencia: 2,4 GHz y 5 GHz.

El estándar 802.11n (todavía en desarrollo) es una ratificación que mejora los previos estándares 802.11 añadiendo la tecnología MIMO que son antenas Multiple-Input Multiple-Output, unión de interfaces de red (Channel Bonding), además de agregación de marco a la capa MAC.

- **Mimo**.- genera cuatro canales de tráfico simultáneos de 72.2 Mbps para enviar y recibir datos a través de la incorporación de varias antenas.

- **Channel Bonding**.- también conocido como canal 40 MHz, puede usar simultáneamente dos canales separados no superpuestos de 20 MHz, lo que permite incrementar enormemente la velocidad de datos transmitidos.

La velocidad real de transmisión se estima que podría llegar a los 600 Mbps, que es 10 veces más rápida que bajo los estándares 802.11a y 802.11g, y cerca de 40 veces más rápida que bajo el estándar 802.11b.

1.6.3. Módulo Wifi Shield de Arduino

El sistema en desarrollo consta además del módulo de adquisición de huella dactilar de una tarjeta que me permita realizar la comunicación inalámbrica, transmitiendo los datos de cada usuario hacia la base de datos de un computador, para esto se ha ocupado la tarjeta Wifi Shield de Arduino. Fig. 15.

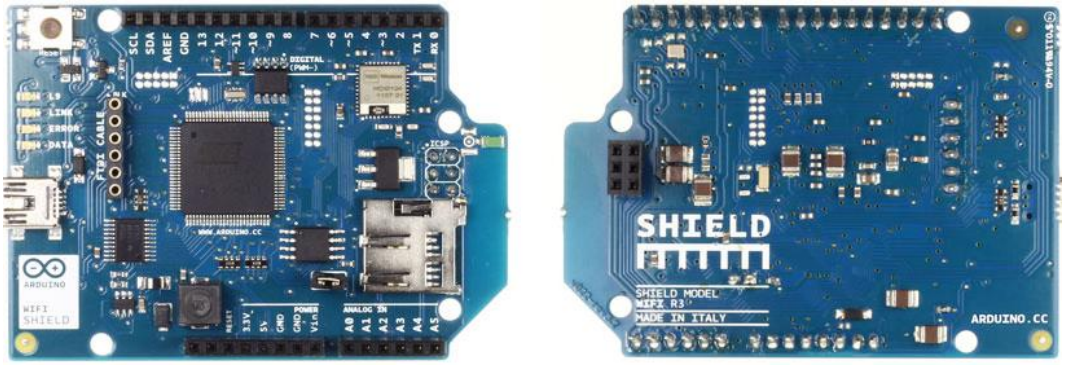

Fig. 15 Vista Frontal de la WIFI SHIELD de Arduino
Fuente: http://arduino.cc/en/Main/ArduinoWiFiShield

El Arduino WiFi Shield permite a una placa Arduino conectarse a Internet a través de la especificación inalámbrica 802.11 (Wi-Fi). Se basa en la HDG104 Wireless LAN 802.11b / g Sistema in-Package. Un 32UC3 Atmega proporciona una red (IP) de pila capaz de TCP y UDP.

El Arduino se comunica a través de la SPI con AVR32 y este a su vez controla el módulo de WIFI, un HDG104 de H & Wireless D. Estas son algunas de sus características:

- Requiere Arduino UNO
- Alimentación: 5V (proporcionado por la tarjeta Arduino)

- Red: 802.11b/g
- Encriptaciones soportadas: WEP y WPA2
- Conexión con Arduino por el puerto SPI
- Socket para tarjeta Micro SD incorporado
- Pines ICSP
- Conexión FTDI para debug
- Conexión Mini-USB para actualizaciones de Firmware

La Fig.16 muestra los pines de conexión de la Wifi Shield con Arduino, para lo que tan solo se necesita montar la placa Wifi Shield sobre la placa Arduino.

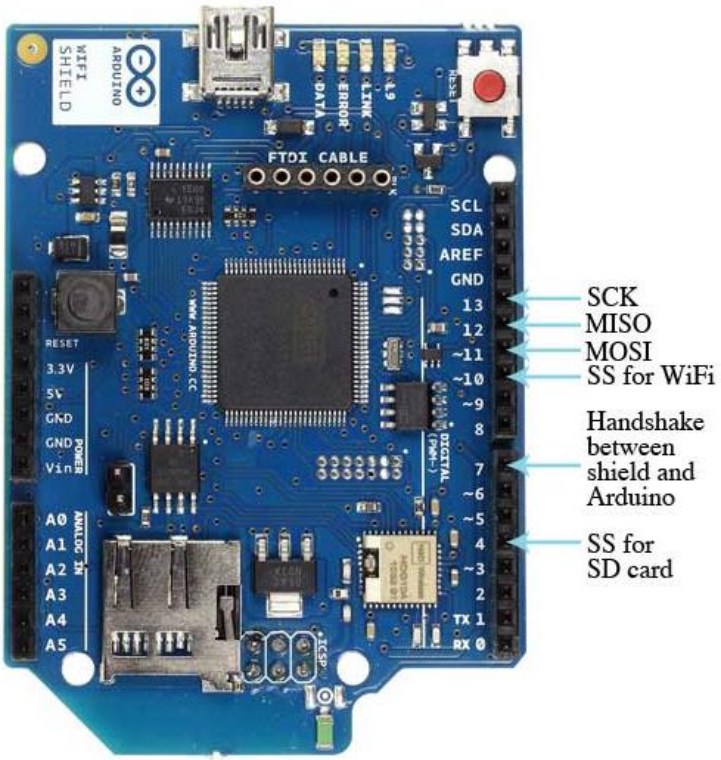

Fig. 16 Diagrama de Conexiones de la Wifi Shield con Arduino
Fuente: http://arduino.cc/en/Guide/ArduinoWiFiShield

El escudo Wifi se conecta a una placa Arduino con largos pines que se extienden a través del escudo, esto mantiene la disposición de las clavijas intacta y permite que otro escudo pueda ser apilado en la parte superior o que conexiones externas sea realizada.

Hay una ranura micro-SD en la tarjeta, que puede ser usado para almacenar archivos para servir a través de la red. Es compatible con el Arduino Uno y Mega. Se puede acceder al códec de lector de tarjetas micro SD a través de la biblioteca SD. Cuando se trabaja con esta biblioteca, SS es el pin 4.

Arduino se comunica tanto con el escudo Wifi del procesador y tarjeta SD con el bus SPI (a través de la cabecera ICSP). Esto es en los pines digitales 11, 12 y 13 sobre el Arduino Uno y los pines 50, 51, y 52 en el Arduino Mega. En ambas tablas, el pin 10 se utiliza para seleccionar el HDG104 y el pin 4 para la tarjeta SD. Estas patillas no se puede utilizar para general I / O. En el Mega, el pin SS es el 53, no se utiliza para seleccionar ya sea el HDG104 o la tarjeta SD, pero debe mantenerse como una salida o la interfaz SPI no funcionará, digital pin 7 se utiliza como un apretón de manos entre el pin escudo WiFi y Arduino, y no debe ser utilizado.

La Fig.17 presenta las interfaces de la Wifi Shield.

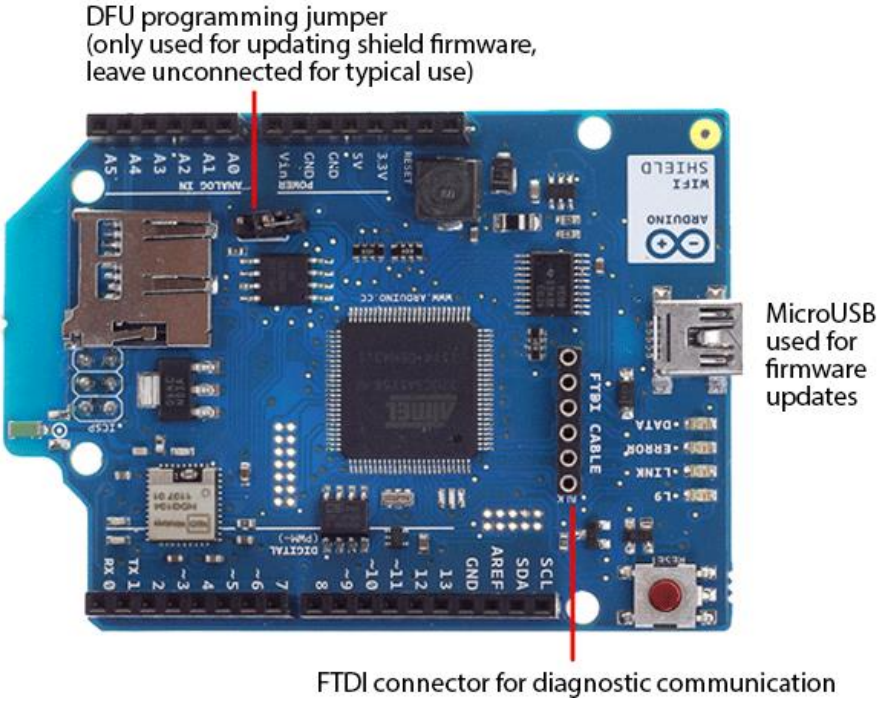

Fig. 17 Puertos del módulo Wifi Shield

Fuente: http://arduino.cc/en/Guide/ArduinoWiFiShield

Hay una interfaz mini USB. Esto no es para la programación de un Arduino adjunto, esta interfaz es ocupada para la actualización de los 32U ATMEGA. El jumper de programación adyacente al bus de alimentación y las entradas analógicas que se ve en la Fig. 17 debe dejarse sin conectar para el uso típico, sólo se utiliza para el modo de programación DFU.
Una conexión FTDI permite la comunicación serial con el 32U con fines de depuración, además el escudo contiene un número de LEDs informativos:

- L9 (amarillo): esto está ligado al pin digital 9
- LINK (verde): indica una conexión a una red
- ERROR (rojo): indica cuando hay un error de comunicación
- DATA (azul): indica los datos que se están transmitiendo / recibiendo

El módulo Wifi Shield se conecta a las redes abiertas, así como los de cifrado utilizando encriptación WEP y WPA2. El módulo Wifi no puede conectarse a redes que utilizan cifrado WPA2 Enterprise.

El SSID (nombre de red) debe ser transmitido por el escudo de la conexión, dependiendo de la configuración del router inalámbrico, necesitará información diferente. Para una red abierta (no cifrada), se necesita mencionar tan solo el SSID.
Para redes que utilizan el cifrado WPA/WPA2 Personal, usted necesita el SSID y la contraseña.

1.6.4. Topologías de redes inalámbricas

Se define como topología a la disposición lógica o a la disposición física de una red, existen tres tipos de tecnología WLAN.

- Ad-hoc
- Infraestructura

1.6.4.1. Red Ad Hoc

Fig. 18 Red Ad-hoc
Fuente: http://ieeestandards.galeon.com/aficiones1573328.html

Una red Ad-Hoc es una red formada por dispositivos de comunicación inalámbricos que se conectan por periodos de duración corta, estos dispositivos se pueden comunicar sin necesidad de ningún AP o infraestructura existente, Fig. 18.

Las configuraciones "Ad hoc", son comunicaciones de tipo punto a punto, solamente los ordenadores dentro de un rango de transmisión definido pueden comunicarse entre ellos. La tecnología es utilizada en varios campos como en el ejército, celulares y juegos de videos, en las redes Ad-hoc, cada terminal de comunicación se comunica con sus compañeros para hacer una red "peer to peer" o punto a punto.

Las redes Ad Hoc consisten en un conjunto de nodos que se comunican mediante enlaces inalámbricos y que no tienen una infraestructura fija, esto implica que no tienen ningún tipo de control centralizado y que por lo tanto son flexibles y fácilmente desplegables.

1.6.4.1.1. Principales características de las redes Ad Hoc

- Terminales autónomos: Cada Terminal se comporta como un nodo autónomo que puede funcionar como emisor, receptor o encaminador.
- Conexiones inalámbricas: No existe ningún tipo de infraestructura fija, los terminales usan el aire como canal de comunicación.

- Funcionamiento distribuido: No existe ningún elemento central que se encargue de la gestión y el control de la red, todos los nodos son iguales y por lo tanto la gestión está distribuida.
- Topología dinámica: Como no existe ninguna infraestructura fija y además los nodos son móviles, la topología de la red puede ser altamente cambiante. Las redes MANET deben adaptarse rápidamente a los cambios de tráfico generado por los nodos, a los distintos patrones de movimientos y a las condiciones de propagación.
- Capacidad variable de los enlaces: Al tratarse de un medio de transmisión compartido el canal de transmisión cambia constantemente los niveles de ruido, atenuación e interferencias. Además, en una transmisión extremo a extremo puede participar varios enlaces distintos y la ruta puede cambiar varias veces en una misma transmisión.
- Consumo de energía: Los nodos son móviles y por lo tanto es de suponer que funcionan con baterías de vida limitada, por esa razón es muy importante que el consumo de energía se reduzca lo máximo posible.

1.6.4.2. Modo Infraestructura

Contrario al modo ad hoc donde no hay un elemento central, en el modo de infraestructura hay un elemento de "coordinación": un punto de acceso o estación base. Si el punto de acceso se conecta a una red Ethernet cableada, los clientes inalámbricos pueden acceder a la red fija a través del punto de acceso. Para interconectar muchos puntos de acceso y clientes inalámbricos, todos deben configurarse con el mismo SSID. Para asegurar que se maximice la capacidad total de la red, no configure el mismo canal en todos los puntos de acceso que se encuentran en la misma área física.

Los clientes descubrirán (a través del escaneo de la red) cual canal está usando el punto de acceso de manera que no se requiere que ellos conozcan de antemano el número de canal.

En redes IEEE 802.11 el modo de infraestructura es conocido como Conjunto de Servicios Básicos (BSS – Basic Service Set). También se conoce como Servidor y Cliente, la Fig. 19 muestra un ejemplo del modo Infraestructura.

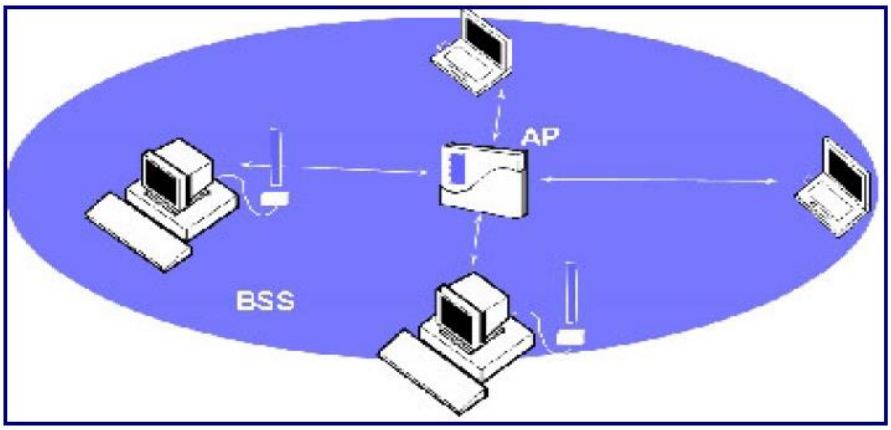

Fig. 19 Modo Infraestructura
Fuente:http://www.itrainonline.org/itrainonline/mmtk/wireless_es/files/04_es_topologia-e-infraestructura_guia_v02.pdf

Al ser una comunicación centralizada si cae el AP ninguno de los dispositivos podrá comunicarse entre sí.

1.6.5. Componentes de redes Inalámbricas

1.6.5.1. Punto de acceso

Un punto de acceso es un "concentrador" inalámbrico. El transmisor/receptor conecta entre sí los nodos de la red inalámbrica y normalmente también sirve de puente entre ellos y la red cableada. Un conjunto de puntos de (coordinados) se pueden conectar unos con otros para crear una gran red inalámbrica, un ejemplo de estos es el Access point de TP-Link de la Fig. 20.

Fig. 20 Router TP-LINK
Fuente: http://zapakshop.bigadda.com/catalog/product/view/id/20268/s/tp-link-54mbps-wireless-access-point-tl-wa500g/category/25/

Desde el punto de vista de los clientes inalámbricos (como las computadoras portátiles o las estaciones móviles), un punto de acceso provee un cable virtual entre los clientes asociados. Este "cable inalámbrico" conecta tanto a los clientes entre sí, como los clientes con la red cableada.

Un punto de acceso debe distinguirse de un enrutador inalámbrico, que es muy común en el mercado actual. Un enrutador inalámbrico es una combinación entre un punto de acceso y un enrutador, y puede ejecutar tareas más complejas que las de un punto de acceso. Considere un enrutador inalámbrico como un puente inalámbrico (entre la red inalámbrica y la red Ethernet) y un enrutador (con características de enrutamiento IP).

Los clientes se conectan a un punto de acceso mediante su nombre. Este mecanismo de identificación se conoce como SSID- Service Set Identifier- (Identificador del Conjunto de Servicio) y debe ser el mismo para todos los miembros de una red inalámbrica específica. Todos los puntos de Acceso y clientes que pertenecen a un mismo ESS -Extended Service Set- (Conjunto de Servicio extendido) se deben configurar con el mismo ID (ESSID).

Cuando hablamos de SSID pensamos en la etiqueta de un punto (socket) de Ethernet. Conectarse a una red inalámbrica con SSID "x" es equivalente a conectar su computador a un punto de red sobre una pared identificado con la etiqueta "x".

1.6.5.2. Clientes inalámbricos

Un cliente inalámbrico es cualquier estación inalámbrica que se conecta a una red de área local (LAN –Local Area Network) inalámbrica para compartir sus recursos. Una estación inalámbrica se define como cualquier computador con una tarjeta adaptadora de red inalámbrica instalada que transmite y recibe señales de Radio Frecuencia (RF). Algunos de los clientes inalámbricos más comunes son las computadoras portátiles, PDAs, equipos de vigilancia y teléfonos inalámbricos de VoIP.

1.6.6. Seguridad

La mayoría de los problemas de seguridad en WLAN son debidos al medio de transmisión utilizado, el aire, que es de fácil acceso para los atacantes, por ello, hay que establecer unos medios para asegurar la privacidad de nuestros datos.

- Medios Físicos
- Medios Lógicos (SW)

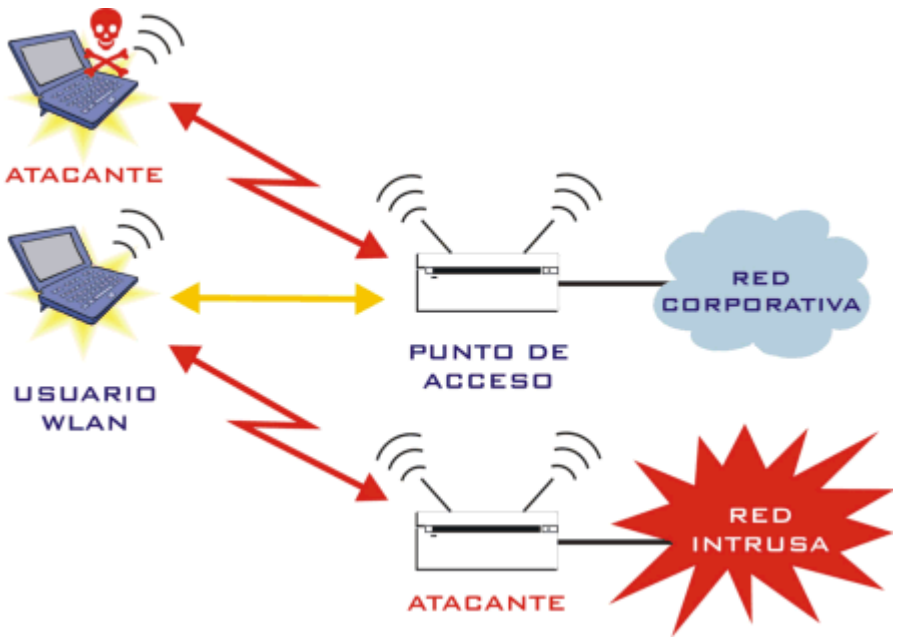

Fig. 21 Ataques a redes inalámbricas
Fuente: http://juancho0125.blogspot.com/2010/12/tipos-de-ataque-redes-inalambricas_09.html

1.6.6.1. Seguridad Lógica

Son técnicas de cifrado e integridad de la información y técnicas de autenticación/ autorización/ Accounting (AAA), estos dos tipos de técnicas pueden complementarse la una con la otra.

- Cifrado e integridad de la información.- Se encargan de mantener la privacidad de nuestros datos y de evitar posibles suplantaciones de personalidad en la comunicación, el cifrado se basa en claves compartidas previamente o que se asignan de forma dinámica.

MÉTODO	DESCRIPCIÓN	NIVEL DE SEGURIDAD
WEP – Wired Equivalent Privacy	Es el estándar de seguridad para redes WiFi, es un mecanismo de seguridad básico	BAJA
WPA – WiFi Protected Access	Método que opera con una contraseña predefinida , tiene cierta vulnerabilidad a ataques pero mucho menor que WEP	MEDIA
WPA2	WPA2 es una combinación de contraseña predeterminada y un software adicional para la creación de llaves únicas, de tal manera que cada usuario posee una llave de autenticación individual	ALTA - RECOMENDABLE

Tabla 6 Métodos de encriptación
Fuente: Los autores

1.7. MODELO OSI

Creada en 1947, la Organización Internacional de Estandarización (ISO, International Standars Organizaion).

Un estándar ISO que cubre todos los aspectos de las redes de comunicación es el modelo de Interconexión de sistemas abiertos (OSI, Open System Interconnection), un sistema abierto es un modelo que permite que dos sistemas diferentes se puedan comunicar independiente ente de la arquitectura subyacente, el objetivo del modelo OSI es permitir la comunicación entre sistemas distintos sin que sea necesario cambiar la lógica del Hardware o el software subyacente. El modelo OSI no es un protocolo es un modelo para comprender y diseñar una arquitectura de red flexible, robusta e interoperable.

El modelo OSI está compuesto por siete niveles ordenados, Fig. 22, al desarrollar el modelo, los diseñadores refinaron el proceso de transmisión hasta los elementos más fundamentales identificaron que funciones tienen usos relacionadas y unieron todas las funciones dentro de grupos discretos que se convirtieron en niveles.

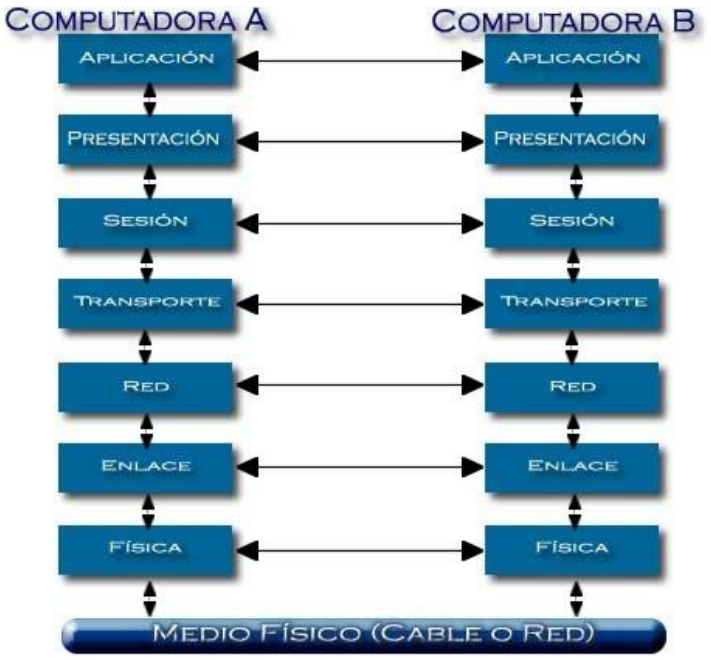

Fig. 22 Niveles del modelo OSI

Fuente: http://www.oocities.org/dralkzta/osi.htm

De acuerdo al lineamiento de esta investigación será hará énfasis solamente en la Capa de Transporte del modelo OSI.

1.7.1. Capa de Transporte

La capa transporte es el cuarto nivel del modelo OSI, es una capa de extremo a extremo (origen-destino) y por lo tanto deberá garantizar una comunicación estable entre los extremos, además de las tareas de inicio, mantenimiento y liberación de las comunicaciones, la capa de transporte se encarga de:

- La transferencia libre de errores de los datos entre el emisor y el receptor
- Mantener el flujo de la red. Es la base de toda la jerarquía de protocolo.
- Proporciona un transporte de datos confiable y económico de la máquina de origen a la máquina destino
- La capa de transporte recibe los datos de la capa de sesión y los multiplexa.

En la capa de Transporte se distinguen elementos como:

- **Direccionamiento.-** El método que normalmente se emplea es definir direcciones de transporte en las que los procesos pueden estar a la escucha de solicitudes de conexión.

- **Establecimiento de una conexión.-** Se envían TPDU (Unidad de Datos del Protocolo de Transporte) con la petición de conexión y esperar a que el otro acepte la conexión.

- **Liberación de una conexión.-** Hay dos estilos de terminación de una conexión: liberación asimétrica y liberación simétrica. La liberación asimétrica es la manera en que funciona el mecanismo telefónico: cuando una parte cuelga, se interrumpe la conexión. La liberación simétrica trata

la conexión como dos conexiones unidireccionales distintas, y requiere que cada una se libere por separado.

- **Control de Flujo y almacenamiento en buffer.-** En esta capa lo que se hace es, si el servicio de red no es confiable, el emisor debe almacenar en un buffer todas las TPDUs enviadas, igual que en la capa enlace de datos. Sin embargo, con un servicio de red confiable son posibles otros arreglos.

- **Multiplexión.-** existen dos tipos de multiplexión (hacia arriba y hacia abajo)

- Multiplexión hacia arriba.- si en un host sólo se dispone de una dirección de red, todas la conexiones de transporte de esa máquina tendrán que utilizarla. Cuando llega una TPDU, se necesita algún mecanismo para saber a cuál proceso asignarla.

- Multiplexión hacia abajo.- para la utilización de circuitos virtuales, que dan más ancho de banda cuando se re- asigna a cada circuito una tasa máxima de datos.

Esta capa presenta dos protocolos principales, uno orientado y otro no orientado a la conexión. Analizaremos brevemente estos dos protocolos para determinar cuál es el más apropiado a usarse en esta investigación, el protocolo no orientado a la conexión es UDP y el protocolo orientado a la conexión es TCP.

1.7.1.1. Protocolo UDP

UDP (Protocolo de Datagramas de Usuario), este protocolo proporciona una forma para que las aplicaciones envíen datagramas IP encapsulados sin tener que establecer una conexión.

UDP transmite segmentos que consisten en un encabezado de 8 bytes seguido por la carga útil, en la Fig. 23 se muestra tal encabezado. Los dos puertos sirven para identificar los puntos terminales dentro de las máquinas de origen y destino. Cuando llega un paquete UDP, su carga útil se entrega al proceso que está enlazado al puerto de destino. Sin los campos de puerto, la capa de transporte no sabría qué hacer con el paquete. Con ellos, entrega los segmentos de manera correcta

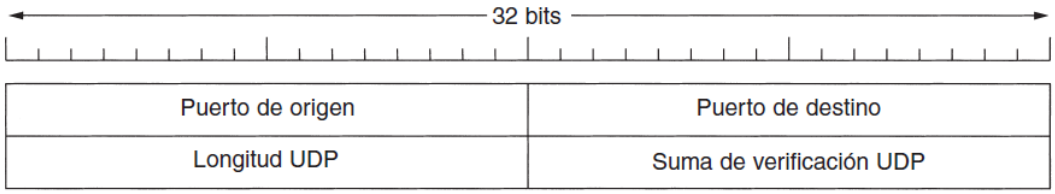

Fig. 23 Encabezado del protocolo UDP

Fuente: Libro Redes de Computadoras

El puerto de origen se necesita principalmente cuando debe enviarse una respuesta al origen, el campo Longitud UDP incluye el encabezado de 8 bytes y los datos. El campo Suma de verificación UDP es opcional y se almacena como 0 si no se calcula (un 0 calculado se almacena como 1s). Su desactivación no tiene sentido a menos que la calidad del servicio de los datos no importe (por ejemplo, en la voz digitalizada).

De igual manera vale la pena mencionar de manera explícita algunas de las cosas que UDP no realiza, por ejemplo no realiza control de flujo, control de errores o retransmisión cuando se recibe un segmento erróneo, proporciona una interfaz al protocolo IP con la característica agregada de demultiplexar varios procesos utilizando los puertos, el código no sólo es simple, sino que se necesitan muy pocos mensajes (uno en cada dirección) en comparación con un protocolo que requiere una configuración inicial.

Una aplicación que utiliza de esta manera a UDP es DNS (el Sistema de Nombres de Dominio), En resumen, un programa que necesita buscar la dirección IP de algún host, por ejemplo, www.cs.berkeley.edu, puede enviar al servidor DNS un paquete UDP que contenga el nombre de dicho host. El servidor responde con un

paquete UDP que contiene la dirección IP del host. No se necesita configuración por adelantado ni tampoco liberación posterior.

UDP es un protocolo simple y tiene algunos usos específicos, como interacciones cliente-servidor y multimedia, pero para la mayoría de las aplicaciones de Internet se necesita una entrega en secuencia confiable. UDP no puede proporcionar esto, para aplicaciones más robustas se necesita otro protocolo. Se llama TCP y es el más utilizado en Internet.

1.7.1.2. Protocolo TCP

TCP (Protocolo de Control de Transmisión) se diseñó específicamente para proporcionar un flujo de bytes confiable de extremo a extremo a través de una inter red no confiable. Una inter red difiere de una sola red debido a que diversas partes podrían tener diferentes topologías, anchos de banda, retardos, tamaños de paquete y otros parámetros. TCP tiene un diseño que se adapta de manera dinámica a las propiedades de la inter red y que se sobrepone a muchos tipos de fallas.

Cada máquina que soporta TCP tiene una entidad de transporte TCP, ya sea un procedimiento de biblioteca, un proceso de usuario o parte del kernel. En todos los casos, maneja flujos TCP e interactúa con la capa IP. Una entidad TCP acepta flujos de datos de usuario de procesos locales, los divide en fragmentos que no excedan los 64 KB (en la práctica, por lo general, 1460 bytes de datos que se ajustan en una sola trama Ethernet con los encabezados IP y TCP), y envía cada fragmento como un datagrama IP independiente. Cuando los datagramas que contienen datos TCP llegan a una máquina, se pasan a la entidad TCP, la cual reconstruye los flujos de bytes originales.

La capa IP no proporciona ninguna garantía de que los datagramas se entregarán de manera apropiada, por lo que corresponde a TCP terminar los temporizadores y retransmitir los datagramas conforme sea necesario. Los datagramas que llegan podrían hacerlo en el orden incorrecto; también corresponde a TCP re

ensamblarlos en mensajes en la secuencia apropiada. En resumen, TCP debe proporcionar la confiabilidad que la mayoría de los usuarios desean y que IP no proporciona.

El servicio TCP se obtiene al hacer que tanto el servidor como el cliente creen puntos terminales, llamados sockets, cada socket tiene un número (dirección), que consiste en la dirección IP del host, y un número de 16 bits, que es local a ese host, llamado puerto. Un puerto es el nombre TCP para un TSAP. Para obtener el servicio TCP, se debe establecer de manera explícita una conexión entre un socket en la máquina emisora y uno en la máquina receptora. Un socket puede utilizarse para múltiples conexiones al mismo tiempo. En otras palabras, dos o más conexiones pueden terminar en el mismo socket. Las conexiones se identifican mediante los identificadores de socket de los dos extremos, es decir (socket1, socket2). No se utiliza ningún otro número de circuitos virtuales ni identificador.
Los números de puerto menores que 1024 se llaman puertos bien conocidos y se reservan para servicios estándar. Por ejemplo, cualquier proceso que desee establecer una conexión a un host para transferir un archivo utilizando FTP puede conectarse con el puerto 21 del host de destino para conectar su dominio FTP.

La Fig. 24 muestra la distribución de un segmento TCP. Cada segmento comienza con un encabezado de formato fijo de 20 bytes. El encabezado fijo puede ir seguido de opciones de encabezado. Tras las opciones, si las hay, pueden continuar hasta 65,535 - 20 - 20 = 65,495 bytes de datos, donde los primeros 20 se refieren al encabezado IP y los segundos al encabezado TCP. Los segmentos sin datos son legales y se usan por lo común para confirmaciones de recepción y mensajes de control.

Realicemos la disección del encabezado TCP campo por campo. Los campos de Puerto de origen y Puerto de destino identifican los puntos terminales locales de la conexión. La dirección de un puerto más la dirección IP de su host forman un punto terminal único de 48 bits. Los puntos terminales de origen y de destino en conjunto identifican la conexión.

Los campos de Número de secuencia y Número de confirmación de recepción desempeñan sus funciones normales. Ambos tienen 32 bits de longitud porque cada byte de datos está numerado en un flujo TCP. La Longitud del encabezado TCP indica la cantidad de palabras de 32 bits contenidas en el encabezado TCP.

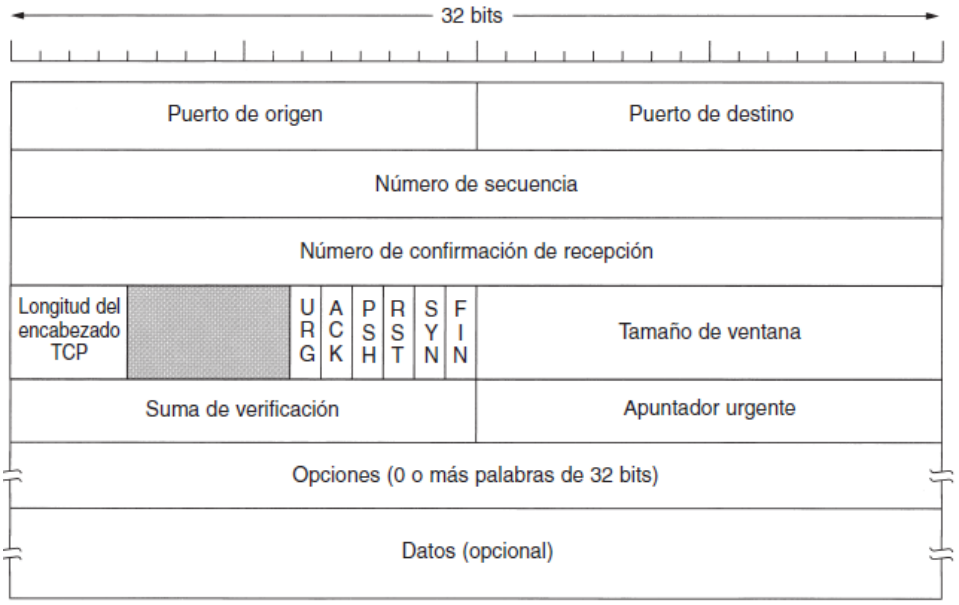

Fig. 24 Encabezado protocolo TCP

Fuente: Libro Redes de Computadoras

1.8. MY SQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento.

Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo, en la Fig.25 se muestra la pantalla principal de MySQL.

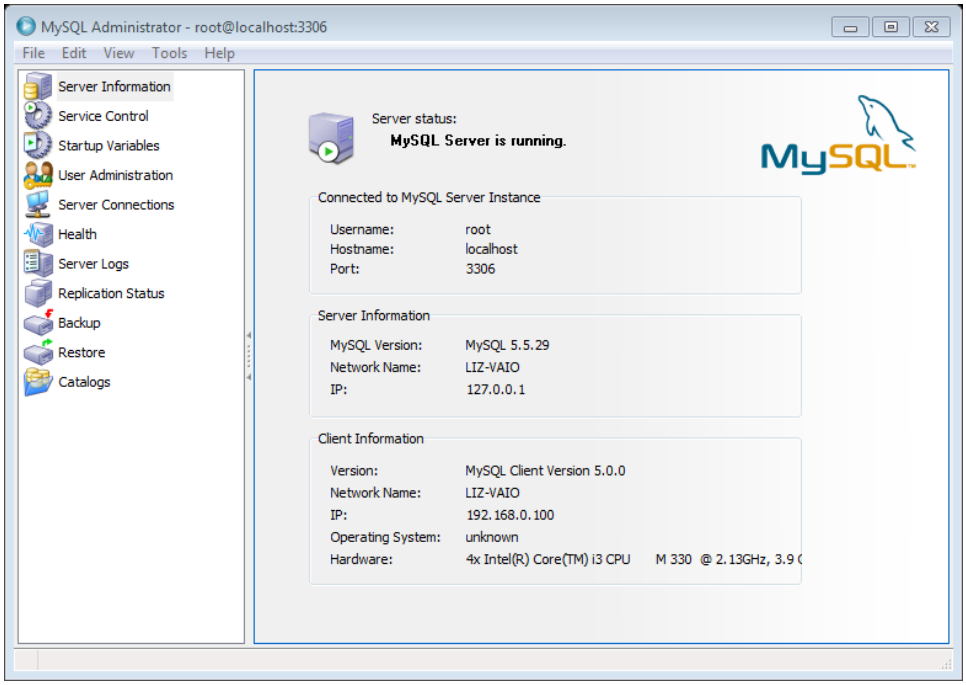

Fig. 25 Pantalla principal de MySQL

Fuente: Autores

Características:

• Está desarrollado en C/C++.

• Se distribuyen ejecutables para cerca de diecinueve plataformas diferentes.

• La API se encuentra disponible en C, C++, Eiffel, Java, Perl, PHP, Python, Ruby y TCL.

• Está optimizado para equipos de múltiples procesadores.

• Es muy destacable su velocidad de respuesta.

• Se puede utilizar como cliente-servidor o incrustado en aplicaciones.

• Cuenta con un rico conjunto de tipos de datos.

• Soporta múltiples métodos de almacenamiento de las tablas, con prestaciones y rendimiento diferentes para poder optimizar el SGBD a cada caso concreto.

• Su administración se basa en usuarios y privilegios.

• Se tiene constancia de casos en los que maneja cincuenta millones de registros, sesenta mil tablas y cinco millones de columnas.

• Sus opciones de conectividad abarcan TCP/IP, sockets UNIX y sockets NT, además de soportar completamente ODBC.

Debido a las características mencionadas se eligió My SQL como motor de búsqueda para la realización de la base de datos de los estudiantes a ser registrados, además es una plataforma que interactúa de una manera sencilla con Visual Basic.Net que es la plataforma de programación escogida para el desarrollo de este software.

De modo de facilitar la programación en My SQL se usará My SQL Front, ofrece un entorno de programación sencillo.

1.8.1. My SQL Front

MySQL-Front es una herramienta final de MySQL que simplifica la creación y modificación de cualquier objeto de base de datos, añadir y cambiar cualquier registro de base de datos y una gran cantidad de pasos de adición de desarrollo aplicaciones con acceso a base de datos en cada idioma en desarrollo.

Hay diferentes elementos visuales para permitir una fácil navegación y cambios rápidos. Si el usuario no conoce el Lenguaje de consulta Corto (SQL), MySQL-Front supera esto - pero permite al usuario aprender viendo las declaraciones utilizadas en SQL. La interfaz simple e intuitiva es fácil de utilizar para los principiantes de base de datos sin ningún conocimiento de SQL, pero permite a un usuario de la energía para hacer todo lo que necesita en una sola vez.

MySQL-Front permite al usuario hacer lo que quiere hacer - sin preocuparse mucho de conocimiento sobre las diferentes incompatibilidades versión MySQL, perdiendo mucho tiempo con la espera de una base de datos o de respuesta que ofrece características inutilizables para la aplicación utiliza MySQL, en la Fig. 26, se muestra el entorno de programación de MySQL Front.

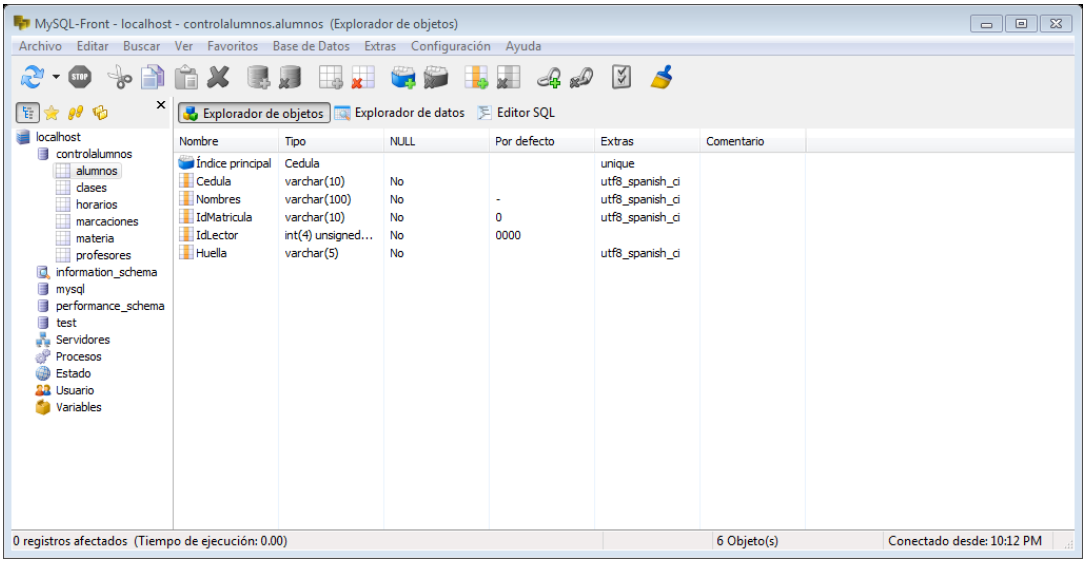

Fig. 26 Entorno de programación de MySQL Front

Fuente: Autores

1.9. VISUAL BASIC

Visual Basic es un lenguaje de programación dirigido por eventos, desarrollado por Alan Cooper para Microsoft. Este lenguaje de programación es un dialecto de BASIC, su primera versión fue presentada en 1991, con la intención de simplificar la programación utilizando un ambiente de desarrollo completamente gráfico que facilitara la creación de interfaces gráficas y, en cierta medida, también la programación misma.

La última versión fue la 6, liberada en 1998, para la que Microsoft extendió el soporte hasta marzo de 2008, pero a partir del 2001 Microsoft propuso abandonar el desarrollo basado en la API Win32 y pasar a un framework o marco común de librerías, independiente de la versión del sistema operativo, .NET Framework, a través de Visual Basic .NET quien fue el sucesor de Visual Basic 6, la imagen de Visual Basic. Net la podemos ver en la Fig. 27.

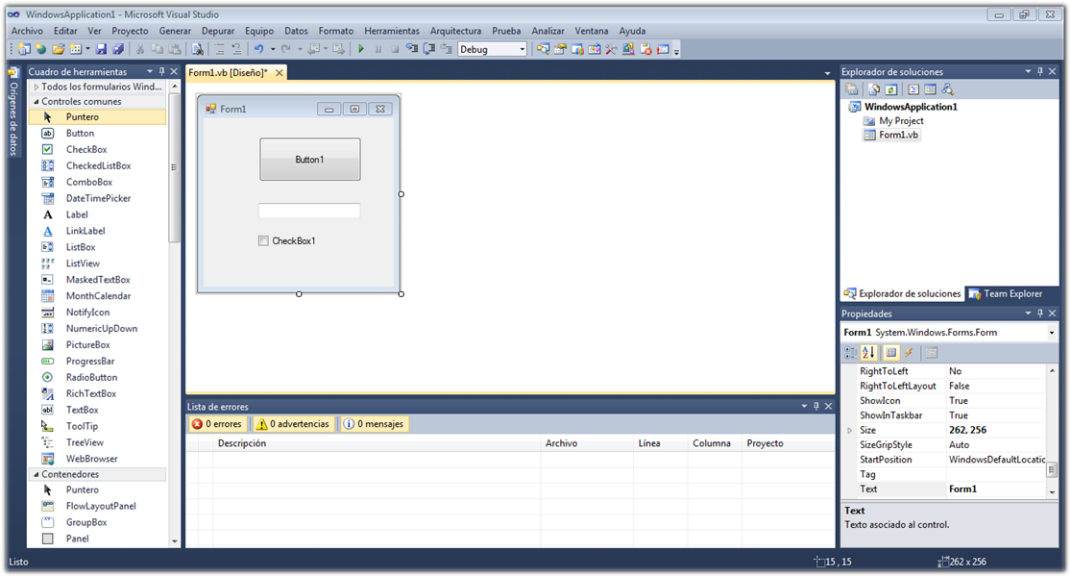

Fig. 27 Visual Basic. Net

Fuente: Autores

Visual Basic .NET (VB.NET), no es compatible hacia atrás con Visual Basic, pero el manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas.

La gran mayoría de programadores de VB.NET utilizan el entorno de desarrollo integrado Microsoft Visual Studio en alguna de sus versiones, al igual que con todos los lenguajes de programación basados en .NET, los programas escritos en VB .NET requieren el Framework .NET o Mono para ejecutarse.

CAPITULO II

2. METODOLOGÍA

Este capítulo presenta la metodología utilizada para el desarrollo de este trabajo.

2.1. TIPO DE ESTUDIO

- De laboratorio.- la investigación es realizada de manera directa dentro del propio lugar de trabajo (área previamente establecida y definida).

- Analítico.- Debe tener un conocimiento claro de cada uno de los elementos y dispositivos que forman parte de las etapas del sistema a estudiar.

- Deductivo.- Proceso de deducción lógica, partiendo siempre de los postulados iniciales para con esto llegar a un nuevo conocimiento.

- Aplicada.- utiliza conocimientos ya adquiridos para en base a ellos dar paso a la investigación del proyecto propuesto.

2.2. POBLACIÓN MUESTRA

- La población a quien se dirige este proyecto, abarcará la comunidad universitaria conformada por: estudiantado, profesores y personal administrativo, ya que los resultados de esta investigación generarán beneficios para cada uno de ellos facilitando su trabajo.

- Al ser un trabajo de investigación debe limitarse a una muestra de la población, por lo que el prototipo se aplicara a la escuela de Electrónica y Telecomunicaciones de la Facultad de Ingeniería.

2.3. OPERACIONALIZACIÓN DE VARIABLES

INDEPENDIENTE	DEPENDIENTE
Implementación del prototipo	Mejorar el control de asistencia de los estudiantes de la Universidad Nacional de Chimborazo

Tabla 7 Operacionalización de variables

Fuente: Autores

2.4. PROCEDIMIENTOS

Esta parte menciona cada uno de los pasos seguidos durante la investigación que han contribuido a cumplir con los objetivos planteados, la Fig. 28 muestra los pasos más representativos, mismos que serán ampliados en este capítulo:

Fig. 28 Procedimiento de la investigación

Fuente: Autores

El capítulo I muestra un análisis previo al desarrolla la investigación, este capítulo explica las soluciones a este problema, en el diagrama de bloques mostrado en la Fig. 29, presenta un resumen de las etapas que conforman el proyecto, mencionado las herramientas utilizadas en cada una de las etapas.

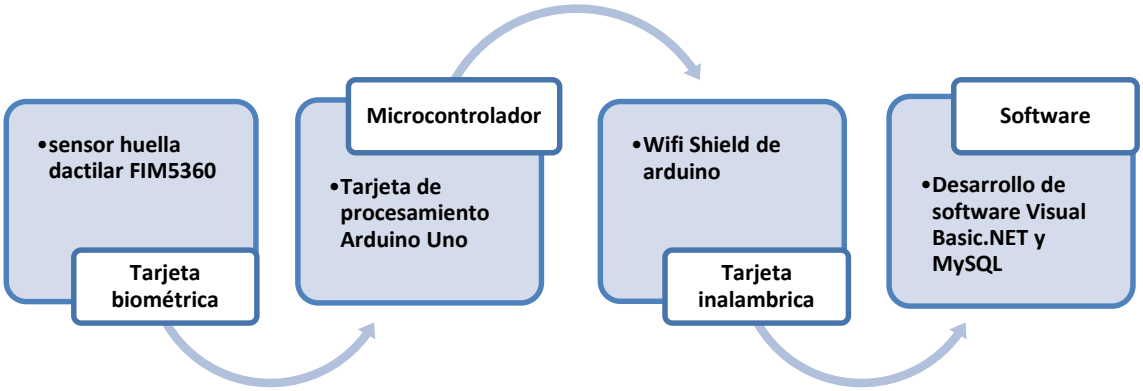

Fig. 29 Diagrama etapas y herramientas del proyecto

Fuente: Autores.

2.4.1. Configuración del módulo FIM5360

La configuración del módulo FIM 5360 utilizamos dos herramientas fundamentales la primera el documento de protocolos colocado en el Anexo 3, la segunda herramienta el software EvTools cortesía de Nitgen, la siguiente sección muestra las principales funciones de este software.

2.4.1.1. Software EvTools

El Software EvTools, propio de Nitgen, trabaja enviando comandos para la ejecución de cada una de las tareas requeridas, sin embargo dado a que no es el principal objetivo de este proyecto se menciona los comandos que serán de más uso. La Fig. 30 presenta la pantalla principal del Software.

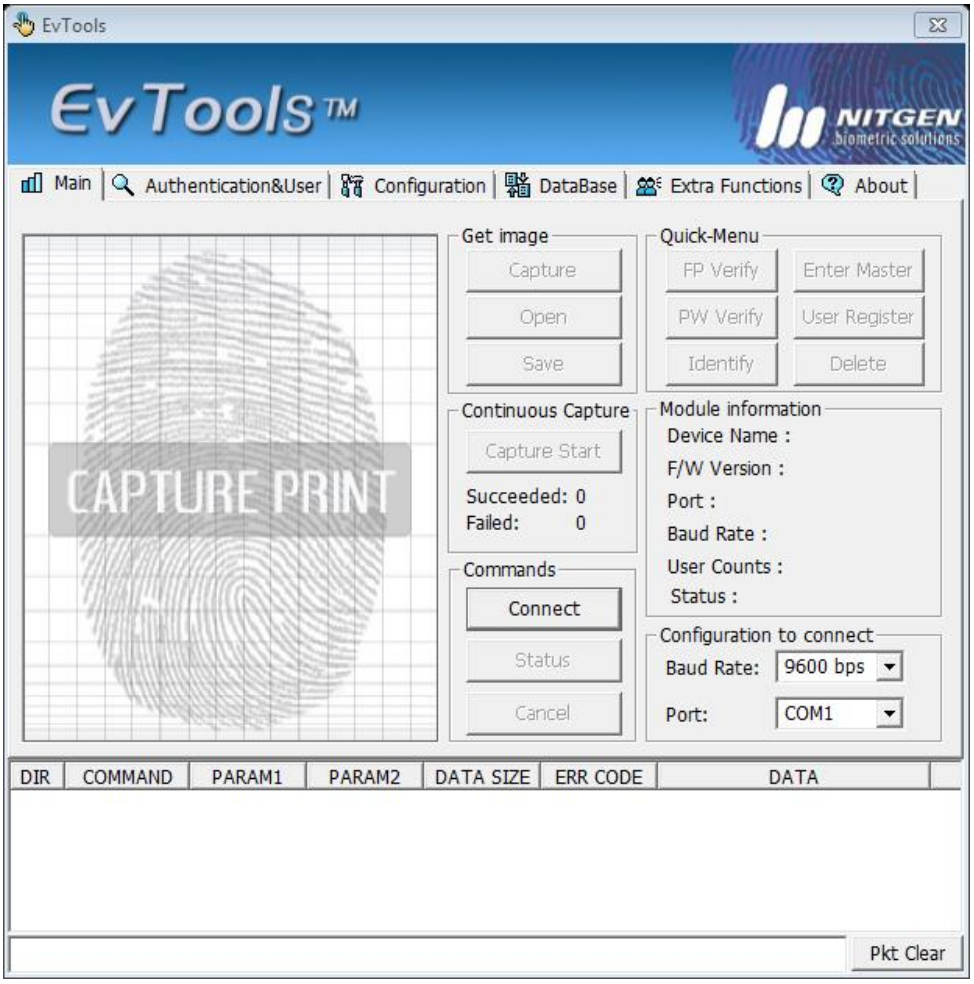

Fig. 30 Muestra del software EvTools

Fuente: Autores

2.4.1.2. Protocolo de comunicación

La Fig. 31 muestra la estructura del paquete de comunicación, el paquete consta de byte inicial, cabecera, datos (opcional), y la suma de comprobación de datos.

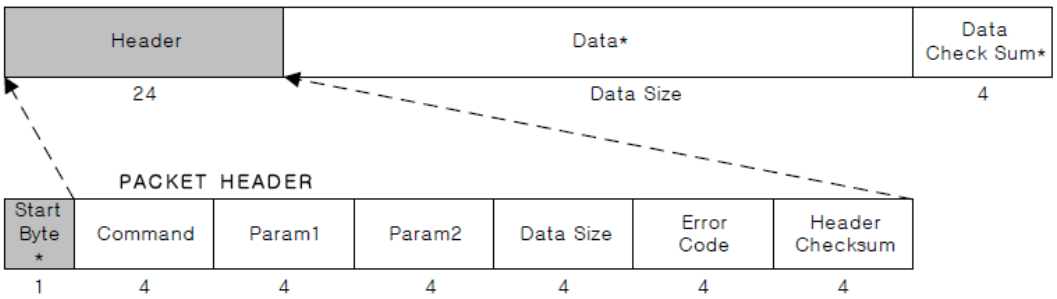

Fig. 31 Estructura del paquete de datos

Fuente: Documento FimProtocol.pdf

El Byte inicial (Start Byte), es 0x7E para todos los casos, además el tamaño máximo de un paquete es 64kByte.

Tamaño (Byte Start) Tamaño + (Header) Tamaño + (datos) Tamaño + (Header Checksum) < 65.536

Si el tamaño de datos es tan grande que el host no puede transportar datos en un solo paquete, el host dividirá los datos en bloques de datos pequeños este es el caso de captura de imagen de huella dactilar en la que la tarjeta capturará la huella dactilar la divide en tramas y la envía trama por trama al PC en donde con ayuda del software serán reconstruidos como imagen pura .RAW y visualizados en el mismo software, el índice de cada paquete tiene el valor de 0 a 255. El tamaño máximo de datos que puede ser enviado se calcula de la siguiente manera.

Max Data Block = 256 x 65.507 = 16.769.792 [bytes]

El Data Checksum resulta de la suma de todos los datos byte, con el fin de crear la cabecera de comprobación, se añaden el byte inicial.

EvTools es un software de múltiples características, está compuesto por pestañas de operación en las que cada sección contiene funciones específicas para el desarrollo del módulo FIM 5360, en este estudio revisaremos la primera pantalla (MAIN) en donde se encuentran detalladas las principales funciones del módulo que serán utilizadas en esta investigación.

2.4.1.3. Main

La Fig. 32 señala las zonas de interés del software EvTools, además explica con más detalle la aplicación de estas funciones.

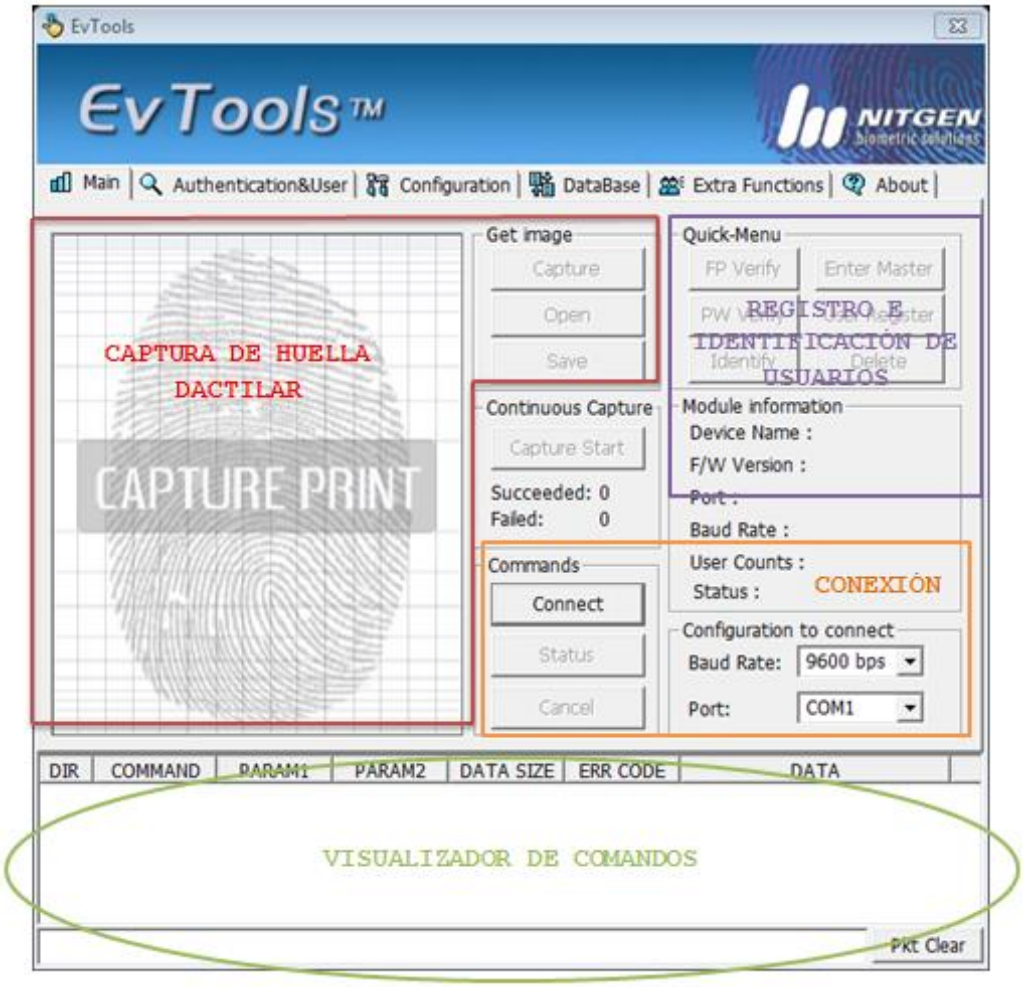

Fig. 32 Detalle de software

Fuente: Autores

2.4.1.4. Conexión

La Fig. 33, muestra la zona de conexionado entre el módulo FIM 5360 y la PC, el software pide el ingreso de parámetros específicos para la comunicación serial como son Puerto de conexión y velocidad de transmisión, una vez ingresados estos parámetros podremos habilitar la conexión con el botón Connect, al hacer esta acción se envía una serie de comandos de verificación, conexión y estado de la tarjeta, por ejemplo el comando CMD_REQUEST_CONNECTION.

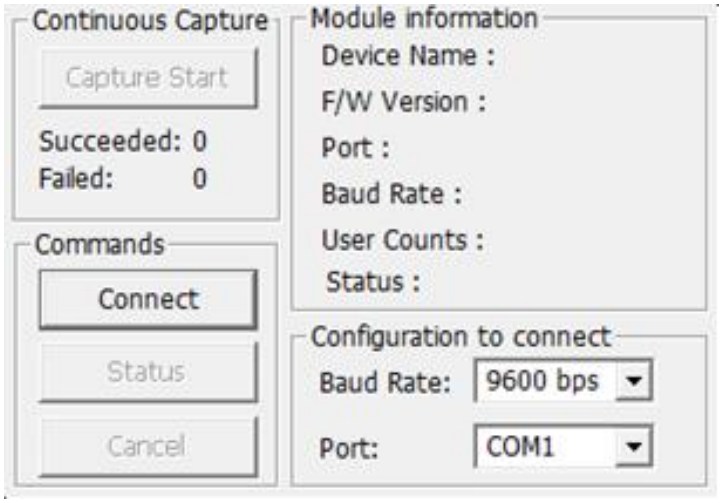

Fig. 33 Conexionado del módulo FIM 5360 y la PC

Fuente: Autores

El módulo FIM 5360 tiene predefinidas las configuraciones iniciales, al momento de enviar el comando CMD_REQUEST_CONNECTION, hace la comprobación de estos parámetros en caso de ser incorrectos produce un error en la transmisión lo que impedirá que la tarjeta se conecte. El detalle del comando se encuentra en la Tabla. 8.

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x01	Command	0x01
Param1	0	Param1	RESULT_SUCCEEDED
Param2	0	Param2	User Count
Data Size	0	Data Size	0
Error Code	0	Error Code	Error code

Tabla 8 Comando CMD REQUEST CONNECTION

Fuente: Protocol Guide. Pdf

El comando está compuesto de los siguientes segmentos:

- Command.- comando específico para cada función, por ejemplo 0x01-CMD REQUEST CONNECTION.

- Param 1 y 2.- llevan valores específicos de acuerdo a las necesidades de cada una de las funciones para este caso y dado que es un comando de comprobación el parámetro 1 y 2 es 0x00.

- Data Size.- como no se envían datos el Data Size es igual 0x00, al igual que el Error Code.

Las pruebas de conexión es necesario el uso de un cable Serial a USB dado que el computador de pruebas no dispone de una interfaz serial, la Fig. 34 muestra el cable utilizado, Belkin.

Fig. 34 Cable convertidor de serial a USB
Fuente: www.solostocks.com

2.4.1.5. Visualizador de comandos

La zona de mayor ayuda es el visualizador de comandos de la Fig. 35, cada uno de los botones llevan instrucciones para la tarjetas, instrucciones que son enviadas a través de comandos escritos en hexadecimal, cada uno de los comandos son detallados en el Anexo 3.

Visualizador de comandos permite ver de una manera muy didáctica la estructura de los comandos enviados a la tarjeta y de igual manera las respuestas que se obtienen de ella.

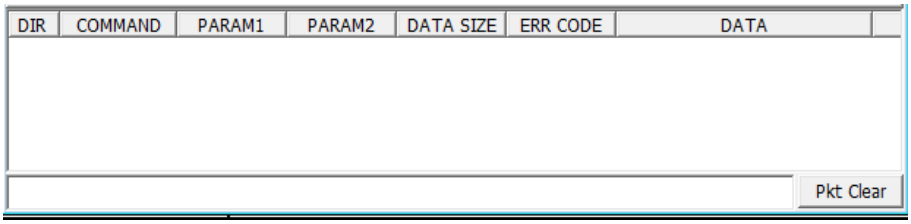

Fig. 35 Visualizador de comandos

Fuente: Autores

2.4.1.6. Zona de captura de huella dactilar.

La Fig. 36 indica la zona de captura de huella dactilar, este módulo consta con un sistema que permite capturar la huella dactilar y almacenarla en un directorio del computador para que posteriormente el usuario pueda hacer el tratamiento de la imagen, la imagen es grabada con la extensión de archivo .RAW (Formato de imágenes sin modificaciones), debido a que el proyecto exige la mayor rapidez de captura y procesamiento de la huella esta función no será utilizada, la función a usarse estará encargada del procesamiento de la imagen y entregara tan solo los datos necesarios para la identificación de usuario.

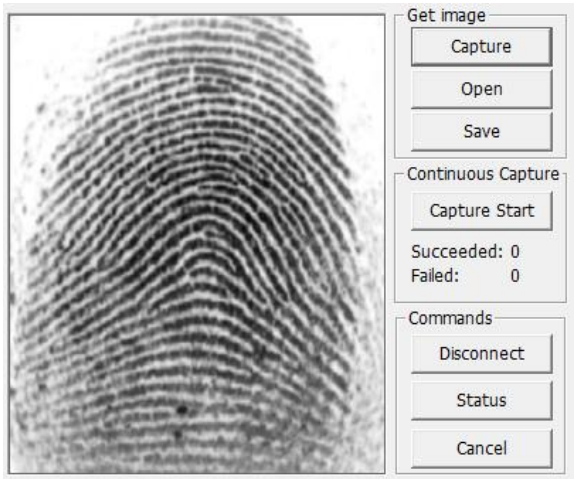

Fig. 36 Zona de captura de huella dactilar

Fuente: Autores

Al mismo tiempo que la huella dactilar es capturada, se muestra en el visualizador que posee el software, esta huella es almacenada en un directorio del computador más no en la memoria de la tarjeta.

2.4.1.7. Modos de Operación.

La Fig. 37 muestra el menú rápido para registro, identificación y verificación de usuarios, este módulo trabaja en dos modos de operación.

- Modo Usuario.- este modo permite realizar las operaciones generales que no afecten a la configuración del módulo, las funciones permitidas en modo usuario son: Verificación por huella dactilar, verificación por contraseña e identificación de usuario.

- Modo Administrador o modo Master.- al estar en modo administrador el módulo permite realizar funciones mucho más específicas que podrían variar la configuración del módulo, algunas de las funciones permitidas en modo administrador son: registro de usuarios, eliminación de usuarios, tener ingreso a la base de datos de huella dactilar, cambios en la configuración del módulo, adquisición del template de huella dactilar, entre otras.

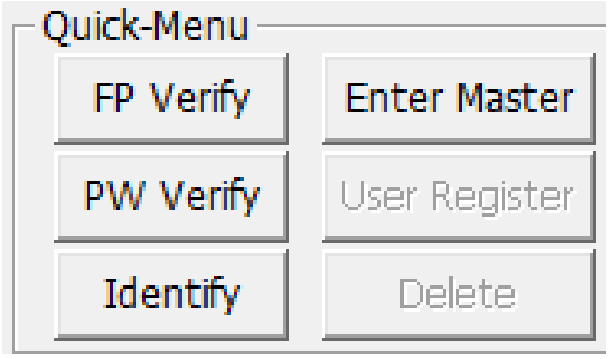

Fig. 37 Menú de registro, identificación y verificación

Fuente: Autores

2.4.1.8. Modo Administrador

Al ser activado el modo Enter Master o modo Administrador el software envía por el puerto serial, el comando CMD_ENTER_MASTER_MODE2.

CMD_ENTER_MASTER_MODE2

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x2F	Command	0x2F
Param1	Master authentication type Master FP verification = 0 Master password verification = 1 Board password verification = 2 Null = 3 Master instant FP verification = 4	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_INVALID_PARAM RESULT_INVALID_ID RESULT_CANCELED RESULT_EXTRACT_FAIL
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	Master authentication type Master FP = 0 Master password = 1 FDA board password = 2 Null = 3 Master FP from host = 4 Master FP from host (FDA01 style) = 5 (FIM10 only)
Data Size	**IF** Master FP Size (A fraction of FPID) **ELSE IF** master password Size (A fraction of FPID + Password) **ELSE IF** device board password Size (A fraction of password) **ELSE IF** Master FP from host Size (A fraction of FPID + Template) **ELSE IF** null 0	Data Size	0
Error Code	X	Error Code	Error code
Data	**IF** Master FP A fraction of FPID	Data	-

Tabla 9 Comando CMD_ENTER_MASTER_MODE2

Fuente: Documento Protocol Guide.pdf

La Tabla. 9 detalla a este comando, está formado por dos secciones, la primera llamada COMMAND PACKET, ayuda a formar el comando que el host deberá enviar al módulo, la segunda sección es la ACKNOWLEDGEMENT PACKET que indica los parámetros de respuesta que envía la tarjeta, para el ingreso a modo master se envía el siguiente comando:

- ✓ Start Byte: 0x7E
- ✓ Command: 0x2F
- ✓ Param1: 0x03
- ✓ Param2: 0x00
- ✓ Data Size: 0x00
- ✓ Error Code: 0x00
- ✓ Header checksum: 0x32

Una vez ingresado a modo master se activa las funciones de registro y borrado de usuarios como lo vemos en la Fig. 38.

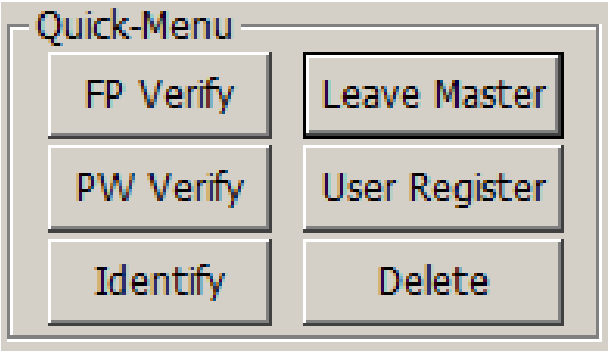

Fig. 38 Menú rápido - funciones activas

Fuente: Autores

2.4.1.9. Registro de Usuarios

El registro de los usuarios (User Register) almacena los datos de huella en la memoria del módulo. La Fig. 39, muestra un diagrama en bloques del proceso para el registro de usuarios.

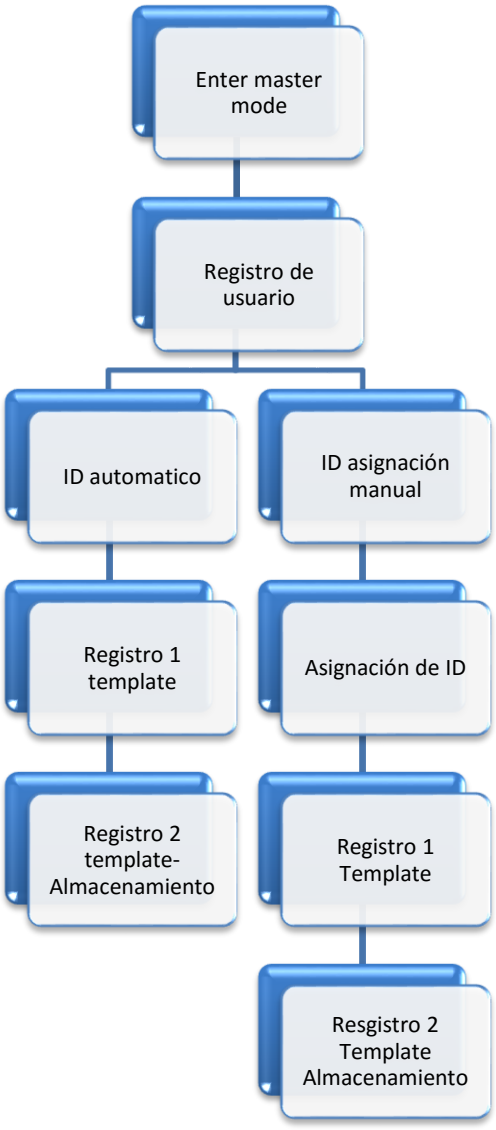

Fig. 39 Proceso de Registro de usuario

Fuente: Autores

La FIM5360 tiene una capacidad de almacenamiento de 1000 usuarios, con registro de dos huellas por cada uno, sin embargo el registro de los usuarios se realiza de dos maneras.

- ID automático.- por cada usuario registrado la tarjeta asigna automáticamente el ID, este ID es un número que puede ir desde el 0000 hasta el 1000 , la ventaja de la asignación automática de un ID, es que permite relacionar de una manera más fácil a una huella dactilar con un usuario esto debido a que la tarjeta no almacena datos personales como

nombre, cédula o direcciones, además de esto tanto en el registro manual como en el registro automático el módulo debe cargar dos huellas por cada usuario, para el registro de la primera huella el comando utilizado será:

- ✓ Start Byte: 0x7E
- ✓ Command: 0x33
- ✓ Param1: 0x00
- ✓ Param2: 0x10
- ✓ Data Size: 0x00
- ✓ Error Code: 0x00
- ✓ Header checksum: 0x43

La segunda huella a registrarse corresponderá al mismo dedo indicado en la primera captura, este paso ayuda a comprobar las características tomadas de la huella en el primer paso, si las características coinciden la huella será almacenada en la memoria de la tarjeta.

- ✓ Start Byte: 0x7E
- ✓ Command: 0x33
- ✓ Param1: 0x00
- ✓ Param2: 0x01
- ✓ Data Size: 0x00
- ✓ Error Code: 0x00
- ✓ Header checksum: 0x34

- ID manual.- por cada usuario registrado la persona encargada del registro asigna manualmente un ID para cada usuario, este ID debe ser de 4 dígitos, los mismos que serán añadidos a la trama que será enviada a la tarjeta, esto realizado de la siguiente manera**:**

  - ✓ Start Byte: 0x7E
  - ✓ Command: 0x33

- ✓ Param1: 0x00
- ✓ Param2: 0x01
- ✓ Data Size: 0xC0
- ✓ Error Code: 0x34
- ✓ Header checksum: 0xF3
- ✓ Data: 0x30 0x30 0x30 0x30
- ✓ Data Checksum: 0xC0(Suma de Datos)

La Tabla 10 muestra el detalle del Comando CMD_REGISTER_FP

2.4.1.10. Eliminación de usuarios

La eliminación de un usuario o DELETE, es un comando indispensable para un administrador de sistema, teniendo en cuenta que pueden producirse fallas humanas o mecánicas al momento del registro de un nuevo usuario, se detalla el comando que será de utilidad para el proceso de eliminación de un usuario:

- ✓ Start Byte: 0x7E
- ✓ Command: 0x22
- ✓ Param1: 0x00
- ✓ Param2: 0x0A
- ✓ Data Size: 0xC0
- ✓ Error Code: 0x00
- ✓ Header checksum: 0xEC
- ✓ Data: 0x30 0x30 0x30 0x30
- ✓ Data Checksum: 0xC0 (Suma de Datos)

La tabla. 11, muestra el detalle del comando CMD_DELETE_FP y las posibles respuestas que puede enviar el módulo.

CMD_REGISTER_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x33	Command	0x33
Param1	0 – User 1 – Master Otherwise – Reserved	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA RESULT_NOT_IN_TIME RESULT_INVALID_PARAM RESULT_USED_ID RESULT_DB_IS FULL RESULT_NOT_MASTER_MODE RESULT_ANOTHER_FINGER RESULT_CANCELED RESULT_EXTRACT_FAIL RESULT_INVALID_SEQUENCE

Param2	Packet Index
	0x00 – Extract 1st Template from sensor with ID and password
	0x10 – Extract 1st Template from sensor with auto-generated ID
	0x01 – Extract 2nd Template from sensor & Save
	0x02 – Extract 2nd Template from sensor & Save with different finger
	(FIM20 emulation mode Only)
	0x03 – Extract 3rd Template from sensor
	0x04 – Extract 4th Template form sensor & save
	0x05 – Extract 4th Template from sensor & save with different finger
		Param2	IF (Param1 == RESULT_SUCCEEDED)
			&& (((Packet Index == 0x01 or 0x02)
			&& (2 templates mode)) ||
			((Packet Index == 0x11 or 0x12)
			&& (4 templates mode)))
			Registered User Count
			(Only valid if succeed)
			ELSE
			0

Data Size	IF (Packet Index == 0)
	Size of (FPID + Password)
	ELSE
	0
		Data Size	0

Error Code	0	Error Code	Error Code

Data	IF (Packet Index == 0)
	FPID + password
	ELSE
	0
		Data	-

Tabla 10 Comando CMD_REGISTER_FP

Fuente: Protocol Guide.pdf

CMD_DELETE_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x22	Command	0x22
Param1	0	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_DATASIZE RESULT_INVALID_DATA REAULT_INVALID_ID RESULT_NOT_MASTER_MODE
Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)	Param2	**IF** (Param1 == Succeeded) Registered User Count **ELSE** 0
Data Size	Size (A fraction of FPID)	Data Size	0
Error Code	0	Error Code	Error Code
Data	A fraction of FPID	Data	-

Tabla 11 Comando CMD_DELETE_FP

Fuente: Documento Protocol Guide.pdf

2.4.2. Verificación VS Identificación

La metodología del reconocimiento de la huella dactilar está dividida en dos procesos diferentes: identificación (1:N) y la verificación (1:1), la mayoría de sistemas de identificación con lectores biométricos pueden realizar estas dos características.

Para el desarrollo de aplicaciones software con lectores biométricos es necesario disponer de unas SDK biométricas, esta herramienta de software permite normalmente desarrollo de aplicaciones de identificación y verificación biométrica.

2.4.2.1. Verificación

El proceso de verificación es un proceso de combinación de uno-a-uno (1:1). El usuario confirma quién es el usuario. Una nueva muestra de la huella dactilar es

tomada del usuario y comparada a la otra previamente registrada o archivada. Si las huellas dactilares coinciden, el usuario es "verificado" entonces se conceden todos los privilegios y accesos del usuario confirmado, es decir que el sistema pudo verificar al usuario.

2.4.2.2. Identificación

El proceso de identificación es un proceso de combinación de uno-a-muchos (1:N), el usuario no precisa confirmar quién es. La nueva muestra de la huella dactilar es tomada del usuario y comparada a una ya existente en el banco de datos de huellas dactilares, registradas o archivadas de todos los usuarios. Cuando es encontrada una combinación, el usuario es "identificado" como un usuario preexistente, o sea, el sistema encuentra quién es.

Esta investigación está enfocada al control de asistencia del alumnado por lo que la técnica adecuada para este caso es la Identificación, cada estudiante colocará su huella y someterá a la identificación de esta de entre una base de datos en la que estarán registrados todos los alumnos de una determinada materia, se detalla a continuación el comando utilizado para este propósito:

- ✓ Start Byte: 0x7E
- ✓ Command: 0x12
- ✓ Param1: 0x01
- ✓ Param2: 0x00
- ✓ Data Size: 0x00
- ✓ Error Code: 0x00
- ✓ Header checksum: 0x13

La Tabla. 12 muestra el detalle del Comando CMD_IDENTIFY_FP

2.4.3. Programación de Arduino

El capítulo anterior analizo cada una de las características de Arduino, esta sección explica porque Arduino fue elegido respecto a sus competidores, dentro de sus ventajas están:

- Asequible - Las placas Arduino son más asequibles comparadas con otras plataformas de microcontroladores, además su costo es relativamente bajo.

- Multi-Plataforma - El software de Arduino funciona en los sistemas operativos Windows, Macintosh OSX y Linux.

- Entorno de programación simple y directo - El entorno de programación de Arduino es fácil de usar para principiantes y lo suficientemente flexible para los usuarios avanzados.

- Software ampliable y de código abierto- El software Arduino está publicado bajo una licencia libre y preparada para ser ampliado por programadores experimentados.

El código de programación de Arduino para este proyecto, básicamente controla las interfaces de comunicación serial entre el módulo de captura de huella dactilar y el módulo de interface inalámbrica.

La Fig. 40, explica mediante un diagrama de bloques las fases de programación.

CMD_IDENTIFY_FP

COMMAND PACKET		ACKNOWLEDGEMENT PACKET	
Command	0x12	Command	0x12
Param1	0x00 – User ID only request 0x01 – User ID and Template Index request 0x02 – User ID and user type request	Param1	RESULT_SUCCEEDED RESULT_FAILED RESULT_INVALID_PARAM RESULT_NOT_IN_TIME RESULT_IDENTIFY_TIMEOUT (FIM01 & FIM20xx only) RESULT_CANCELED RESULT_EXTRACT_FAIL
Param2	0	Param2	(Packet Index (0~N) << 8) + (Max Packet Index N)

Data Size	0	Data Size	IF (Param1 == Succeeded)
			IF (Command Param1 = 0x00)
			Size of FPID (various between devices)
			ELSE IF (Command Param1 = 0x01)
			Size of (FPID + Template Index)
			ELSE IF (Command Param1 = 0x02)
			Size of(FPID + User Type)
			ELSE
			0
			ELSE
			0

Error Code	0	Error Code	Error Code

Data	-	Data	IF (Param1 == Succeeded)
			IF (Command Param1 = 0)
			FPID
			ELSE IF (Command Param1 = 1)
			(FPID + Template Index)
			ELSE
			0
			ELSE

Tabla 12 Comando CMD_IDENTIFY_FP

Fuente: Documento protocol Guide. pdf

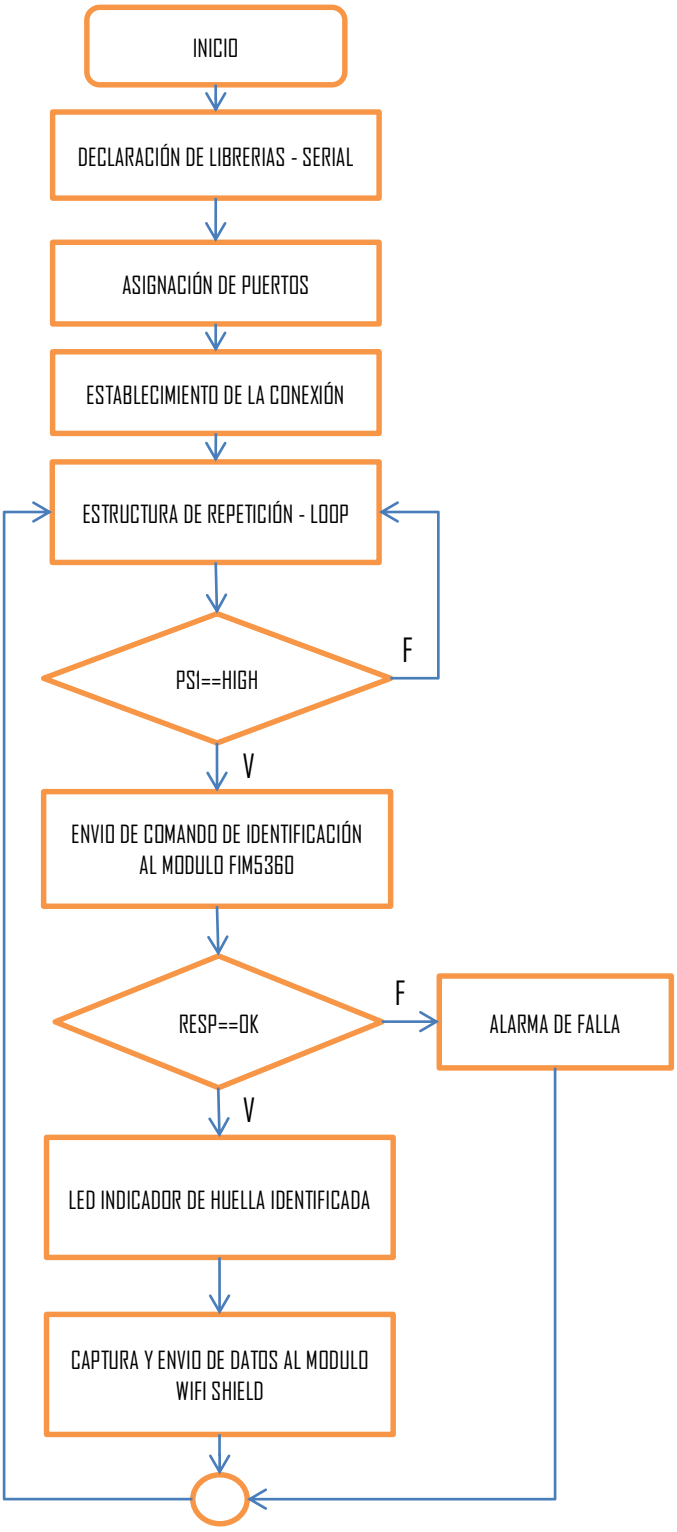

Fig. 40 Diagrama de Bloques de la programación de Arduino

Como indica el diagrama de bloques el primer paso a realizar en toda programación y en cualquier lenguaje que se lo realice, es la declaración de variables y librerías:

```
#include <SPI.h>
#include <WiFi.h>
#include <SoftwareSerial.h>
#define rxPin 2
#define txPin 3
int son=5;
int led = 6;
int push = 8;
int conect = 9;
int verifi=0;
char ssid[] = "ALUMNOS";
char pass[] = "12345678";
IPAddress server(192,168,0,100);  // numeric IP  (no DNS)
SoftwareSerial miSerial =  SoftwareSerial(rxPin, txPin);
```

La tarjeta Arduino posee una interfaz serial física que corresponde a los pines 0 y 1 mismos que serán utilizados para labores de administración de la red en caso de ser necesaria la monitorización del sistema y transmisión de los datos, el módulo Fim5360 se conecta a la tarjeta Arduino a través de un puerto serial de igual manera, para lo cual fue necesario su configuración por software, es declarada la librería Software Serial, adicional a esto hay que declarar los puertos que serán ocupados en la comunicación serial, en este caso se designa a los puertos 2 y 3, como se observa en las líneas de código anteriores, el pin 9 está asignado a un led que nos indicara el estado de la conexión al equipo configurado como servidor.

El pin 8 está asignado a un pulsador que controlara la orden de identificación de un usuario, los pines 5 y 6 son indicadores, el pin 5 está conectado a un buzzer que emitirá un tono cuando el usuario no sea identificado, por el contrario el pin 6 está conectado a un led el cual se enciende cuando la identificación es exitosa.

En este paso hay que declarar una variable que contenga e SSID de la red a conectarse y la contraseña de la misma, dado que estos datos serán utilizados para futuras actividades de conexión a la red, para el caso de este proyecto no bastará

con que el equipo esté conectado a la red sino que también es necesario que se encuentre enlazado al computador servidor, para esto es necesario la asignación de la dirección IP y el número de puerto del computador servidor, por ejemplo (192.168.1.100/23).

El segundo paso es la declaración de puertos, aunque parte de este proceso se muestra en las líneas de código anteriores, en este paso cada puerto es declarado como entrada o salida y los puertos seriales son inicializados, la configuración de los puertos seriales se los ha mantenido en 9600bps que son las especificación por defecto, además de que después de las pruebas de comunicación realizadas entre la placa Arduino y el módulo de huella dactilar este baud rate es el apropiado para la transmisión.

```
void setup() {
   pinMode(push, INPUT);
  pinMode(rxPin, INPUT);
  pinMode(txPin, OUTPUT);
  pinMode(son, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(conect, OUTPUT);
  pinMode(sd, OUTPUT);
  Serial.begin(9600);
  miSerial.begin(9600);
  digitalWrite(sd, HIGH);
        while ( status != WL_CONNECTED) {
                Serial.print("Conectando a la red: ");
                Serial.println(ssid);
                 status = WiFi.begin(ssid, pass);
                 delay(1000);
        }
   printWifiStatus();
   while (!client.connected()) {
   Serial.println("Conectando al servidor....");
   client.connect(server, 23);
```

El tercer paso está basado en una estructura de repetición Loop, el led conectado al pin 9 permanecerá encendido mientras el equipo se encuentre conectado a la maquina servidora, si esta verificación es correcta el led asociado al pin 9 “conect” permanecerá encendido, según la lectura del estado del pulsador PS1, si es activado, el comando previamente definido será enviado a través del puerto serial hacia el módulo de huella dactilar y esperará el comando de respuesta el cual será leído por el puerto serial y almacenado en un vector llamado “ing” para que sus datos puedan ser utilizados posteriormente, cuando un dato ha sido receptado correctamente por la FIM 5360, esta envía un comando de confirmación.

```
void loop() {
 while (client.connected()) {
  digitalWrite(conect, HIGH);
  verifi=digitalRead(push);
  char comand[25]={126,0,0,0,18,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,19};
  if (verifi == HIGH) {
        for (i=0;i<25;i=i+1){
              miSerial.print(comand[i]);
              delay (20);
        }
        delay(2000);
        char
        ing[40]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
        0,0};
         for (i=0;i<40;i=i+1){
              char dat = miSerial.read();
              ing[i]=(dat);
              delay (50);
```

De acuerdo a la estructura del comando analizado en la sección anterior si el comando fue exitoso el parámetro 1 correspondiente al dato 8 del vector debe ser igual a 1, es así que después de ser almacenado el comando en un vector es necesaria la comprobación de este parámetro, si la respuesta es afirmativa el

módulo Wifi Shield envía los datos del 25 al 28 del comando de respuesta que corresponden al ID, estos serán enviados inalámbricamente al PC para la validación del usuario, de la misma manera el puerto asignado al led indicador será puesto en alto para indicar que la identificación del usuario fue realizada correctamente, este dato será enviado al puerto serial para casos de monitoreo.

```
if (ing[8]==01){
      delay(100);
       client.write(ing[25]);
      client.write(ing[26]);
      client.write(ing[27]);
      client.write(ing[28]);
      Serial.write(ing[25]);
      Serial.write(ing[26]);
      Serial.write(ing[27]);
      Serial.write(ing[28]);
      digitalWrite(led, HIGH);
      delay (500);
      digitalWrite(led, LOW);
       }
```

Por el contrario si el dato 8 del vector de respuesta no es igual a 1 significara que se produjo un error en la identificación del usuario, si esto ocurre no enviará ningún dato al módulo Wifi Shield, y activa el puerto 5 que está conectado a un buzzer el cual funcionará a manera de un alarma indicadora de error para el usuario, para fines de mantener activa la conexión fue incluida dentro de la programación el envío de un carácter cualquiera de manera constante, para este caso fue elegido el carácter “A”.

```
    else{
      digitalWrite(son,HIGH);
      delay(50);
      digitalWrite(son,LOW);
      delay(50);
```

```
  digitalWrite(son,HIGH);
  delay(50);
  digitalWrite(son,LOW);
  delay(50);
  digitalWrite(son,HIGH);
  delay(50);
  digitalWrite(son,LOW);
 }
client.write("A");
 }
   digitalWrite(conect, LOW);
 }
```

2.4.4. Configuración del módulo de comunicación inalámbrica

Existen dos maneras de configuración del módulo Wifi Shield de Arduino, la primera opción es configurarlo en modo Servidor y la segunda es un modo de configuración en donde el módulo Wifi actuará en modo cliente, para la configuración del módulo en cualquiera de los dos modos es necesario descargar las librerías Wifi, que se encuentran en la página principal de Arduino y agregar estas librerías al directorio raíz del software Arduino, este paso es necesario dado en la sección anterior la comunicación entre la placa Arduino y la Wifi Shield se la realiza por una interfaz SPI, al ser cargadas estas librerías serán importadas las librerías SPI adicionalmente.

Para el desarrollo de la investigación fue necesario configurar el módulo Wifi Shield en modo cliente de manera que este pueda enlazarse a un computador servidor que permanecerá escuchando el tráfico entrante a este y enviará los datos recibidos desde el equipo a la base de datos.

Los comandos utilizados con este fin están detallados en la Tabla. 13, de igual manera se los puede revisar en la Programación de Arduino especificada en la sección anterior.

COMANDO	DETALLE	SINTAXIS
WiFiClient	Crea un cliente que puede conectarse a Internet o a una dirección IP y el puerto definido en client.connect ().	WiFiClient()
Connected	Comprueba la conexión del cliente al servidor, sin embargo hay que tener en cuenta que un cliente se considera conectado si la conexión se ha cerrado pero todavía hay datos leídos.	Client.connected()
Connect()	Conectarse a la dirección IP y el puerto especificado en el servidor, también es compatible con las búsquedas de DNS a través de un nombre de dominio.	*client*.connect(ip, port) *client*.connect(URL, port)
Write()	Escribir datos en el servidor que está conectado el cliente, los datos a ser escritos pueden ser del tipo Char o Byte	*client*.write(data)
println	Los datos de impresión, seguido de un retorno de carro y salto de línea, en el servidor al que un cliente está conectado. Imprime números como una secuencia de dígitos, cada carácter ASCII (por ejemplo, el número 123 se envía como los tres caracteres '1 ', '2', '3 ').	*client*.println() *client*.println(data) *client*.print(data, BASE)

Tabla 13 Listado de comandos WifiClient
Fuente: Autores

2.4.5. Diseño de la tarjeta de conversión de nivel TTL a RS232

El capítulo 1, mostró las características físicas de cada una de las tarjetas, sus niveles de voltaje de entrada y salida no siempre son iguales, por ejemplo la tarjeta FIM 5360 trabaja con niveles de voltaje RS232, la placa Arduino trabaja con niveles de voltaje TTL, la comunicación entre estos dos módulos se complica debido a que sus niveles de voltaje no son iguales, la solución a este problema es la implementación de un circuito de acople de niveles de voltaje el uso del circuito integrado MAX232 será la solución al problema, los dispositivos adicionales como pulsador, led indicador, buzzer de alarma y switch de selección, han sido incluidos en esta placa para una mejor organización del sistema, la Fig. 41 muestra el diagrama esquemático de esta placa, la Fig. 42 muestra el diseño de la placa. La Fig. 43 muestra una imagen de la elaboración de esta placa.

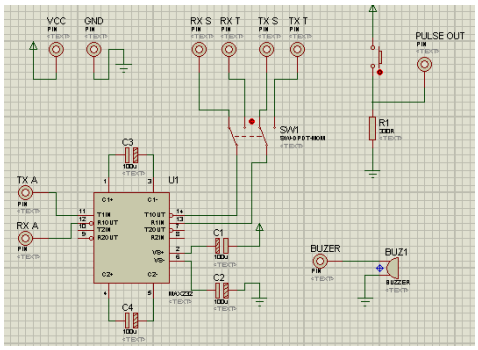

Fig. 41 Diagrama esquemático de la placa de interfaces

Fuente: Autores

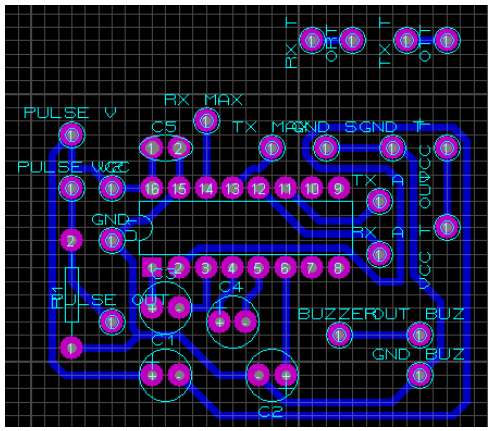

Fig. 42 Diseño de la placa de interfaces

Fuente: Autores

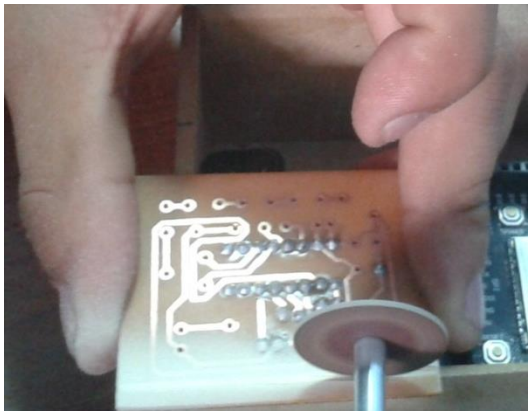

Fig. 43 Elaboración de la placa TTL-RS232

Fuente: Autores

2.4.6. Software de comunicación inalámbrica

La comunicación inalámbrica con el dispositivo debe estar permanentemente activo, siempre y cuando se vaya a hacer el proceso de identificación de los usuarios, la Fig. 44, muestra la pantalla del módulo de Recepción de datos inalámbricos, el único dato que solicita es el puerto de red por el cual va a escuchar el trafico el servidor, este puerto debe ser especificado en la programación en Arduino detallado en la sección anterior, el puerto de red es una interfaz para comunicarse con un programa a través de una red.

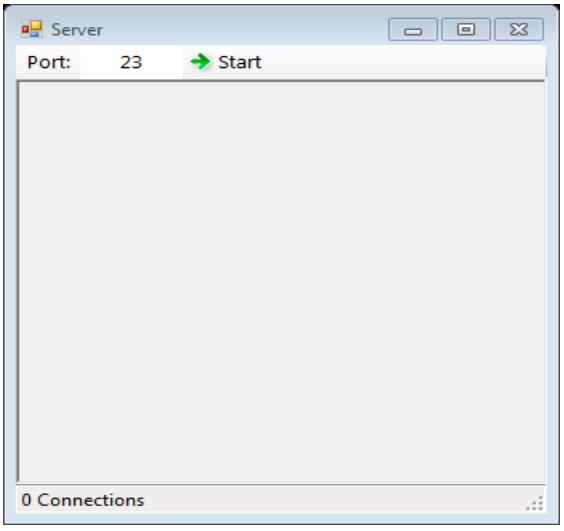

Fig. 44 Apariencia del módulo SERVER

Fuente: Autores

Una vez ingresado el número del puerto es necesario elegir la opción “Start” y el programa iniciara, este puede desactivarse con el botón “Stop”.

2.4.7. Creación de una base de datos – MY SQL

La base de datos es creada con el apoyo de MySQL Front, que ofrece un entorno de programación sencillo, esta sección analiza el procedimiento para la creación de la base de datos del sistema, empezaremos entonces por definir cuáles son las tablas a ser creadas y los datos que requieren ser almacenados dentro de estas.

Esta investigación está dirigida al control de asistencia de alumnado por lo tanto los datos más relevantes serán: materias, profesores, horarios y por supuesto los datos personales del estudiante, la Fig. 45 muestra un detalle de las tablas y los datos que constan dentro de cada uno, de igual manera la Fig. 46 contiene el diagrama necesario para la creación de una base de datos.

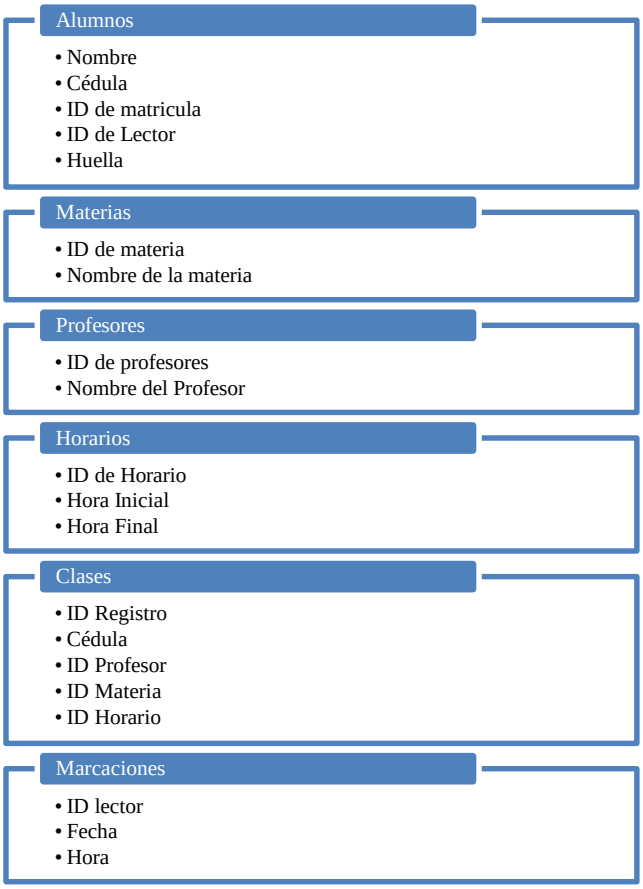

Fig. 45 Menú de Variables de la base de datos

Fuente: Autores

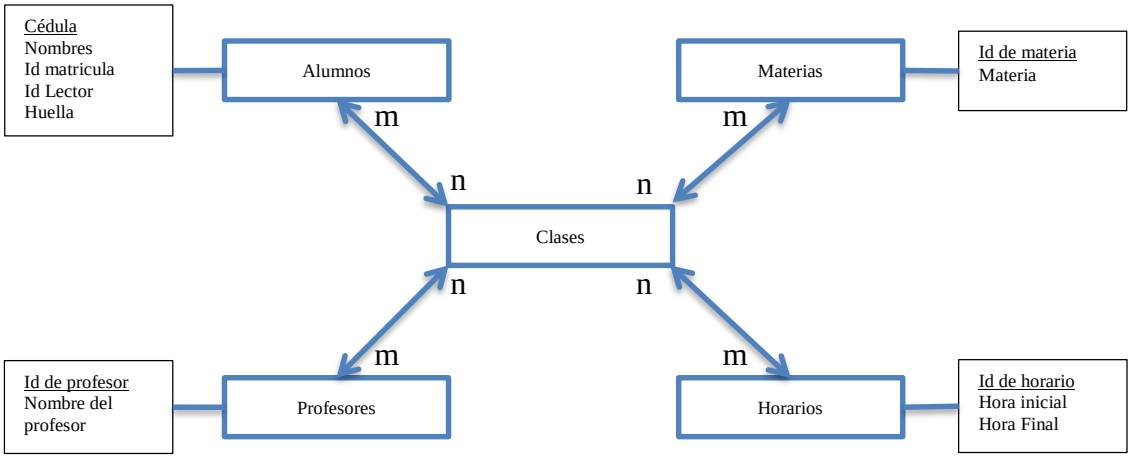

Fig. 46 Relación de Entidades

Fuente: Autores

Las siguientes imágenes mostraran paso a paso el proceso de creación de la base de datos.

La Fig. 47 muestra la pantalla principal de MySQL Front en donde hay que la opción de crear nueva base de datos, para iniciar con este proceso.

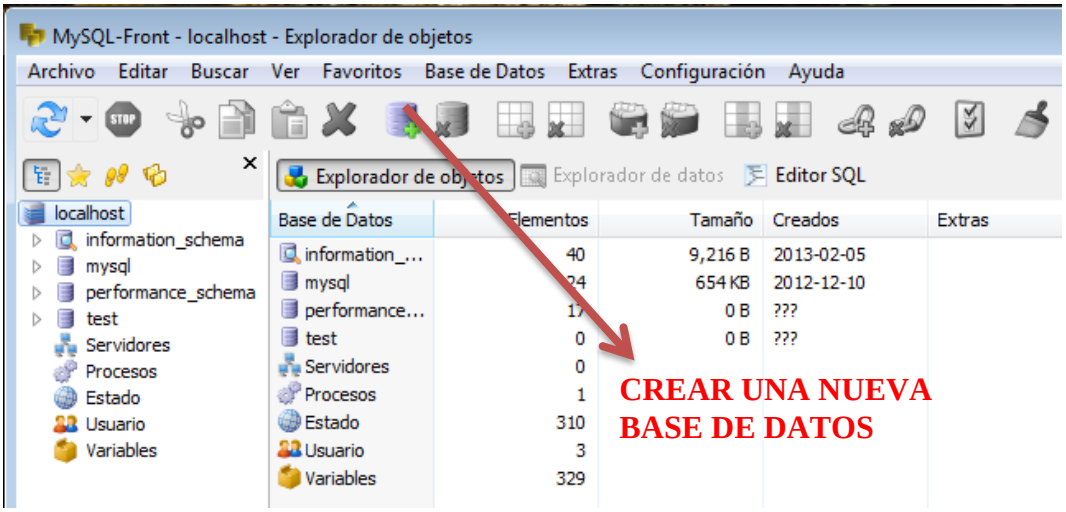

Fig. 47 Creación de una nueva base de datos

Fuente: Autores

Se despliega la pantalla de la Fig. 48, esta pide asignar un nombre a la base de datos, el juego de caracteres hace referencia al tipo de datos a manejar, al ser este

variado (enteros, caracteres y tiempo), estos están contemplados dentro de la opción utf8.

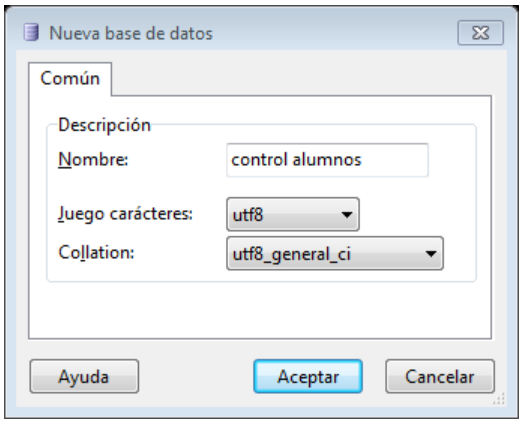

Fig. 48 Nombre de la Base de datos

Fuente: Autores

Después de este paso aparecerá en la pantalla principal la base de datos creada, Fig. 49, lo siguiente es asignar las tablas y las variables de esta base de datos, el proceso es similar para cada tabla, de este modo la explicación está concentrada en la Tabla de alumnos.

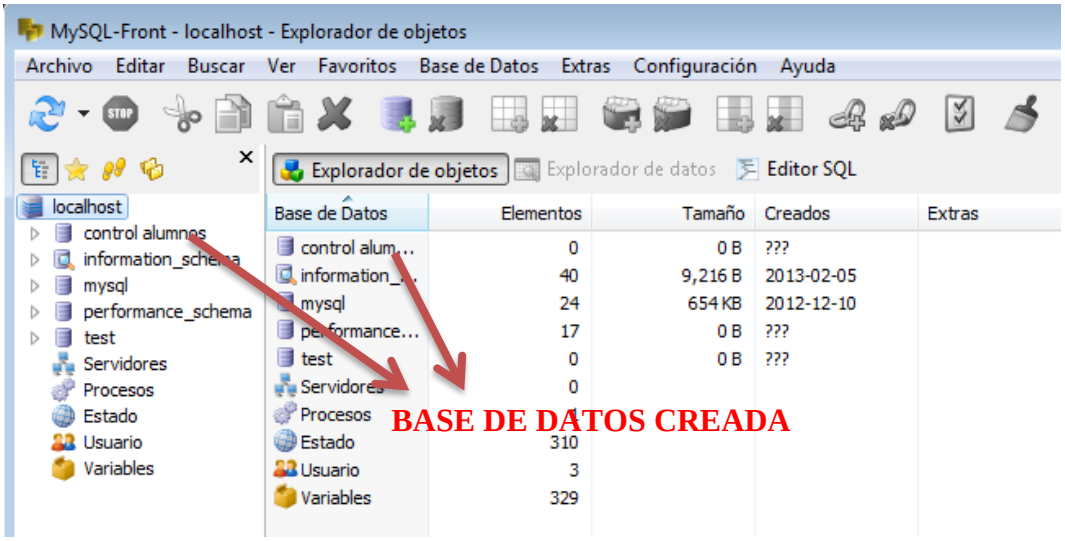

Fig. 49 Base de datos creada

Fuente: Autores

El botón agregar tabla creará una nueva Tabla, Fig. 50.

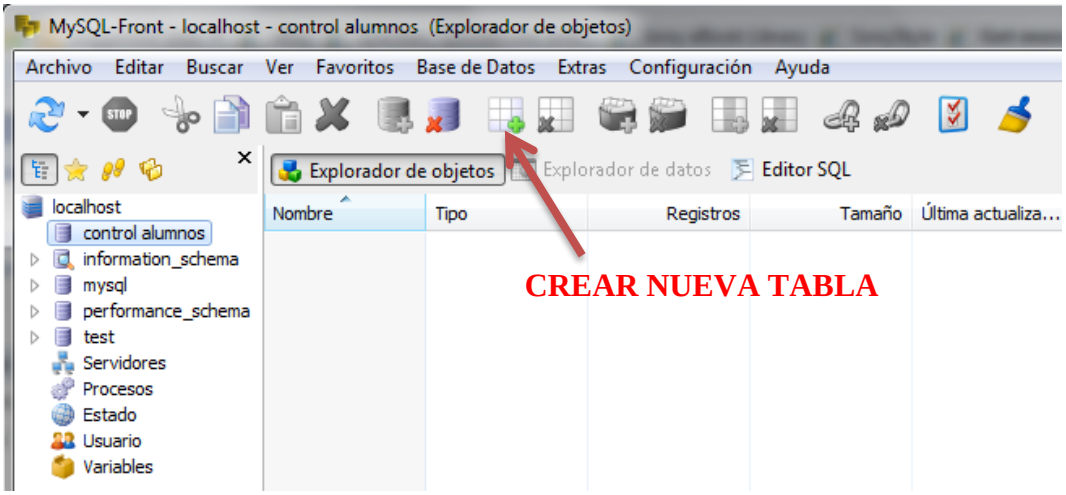

Fig. 50 Creando una tabla

Fuente: Autores

Se abre un cuadro como la Fig. 51, en donde solicita la asignación de nombre y sus características, al completar esta información regresa a la pantalla principal de MySQL Front, Fig. 52, con la tabla ya creada.

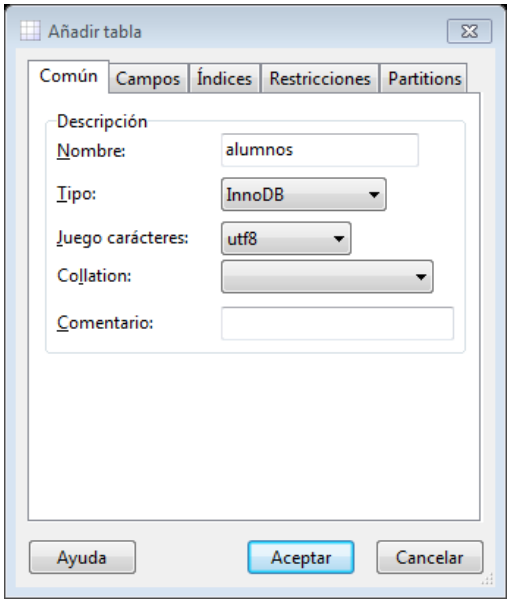

Fig. 51 Creación de Tabla

Fuente: Autores

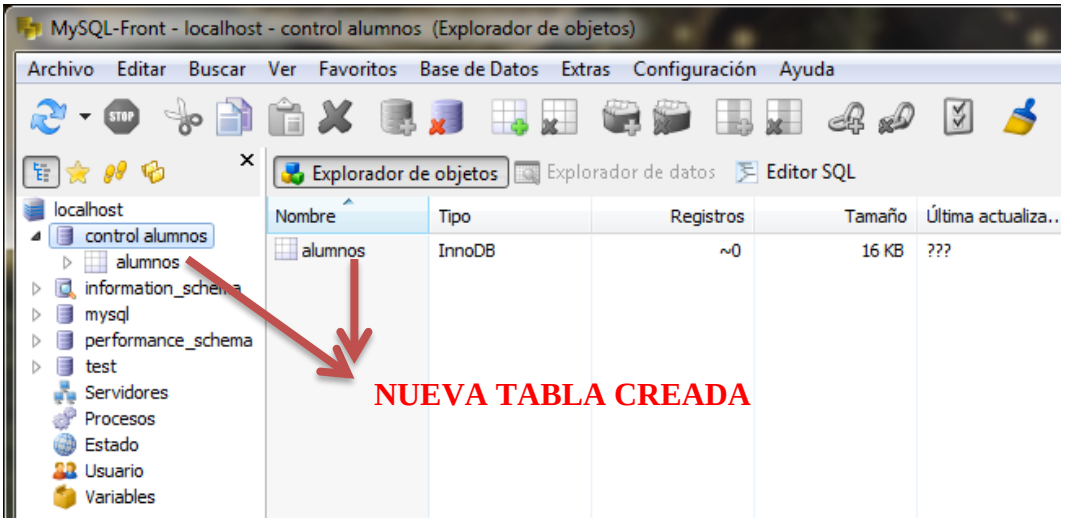

Fig. 52 Nueva Tabla creada

Fuente: Autores

Una vez creada la tabla en la base de datos, el paso siguiente es crear los campos correspondientes en cada una de las tablas, se crean todos los campos especificados en la Fig. 45 y Fig.46, en sus tablas correspondientes, en la tabla Alumnos los campos a crear son: nombre, cédula, Id de matrícula, Id de lector, Huella, para crear un nuevo campo hace falta seleccionar el botón de añadir campo, Fig. 53.

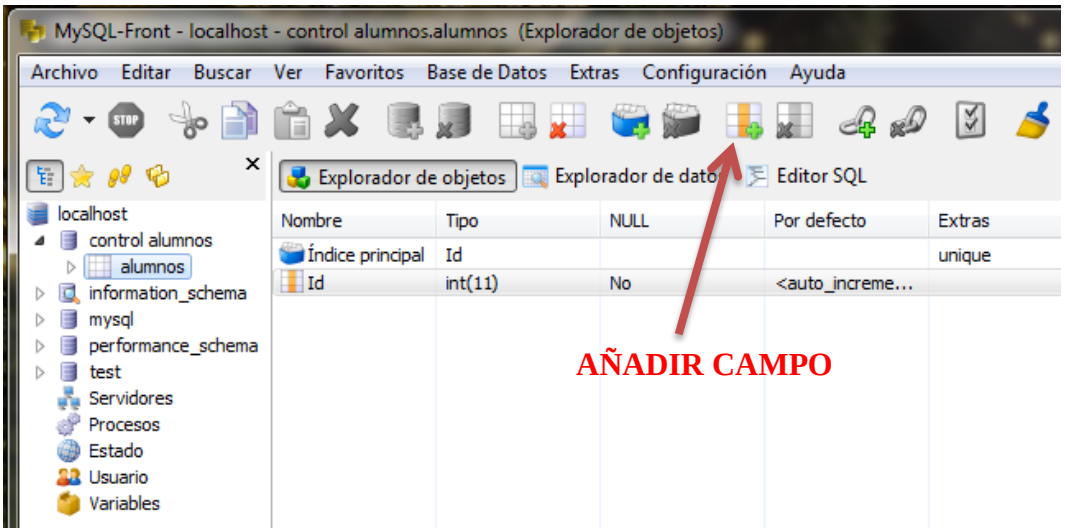

Fig. 53 Añadir un nuevo campo a la Tabla Alumnos

Fuente: Autores

Seleccionado el botón de añadir un nuevo campo el software presenta la pantalla mostrada en la Fig. 54, de la misma manera es necesario agregar los parámetros: posición, nombre, tipo de variable la longitud de caracteres a ingresar, realizado este paso se retornara a la pantalla principal en donde se visualizan los campos creados, Fig. 55.

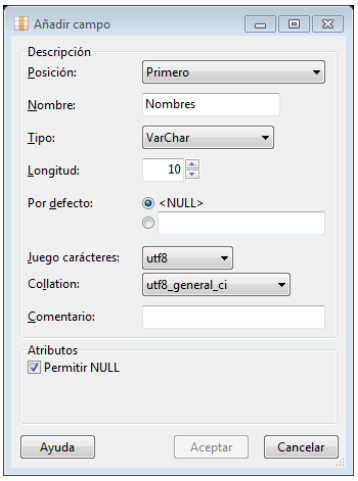

Fig. 54 Añadir nuevo campo

Fuente: Autores

Este procedimiento se ejecutara para cada uno de los campos y tablas a crearse dentro de la base de datos.

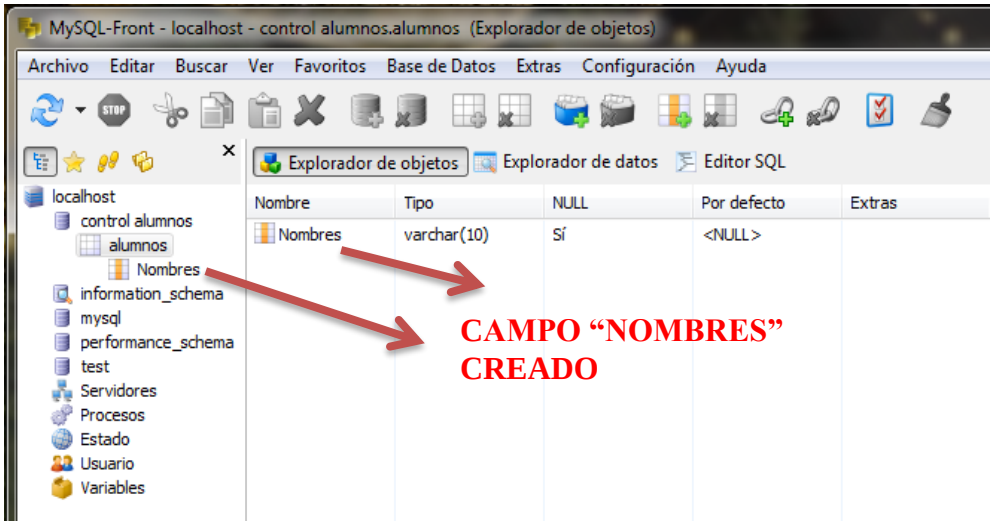

Fig. 55 Nuevo campo creado

Fuente: Autores

2.4.8. Desarrollo del software de control de asistencia de alumnado-Visual Basic.Net

Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que puede ser considerado una evolución de Visual Basic implementada sobre el framework.NET.

El software desarrollado en Visual Basic contiene formularios iguales a los de la base de datos de manera que tengan enlace directo, en esta sección serán analizados cada uno de los formularios y su programación, el módulo principal está representado en la Fig. 56

Fig. 56 Pantalla Principal del Software

Fuente: Autores

La Fig. 57, muestra el menú del sistema y sub menús presentes en el interior de cada uno de ellos.

MODULO ALUMNOS	Registrar huella
	Borrar Huella
	Borrar Datos
	Insertar Registro
	Actualizar Registro
	Eliminar Registro
	Exportar a Excell
MODULO PROFESORES	Insertar Registro
	Actualizar Registro
	Eliminar Registro
	Exportar a Excel
MODULO MATERIAS	Insertar Registro
	Actualizar Registro
	Eliminar Registro
	Exportar a Excel
MODULO HORARIOS	Insertar Registro
	Actualizar Registro
	Eliminar Registro
	Exportar a Excel
MODULO CLASES	Asignar alumno
	Asignar Materia
	Asignar Profesor
	Asignar Horario
	Insertar Registro
	Actualizar Registro
	Eliminar Registro
	Exportar a Excel
MODULO ASISTENCIA	Buscar Registro
	Exportar a Excel
	Especificación de Materia, Profesor y Horario
MODULO DE REPORTES	Fecha Inicial
	Fecha Final
	Buscar Registro
	Exportar a Excel
	Especificación de Materia, Profesor y Horario

Fig. 57 Menú y Submenú del sistema

Fuente: Autores

2.4.9. Módulo Alumnos

La labor principal será el registro de los usuarios, Fig. 58, las variables identificadas en este módulo son:

- ✓ Cédula
- ✓ Nombres
- ✓ ID Matricula
- ✓ ID Lector

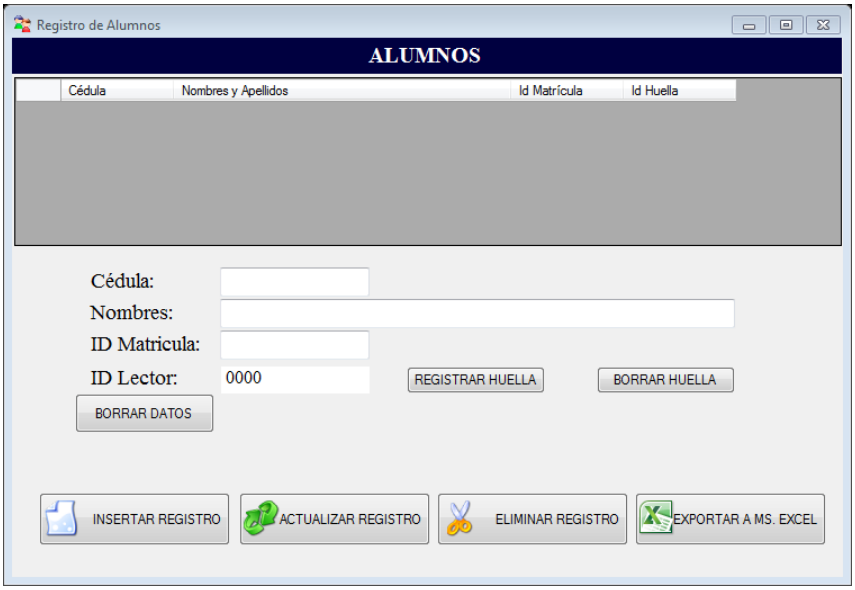

Fig. 58 Módulos de Alumnos

Fuente: Autores

Las opciones manejadas son:

- Ingresar registro.- una vez cargados los datos personales y registrada la huella dactilar del estudiante, hay que finalizar este proceso de registro con el botón insertar registro, la información será almacenada en la base de datos del computador, la huella dactilar por su parte será almacenada en el módulo FIM5360 compartiendo el mismo ID Lector que se almaceno en la base de datos del computador.

Existirán ocasiones en las que por falla humana se ingrese erróneamente los datos de un usuario o la huella dactilar, para estos casos existen opciones como:

- Actualizar Registro.- esta opción me permite realizar modificaciones a los datos de los usuarios ya registrados.

- Eliminar Registro.- Elimina el registro completo del estudiante, previo a este paso se debe primeramente eliminar la huella dactilar.

- Borrar Huella.- permite borrar solo la huella dactilar sin que sea alterado los datos personales del estudiante.

- Exportar a Excel.- Exportar a un archivo Excel la nómina de los usuarios registrados y sus datos.

Estas opciones están presentes en las demás tablas, por lo solo serán analizadas las líneas de código del módulo de registro de alumnos.

```
Dim dato As Integer
Dim buffer(450) As Byte
Dim texto1 As String


Dim ver As String
Dim num As Char
Dim Baud As Integer
Dim comand1(24), comand2(24), comand4(24), comand5(24), comand6(24), comand7(24), comand8(38) As Integer
Dim comand3(54) As Int64
```

De la misma manera que se lo hizo para la programación de la placa Arduino, primeramente se declaran las variables globales a ser utilizadas, cada variable tiene su propio tipo de datos, char, integer o byte respectivamente.

Las siguientes líneas de código son los comandos de uso exclusivo para la tarjeta biométrica, para poder enviar los comandos y que estos sean reconocidos por la tarjeta hay que realizar un cambio en los mismos, el comando hexadecimal se

cambia por su equivalencia en número decimal y es enviado por el puerto serial como carácter de este modo el reconocimiento de los datos en el módulo FIM5360 es exitoso.

```
comand1 = {126, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1} ' Comando CMD_REQUEST CONECTION

comand4 = {126, 0, 0, 0, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 22} ' Comando CMD_GET_TEMPLATE

comand2 = {126, 0, 0, 0, 47, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 50}  ' Comando CMD_ENTER_MASTER_MODE2

comand3 = {126, 0, 0, 0, 51, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 26, 0, 0, 0, 0, 0, 0, 0, 77, 48, 48, 48, 48, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 192} ' Comando CMD_REGISTER_FP

comand5 = {126, 0, 0, 0, 104, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 104}  ' Comando CMD_GET_IMAGE_QUALITY

comand6 = {126, 0, 0, 0, 51, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 52}  ' Comando CMD_REGISTER_FP2

comand7 = {126, 0, 0, 0, 18, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 19}  ' Comando CMD_IDENTIFY_FP

comand8 = {126, 0, 0, 0, 34, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10, 0, 0, 0, 0, 0, 0, 0, 44, 48, 48, 48, 48, 0, 0, 0, 0, 0, 0, 0, 0, 0, 192}  ' Comando CMD_DELETE_FP
```

Las siguientes líneas de código realizan una consulta a la base de datos de MySQL, de modo de hacer un reconocimiento de las variables y que cada dato ingresado sea registrado en la base de datos sin errores.

```
DgAlumnos.DataSource = Consultar_Tabla_MySQL("Select * from alumnos;")
    DgAlumnos.Columns(0).Width = 100
    DgAlumnos.Columns(0).HeaderText = "Cédula"
    DgAlumnos.Columns(1).Width = 300
    DgAlumnos.Columns(1).HeaderText = "Nombres y Apellidos"
    DgAlumnos.Columns(2).Width = 100
    DgAlumnos.Columns(2).HeaderText = "Id Matrícula"
    DgAlumnos.Columns(3).Width = 100
    DgAlumnos.Columns(3).HeaderText = "Id Huella"

        DgIdLector.DataSource = Consultar_Tabla_MySQL("Select
        max(IdLector)+1 as IDL FROM alumnos")

  If Not IsDBNull(DgIdLector.Rows(0).Cells(0).Value) Then
    TxtIdLect.Text = DgIdLector.Rows(0).Cells(0).Value
  Else
```

2.4.9.1.1. Registro de Huella

Como ya se mencionó en una sección anterior el registro de usuarios es realizado de forma manual, el ID de usuario debe ser declarado por el administrador del sistema debiendo ingresar cada ID compuesto por cuatro dígitos que pueden ir desde el 0000 hasta el 9999, para evitar errores de digitación se ha considerado dentro de la programación el ingreso del ID de modo incremental , programando esto desde la base de datos, este ID deberá sumarse al comando y luego ser enviado, todo esto es obtenido con las siguientes líneas de comandos.

```
comand3 = {126, 0, 0, 0, 51, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 26, 0, 0, 0, 0, 0, 0, 0, 77,
48, 48, 48, 48, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 192}

comand3(25) = comand3(25) + Val(Mid(TxtIdLect.Text, 1, 1))
TextBox1.Text = TextBox1.Text & comand3(25)
comand3(26) = comand3(26) + Val(Mid(TxtIdLect.Text, 2, 1))
```

```
TextBox1.Text = TextBox1.Text & comand3(26)
comand3(27) = comand3(27) + Val(Mid(TxtIdLect.Text, 3, 1))
TextBox1.Text = TextBox1.Text & comand3(27)
comand3(28) = comand3(28) + Val(Mid(TxtIdLect.Text, 4, 1))
TextBox1.Text = TextBox1.Text & comand3(28)
```

El comando final o de Checksum es la suma del ID de lector para lo cual disponemos de la siguiente línea de comando.

```
comand3(54) = comand3(25) + comand3(26) + comand3(27) + comand3(28)

TextBox1.Text = TextBox1.Text & comand3(54)
```

Este procedimiento es el mismo para la eliminación de usuario, lo único que deberá variar será el código enviado a la tarjeta.

Las siguientes líneas de código son usadas para la verificación de los datos, cada vez que es enviado un comando a la tarjeta, esta regresa un comando de confirmación en el que especifica parámetros que indican si la acción fue exitosa o no, de ser exitoso continua con la acción, por el contrario de no serlo el software enviará un mensaje de error de conexión con el dispositivo, esta instrucción es colocada después del envío de cada comando con el fin de verificar que la recepción haya sido correcta, tan solo habrá que variar la condición de acuerdo al comando enviado.

```
For v As Integer = 0 To 24
        TextBox1.Text = TextBox1.Text & SERIAL.ReadChar()
        If v = 8 Then
           If TextBox1.Text <> "126000470001" Then
                    MsgBox("Error de coneccion con el dispositivo",
        MsgBoxStyle.Critical, "Error dispositivo")
                      Exit Sub
```

```
                End If
        End If
Next v
```

Las líneas de código siguientes corresponden a los comandos de registro de huella dactilar.

```
MessageBox.Show("Coloque su dedo en el sensor")

    For v As Integer = 0 To 54
       num = Convert.ToChar(comand3(v))
       SERIAL.Write(num)
    Next

    MessageBox.Show("Verifique su huella")
    For v As Integer = 0 To 24
       num = Convert.ToChar(comand6(v))
       SERIAL.Write(num)
    Next
```

Una vez seleccionada la opción de registro de huella mostrara un mensaje en el que pide colocar el dedo en el sensor, con lo que disminuye los errores por exceso en tiempo de captura, con este comando el módulo capturará la huella dactilar, para que el proceso quede completo necesitamos verificar la huella de usuario para esto el software muestra un segundo mensaje en el que pedimos "verificar la huella" con lo que el módulo capturará la segunda huella y la almacenará en la base de datos completando de este modo el proceso de registro.

```
For v As Integer = 0 To 24
 TextBox1.Text = TextBox1.Text &
 SERIAL.ReadChar()

  If v = 8 Then
   If TextBox1.Text <> "126000510001" Then
```

```
        MsgBox("Error registro no almacenado", MsgBoxStyle.Critical,
        "Error dispositivo")
        Exit Sub

    Else

      TxtHuella.Text = "SI"
        Ejecutar_Transaccion_MySQL("UPDATE alumnos SET Cedula='"
        & TxtCed.Text & "', Nombres='" & TxtNomb.Text & "', IdMatricula='" &
        TxtIdMat.Text & "', IdLector='" & TxtIdLect.Text & "',Huella='" &
        TxtHuella.Text & "' Where Cedula='" & TxtCed.Text & "';")
         DgAlumnos.DataSource = Consultar_Tabla_MySQL("Select * from
        alumnos;")

     End If
    End If
   Next v
```

En las líneas de comando anteriores está ejecutada la verificación del comando de registro que me devuelve el módulo una vez que haya comprobado que el usuario es almaceno correctamente, se realiza una comparación del comando de respuesta, si el comando de respuesta es distinto de “126000510001”, significara que hay un error en la transmisión, es enviado entonces un mensaje de error del dispositivo, cuando la huella es registrada correctamente asignara un valor texto “SI” como indicador de comando exitoso, por medio de los comandos de actualización de tabla de MySQL permite que este valor sea actualizado en la base de datos con lo que indica que el usuario tiene ya registrada la huella en la base de datos.

2.4.9.1.2. Ingresar Registro

Se utilizará para registrar los datos personales de cada usuario, puede incluir la huella dactilar o puede hacer un registro independiente e ingresar posteriormente la huella, las líneas de comando que acompañan a esta acción son las siguientes:

```
Private Sub ButInsert_Click(sender As System.Object, e As System.EventArgs)
Handles ButInsert.Click

Ejecutar_Transaccion_MySQL("insert into alumnos (Cedula, Nombres,
IdMatricula, IdLector) values ('" & TxtCed.Text & "','" & TxtNomb.Text & "','" &
TxtIdMat.Text & "'," & TxtIdLect.Text & ");")

DgAlumnos.DataSource = Consultar_Tabla_MySQL("Select * from alumnos;")

End Sub
```

Lo que hace es ejecutar los comandos que permitan editar la base de datos, para el caso de registro utilizará el comando "Ejecutar_Transaccion_MySQL" seguido a este comando señalaremos los títulos de cada columna de la base de datos y el lugar de donde se extrae los datos para ser almacenados, en este caso es el uso de varios Textbox, que permitirán al usuario hacer el ingreso manual de los datos a través de la edición en estos Textbox, la última línea de código permite consultar el estado actual de la tabla con los últimos datos recogidos.

2.4.9.1.3. Actualizar registro

Cuando los datos no han sido ingresados correctamente, o se han registrados de una manera incompleta el usuario debe poder manejar una opción que le permita modificar los datos ya ingresados, para esto el sistema dispone de un botón de actualización, las líneas de este código se enfocan en leer cada una de la entradas de texto y actualizarlos a la base de datos.

```
Private Sub ButUpdate_Click(sender As System.Object, e As System.EventArgs)
Handles ButUpdate.Click

Ejecutar_Transaccion_MySQL("UPDATE alumnos SET Cedula='" &
TxtCed.Text & "', Nombres='" & TxtNomb.Text & "', IdMatricula='" &
```

TxtIdMat.Text & "', IdLector='" & TxtIdLect.Text & "',Huella='" & TxtHuella.Text & "' Where Cedula='" & TxtCed.Text & "';")

DgAlumnos.DataSource = Consultar_Tabla_MySQL("Select * from alumnos;")

End Sub

Este comando se repetirá en varias partes del software de modo que cada cambio que realizado en la información de un usuario pueda ser visto por el administrador de manera inmediata reduciendo errores a causa de este motivo.

2.4.9.1.4. Eliminar registro

La administración del sistema tiene situaciones en las que es más conveniente eliminar el usuario en lugar de editar todos sus datos, para una correcta eliminación del usuario primero se realiza la eliminación de la huella dactilar de modo que no se produzcan errores por huella repetidas, en estas líneas de comando también se distingue el uso de actualización o consulta de tabla.

Private Sub ButElemina_Click(sender As System.Object, e As System.EventArgs) Handles ButElemina.Click

Ejecutar_Transaccion_MySQL("DELETE FROM alumnos WHERE Cedula='" & TxtCed.Text & "';")

DgAlumnos.DataSource = Consultar_Tabla_MySQL("Select * from alumnos;")

End Sub

2.4.9.1.5. Exportar a Excel

Como una acción adicional el sistema dispone de un botón que permite exportar los datos correspondientes a esta tabla hacia un archivo de Excel, permitiendo así

que los datos sean manipulables por los usuarios, los datos exportados a Excel muestran los usuarios registrados en la base de datos y los títulos de cada columna creada, Fig. 59.

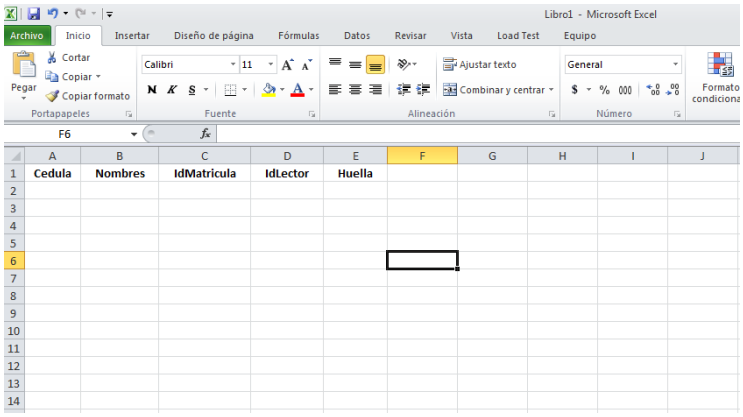

Fig. 59 Apariencia de datos en Excel

Fuente: Autores

2.4.9.2. Módulo de Profesores

La Fig. 60, muestra la apariencia del módulo de registro de profesores.

Fig. 60 Modulo de registro de profesores

Fuente: Autores

Este módulo solicita el registro de cada profesor con un ID correspondiente a cada uno, las opciones que este comando maneja son similares a las mencionadas en el módulo de Alumnos, por lo que no se repetirá la explicación de cada botón.

2.4.9.3. Módulo de Materias

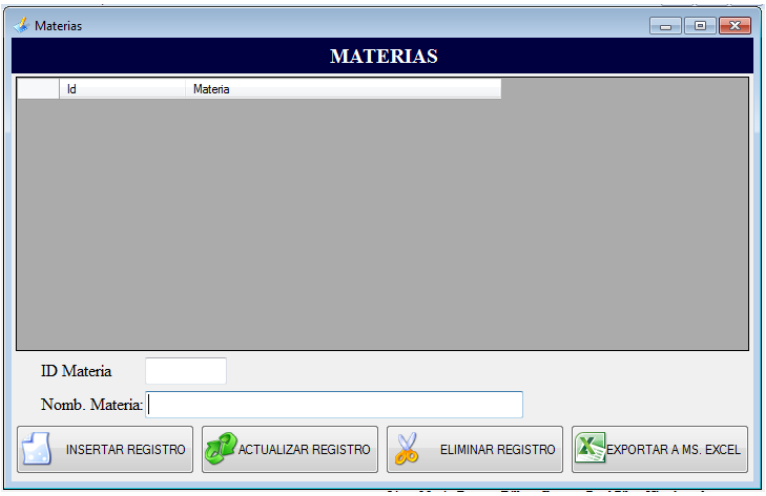

Fig. 61 Módulo registro de Materias

Fuente: Autores

Este módulo solicita el ingreso de materias, con un respectivo ID para cada una, dato que será requerido posteriormente en el módulo clases, la Fig. 61 muestra la apariencia de este módulo.

2.4.9.4. Modulo Horarios

Fig. 62 Módulo registro de horarios

Fuente: Autores

La Fig. 62, muestra la apariencia del módulo de registro de horarios de clases, este módulo tiene una particularidad, maneja los datos de tiempo, es decir la hora de inicio y fin, que serán cargadas junto con un ID para identificarlas, previamente en la base de datos hay que delimitar las variables y de qué tipo serán estas, las variables Hora Inicio y Hora Fin: son del tipo Time.

```
Private Sub ButInsert_Click(sender As System.Object, e As System.EventArgs)
Handles ButInsert.Click

Ejecutar_Transaccion_MySQL("insert into horarios (HoraIni,HoraFin) values ('"
& TxtHoraIni.Text & "','" & TxtHoraFin.Text & "');")

DgHorarios.DataSource = Consultar_Tabla_MySQL("Select * from horarios;")

End Sub
```

2.4.9.5. Módulo de Clases

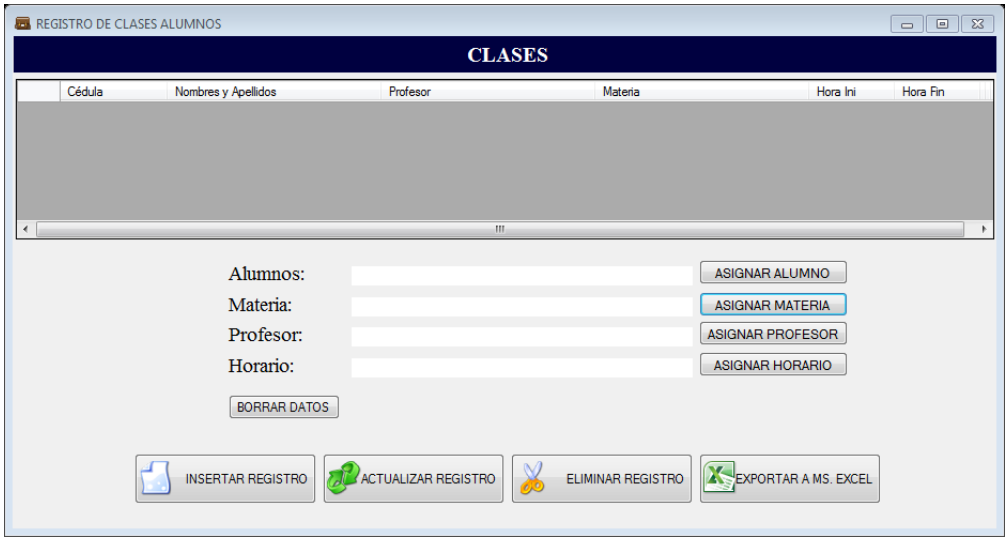

Fig. 63 Apariencia del módulo Clases

Fuente: Autores

Este módulo Fig. 63, permite concatenar todos los datos ingresados anteriormente, hay que asignar a cada alumno a la asignatura correspondiente, asignar el profesor y la hora de cada asignatura, hay que enfatizar que el módulo de horario no se especifica fechas ni tampoco días, por tanto el registro de un estudiante en una determinada asignatura deberá ser replicado para cada una de las horas semanales que recibe esa clase, cumplido con este módulo queda finalizado el proceso de inscripción y asignación de materias a cada usuario además al igual que en los anteriores puede exportarse los datos a Excel para que sean operados o manipulados.

2.4.9.6. Módulo de Asistencia

Este módulo presenta la información de la base de datos de acuerdo al control de asistencia de cada alumno, es un módulo enteramente de consulta, en donde el usuario deberá seleccionar en cada ítem dependiendo de su necesidad.

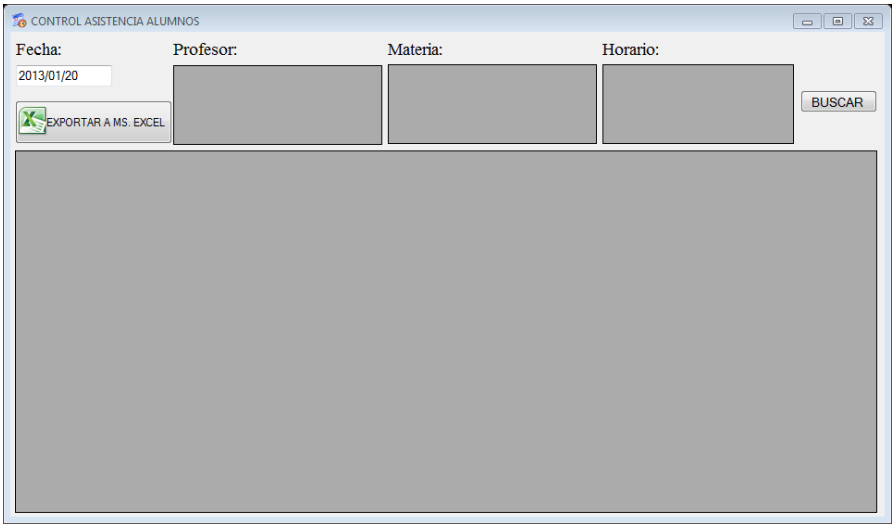

Fig. 64 Apariencia del módulo Asistencia

Fuente: Autores

En la Fig. 64, se identifica:

- Fecha.- seleccionar el día a conocer, la fecha deberá ser ingresada manualmente y con el siguiente formato, AAAA/MM/DD.

- Profesor.- seleccionar el profesor que dicta la asignatura a ser consultada.
- Materia.- seleccionar la asignatura del profesor seleccionado anteriormente, puede parecer redundante el requerimiento pero puede darse el caso de que existan asignaturas duplicadas con profesores distintos, por ejemplo: Física, Análisis Matemático, Telemática.
- Horario.- seleccionar la hora correspondiente a la asignatura dictada que va a ser consultada, debido a que las asignaturas son tomadas varias horas a la semana.

Una vez seleccionado todos los parámetros, el botón de consulta o "Search" deberá ser activado, como resultado desplegará una lista de los estudiantes que estén registrados bajo estos parámetros, una de estas columnas de la lista contiene la información de control de asistencia de cada estudiante registrado, para esto existen dos opciones: "asistió y no asistió", lo que significa que el estudiante puede registrarse en cualquier momento de la hora clase.

Al igual que en los otros módulos aquí también se tiene un botón de exportación a tablas de Excel para que los datos puedan ser impresos.

2.4.9.7. Módulo de Reportes

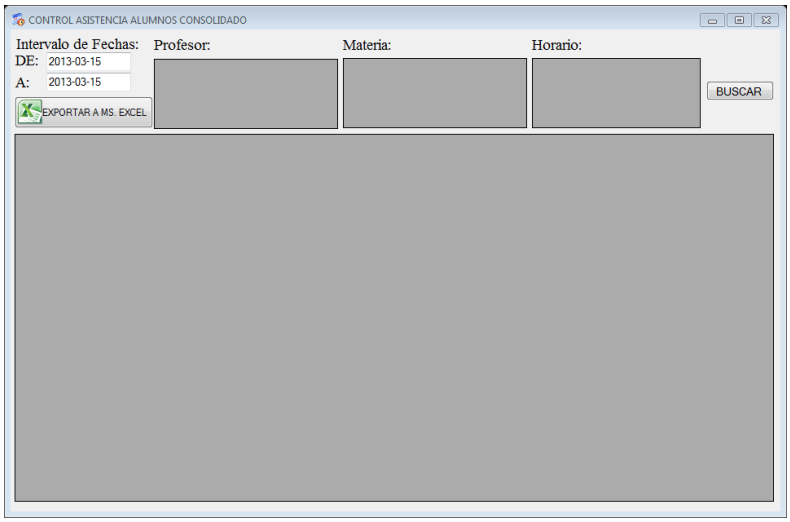

Fig. 65 Pantalla principal del sistema de control de alumnos
Fuente: Autores

Este es un módulo adicional que se ha agregado al sistema, permite obtener un informe de la asistencia del alumnado estableciendo un rango de consulta, para lo cual se pide ingresar además de los datos ya conocidos, el rango de fecha a ser consultado, en el formato de AAAA-MM-DD, Fig. 65, partiendo de una fecha inicial de consulta a una fecha final, cuando una asistencia sea registrada será representada con un "1", y con un "0" en caso de falta, el total de la asistencia resulta de la suma de todos los días consultados dentro del rango.

2.4.10. Montaje Final del Equipo

Para las pruebas iniciales del equipo las conexiones se encuentran montadas en un proto board que permite la movilidad en las conexiones y ajustes en las interconexiones de los distintos módulos, Fig.66, el esquema del equipo en su totalidad es explicado en el diagrama de bloques de la Fig.67 y su diagrama esquemático de conexiones se encuentra en el Anexo 2.

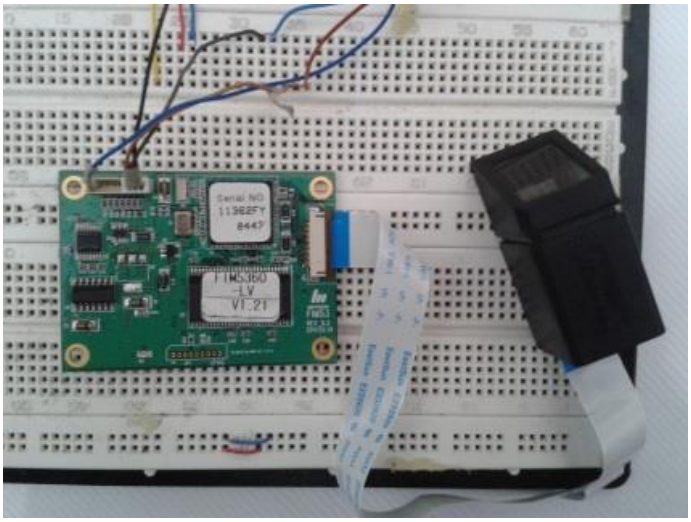

Fig. 66 Conexionado en Proto Board
Fuente: Autores

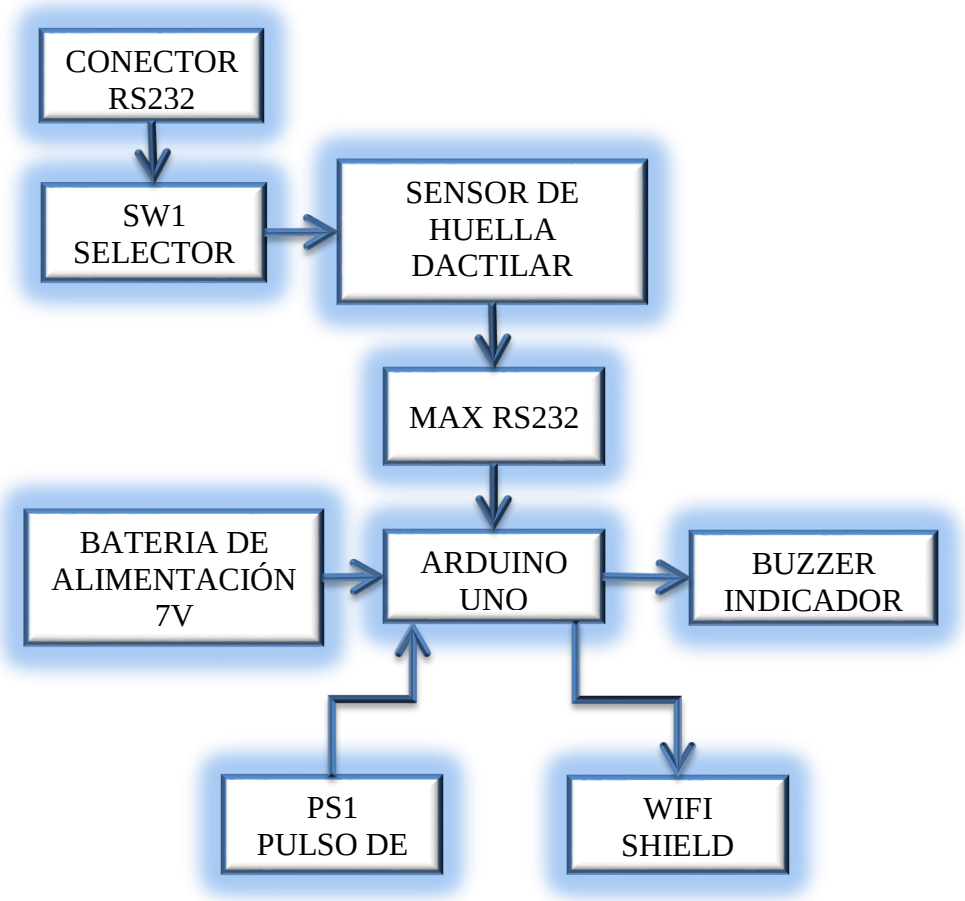

Fig. 67 Diagrama de bloques del funcionamiento del equipo completo
Fuente: Autores

Una vez corridas las pruebas en datos simulados y comprobado la efectividad del mismo se procede a realizar las pruebas en datos reales con el alumnado de las asignaturas ya antes mencionadas, para poder realizar estas pruebas fue necesario montar todo el sistema en una estructura que permita su fácil manipulación, el equipo es montado en una carcasa de madera donde se distribuye cada uno de los módulos teniendo en cuenta la función, dimensión y espacio entre cada uno de ellos. Las Fig.68 y Fig.69, muestran la imagen inicial de la caja en donde se montaran todos los dispositivos.

Las dimensiones de la caja son: 8 x 4.8 x 10.58 cm, cabe recalcar que es un montaje previo de los módulos para determinar los espacios requeridos para cada una de ellas.

Fig. 68 Pruebas iniciales del montaje del Equipo
Fuente: Autores

Fig. 69 Vista Frontal de la distribución del equipo
Fuente: Autores

Una vez determinado el lugar de montaje de cada uno de los módulos, la caja fue pintada para mejorar la presentación del equipo, posterior a esto viene el montaje final con todos los módulos completos, la alimentación inicial del sistema estuvo a cargo de una batería de 9V de uso común, el consumo total del equipo es de 700mA, 5Vcd, la alimentación es entregada a la placa Arduino quien se encarga de la distribución de los voltajes, esta placa cuenta con un regulador de voltaje interno que permite descender el voltaje de 9 Vcd a 5Vcd y 3.3Vcd, al correr el sistema el consumo de las baterías es muy acelerado, la solución fue remplazar la

batería inicial por una batería de 7Vcd – 1100mA, reciclada de una filmadora, Fig. 70.

Fig. 70 Batería de filmadora 7Vcd – 1100mA

Fuente: Autores

Superado el inconveniente, empieza las perforaciones en la caja para el conector RS232, switch de encendido, selector de modo de operación, led indicador, sensor de huella dactilar, antena de transmisión, estos dispositivos mencionados van soldados a la placa de acoplamiento TTL a RS232, con la finalidad de darle mayor soporte al equipo, Fig.71.

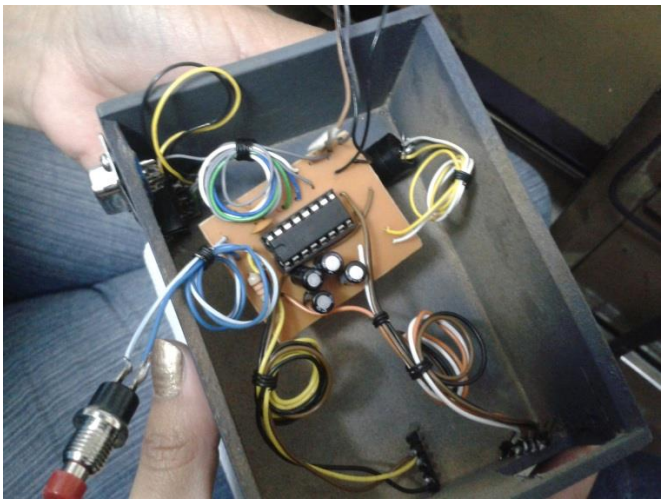

Fig. 71 Montaje de placa de acoplamiento de interfaces

Fuente: Autores

La conexión de la placa de acoplamiento de interfaces hacía la placa Arduino es mediante espadines de manera que tenga facilidad de conexión en caso de realizar alguna reparación, el montaje de las dos placas está en la Fig. 72

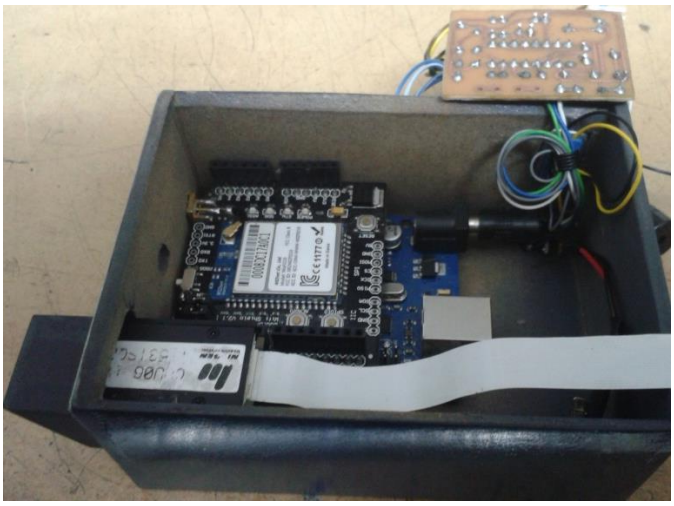

Fig. 72 Montaje de la placa Arduino

Fuente: Autores

La prueba de conexión final es realizada con la finalidad de comprobar que todo este correctamente instalado antes de fijar las tarjetas a la caja de montaje, Fig.73.

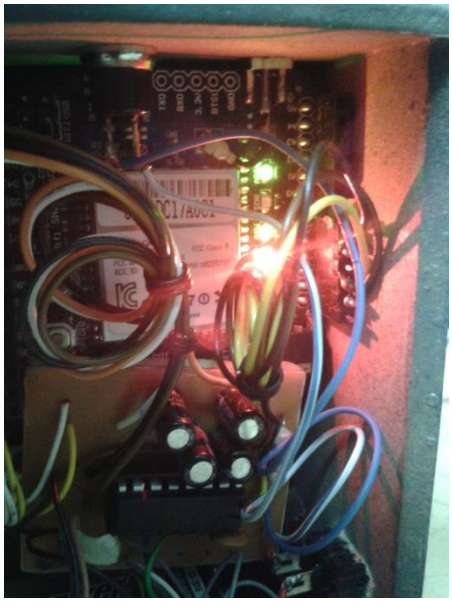

Fig. 73 Pruebas de conexión

Fuente: Autores

Fijadas las placas, la presentación del equipo es fundamental, un sello representativo de la Universidad Nacional de Chimborazo fue agregado a la parte frontal del equipo, las siguientes imágenes presentan una vista de cada uno de los perfiles del equipo identificando todas sus partes, procurando además que el tamaño del equipo sea reducido, para mejorar su portabilidad, sin que eso afecte su eficiencia.

Fig. 74 Tamaño del equipo

Fuente: Autores

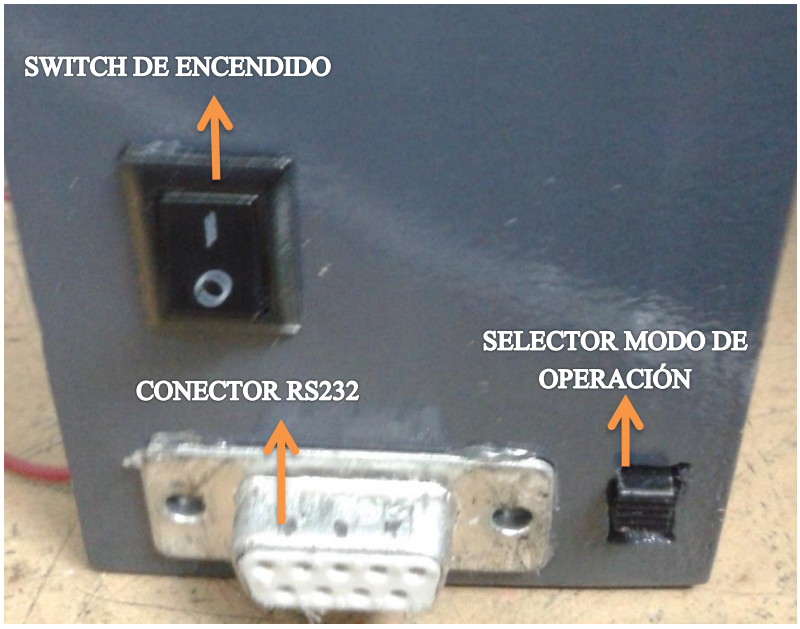

Fig. 75 Vista de perfil

Fuente: Autores

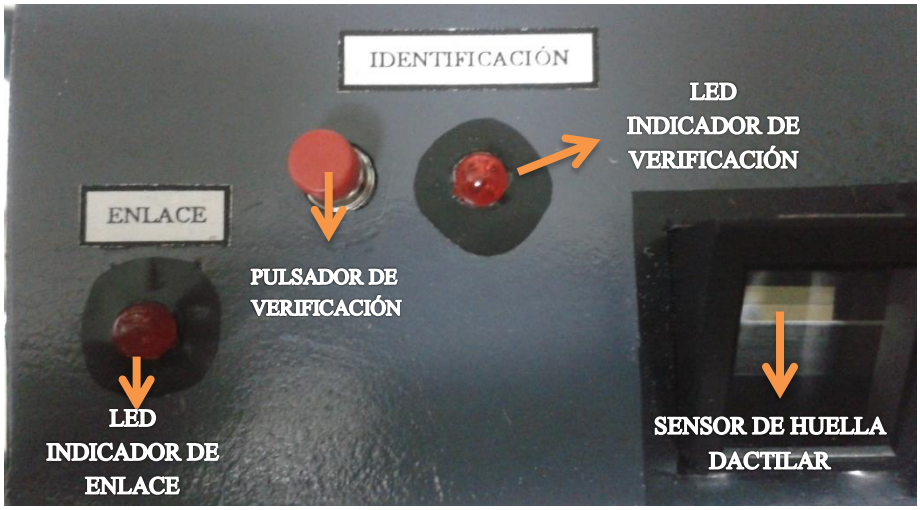

Fig. 76 Vista Posterior

Fuente: Autores

Fig. 77 Vista Frontal del equipo terminado

Fuente: Autores

2.4.11. Conexionado a la Red

Para la conexión a la red es necesario primeramente configurar el equipo que se encargará de generar la red, para este caso se utilizó un Router TP-LINK, con la siguiente configuración:

- SSID: ALUMNOS
- Encriptación; WAP2-PSK
- Password: 12345678

CAPITULO III

3. RESULTADOS

Este capítulo hace referencia a pruebas del equipo realizadas en datos reales. Las pruebas en datos reales fueron realizadas con los estudiantes correspondientes a las asignaturas del Ing. Fabian Gunsha docente de la Facultad de Ingeniería y tutor de esta investigación, dado a que presto las facilidades de acceso a las aulas en horarios de clase para el registro del alumnado y posterior control de asistencia del mismo.

Las asignaturas, así como el horario de clases lo muestra la Tabla 14.

Hora	Lunes	Martes	Miércoles	Jueves	Viernes
MAÑANA					
7:10 – 8:50	Telemática	Microprocesadores	Telemática	Telemática	
8:50 – 10:30			Microprocesadores	Microprocesadores	S. Control
12:10 – 13:50				S. Control	
TARDE					
17:30 – 19:10		Electrónica I			
19:10 – 20:50				Electrónica I	

Tabla 14 Horario de clases Ing. F. Gunsha
Fuente: Autores

Los alumnos por materia están detallados a continuación:

Telemática	14
Microprocesadores	42
Electrónica I	32
S. Control Automático	26

Los estudiantes de estas materias suman 114, sin embargo este no es el total ya que no concuerda con los estudiantes registrados, esto debido a que existen estudiantes matriculados en varias materias haciendo que su registro sirva para todas ellas.

La Fig. 78, presenta una gráfica de conjuntos en donde los puntos convergentes muestran las asignaturas con estudiantes comunes.

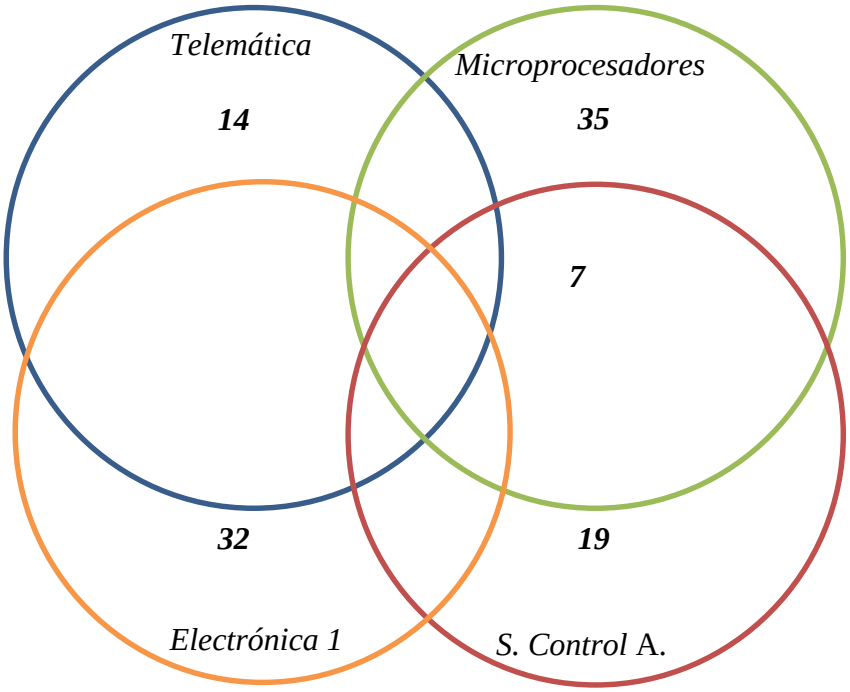

Fig. 78 Relación Estudiantes Vs Asignaturas

Fuente: Autores

3.1. REGISTRO DE ESTUDIANTES

El registro de los estudiantes es realizado con el sistema completo, lo que incluye: Equipo de registro de huellas dactilares, computador, software de registro, cable USB – Serial, Fig. 79.

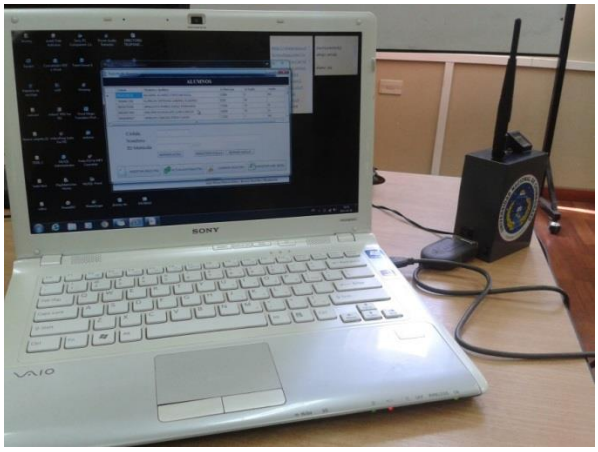

Fig. 79 Sistema completo para el registro de alumnado

Fuente: Autores

El registro de los estudiantes fue realizado en las horas clase de acuerdo al horario indicado en la tabla 15, las Fig. 80, 81, 82 y 83 muestran las evidencias del registro.

Fig. 80 Aula de Electrónica I

Fuente: Autores

Fig. 81 Registro a los estudiantes de S. Control

Fuente: Autores

Fig. 82 Registro a los estudiantes de Electrónica I

Fuente: Autores

Fig. 83 Registro de usuarios en el módulo alumnos

Fuente: Autores

El resultado de esta actividad consiguió un total de 107 estudiantes registrados, de estos 3 tuvieron errores que fueron superados, con estos datos se aplicó una regla de tres para determinar el porcentaje de error producido en las pruebas de registro.

$$\textit{Total de estudiantes registrados} = 107 \qquad 100\%$$
$$\textit{Total de errores producidos} = 3 \qquad X$$

$$X = \frac{3 * 100\%}{107} = 2.80\ \%$$

La Fig. 84, muestra en un diagrama de pastel la relación de error producido.

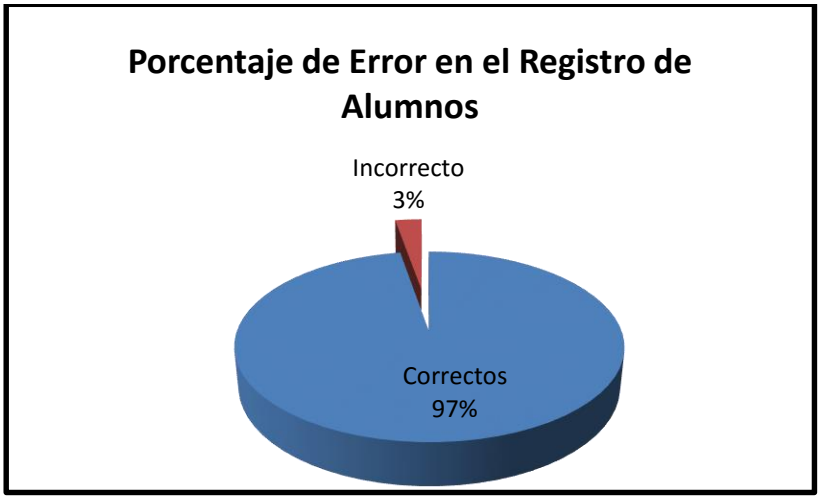

Fig. 84 Porcentaje de error en el registro de alumnos

Fuente: Autores

Este porcentaje de error puede parecer alto sin embargo tomado en consideración el motivo que los produjo, ayudara a demostrar que no lo es.

Es necesario mencionar los motivos que causaron estos errores para que no vuelvan a producirse en futuras ocasiones.

- Posición incorrecta del dedo sobre el sensor
- Impurezas, grasa o humedad presente en el dedo
- Incorrecta operación del sistema de registro de huella dactilar

3.2. CONTROL DE LA ASISTENCIA DEL ALUMNADO (IDENTIFICACIÓN)

Por otro lado las pruebas de control de asistencia del alumnado usando el método investigado que corresponde a la identificación (1:N), fueron realizadas con los estudiantes registrados en cada una de las materias antes mencionadas, de este modo obtuvo un total de 114 pruebas, debido a que las aulas de la Facultad de ingeniería no son uniformes en su área, ofrecen varios entornos para la realización de estas pruebas, la Tabla 15, muestra una relación de los resultados respecto al rango de cobertura y la intensidad de señal de la red inalámbrica.

Distancia	Nivel de Señal	Resultado
2 m	Excelente	Exitoso
5 m	Excelente	Exitoso
10 m	Excelente	Exitoso
12 m	Buena	Exitoso
15 m	Buena	Exitoso

Tabla 15 Relación distancia – nivel de señal
Fuente: Autores

Para determinar el porcentaje de error de las pruebas de control de asistencia (identificación), fue utilizado el mismo método que para el registro de usuarios obteniendo así un total de dos errores, de 114 pruebas.

$$\textit{Total de pruebas realizadas} = 114 \quad 100\%$$
$$\textit{Total de errores producidos} = 2 \quad X$$

$$X = \frac{2 * 100\%}{114} = 1.75\ \%$$

La Fig. 85, muestra en un diagrama de pastel la relación de error producido para la identificación de usuarios.

Fig. 85 Relación de error producido en el control de asistencia

Fuente: Autores

El porcentaje de error resultante es relativamente bajo, considerando que fueron producto de que los dedos de los usuarios se encontraban con grasa, mismos que fueron superados después de limpiarlos.

Como resultado de la pruebas el software, mostró la nómina de usuarios asistidos a clases, de la misma manera fue reportado la inasistencia de algunos alumnos que no asistieron el día de la realización de las pruebas, las Figuras 86 y 87 muestran estos resultados.

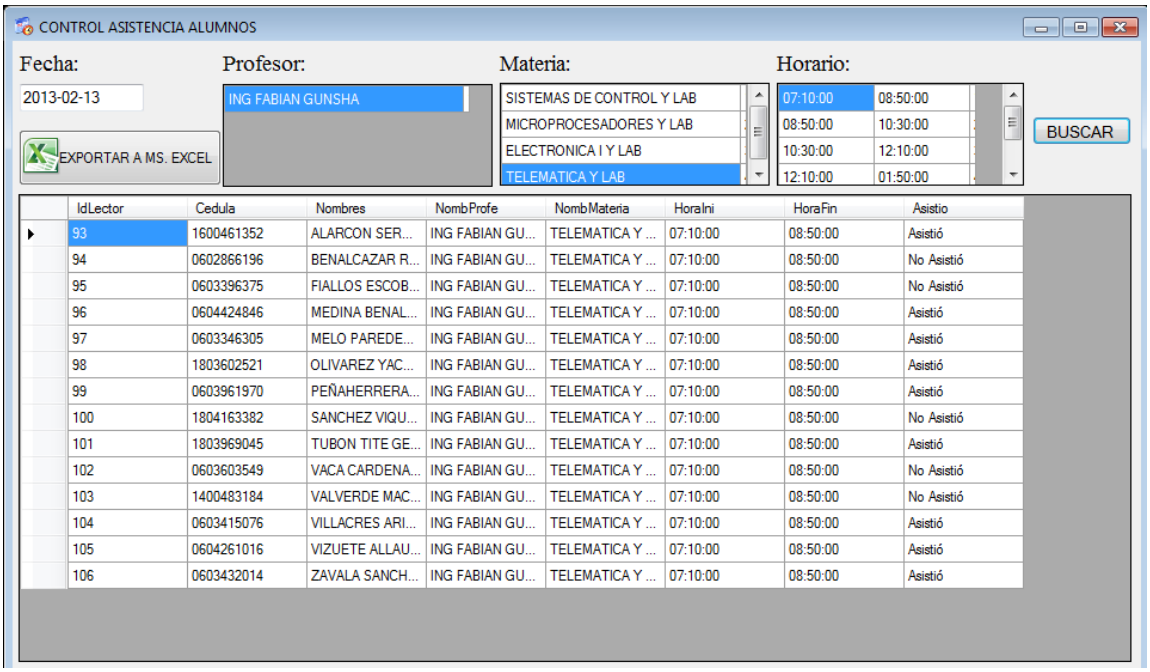

Fig. 86 Nómina de estudiantes de la Asignatura de Telemática

Fuente: Autores

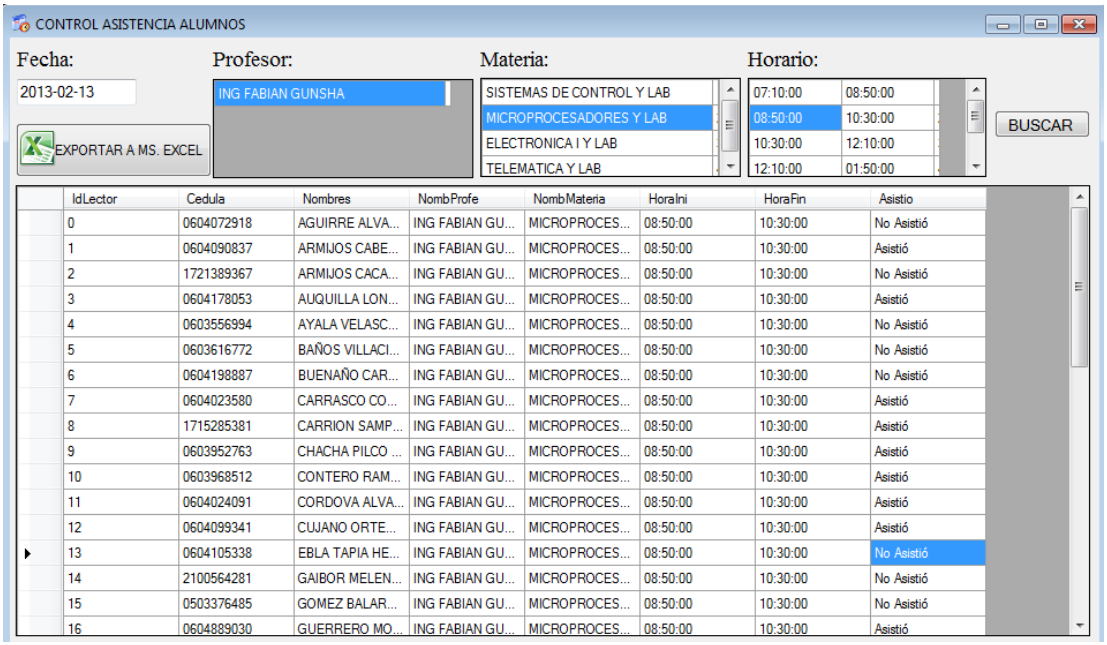

Fig. 87 Nómina de estudiantes de la Asignatura de Microprocesadores

Fuente: Autores

3.3. ANÁLISIS FINANCIERO

La tabla. 16 muestra el costo total del proyecto, de donde se deduce que la inversión que representa la implementación de este equipo es reducida en comparación con los beneficios y la eficiencia que representa.

COSTO DE LA INVERSION			
CANTIDAD	**DETALLE**	**COSTO UNITARIO**	**COSTO TOTAL**
1	Modulo Huella Dactilar FIM5360	195.00	195.00
1	Tarjeta Arduino Uno	33.00	33.00
1	Módulo Wifi Shield de Arduino	110.00	110.00
1	Baquelita	1.00	1.00
1	Cloruro Férrico	1.00	1.00
1	Max232	2.50	2.50
1	Pulsador	0.25	0.25
1	Interruptor	0.45	0.45
4	Capacitores 10uf	0.15	0.60
1	Zumbador	1.00	1.00
1	Led	0.15	0.15
1	resistencia	0.08	0.08
1	Conector RS232	0.80	0.80
1	Cable USB a RS232	25.00	25.00
1	Caja de montaje	8.00	8.00
1	Switch selector	0.45	0.45
1	Cableado	0.50	0.50
1	Batería	25.00	25.00
1	Conector para batería	0.30	0.30
1	Espadines	0.40	0.40
1	Sello identificativo	0.50	0.50
1	Porta baterías	1.00	1.00
1	Zócalo	0.60	0.60
1	Cable USB para Arduino	5.00	5.00
COSTO TOTAL			**412.58**

Tabla 16 Presupuesto Final
Fuente: Autores

4. DISCUSIÓN

El trabajo de investigación llevado a cabo en esta tesis no ha pretendido resolver el problema del reconocimiento automático de los sistemas basados en la huella dactilar. Como ya se ha dicho, uno de los objetivos importantes, aquí planteados, ha sido el mejoramiento de los sistemas de control de personal o alumnado, se profundizo en el estudio de los sistemas basados en la extracción de minucias, y del trabajo realizado han surgido algunas propuestas que, como otras contribuyen al acercamiento de la solución. Varias líneas de investigación, al día de hoy, suponen todavía un gran esfuerzo por parte de los investigadores en el campo de la biometría, y más concretamente, en el reconocimiento de las huellas dactilares. Todas ellas tienen como factor común el reconocimiento de patrones ante las condiciones adversas que pueden aparecer durante el proceso de adquisición.

El sistema de verificación implementado con la base de datos MY SQL. A la presente fecha, la base de datos lleva almacenado la validación de las huellas dactilares de aproximadamente 100 usuarios, el equipo diseñado tiene cobertura de aproximadamente 50m a la redonda en espacios libres, sin embargo al tener acceso a una red este rango puede extenderse. El sistema opera con una seguridad WPA2 PSK, sin embargo puede mejorarse configurando el router en modo de filtrado de MAC.

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

- Los diferentes tipos de tecnologías existentes hasta determinar las más apropiadas para el desarrollo del proyecto, de lo cual fue elegido la verificación por huella dactilar como método de seguridad y la comunicación wireless como mejor método para la transmisión inalámbrica.

- El equipo de autenticación biométrica basado en huellas digitales y transmisión Inalámbrica con la ayuda de un Sensor Biométrico, una tarjeta wireless para transmisión de datos, que resulto liviano y fácilmente transportable.

- De acuerdo a los resultados obtenidos en la fase de registro, aplicado a un total de 107 estudiantes, se determinó que el sistema tiene un grado de confiabilidad de 97.2% en esta fase de uso, y un 2.8% de error debido al manejo incorrecto del sistema.

- La pruebas de control de asistencia (identificación de usuarios), fue aplicada a un total de 114 estudiantes, en lo que el sistema mostro una confiabilidad de 98.25% y un 1.75% de error, debido a impurezas existentes en las huellas dactilares.

- El desarrollo del presente proyecto ha dado una solución al problema de control de asistencia de alumnado, brindando con esto una alternativa segura, de fácil operación y portabilidad.

5.2. Recomendaciones

- Es necesario realizar una limpieza periódica del dispositivo debido a la humedad, grasa o polvo de cada dedo, la limpieza se debe realizar con el dispositivo apagado y con un paño seco.

- Verificar que el sistema sea compatible con el hardware y software del computador para su correcto desempeño.

- Durante la programación de la interfaz gráfica es indispensable definir cada una de las operaciones que se ejecutara en cada ventana, ya que esto permite tener una mejor visión del objetivo de desarrollo del sistema.

- Es necesario realizar un estudio minucioso sobre la relación entre tablas de la base de datos, ya que de ello depende que la información sea consistente en todo momento, además de la rapidez con la que se acceda a la información.

6. PROPUESTA

6.1. Título de la propuesta

DISEÑO E IMPLEMENTACION DE UN CONTROL DE ASISTENCIA INALAMBRIO POR HUELLA DACTILAR"

6.2. Introducción

Previamente analizamos el problema en el que nos vamos a enfocar en resolver, es así como se escogió el bajo rendimiento de los estudiantes de la Escuela de Ingeniería Electrónica y Telecomunicaciones, su causante son múltiples factores, dentro de los cuales se destaca la inasistencia de los alumnos a sus horas de clase, lo que responde a la vulnerabilidad de los sistemas de control de asistencia.

Hasta el día de hoy las herramientas con las que cuenta un docente para realizar el control de asistencia son: registros de asistencia y hojas firmadas. En el primer caso involucra al docente quien tiene que llevar consigo un registro o leccionario y cada inicio de la clase debe dedicar parte del tiempo al control de la asistencia, este sistema aparentemente sencillo no garantiza del todo la seguridad de los datos ya que una pérdida de este registro involucrará que todos los datos correspondientes a la asistencia de un estudiante sean extraviados, dejando al docente sin esta importante información que respalda las calificaciones obtenidas por el estudiante.

El segundo caso aún más inseguro y vulnerable se refiere a las hojas sueltas que son pasadas por el docente a cada estudiante para que registre su asistencia con datos como nombre, cédula y firma, este método permiten al docente dedicar mayor tiempo a la exposición de su clase, lo que puede parecer una ventaja lo convierte también en la mayor vulnerabilidad, debido a que la asistencia corresponde de manera entera al estudiante quien puede fácilmente registrar a una

persona no asistente ya que no se cuenta con ningún método que garantice la correspondencia de la firma a su propietario.

La solución planteada contempla las ventajas de los sistemas anteriormente diseñados pero reforzando aquellos problemas que los vuelve vulnerables, se propone el diseño de un sistema de control de asistencia de alumnado con el uso de sensores de huella dactilar y comunicación inalámbrica.

6.3. Objetivos

6.3.1. General.-

Diseñar e implementar un control de asistencia inalámbrico por huella dactilar.

6.3.2. Específicos:

- Elaborar un sistema de control de asistencia de alumnado que ofrezca seguridad e inviolabilidad del sistema.

- Elaborar un dispositivo electrónico detector de huellas digitales, que sea inalámbrico, liviano y fácilmente transportable.

- Contribuir con los docentes con un sistema de control de asistencia de alumnado de fácil operación.

6.4. Fundamentación Científico –Técnica

La investigación realizada ha concluido en el uso de las siguientes tecnologías.

6.4.1. Módulo de Huella Dactilar FIM5360

FIM5360 es un módulo de reconocimiento de huella digital autónomo compuesto por un sensor óptico y una placa de procesado.
Mediante la incorporación de una CPU de gran velocidad y un algoritmo de reconocimiento de huella optimizado, el FIM5360 ofrece una alta capacidad de reconocimiento y una gran velocidad para operaciones de identificación 1:N, y para la carga y descarga de datos, proporcionando las condiciones óptimas para su aplicación en sistemas de control de acceso.

El FIM5360 dispone de entradas digitales para registro de huellas, identificación, borrado parcial o completo y reset, ofrece un entorno de desarrollo cómodo y seguro para aplicaciones on-line y off-line.

6.4.2. Placa Arduino

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinares.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida, los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, ATmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque (boot loader) que corre en la placa.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a software del ordenador, las placas se pueden montar a mano o adquirirse. El entorno de desarrollo integrado libre se puede descargar gratuitamente.

Al ser open-hardware, tanto su diseño como su distribución es libre. Es decir, puede utilizarse libremente para el desarrollo de cualquier tipo de proyecto sin haber adquirido ninguna licencia.

6.4.3. Módulo Wifi Shield de Arduino

El módulo Wifi Shield de Arduino, es un módulo poder-consumidor bajo de WIFI que se aplica con tecnología dinámica de la gestión del poder.

El módulo inalámbrico de WIFI Proporciona el enlace de la comunicación del puerto serie de TTL a la comunicación de la radio de IEEE802.11b/g/n. Cualquier dispositivo con los puertos serie de TTL se puede conectar fácilmente con este módulo de WIFI, controlar y manejar remotamente a través de una red inalámbrica. Los diferentes tipos de protocolos de comunicación y de algoritmos de encriptación se integran con el módulo. La arquitectura de Arduino le permite integrar fácilmente este módulo en cualquier proyecto basado Arduino.

6.5. Descripción de la propuesta

La identificación a través de sensores de huella dactilar (sensores biométricos) garantiza la seguridad en el sistema y la comunicación WiFi (comunicación inalámbrica), proporciona el ahorro de tiempo para el docente, tiempo que puede ser utilizado para ampliar los contenidos de su materia, el sistema de control de asistencia consta no solo con la parte de hardware sino también lo hace con una parte de software desarrollado en lenguajes de programación robustos (Visual Basic. Net), este está apoyado en una base de datos desarrollada en MySQL, el sistema en conjunto es muy confiable y de fácil operación aparte que ofrece la seguridad de que en caso de pérdida del equipo, los datos registrados en la base de datos no corran ningún peligro, ya que estos están respaldados en la computadora del docente o en la computadora de secretaria según el uso que se le dé al sistema.

6.6. Diseño Organizacional.

6.7. Monitoreo y Evaluación de la propuesta

La evaluación de la propuesta se la hará por parte del tutor de esta propuesta, ya que ha visto de cerca el desarrollo de la misma y esta la tanto de todas sus características.

Por otro lado y de entrar en uso este sistema, su monitoreo deberá encargarse al docente encargado de la utilización del equipo, quien aparte de ser usuario del equipo se convertirá en administrador del sistema, y será la persona que analice posibles cambios o recomendaciones para la eficiencia del sistema, que pueden ser tratados en futuros estudios.

7. BIBLIOGRAFÍA

- Andrew S. Tanenbaum, 2006, REDES DE COMPUTADORAS, Cuarta edición.
- Behrouz A. Forouzan, TRANSMISIÓN DE REDES DE COMUNICACIONES, Cuarta edición, 2003.
- http://arduino.cc/en/Main/Software
- Protector V2.2 de WiFi para Arduino (viruta 802.11 b/g/n) WizFi210 de la comunicación del puerto serial de la TTL a la radio de IEEE802.11b/g/n, http://es.aliexpress.com/item/WiFi-Shield-V2-2-For-Arduino-802-11-b-g-n-WizFi210-chip-from-TTL-serial/683666090.html
- Wifi Shield para Arudino, Caracteristicas, http://www.dfrobot.com/wiki/index.php/WiFi_Shield_V2.2_For_Arduino_(SKU:TEL0047)
- Wifi Shield V21 SCH_2, http://www.dfrobot.com/image/data/TEL0047/Wifi%20Shield%20V21%20SCH.pdf
- WizFi210-User_Manual_EN_V1.12_2, http://www.dfrobot.com/image/data/TEL0047/WizFi210-User_Manual_EN_V1.12.pdf
- WizFi210-QuickStartGuide_EN_V1.0_2.pdf, http://www.dfrobot.com/image/data/TEL0047/WizFi210-QuickStartGuide_EN_V1.0.pdf
- Software Wizfi ConfigTool, http://wiznet.co.kr/Sub_Modules/en/product/product_detail.asp?Refid=481&page=1&cate1=5&cate2=43&cate3=0&pid=1137&cType=2
- Software Arduino, http://arduino.cc/en/Main/Software
- Arduino uno, http://arduino.cc/en/Main/ArduinoBoardUno
- Topología de redes inalámbricas, http://www.slideshare.net/alandk/topologias-inalambricas-presentation
- Componentes y topologías de una red inalámbrica, http://ieeestandards.galeon.com/aficiones1573328.html

- Topología e infraestructura básica de redes inalámbricas, http://www.itrainonline.org/itrainonline/mmtk/wireless_es/files/04_es_topologia-e-infraestructura_guia_v02.pdf
- Seguridad en una red Wifi, http://www.telmex.com/mx/hogar/internet/seguridad-wifi.html

8. APÉNDICES Y ANEXOS

ANEXO 1