



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES
Y REDES

“IMPLEMENTACIÓN DE UN PROTOTIPO DE UNA RED
INALÁMBRICA DE SENSORES PARA LA IDENTIFICACIÓN DE
PERSONAS Y ACCESO A HISTORIAS CLÍNICAS BASADO EN
TARJETAS DE DESARROLLO”

Trabajo de titulación presentado para optar al grado académico de:

INGENIERA EN ELECTRÓNICA, TELECOMUNICACIONES Y
REDES

AUTORAS: MARLLORY CORINA COBOS MALDONADO
MAYRA ESTEFANÍA ORTIZ JIMÉNEZ

TUTOR: ING. JOSÉ ENRIQUE GUERRA SALAZAR

Riobamba-Ecuador

-2017-

@2017, Marllory Corina Cobos Maldonado, Mayra Estefanía Ortiz Jimenez.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA
EN TELECOMUNICACIONES Y REDES

El Tribunal del Trabajo de Titulación certifica: “IMPLEMENTACIÓN DE UN PROTOTIPO DE UNA RED INALÁMBRICA DE SENSORES PARA LA IDENTIFICACIÓN DE PERSONAS Y ACCESO A HISTORIAS CLÍNICAS BASADO EN TARJETAS DE DESARROLLO”, de responsabilidad de Marllory Corina Cobos Maldonado y Mayra Estefanía Ortiz Jiménez ha sido minuciosamente revisado por los miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

Derechos compartidos

NOMBRE	FIRMA	FECHA
Ing. Washington Luna		
DECANO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	_____	_____
Ing. Franklin Moreno		
DIRECTOR DE ESCUELA DE INGENIERÍA ELECTRÓNICA EN TELECOMUNICACIONES Y REDES	_____	_____
Ing. José Guerra		
DIRECTOR DEL TRABAJO DE TITULACIÓN	_____	_____
Ing. Edwin Altamirano.		
MIEMBRO DEL TRIBUNAL	_____	_____

Nosotras, Marllory Corina Cobos Maldonado y Mayra Estefanía Ortiz Jiménez declaramos ser los autores del presente trabajo de titulación: “IMPLEMENTACIÓN DE UN PROTOTIPO DE UNA RED INALÁMBRICA DE SENSORES PARA LA IDENTIFICACIÓN DE PERSONAS Y ACCESO A HISTORIAS CLÍNICAS BASADO EN TARJETAS DE DESARROLLO”, que fue elaborado en su totalidad por nosotros, bajo la dirección del Ingeniero José Guerra haciéndonos totalmente responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

Marllory Corina Cobos Maldonado

Mayra Estefanía Ortiz Jiménez

DEDICATORIA

Esta tesis se la dedico a mi madre Betty Maldonado por se el pilar mas importante de mi vida, ya que con su paciencia, amor incondicional y con su sacrificio hicieron posible la culminación de esta etapa de mi vida. A mi padre Luis Cobos, ya que con sus enseñanzas y sus palabras han sabido apoyarme y guiarme para alcanzar mis metas. A mis hermanos Vladimir, Nadeshka y Sebastian por estar siempre presentes y alegrar cada día y cada paso que dado.

A mi abuelita Corina por ser mi segunda madre quien siempre supo aconsejarme, escucharme y por brindarme su cariño de abuela y de mamá, para lograr mi superación personal y academica, gracias mami Cori. A mi tío Victor Maldonado que desde el cielo me guía el cual siempre me motivó a seguir adelante y a quien prometí que terminaría mis estudios. Promesa cumplida.

A mis primas Heidi y Belen que son como mis hermanas mayores, las cuales han siempre han sabido estar para mí viéndome crecer y estuvieron listas para brindarme su ayuda en todo momento. A mi madrina, prima y hermana mayor Salomé que es una persona muy importante en mi vida, quien supo brindarme su cariño, sus consejos, y cuidarme en cada momento y a pesar de la distancia siempre estuvo presente en cada uno de mis logros.

A mis amigos ya que con ellos he compartido muchas experiencias a lo largo de mi vida, recuerdos inolvidables los cuales se quedan en el corazón.

Marllory

Esta tesis se la dedico a Dios, por permanecer constantemente en mi vida, brindándome la fortaleza y la sabiduría necesaria para culminar esta etapa. A mis padres, Luis Alfonso Ortiz y Mary Lily Jiménez, quienes día a día han sido el más grande ejemplo de perseverancia, comprensión y constancia, ya que con su amor, esfuerzo y apoyo incondicional me han dado la mejor herencia al permitirme cumplir esta meta tan anhelada. A mis hermanos Orlando, Gladys, Marcelo, y Miryam que han sido un sostén fundamental porque con sus consejos y su constante motivación han sido mi modelo a seguir. A mis hermosos sobrinos Josue, Mateo, Luisito, Yaritza, Mía y Jhordana, quienes han sabido brindarme el cariño más sincero y puro, sacándome una sonrisa en los momentos más difíciles.

Además a mis grandes amigos, Marllo, Gaby, Katy, Jomy, Daniel, Andres, Felipe, Cristian y Jhony quienes durante toda la carrera han sido como una mano derecha, apoyándonos mutuamente y brindándonos una amistad única, inolvidable y trascendental, los llevare siempre en el corazón. Gracias a todos ellos por hacer posible ahora este logro.

Mayra

AGRADECIMIENTO

Agradecemos a Dios por habernos permitido culminar esta etapa importante de nuestras vidas, a nuestros queridos padres y hermanos porque han sido un pilar fundamental y nos han brindado un apoyo incondicional a lo largo de nuestra carrera. A nuestros amigos, quienes sin esperar nada a cambio compartieron con nosotras grandes momentos, supieron estar a nuestro lado e hicieron de esta experiencia una de las mas especiales.

De la misma manera, a la Escuela Superior Politécnica de Chimborazo, así como también a la Escuela de Ingeniería Electrónica en Telecomunicaciones y Redes ya que durante esta etapa se convirtió en nuestro segundo hogar.

Con una expresión de gratitud a nuestros profesores quienes nos incentivaron en muchos sentidos a seguir adelante, guiándonos con sus conocimientos para alcanzar esta meta. Y sobre todo al Ing. José Guerra Salazar e Ing. Edwin Altamirano Santillán quienes supieron guiarnos, y con su apoyo incondicional y su ayuda han hecho posible la culminación con éxito del presente trabajo de titulación.

Marllory Cobos y Mayra Ortiz.

TABLA DE CONTENIDO

PORTADA	I
DERECHO DE AUTOR	II
FIRMA DE RESPONSABILIDADES	III
RESPONSABILIDAD DE AUTORES	IV
DEDICATORIA	V
AGRADECIMIENTO	V
TABLA DE CONTENIDO	VII
ÍNDICE DE TABLAS	X
ÍNDICE DE FIGURAS	XI
INDICE DE ABREVIATURAS	XIII
RESUMEN	XVI
SUMMARY	XVII

INTRODUCCIÓN	1
---------------------------	---

CAPÍTULO I

1. MARCO TEÓRICO	4
1.1. Red de Sensores Inalámbricos	4
1.1.1. Nodos Sensores	5
1.1.2. Gateway	5
1.1.3. Estación base	5
1.1.4. Topología	6
1.1.5. Tecnologías para transmisión de datos	7
1.2. Sistemas de Identificación Automática.....	8
1.2.1. Sistema de códigos de barras	8
1.2.2. Procedimientos biométricos	9

1.2.3.	Tarjetas inteligentes (Smart Cards)	10
1.2.4.	Sistemas RFID	11
1.2.4.1.	Definición de sistemas RFID	11
1.2.4.2.	Tipos de Lectores de Frecuencias	13
1.2.5.	Comparación de los diversos sistemas de identificación	13
1.3.	Tarjetas de Desarrollo	14
1.3.1.	Arduino	15
1.3.2.	Raspberry Pi 2	15
1.3.3.	Galileo	16
1.3.4.	Comparación de las diversas tarjetas de desarrollo.	16

CAPÍTULO II

2.	IMPLEMENTACIÓN; HARDWARE Y SOFTWARE DEL PROTOTIPO	19
2.1.	Concepción general del Sistema	19
2.2.	Requerimientos Hardware para el sistema	20
2.2.1.	Diseño de la Arquitectura del Módulo Lector	20
2.2.2.	Descripción de los dispositivos seleccionados.	20
2.2.3.	Esquema de conexión del Módulo Lector	22
2.2.4.	Alimentación del prototipo	24
2.3.	Requerimientos Software para el sistema	24
2.3.1.	MySQL	24
2.3.2.	Dreamweaver	26
2.3.3.	Android Studio	31
2.3.4.	Arduino	34

CAPITULO III

3.	PRUEBAS Y RESULTADOS DEL SISTEMA	36
----	---	-----------

3.1.	Pruebas del hardware implementado.	36
3.1.1.	Tiempo de respuesta del dispositivo del lector RFID.	36
3.1.2.	Tiempo de respuesta de autenticación y búsqueda de datos.	38
3.1.3.	Distancia de Operación Bluetooth.	38
3.1.4.	Consumo del prototipo	39
3.2.	Pruebas del software del prototipo implementado.	40
3.2.1.	Aplicación Móvil.....	40
3.2.2.	Página Web	42
3.3.	Análisis económico del prototipo.	46
CONCLUSIONES		48
RECOMENDACIONES		50
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-1:	Tipos de tarjetas inteligentes.....	10
Tabla 2-1:	Bandas de frecuencia y usos en etiquetas RFID.....	13
Tabla 3-1:	Comparación de los diversos sistemas de identificación.....	14
Tabla 4-1:	Comparativa de las diferentes tarjetas de desarrollo.....	17
Tabla 5-1:	Modelos de tarjetas de desarrollo Arduino.....	18
Tabla 1-2:	Esquema de conexión Hardware.....	23
Tabla 1-3:	Mediciones tiempo de respuesta RFID en llaveros, pulseras y tarjetas.....	37
Tabla 2-3:	Mediciones de tiempo de respuesta de autenticación y búsqueda.....	38
Tabla 3-3:	Consumo de corriente y voltaje en el módulo lector.....	39
Tabla 3-3:	Presupuesto del prototipo implementado.....	47

ÍNDICE DE FIGURAS

Figura 1-1:	Elementos de una Red de Sensores Inalámbricos	4
Figura 2-1:	Arquitectura de un Nodo Sensor.....	5
Figura 3-1:	Topologías de una Red a) Estrella b) Malla c) Híbrida.....	6
Figura 4-1:	Esquemas de los sistemas más importantes de auto-identificación.....	8
Figura 5-1:	Código de Barras de un producto.	8
Figura 6-1:	Tipos de Sistemas Biométricos a) Iris b) Huella Dactilar c) Facial	9
Figura 7-1:	Componentes RFID.....	11
Figura 8-1:	Etiqueta RFID.	11
Figura 9-1:	Funcionamiento sistema RFID.	12
Figura 10-1:	Placa Raspberry Pi 2.....	15
Figura 11-1:	Placa Intel Galileo	16
Figura 1-2:	Concepción general del sistema.....	19
Figura 2-2:	Diagrama de bloques del prototipo.....	20
Figura 3-2:	Arduino Pro-Mini - Aspecto físico y distribución de entradas y salidas	21
Figura 4-2:	Modulo RFID HF 13.56MHz	21
Figura 5-2:	Modulo Bluetooth HC-06.....	22
Figura 6-2:	Regulador de Voltaje LM7833	22
Figura 7-2:	Esquema de conexión del módulo lector del Prototipo.....	23
Figura 8-2:	Partes de Batería RPI PowerPack V1.0	24
Figura 9-2:	Diagrama de base de datos: Datos personales e Historias Clínicas.....	25
Figura 10-2:	Control de personal en la base de datos.....	26
Figura 11-2:	Parte de la codificación – declaración SQL para una búsqueda.....	26
Figura 12-2:	Diagrama de flujo página Web 1/3.....	28
Figura 13-2:	Diagrama de flujo página Web 2/3.....	29
Figura 14-2:	Diagrama de flujo página Web 3/3.....	30
Figura 15-2:	Distribución gráfica de la página web.	31
Figura 16-2:	Fragmento de código programado – Configuración de botones.....	31
Figura 17-2:	Diagrama de flujo Aplicación Móvil 1/2.....	32
Figura 18-2:	Diagrama de flujo Aplicación Móvil 2/2.....	33
Figura 19-2:	Pantalla resultante de la Aplicación Móvil.	34
Figura 20-2:	Código utilizado en el módulo lector – Arduino Pro-mini.....	34
Figura 21-2:	Diagrama de flujo Programación de arduino.	35
Figura 1-3:	Parte interna y externa del módulo lector implementado.....	36
Figura 2-3:	Rangos de funcionamiento respecto a la distancia en espacio libre.	39

Figura 3-3:	Medición de consumo Batería Raspberry Pi PowerPack V1.0.	40
Figura 4-3:	Icono de la aplicación móvil – Identificate Ecuador.	40
Figura 5-3:	Pantalla de autenticación.	41
Figura 6-3:	Búsqueda datos personales.	41
Figura 7-3:	Lectura de historia clínica realizada en una persona.	42
Figura 8-3:	Prueba de registro de usuarios.	43
Figura 9-3:	Menú de autenticación dentro del sistema.	43
Figura 10-3:	Autenticación del administrador.	43
Figura. 11-3:	Búsqueda realizada por el policía – Datos personales.	44
Figura 12-3:	Formularios de registro de historias clínicas.	45
Figura 13-3:	Listado de Historias Clínicas – Eliminación y modificación.	46
Figura 14-3:	Partes del prototipo implementado.	46

INDICE DE ABREVIATURAS

ESPOCH:	Escuela Superior Politécnica de Chimborazo
RFID:	Radio Frequency Identification, (identificación por radiofrecuencia)
ADN:	Acido desoxirribonucleico
COIP:	Código Orgánico Integral Penal
IVC:	Formulario para la Identificación de Víctimas de Catástrofes
OMS:	Organización mundial de la salud
ANT:	Agencia Nacional de Tránsito.
MSP:	Ministerio de Salud Pública.
RIS:	Redes Inalámbricas de sensores.
WSN:	Wireless Sensor Networks (Red inalámbrica de sensores).
GSM:	Global System for Mobile communications (Sistema global para las comunicaciones móviles).
CDMA:	Code Division Multiple Access (Acceso múltiple por división de código).
RF:	Radio frecuencia.
IEEE:	Institute of Electrical and Electronics Engineers (Instituto de Ingeniería Eléctrica y Electrónica).
WPAN:	Wireless Personal Area Network (Red Inalámbrica de Área Personal)
VPN:	Virtual Private Network (Red privada virtual)
EAN:	European Article Number
UPC:	Universal Product Code (Codigo Universal de producto).
FRT:	Face recognition techniques (técnicas de reconocimiento facial)
EEPROM:	Electrically Erasable Programmable Read-Only Memory (ROM Programable y borrrable eléctricamente)
RAM:	Random Access Memory (Memoria de acceso aleatorio)
ROM:	Read-only memory (Memoria de solo lectura)

LF:	Low Frequency (Baja Frecuencia).
HF:	High Frequency (Alta Frecuencia).
UHF:	Ultra High Frequency (Ultra Alta Frecuencia).
ATMEL:	Tecnología avanzada para la memoria y la lógica.
PCB:	Polychlorinated biphenyls (Bifenilos policlorados).
HDMI:	High-Definition Multimedia Interface (Interfaz multimedia de alta definición).
LCD:	Liquid Crystal Display (Pantalla de cristal líquido).
SD:	Tarjeta micro (Secure Digital).
USB:	Universal Serial Bus (Bus Universal en Serie).
CSI:	Camera Serial Interface (Interfaz Serie para Cámaras).
DSI:	Display Serial Interface (Interfaz Serial de Visualización).
JTAG:	Joint Test Action Group
WIFI:	Wireless Fidelity (Fidelidad inalámbrica).
SPI:	Serial Peripheral Interface
GND:	Ground (tierra)
VCC:	Voltaje de corriente continua. TXD: Transmit Data (Transmisor)
RXD:	Receive Data (Receptor)
SDA:	Serial Data Input/output (Interfaz serial para la entrada o salida de datos).
SCK:	Serial Clock
MOSI:	Master Output Slave Input, or Master Out Slave In.
MISO:	Master Input Slave Output, or Master in Slave Out.
IRQ:	Interrupt Request (Requerimiento de Interrupción).
RST:	Reset (Reiniciar)
LED:	Light-emitting diode (diodo emisor de luz).
ISV:	Proveedores independientes de <i>software</i>
OEM:	Fabricantes de equipos originales

PHP:	Hypertext Preprocessor
SQL:	Structured Query Language (Lenguaje de Consulta Estructurada).
HTML:	HyperText Markup Language (Lenguaje de marcas de hipertexto).
GIF:	Graphics Interchange Format (formato de compresión de imagen).
CSS:	Cascading Style Sheets (Hojas de Estilo en Cascada).
JPEG:	Joint Photographic Experts Group (Grupo Conjunto de Expertos en Fotografía).
XML:	Extensible markup language (Lenguaje de Marcado Extensible).
XSL:	Extensible Stylesheet Language (Lenguaje extensible de hoja de estilo).
IDE:	Entorno de desarrollo integrado
URL:	Uniform Resource Locator (Localizador Uniforme de Recursos).
IP:	Internet Protocol (Protocolo de Internet).
FTP:	File Transfer Protocol (Protocolo de Transferencia de Ficheros).

RESUMEN

La presente investigación realizada en la ESPOCH, consiste en estructurar un prototipo de una red inalámbrica de sensores para la identificación de personas y acceso a historias clínicas, que se administran desde un gestor de base de datos en la web, basándose en tarjetas de desarrollo Arduino y RFID como sistema de identificación, la información se registra en una plataforma web y es accesada por medio de una aplicación móvil desarrollada en Android Studio. Para la selección de los dispositivos se utilizó una metodología comparativa en base a la funcionalidad, precio, tamaño, distancia de funcionamiento. La transmisión de datos se realiza por medio de tecnologías Wifi, Radio Frecuencia y Bluetooth, que en conjunto permiten la lectura y el acceso a la información, por medio de interfaces gráficas amigables. El funcionamiento del sistema se basa en la lectura de un código único que es asignado a una persona el mismo que constituye un nodo llave de la red y es comparado con la información ubicada en la base de datos con un módulo lector, para extraer información relacionada a datos personales, datos patológicos, antecedentes familiares y hábitos tóxicos, información necesaria para personal policial y paramédicos. De las pruebas realizadas se obtuvo que la conexión Bluetooth en espacios libres se realiza sin pérdidas hasta 26 metros, para la comunicación WIFI se registró 1.4 segundos como tiempo de respuesta del sistema de identificación, y 3.5 segundos en caso de utilizar datos móviles. Se concluye que la red implementada es una herramienta de ayuda rápida, de fácil manejo y útil para la obtención de datos de personas involucradas en un evento que requiera identificación. Se recomienda que para futuras implementaciones se realice la selección de elementos electrónicos que permitan la miniaturización del dispositivo identificador y del módulo lector para mejorar su manipulación y uso.

Palabras Clave: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INGENIERÍA EN TELECOMUNICACIONES>, <INGENIERÍA ELECTRÓNICA>, <RED DE SENSORES INALÁMBRICOS>, <IDENTIFICACIÓN DE PERSONAS>, <TRANSMISIÓN DE DATOS> <SISTEMAS DE IDENTIFICACIÓN [RFID]>, <TARJETAS DE DESARROLLO [ARDUINO]>

SUMMARY

The present research carried out in ESPOCH, consists of structuring a prototype of a wireless network of sensors for the identification of people and access to clinical histories, which are administered from a database manager on the web, based on development cards Arduino and RFID as an identification system, the information is registered on a web platform and accessed through a mobile application developed in Android Studio. For the selection of the devices a comparative methodology was used based on the functionality, price, size and operating distance. The data transmission is carried out by means of Wi-Fi, Radio Frequency and Bluetooth technologies, which together allow the reading and access to information, through friendly graphical interfaces. The operation of the system is based on the reading of a unique code that is assigned to a person the same that constitutes a key node of the network and is compared with the information located in the database with a reader node, to extract information related to personal data, pathological data, family history and toxic habits, information necessary for police personnel and paramedics. From the tests carried out, it was obtained that the Bluetooth connection in free spaces is performed without losses up to 26 meters, for WIFI communication, 1.4 seconds was recorded as the response time of the identification system, and 3.5 seconds when using mobile data. It is concluded that the implemented network is a quick, easy to use and useful tool for obtaining data from people involved in an event that requires identification. It is recommended for future implementations the selection of electronic elements that allow the miniature making of the identification device and the reader module to be improved to be manipulated and used.

Key words: < TECHNOLOGY AND SCIENCE OF ENGINEERING>, <ENGINEERING IN THELECOMMUNICATIONS>, <ELECTRONIC ENGINEERING>, <WIRELESS SENSORS NETWORK>, <IDENTIFICATION OF PEOPLE>, <DATA TRANSMISSION>, <IDENTIFICATION SYSTEMS [RFID] >, <DEVELOPMENT CARDS [ARDUINO]>

INTRODUCCIÓN

La necesidad de agilizar el acceso a las historias clínicas e identificación de víctimas involucradas en accidentes de tránsito y catástrofes naturales es evidente. Según la Organización Mundial de la Salud cada año a nivel mundial hay 1,25 millones de personas fallecidas y de 20 a 50 millones de personas que sufren traumatismos no mortales (Organización Mundial De La Salud, 2016, <http://www.who.int/mediacentre/factsheets/fs358/es/>). En el Ecuador según la Agencia Nacional de Tránsito en el 2016, 644 personas fallecieron en accidentes de tránsito y 6929 resultaron heridas, de las cuales en la provincia de Chimborazo se tiene 19 y 73 personas, respectivamente (Agencia Nacional de Tránsito, 2016, <http://www.ant.gob.ec/index.php/descargable/file/3663siniestros mayo2016>).

Por otra parte, en el terremoto ocasionado el 16 de Abril del 2016, de acuerdo al Instituto de Seguridad Social de la Policía Nacional y de las Fuerzas Armadas hay 663 personas fallecidas, 9 desaparecidas, 6274 heridas y 113 que han sido rescatadas con vida (Secretaría de Gestión de Riesgos, 2016, <http://www.gestionderiesgos.gob.ec/informesdesituacionactualterremotomagnitud78/>). Igualmente la identificación inadecuada de los pacientes es una causa importante de los problemas y complicaciones asociadas a errores en la asistencia al ingresar un paciente víctimas de catástrofes y accidentes de tránsito; la mala identificación conlleva riesgos importantes a la hora de la prescripción dispensación y administración de fármacos. Según la Organización Mundial de la Salud dos de cada mil muertes en hospitales son productos de errores de medicación al momento del ingreso. (Roldán, 2001, pp. 15-17)

Dentro del ámbito de actuación judicial y policial es donde más evidente resulta la necesidad de utilizar métodos de identificación personal fiables seguros y rápidos. Cuando se trata de cadáveres, a las necesidades legales de establecer esa identidad por cuestiones de herencias y sucesiones, se suman, además, razones humanitarias y motivos religiosos que inducen a las personas a honrar a sus familiares fallecidos (Carbajo, 2011, pp. 20); cuando se trata de personas heridas la identificación de datos personales e historia clínica de las mismas serán trascendentales para precautelar su vida.

La investigación esta en concordancia con el Plan Nacional del Buen Vivir, en el objetivo 10 que indica “*Impulsar la transformacion de la matriz productiva*”, cita la necesidad de diversificar el

conocimiento y la innovación como elemento de cambio, desarrollo científico y tecnológico para la transformación de la matriz productiva, sustentando así el presente trabajo de titulación.

En una publicación del Diario “EL TELEGRAFO” en el año 2013, cita que Ecuador busca pulir protocolos para la gestión de cadáveres en catástrofes, en ese año se llevó a cabo la primera reunión multisectorial sobre gestión de cadáveres en catástrofes, los expertos discutieron qué normas se debe cumplir para la ubicación, identificación y entierro de los cuerpos fallecidos en cualquier tipo de desgracias.

Llegando a la conclusión que el Ecuador es un país vulnerable a erupciones volcánicas, tsunamis, terremotos y accidentes, por ello es importante que las instituciones construyan protocolos de atención para manejo de restos, restablecimiento de contacto con familiares de las víctimas, recursos forenses, entre otros temas. “Lo ideal es la recolección de los fallecidos, luego levantar sus datos y transportarlos a un lugar adecuado para enterrarlos, con conocimientos de los familiares” (Diario El Telégrafo, 2013, <http://www.eltelegrafo.com.ec/noticias/judicial/13>).

Asimismo, en el caso de heridos en accidentes de tránsito o catástrofes el Ministerio de Salud Pública se compromete a velar por la salud de los ecuatorianos, sobre todo en casos fortuitos, basándose en la Constitución en su artículo 359 el cual establece: “El sistema nacional de salud comprenderá las instituciones, programas, políticas, recursos, acciones y actores en salud; abarcará todas las dimensiones del derecho a la salud; garantizará la promoción, prevención, recuperación y rehabilitación en todos los niveles; y la propiciará la participación ciudadana el control social” (Ministerio de Salud Pública, 2016, <http://www.salud.gob.ec/ministeriodesaludpublicagarantizaatencionycoberturadegastosaheridostrasaccidenteenpapallacta/>). Mostrándonos así la importancia del conocimiento de la historia clínica de una víctima.

Considerando la necesidad de que existan dispositivos que permitan identificar personas y acceder a historias clínicas, se debe implementar un sistema electrónico que suministre información de datos personales e historial médico. Esta red de identificación puede hacer uso de sensores inteligentes con comunicación inalámbrica, reduciendo el tiempo de identificación y el fácil acceso a la información requerida de manera oportuna.

Por tal motivo, el presente trabajo de titulación busca estructurar un sistema de identificación de personas, que trabaje como nodos inalámbricos inteligentes en una red, de bajo costo; basándose en elementos tecnológicos adecuados a nuestro medio, que permitan la captura e identificación automática de datos con dispositivos móviles. La investigación propone que una persona con un código constituya un nodo llave de la red, con el cual se podrá acceder a sus datos personales e historia clínica, en caso de requerir una consulta médica o ser víctima de un evento que requiera identificación.

El trabajo con Sensores inalámbricos requirió de un control por medio de un dispositivo para determinar la forma de funcionamiento y las acciones a realizar, para esto se recurrió al uso de tarjetas de desarrollo debido a su alto grado de compatibilidad, bajo costo y disponibilidad en el mercado. La integración de estas tecnologías hizo viable desarrollar un sistema de identificación de personas basado en una red inalámbrica con datos actualizados.

Por lo tanto, la presente investigación tiene como objetivo el implementar un prototipo de una red inalámbrica de sensores para la identificación de personas y acceso a historias clínicas basado en tarjetas de desarrollo. Como objetivos específicos se plantea para la investigación:

- Realizar un estudio y selección de la tecnología y dispositivo a emplear para la identificación de personas.
- Analizar el tipo de comunicación y tarjetas de desarrollo necesarias para la implementación que se ajusten a los requerimientos de la red.
- Implementar el *software* necesario para los nodos de la red de identificación y los dispositivos móviles.
- Realizar las pruebas y afinamiento de la red y equipos que conforman el sistema.

CAPÍTULO I

1. MARCO TEÓRICO

En este capítulo se investigan las características de las redes de sensores inalámbricos, su topología, las tecnologías existentes para comunicación entre nodos. Se especifica también las características y el respectivo análisis de los sistemas de identificación más conocidos y de las tarjetas de desarrollo utilizadas para el envío y recepción de datos.

1.1. Red de Sensores Inalámbricos

Conocidas como WSN (Wireless Sensor Networks, por sus siglas en inglés), son pequeños dispositivos autónomos de bajo costo y con gran tiempo de duración, llamados nodos de sensores, capaces de procesar, recopilar y distribuir información específica de forma inalámbrica en un área determinada. Constan de un microcontrolador, una fuente de energía, un radio tranceptor y un elemento sensor (Flores, 2012, pp. 22). Son utilizadas en múltiples aplicaciones, entre ellas la supervisión remota, inclusive haciendo que sistemas inalámbricos sean parte de sistemas cableados (National Instruments, 2009, <http://www.ni.com/white-paper/7142/es/>).

Una red de sensores inalámbricos está formada una estación base, un Gateway y nodos sensores (Flores, 2012 p. 23), como se aprecia en la *Figura 1-1*.

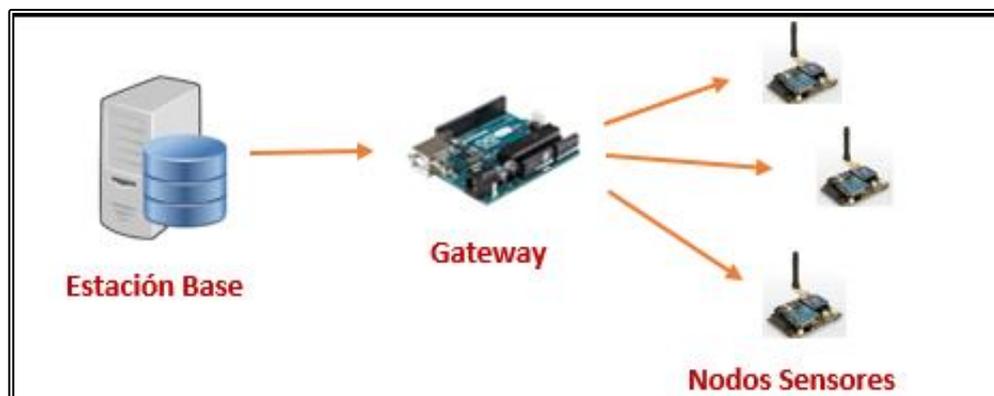


Figura 1-1: Elementos de una Red de Sensores Inalámbricos

Fuente: COBOS, Marllory & ORTIZ, Mayra, 2017.

1.1.1. *Nodos Sensores*

Conocidos también como Motas, representan la unión que existe entre los distintos dispositivos que permiten el proceso de obtención e intercambio de información con otros nodos y elementos de la red (Yépez y Campaña, 2016, pp. 30), sus componentes se muestran en la *Figura 2-1*. Se observa un microprocesador que se encarga de la administración de tareas, una memoria que almacena datos temporalmente, un receptor/ transmisor de radio con su antena, la batería que proporciona la energía a los componentes y uno o varios sensores que dependen de la aplicación (velocidad, temperatura, sonido, etc) pueden ser fijos o móviles (Ramírez, 2012, pp. 9).

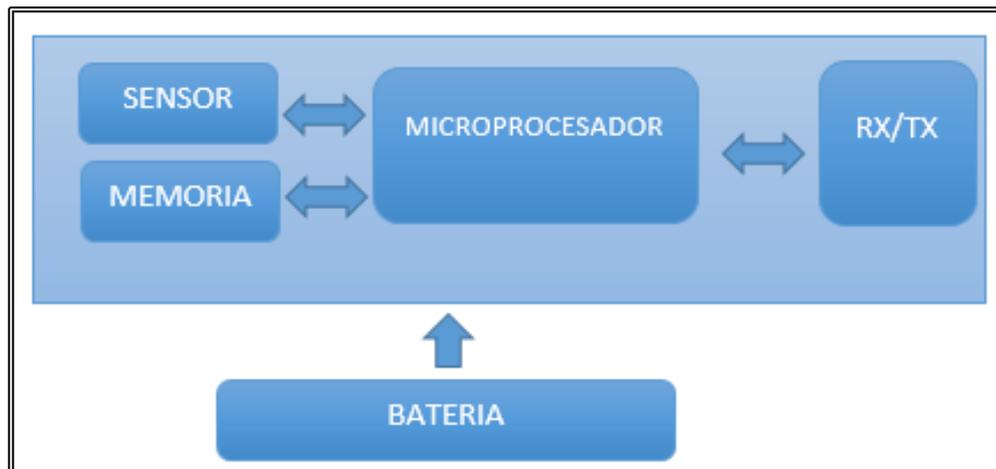


Figura 2-1: Arquitectura de un Nodo Sensor

Fuente: (Ramírez, 2012 pp. 9) **Modificado:** COBOS, Marlory & ORTIZ, Mayra 2017

1.1.2. *Gateway*

Conocido también como Puerta de Enlace, es también un nodo y no contiene sensores (Maroto, 2010, pp. 35). Permite la interconexión entre la red de sensores y la de datos es decir que actúa como un puente entre diferentes redes.

1.1.3. *Estación base*

Es un ordenador o servidor en el cual se recolectan y almacenan los datos, los usuarios pueden accederlo remotamente para observar y analizar datos, o en función a los requerimientos de la aplicación de la red (Maroto, 2010, pp. 36).

1.1.4. Topología

Es la configuración de los componentes *hardware* de la red y como los datos son transmitidos en ella. Dependen de la aplicación de la red. Las topologías de redes de sensores inalámbricos más utilizadas son: topología en estrella, en malla e híbrida (Maroto, 2010, pp. 36), mostradas en la *Figura 3-1*.

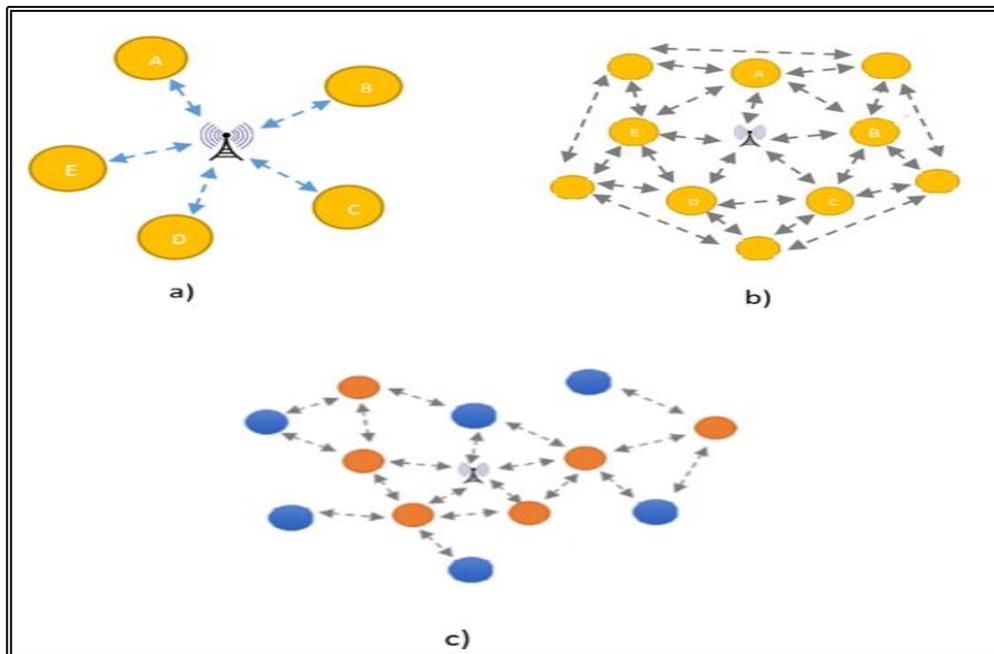


Figura 3-1: Topologías de una Red a) Estrella b) Malla c) Híbrida.

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

Estrella.- Se caracteriza en que todos los nodos sensores están conectados directamente con el Gateway, es decir que la información enviada da un solo salto hacia su destino. Los nodos sensores son iguales, requieren de un Gateway que se encarga de recibir los datos y transmitirlos al exterior (Maroto, 2010, pp. 36). Presenta un menor gasto de energía, pero está limitada por la distancia de transmisión entre cada nodo, que usualmente es de 30 a 100 metros. Una de sus grandes desventajas es que si uno de los nodos sensores se avería la información se pierde al no tener un camino alternativo (Maroto, 2010, pp. 37). En la *Figura 3-1a* podemos apreciar su estructura.

Malla.- Es un sistema multisalto, cada nodo puede enviar y recibir información de otro nodo y del Gateway. Su gran ventaja es que es altamente tolerante a fallos. El tiempo de transmisión varía dependiendo del número de nodos y la distancia entre ellos (Maroto, 2010, pp. 37). Su estructura es en forma de rejilla como se observa en la *Figura 3-1b*.

Híbrida estrella-malla.- Combina las ventajas de las anteriores, como el bajo consumo de energía de la topología estrella, y la posibilidad de cubrir una gran extensión y tolerar fallos de la topología en malla. La red se amplía por medio de routers en malla creandose una red en estrella alrededor

de ellos. Los nodos finales se conectan a los routers más cercanos ahorrando energía (Maroto, 2010, pp. 37). En la *Figura 3-1c* se aprecia la estructura de esta topología.

1.1.5. Tecnologías para transmisión de datos

En una red de sensores inalámbricos, los nodos se comunican por ondas electromagnéticas, ya que trabajan sin utilizar medios guiados. Las tecnologías para transmitir datos se ordenan de acuerdo a la distancia de cobertura. Primero van las comunicaciones satelitales, luego la telefonía celular, en tercer lugar están Wimax y Wifi basadas en protocolos de internet ó Zigbee que interconecta sensores. En un cuarto lugar se encuentran los protocolos que enlazan dispositivos en una red de área personal (PAN) como Bluetooth. Y finalmente las que utilizan una antena transmisora como la tecnología RFID, que es para identificación por radio frecuencia (Fernández et.al, 2009, pp. 28). Los medios de transmisión inalámbrica más utilizados actualmente son los de Radio Frecuencia, Zigbee, Bluetooth y Wi-fi.

Radio Frecuencia (RF).- Tienen una gran variedad de aplicaciones debido a que pueden atravesar obstáculos y pueden ser de corto y medio alcance. Su rango de trabajo depende de la frecuencia, la potencia de salida, la sensibilidad de recepción, las características de la antena y el entorno en el que se desenvuelve el sistema (Mayné, 2009, pp. 10). Utilizan frecuencias que van entre 433 MHz a 2,4 GHz, y bajo consumo de energía.

Bluetooth.- Es un sistema de comunicación inalámbrico de alcance limitado, es robusta y bajo consumo de potencia. Consiste en un receptor y un emisor de RF que funcionan en una banda hasta 2.4 GHz. Al transmitir la información a un corto alcance la comunicación se hace más segura y menos susceptible a interferencias (Bellido et al, 2004, <http://www.tecnicaindustrial.es/tifrontal>).

Wi-Fi.- Se basa en el estándar 802.11 IEEE, trabaja en una banda de 2,4 GHz. Es de bajo costo, velocidad aceptable y es compatible con la mayoría de dispositivos actuales (Yaagoubi, 2012, pp. 10).

Zigbee.- Se basa en el estándar IEEE 802.15.4 para WPAN de bajo poder y bajo costo, busca ser más simple y menos costosa que otras VPN. Es dirigida en especial para las aplicaciones de RF que demandan una velocidad de transferencia baja, bajo consumo y seguridad (Jaramillo et al, 2011, pp. 15-16).

1.2. Sistemas de Identificación Automática

Se encuentran en constante desarrollo, perfeccionamiento y aumento de sus aplicaciones, tomando como ejemplos claros la utilización para el acceso a cuentas bancarias, áreas restringidas, computadoras, información personal, entre otros (Alvarado, 2008, pp. 1). Al ser automáticos, se cometen menos errores incrementando la confiabilidad y la eficiencia ahorrando tiempo. Su funcionamiento requiere de dos componentes fundamentales: un elemento codificado y un elemento con capacidad de reconocerlo. En la *Figura 4-1* se observa los esquemas de los sistemas más importantes de auto-identificación.

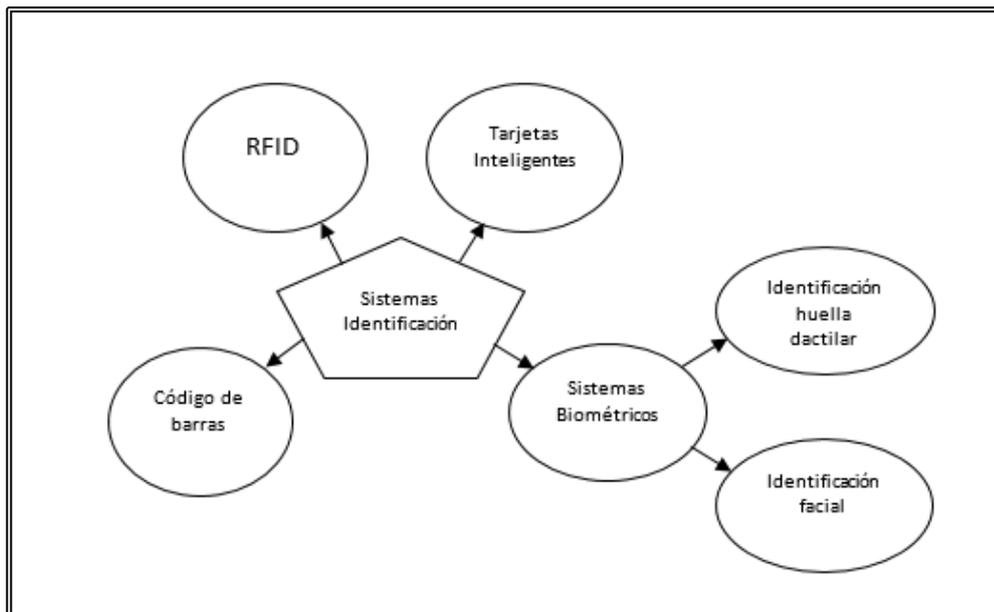


Figura 4-1: Esquemas de los sistemas más importantes de auto-identificación
Fuente: (Alvarado, 2008 pp. 2) **Modificado:** COBOS, Marllory & ORTIZ, Mayra 2017

1.2.1. Sistema de códigos de barras

Es el sistema más utilizado. Consiste en un código binario comprendido por una serie de barras y espacios configurados paralelamente como se muestra en la *Figura 5-1*. La secuencia que emplea tiene forma numérica o alfanumérica, y es leída por un scanner óptico láser, que se basa en la diferente reflexión que sufre la luz en las barras negras y blancas (Almonacid, 2007, pp. 2).



Figura 5-1: Código de Barras de un producto.
Fuente: (Código de Barras, 2017, <https://codigos-de-barras.es/imagenes-de-muestra/>)

El más usado es el código EAN (por sus siglas en inglés European Article Number), se diseñó especialmente para el sector alimenticio. Evolucionó del código UPC (por sus siglas en inglés Universal Product Code) estadounidense. Actualmente estos códigos son compatibles (Almonacid, 2007, pp. 3).

Fue inventado hace más de 25 años, utilizado especialmente en la identificación de productos. Como características necesita visibilidad para funcionar, identifica un tipo de producto, no una unidad, los equipos son de fácil conexión y sus códigos de bajo costo pero se pueden dañar y romper fácilmente (Ciudad, y Samá, 2005, pp. 10).

1.2.2. *Procedimientos biométricos*

Identifican personas se basan en la comparación de las características físicas individuales y que no admite equivocación. Los métodos más utilizados son: huella dactilar, reconocimiento por iris, y facial como se aprecia en la *Figura 6-1*.

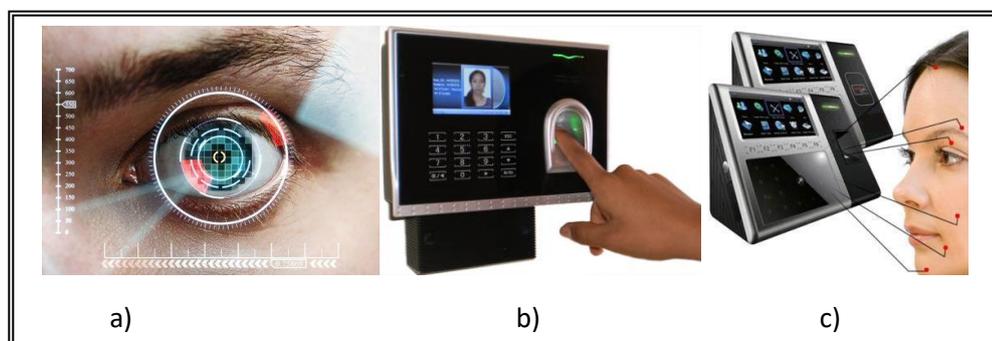


Figura 6-1: Tipos de Sistemas Biométricos a) Iris b) Huella Dactilar c) Facial
Fuente: Cobos, Marllory & Ortiz, Mayra 2017

Sistemas de identificación por huella dactilar

Destacan la presencia de un conjunto de crestas o partes donde la piel se eleva sobre las partes más bajas o valles existentes entre las crestas en los dedos de las personas (Espinosa, 2001, pp. 59). Entre sus características tenemos que son sistemas no invasivos, de elevada aceptación popular, de captura fácil y bajo costo, requiere una elevada resolución de la imagen digital, y la gente que trabaja con sus manos pueden presentar deformaciones de la huella.

Sistemas de identificación facial.

Están incluidos dentro de las técnicas FRT (por sus siglas en inglés Face Recognition Thecniques). Se pueden clasificar en dos categorías según el tipo de aproximación holística o

analítica (Espinosa, 2001, pp. 58-59). Entre sus características tenemos que es un sistema no invasivo, permite la identificación de personas en movimiento, sistema con posibilidad de camuflaje, es susceptible a problemas de iluminación y es vulnerable al reconocimiento de sujetos que se han sometido a operaciones de cirugía plástica.

1.2.3. *Tarjetas inteligentes (Smart Cards)*

Es una tarjeta plástica con un tamaño definido (Estándar ISO 7816), con un microchip capaz de hacer diferentes cálculos, guardar información y manejar programas, protegidos a través de mecanismos avanzados de seguridad. Su clasificación es de acuerdo al tipo de circuito (chip) implantado; las dos categorías principales son: tarjetas de memoria y tarjetas de microprocesador. Las tarjetas de memoria no realizan procesamiento de información y su tecnología de almacenamiento es EEPROM (Memoria Borrable y Programable) o Flash, no necesitan de energía eléctrica para resguardar los datos. Cuentan con capacidades promedio de 24 KB (Romo, 2015, pp. 4).

Las tarjetas de microprocesador tienen capacidad para alterar datos, además, cada sección puede contener una función específica. Tiene un microprocesador de 8 bits con una velocidad máxima de 5 Mhz, organizan la información mediante un Sistema Operativo de Tarjeta (COS), el cual controla el acceso a secciones específicas de la tarjeta a través de un código de instrucciones reducido y encriptado (Romo, 2015, pp. 5). En la *Tabla 1-1* se aprecia los tipos de tarjetas inteligentes.

Tabla 1-1: Tipos de tarjetas inteligentes.

TIPOS DE TARJETAS INTELIGENTES	SEGÚN SUS CAPACIDADES
	<ul style="list-style-type: none"> Memoria Microprocesadas Criptográficas
SEGÚN LA ESTRUCTURA DE SU SISTEMA OPERATIVO	<ul style="list-style-type: none"> Tarjetas de memoria Basadas en sistemas de ficheros Java Cards
SEGÚN LA INTERFAZ	<ul style="list-style-type: none"> ID 000 ID 00 ID 1
SEGÚN EL FORMATO	<ul style="list-style-type: none"> Tarjeta inteligente de contacto Tarjetas inteligentes sin contacto Tarjetas híbridas y duales

Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

1.2.4. Sistemas RFID

1.2.4.1. Definición de sistemas RFID

RFID (por sus siglas en inglés Radio Frequency Identification) es un método de almacenamiento y recuperación de datos remoto que usa dispositivos denominados etiquetas RFID (Fundación Telefónica, 2008, pp. 135). Basa su funcionamiento en los datos almacenados del objeto al que se encuentra incluido; es inducida a generar una señal de radiofrecuencia con dicha información. Está señal es captada por un lector RFID, el cual se encarga de leer la información y pasarla en formato digital a la aplicación específica (Jurado y Salazar, 2013, pp. 45). Los componentes de un sistema RFID están representados en la *Figura 7-1*:

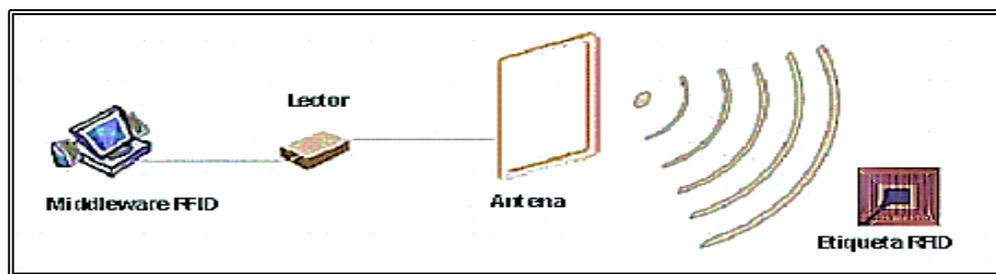


Figura 7-1: Componentes RFID

Fuente: (Monzon, 2014, <http://logistica-sencilla.blogspot.com/2014/09/tecnologiaque-es-la-rfid.html>)

Etiqueta RFID.- Una etiqueta RFID consta de tres elementos una antena, un circuito integrado y un almacenador de energía, mostrados en la *Figura 8-1*.

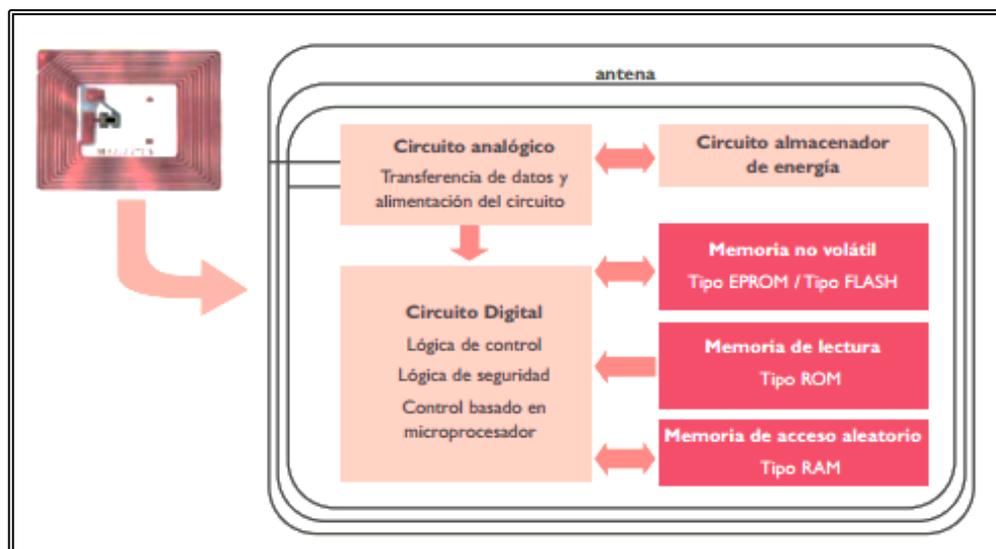


Figura 8-1: Etiqueta RFID.

Fuente: (Fernández et al, 2006 pp. 48)

La antena permite realizar la comunicación entre la etiqueta y el lector. Su tamaño limita la distancia máxima a la que puede realizarse la lectura. El circuito integrado es un circuito mixto analógico-digital. La parte analógica se encarga de controlar la alimentación, la comunicación por radiofrecuencia y la parte digital gestiona la información almacenada en las etiquetas que pueden ser activas y pasivas (Fernández et al, 2006, pp. 46).

En las etiquetas activas se usa una batería para alimentar el circuito, la principal ventaja es un mayor rango de lectura al poder emitirse la información con mayor potencia. En las etiquetas pasivas, el elemento almacenador de energía es un condensador, el cual se carga con la energía emitida por el lector y luego utiliza dicha energía para responder. Por ello es corta la distancia entre el lector y la etiqueta. Debido a esto son más utilizadas, aplicándose en campos tan diversos como la identificación de animales, llaves de contacto de automóviles, identificación de productos en cadenas de montaje, control de accesos, cronometraje de carreras, etc. (Fernández et al, 2006 pp. 47-48).

Lector RFID.- La estructura del equipo de lectura es similar a la de las etiquetas: es necesaria una antena para comunicarse con la etiqueta y un circuito para gestionar la comunicación. Este circuito dispone de un interfaz estándar, como por ejemplo RS-232 o CompactFlash, para conectarse a un ordenador. Además existen en el mercado equipos lectores con la antena integrada y equipos que admiten antenas externas, que pueden seleccionarse en función de la aplicación (Fernández et al, 2006, pp. 49).

Software.- Es el sistema de proceso de datos. Las etiquetas sólo de lectura devuelven un código único grabado al fabricar el chip. Las etiquetas de lectura/escritura, describen una mayor capacidad de almacenamiento que van de 32, 256 o 2048 bits, en la *Figura 9-1* se resume en forma gráfica el Funcionamiento del sistema RFID (Fernández et al, 2006, pp. 49).

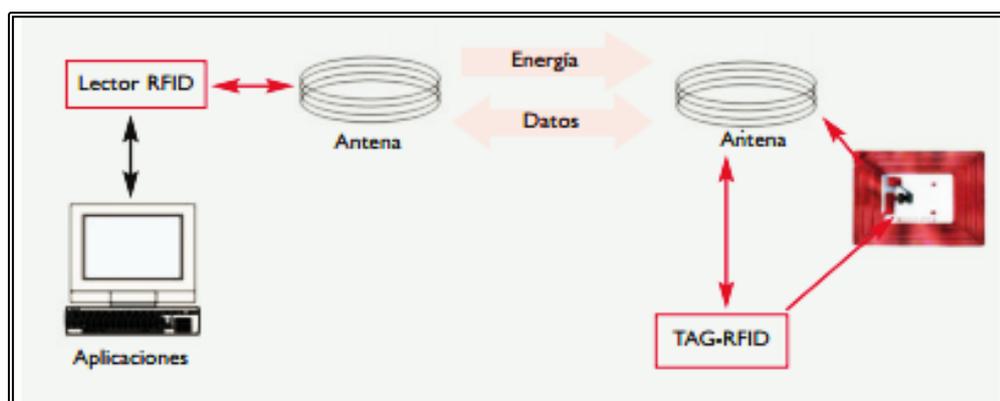


Figura 9-1: Funcionamiento sistema RFID.

Fuente: (Fernández et al, 2006 pp. 49)

1.2.4.2. Tipos de Lectores de Frecuencias

Las etiquetas RFID son utilizadas en base a su frecuencia de operación, ya que de esto depende la distancia de lectura entre etiqueta y lector. Trabajan con tres bandas principales de frecuencias, que se resumen en la *Tabla 2-1*.

Tabla 2-1: Bandas de frecuencia y usos en etiquetas RFID.

Banda de Frecuencia	Características	Aplicaciones Típicas
Baja 100-500 KHz	Lectura para corta y media distancia Sistemas con tags económicos Velocidad de lectura baja	Control de acceso Identificación de animales Control de existencias Inmovilizadores de automóviles
Intermedia 10-15 MHz	Lectura para corta y media distancia Potencialmente barato Velocidad de lectura media	Control de acceso Tarjetas Inteligentes
Alta 850-950 MHz 2.4-5.8 GHz	Lectura para corta y media distancia Velocidad de lectura alta Línea de vista requerida Tecnología cara	Supervisión en sistemas ferroviarios y automotriz Acceso y control de peaje.

Fuente: (Fernández et al, 2006 pág. 50) Modificado: COBOS, Marllory & ORTIZ, Mayra 2017

1.2.5. Comparación de los diversos sistemas de identificación

Una vez analizada en forma individual cada uno de los sistemas de identificación, se procede a realizar una comparativa entre ellos con sus principales características presentándose los resultados en la *Tabla 3-1*.

Tabla 3-1: Comparación de los diversos sistemas de identificación.

	Código de Barras	Tarjetas Inteligentes	Sistemas Biométricos	RFID Pasivo	RFID Activo
Modificación de la información	No Modificable	Modificable	No Modificable	Modificable	Modificable
Seguridad de los Datos	Mínima	Alta	Alta	Variable (baja a alta)	Alta
Capacidad de Almacenamiento de Datos	Lineales(8-30 caracteres) 2D hasta 7200 caracteres	Hasta 8MB	No aplica	Hasta 64KB	Hasta 8MB
Precio	Bajo	Medio-Bajo	Alto	Medio (menos de \$1 por tag)	Muy Alto (\$10 a \$100 por tag)
Estándares	Estables	Estables	No estándar	Evolucionando hacia el estándar	Propietario y en evolución hacia el estándar
Ciclo de Vida	Corto	Largo	Indefinido	Indefinido	Depende de la batería (3 a 5 años)
Distancia de Lectura	Línea de vista (hasta 1.5m)	Requiere contacto	Depende del biométrico	No Requiere línea de vista ni contacto Hasta 1m	No Requiere línea de vista ni contacto Hasta 10m
Interferencia Potencial	Cualquier modificación en las barras y objetos entre el código y el lector	Bloqueo del contacto	Puede ser bloqueo del contacto, o bloqueo de línea de vista e inclusive el ruido	Ambientes o campos que afecten la transmisión de radio frecuencia.	La interferencia es muy limitada, debido a la potencia de transmisión

Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

Basándose en los datos de la *tabla 1-1* se decidió utilizar el sistema RFID pasivo en el prototipo realizado, ya que la ventaja obvia de este tipo de etiquetas debido al ahorro de espacio, la duración prácticamente ilimitada de la etiqueta, su menor costo y tomando en cuenta que es una tecnología innovadora. A diferencia de los sistemas mencionados no es necesario que exista línea de visión directa, debido a que la señal de RF es capaz de viajar a través de la mayoría de los materiales. La distancia de lectura depende de la frecuencia del sensor que se utilice.

1.3. Tarjetas de Desarrollo

Son placas electrónicas que dan soluciones y facilitan la creación de aplicaciones muy útiles para la sociedad, brindan un alto nivel de compatibilidad con los dispositivos y sensores. Con el avance

de la tecnología se han ido desarrollando distintos tipos de tarjetas de desarrollo, en la actualidad las más utilizadas son Arduino, Libelium-Waspote, Raspberry Pi, Galileo.

1.3.1. *Arduino*

Son dispositivos que facilitan la conexión entre el mundo analógico con el digital, permiten controlar todo tipo de sensores. Simplifica el proceso de trabajar con microcontroladores y posee una plataforma de *hardware* de código abierto (Manzano y Tapia, 2013, pp. 25).

Una de las características que diferencia Arduino de otras tarjetas y microcontroladores, es su entorno de desarrollo y lenguaje de programación, las placas en las que se ejecutan asegura su compatibilidad y sencillez (Doutel, 2015, <https://www.xataka.com/especiales>). Estas placas se basan en un circuito impreso que contiene un microcontrolador de marca “ATMEL” y cuenta con entradas y salidas analógicas y digitales. Sus características más importantes son: multiplataforma, es factible, tiene flexibilidad en el *software* y *hardware* y es de código abierto (Manzano y Tapia, 2013, pp. 28).

1.3.2. *Raspberry Pi 2*

Es un mini computador, totalmente compatible con modelos anteriores y con una versión de Windows 10 adaptada a esta tarjeta de desarrollo. Son más potentes y eficientes que sus antecesores, con 1 GB de RAM, salida de video de 1080p, un socket microSD, 4 puertos USB 2.0, GPIO de 40 pines, su voltaje de operación es de 5V a 2A y tiene unas dimensiones de 86 x 56 x 20 mm. En la *Figura 10-1* se visualiza una Placa Raspberry Pi 2 (J.Pomeyrol, 2015, <http://www.muylinux.com/2015/02/02/raspberrypi2>).

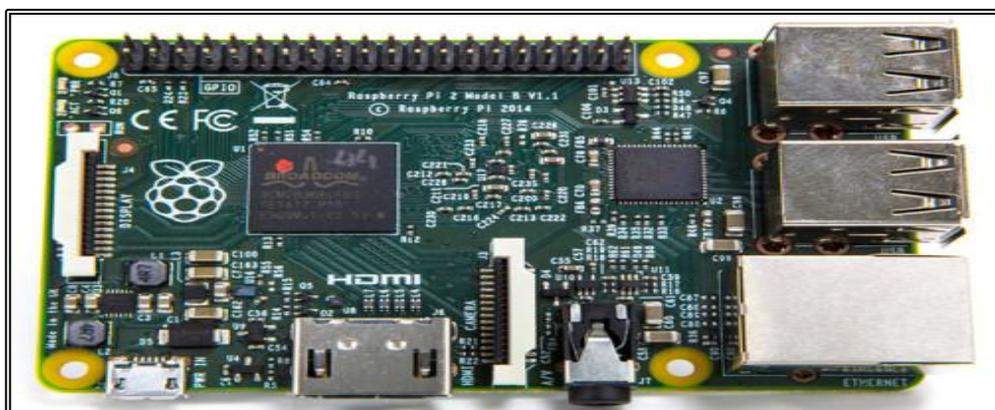


Figura 10-1: Placa Raspberry Pi 2.

Fuente: (J.Pomeyrol, 2015, <http://www.muylinux.com/2015/02/02/raspberrypi2>.)

1.3.3. Galileo

Trabajan bajo una arquitectura Intel, es una herramienta muy interesante en el campo de la Domótica, tiene un procesador de bajo consumo y de bajo costo. Ejecuta un sistema operativo Linux, que contiene librerías de Arduino, gracias a eso es escalable y con un *software* reutilizable. Puede ser programada desde sistemas operativos Mac OS, Microsoft Windows y Linux (Maturana, 2013, <https://www.xataka.com/makers/intelgalileo>). En la *Figura 11-1* se puede apreciar un ejemplo de placa Intel Galileo.



Figura 11-1: Placa Intel Galileo

Fuente: (García, 2015, pp. 10)

1.3.4. Comparación de las diversas tarjetas de desarrollo.

Una vez analizadas cada una de las tarjetas de desarrollo, se realizó una comparativa de sus principales características las que se aprecian en la *Tabla 4-1*.

Tabla 4-1: Comparativa de las diferentes tarjetas de desarrollo.

	Intel Galileo	Raspberry Pi	Arduino
Procesador	SoC Quark X100	Broadcom BCM2836 ARM Cortex-A7	ATMega328
E/S Analógicas	6	-	6
E/S Digitales	14	8	14
Memoria RAM	512 KB	1 GB	2 KB
Memoria Flash	8 MB	-	32 KB
Voltaje de operación	3.3 V / 5 V	3.3 V / 5 V	5 V
Voltaje de entrada	5 V	5 V	7 – 12 V
USB	2	4	1
Ethernet	10/100	10/100	No contiene
Precio	\$90	\$50	\$30
Entorno de desarrollo	Arduino IDE	Linux, IDLE, OpenEmbedded, QEMU, Scratchbox, Eclipse, Windows.	Arduino IDE
Velocidad	400 MHz	900 MHz	16 MHz
Sistema Operativo	Linux, Windows	Distribuciones de Linux, Windows 10.	Ninguno

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

De la *Tabla 4-1* se obtiene que, la tarjeta de desarrollo que se utiliza en el presente prototipo es arduino, debido a su bajo costo y prestaciones, además de que su entorno de desarrollo es mas amigable que los entornos utilizados en Raspberry Pi, debido a su popularidad en el mercado existe una amplia gama de documentación bibliográfica y ejemplos con los cuales se puede aprender y entender su programación, y su bajo consumo lo hace ideal para aplicaciones de usos de larga duración.

Dependiendo de la aplicación y de la demanda que se requiera podemos encontrar varios modelos de tarjetas Arduino que se podrán apreciar en la *Tabla 5-1*.

Tabla 5-1: Modelos de tarjetas de desarrollo Arduino.

Modelo	UNO	NANO	MEGA 2560	YUN	PRO-MINI
Microcontrolador	ATmega 328 P	ATmega168 o ATmega 328	ATmega2560	ATmega32U4	ATmega 328 P
Alimentación	7 -12 VDC	7 -12 VDC	7 -12 VDC	5 VDC	5 VDC
Corriente	20 mA	40 mA	20 mA	40 mA	40 mA
Pines Analógicos	6	8	16	12	8
Pines Digitales	14	14	54	20	14
Pines PWM	6	0	15	7	6
Memoria	32 KB	16,32 KB	256 KB	32 KB	32 KB
Dimensiones	68.6 x 53.4 mm	45 x 18 mm	101.52 x 53.3 mm	73 x 53 mm	30 X 18 mm
Conector	Conector USB, Power Jack ICSP Boton de Reset	Conector mini-B USB, Boton de Reset	Conector USB, Power Jack ICSP Boton de Reset	Microprocesador Linux, Ethernet, Wifi, USB, Lectora SD, RAM y Flash Memory, 3 Botones de Reset	Por medio de una tarjeta o un cable FTDI
Imagen					

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

De la tabla 5-1 se obtuvo que la tarjeta Pro-mini es ideal para el uso en la implementación de este prototipo, debido a su tamaño, bajo consumo de energía y su número de terminales abastece las necesidades del proyecto.

CAPÍTULO II

2. IMPLEMENTACIÓN; *HARDWARE Y SOFTWARE DEL PROTOTIPO*

En este capítulo se presenta la concepción general del sistema de indentificacion de personas, brinda una descripción de los requerimientos *hardware*, requerimientos *software* para la implementación, y los respectivos diagramas de flujo de las distintas etapas *software* que conforman el sistema.

2.1. Concepción general del Sistema

La concepción general del sistema se encuentra en la *Figura 1-2* en donde se aprecia que está constituido por tres nodos: el nodo llave es la persona que posee el código almacenado en distintos tipos de etiquetas (pulseras, llaveros y tarjetas), el nodo lector está formado por el módulo lector y la aplicación móvil y el nodo de almacenamiento consta de la base de datos y la página web que por medio de internet permite ingresar datos, actualizarlos y visualizarlos. El módulo lector obtiene el código del nodo llave por medio del módulo RFID integrado a una tarjeta de desarrollo en conjunto con un módulo Bluetooth que transmite el código hacia la aplicación móvil, la cual obtiene la información del usuario de la base de datos para visualizarla por medio de una conexión inalámbrica Wifi.

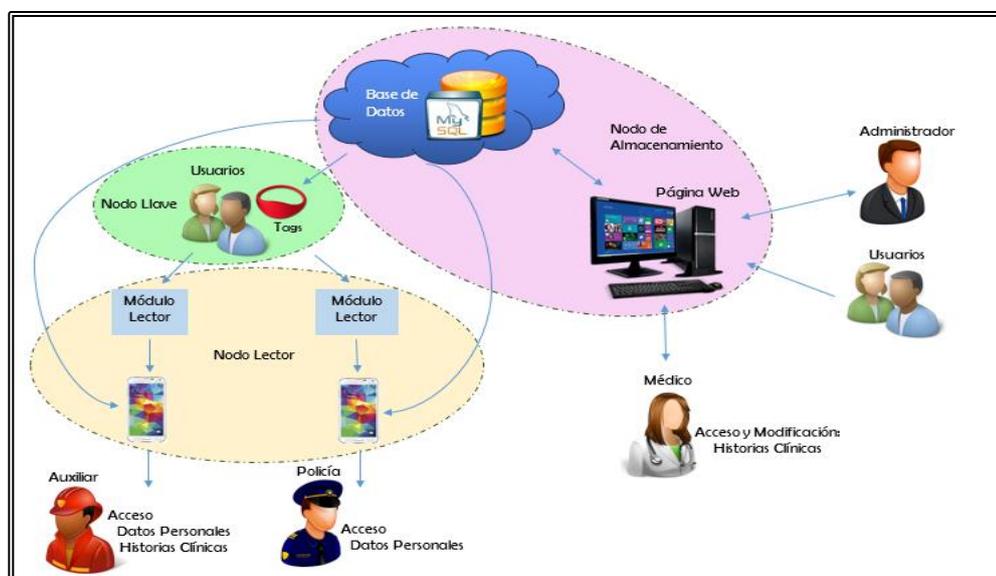


Figura 1-2: Concepción general del sistema

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

2.2. Requerimientos *Hardware* para el sistema

2.2.1. *Diseño de la Arquitectura del Módulo Lector*

El diagrama de bloques del módulo lector se los presenta en la *Figura 2-2* consta de los bloques de procesamiento, alimentación, adquisición de datos, codificación y transmisión de datos. La lectura se realiza con la interacción del bloque de adquisición y el bloque de codificación, los cuales están conformados por el módulo de lectura RFID y la etiqueta, respectivamente. El Bloque de alimentación está conformado por la batería y las terminales de carga lo que permite el funcionamiento del dispositivo electrónico, el bloque de transmisión contiene un módulo Bluetooth el cual permite conectar el dispositivo a la aplicación móvil y el bloque de procesamiento consta de un Arduino PRO-mini que permite la interacción y operación de los demás bloques.

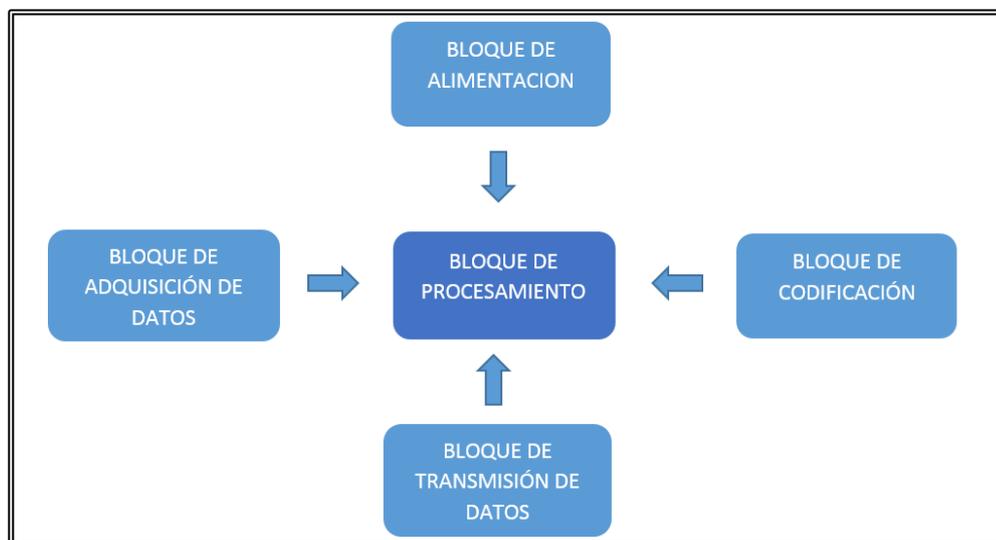


Figura 2-2: Diagrama de bloques del prototipo.

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

2.2.2. *Descripción de los dispositivos seleccionados.*

Arduino Pro Mini.- Cuenta con un microcontrolador ATmega 328 P. Tiene un regulador de voltaje de 5 v, su corriente de operación es de 40 mA, es pequeño, delgado, flexible, con una alta compatibilidad y su costo es muy reducido en relación a los demás. Su tablero viene sin cabeceras de pre-montado por lo que es necesario el uso de diferentes tipos de conectores o una soldadura directa de cables (Arduino, 2017, <https://www.arduino.cc/en/Main/ArduinoBoardProMini>). En la *Figura 3-2*

podremos apreciar su aspecto físico y distribución de entradas y salidas (Sparkfun, 2017, <https://www.sparkfun.com/products/11114>).

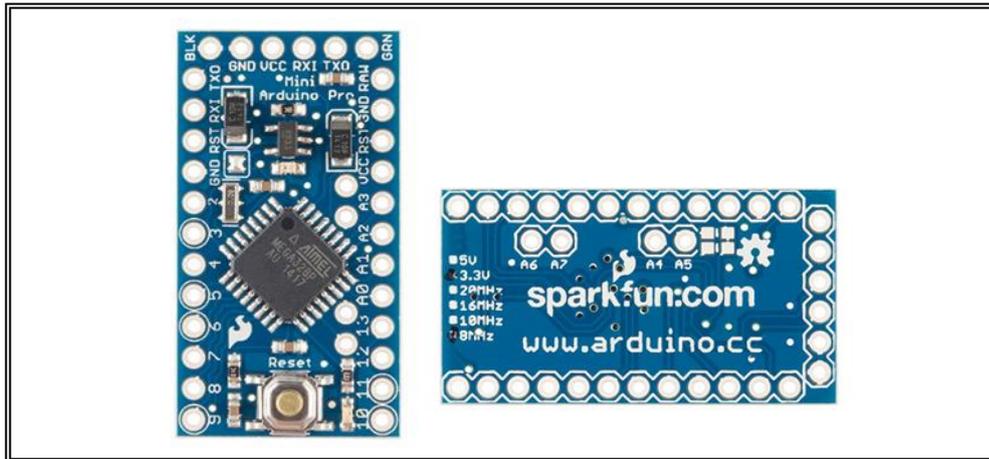


Figura 3-2: Arduino Pro-Mini - Aspecto físico y distribución de entradas y salidas
Fuente: (Sparkfun, 2017, <https://www.sparkfun.com/products/11114>)

RFID MF522-ED.- El sistema seleccionado para realizar el prototipo opera a 13,56 MHz con rangos de lectura de 0 a 60mm, utiliza 3.3V como voltaje de alimentación, una corriente máxima de 30mA y se controla a través del protocolo SPI, por lo que es compatible con casi cualquier microcontrolador. La velocidad de transmisión de datos es de 10Mbit/s y sus dimensiones de 40 x 60 mm (Electronilab, 2016, <https://electronilab.co/tienda/modulorfid>). En la *Figura 4-2* se aprecia el módulo RFID 13.56MHz.



Figura 4-2: Módulo RFID HF 13.56MHz
Fuente: (Electronilab, 2016, <https://electronilab.co/tienda/modulo-rfid-rc522-13-56mhz-rc522/>)

Bluetooth HC-06.- Es sencillo ya que brinda una comunicación fácil entre el móvil y un microcontrolador, tiene funcionalidad de modo esclavo. Permite una conexión sencilla y sin problemas mediante comandos AT a través de una puerta serial. El Módulo Bluetooth HC-06 se

puede apreciar en la *Figura 5-2*, el cual ha sido seleccionado como módulo de comunicación para el prototipo. Sus características más importantes son su frecuencia de operación que es de 2.4 GHz, tiene una antena PCB incorporada, su alcance es de 5 a 10 m , consta de 4 pines de acceso, mide 17 x 40 mm aproximadamente y trabaja a un voltaje de 3.6 a 6 V con un consumo de corriente de 30 a 40 mA.

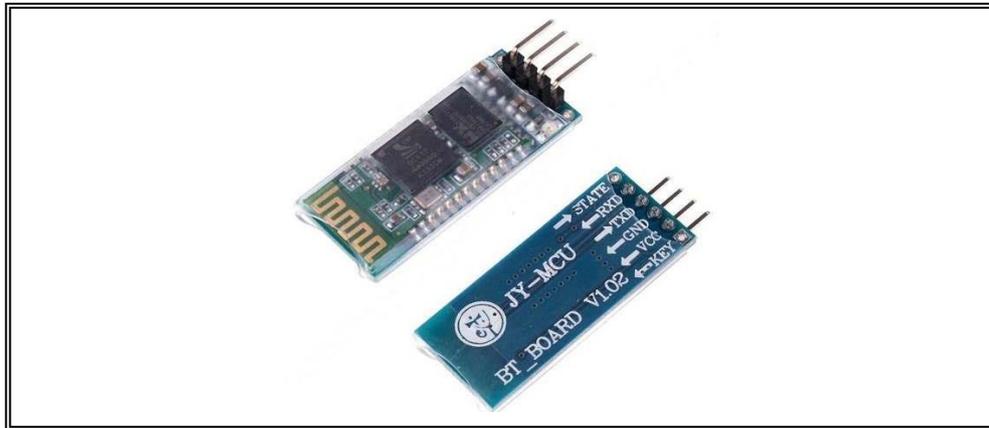


Figura 5-2: Modulo Bluetooth HC-06

Fuente: (Bot Science, 2015, http://botscience.net/store/index.php?route=product/product&product_id=69)

Regulador de Voltaje LM7833.- Es un regulador con una tensión de salida de 3.3V fijo. Brinda gran estabilidad y protección. Pueden dar salida a una corriente máxima de 800 mA (Teslabem, 2016, <http://teslabem.com/lm7833>). En la *Figura 6-2* se puede apreciar el regulador de voltaje LM7833.

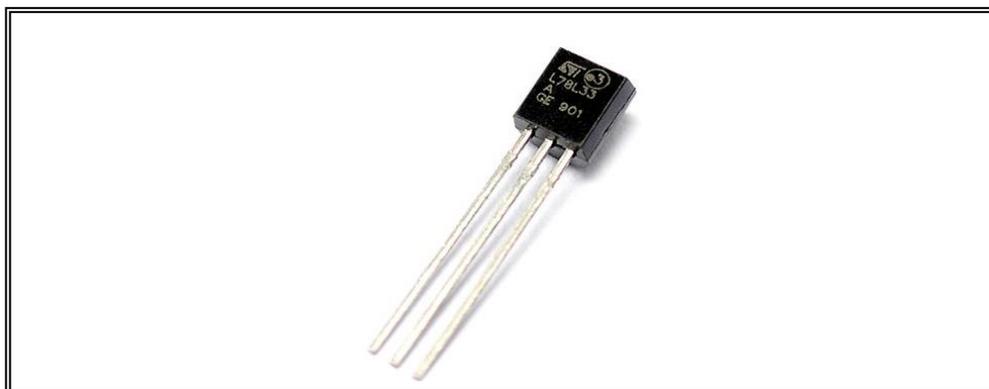


Figura 6-2: Regulador de Voltaje LM7833

Fuente: (lenk, 2013, http://www.ienk.com/document_news_info.html?products_id=3824)

2.2.3. *Esquema de conexión del Módulo Lector*

Las conexiones realizadas en el módulo lector del prototipo se puede observar en la *Tabla 1-2*. El equipo de procesamiento es el Arduino Pro-Mini y tiene sus terminales conectadas a los componentes de la siguiente manera:

- El Sensor de radio frecuencia RFID se encuentra conectado como se observa en la Tabla 1-2. Debido a que la tensión necesaria para el funcionamiento de módulo de RF es de 3.3 - 3.7 Voltios, la terminal del voltaje se encuentra conectada a un regulador de tensión LM7833 que es conectado a la salida de 5 V del Arduino.

Tabla 1-2: Esquema de conexión *Hardware*.

TERMINAL ARDUINO PRO-MINI	TERMINAL MÓDULO RFID
10	SDA
13	SCK
11	MOSI
12	MISO
Gnd	Gnd
-	IRQ
9	RST
3.3 V	3.3 V

Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

- El módulo Bluetooth se conecta a 5 V y GND del arduino. Sus terminales Rx y Tx se conectan a las terminales Tx y Rx en el arduino, respectivamente.
- El diodo LED se conecta a las salidas VCC y GND del arduino, este se enciende al realizar la lectura del código.

En la *Figura 7-2* se visualiza el esquema de conexión del módulo lector del prototipo.

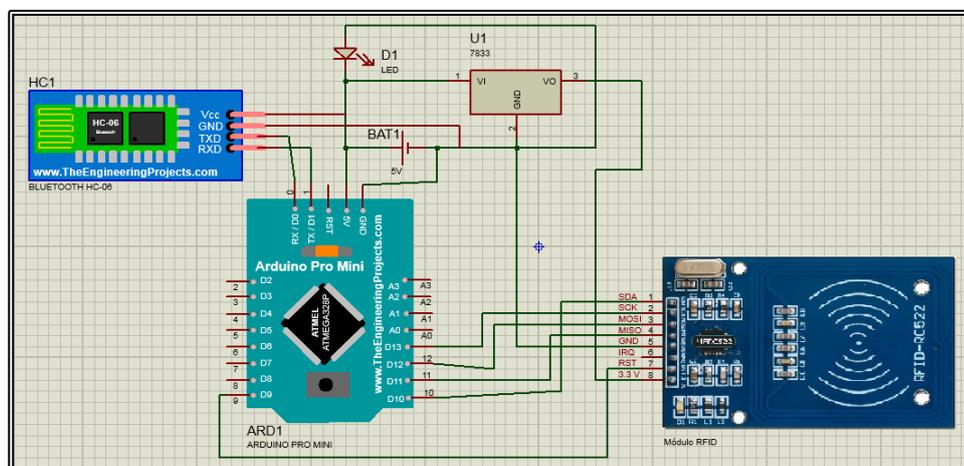


Figura 7-2: Esquema de conexión del módulo lector del Prototipo.

Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

2.2.4. Alimentación del prototipo

Se utilizó una Batería RPI PowerPack V1.0 de potencia diseñada para Raspberry Pi, permitiendo que esta tarjeta pueda trabajar incluso hasta 9 horas. Al utilizarlo con otro tipo de tarjetas de desarrollo con un menor consumo de energía dura un tiempo mayor (Sunfounder, 2017, <https://www.sunfounder.com/batterypowerpack.html>). Esta placa dispone de 2 puertos USB, con una corriente de salida de 1.8 A, una tensión de salida 5.1 V. En la *Figura 8-2* se puede apreciar cada una de las partes de la batería RPI PowerPack V1.0:

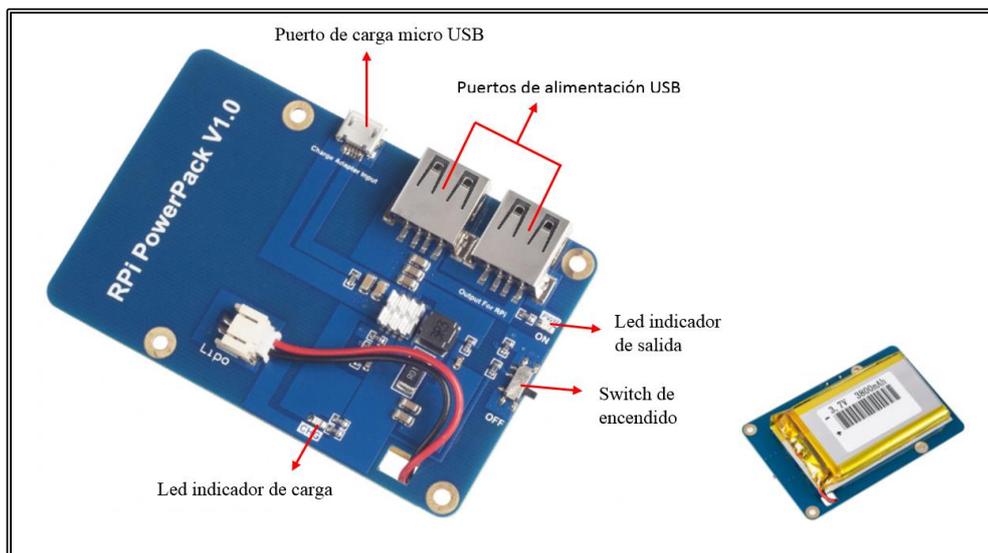


Figura 8-2: Partes de Batería RPI PowerPack V1.0

Fuente: (Sunfounder, 2017, <https://www.sunfounder.com/battery-power-pack.html>)

2.3. Requerimientos *Software* para el sistema

2.3.1. *MySQL*

Es una base de datos de código abierto, su rendimiento, confiabilidad y facilidad de uso la ha convertido en la principal opción de base de datos para aplicaciones basadas en la Web (Oracle, 2017, <https://www.oracle.com/lad/mysql/index.html>). Para la administración de MySQL a través de la Web se utiliza PhpMyAdmin. Las operaciones de uso frecuente como gestión de bases de datos, tablas, columnas, y relaciones, se pueden realizar a través de la interfaz de usuario, y tiene la capacidad de ejecutar directamente cualquier sentencia SQL (PhpMyAdmin, 2017, <https://www.phpmyadmin.net/>). Para la organización, gestión y administración de los datos utilizados en el prototipo de identificación se procedió a la creación de una base de datos en MySQL por medio de phpMyAdmin con una distribución de tablas mostradas en la *Figura 9-2*.

Diagrama de base de datos Entidad-Relación del sistema propuesto.

En la *Figura 9-2* se puede apreciar la organización del diagrama de la base de datos utilizada para el ingreso de datos personales e historias clínicas de los usuarios, las relaciones existentes entre tablas las cuales ayudan al manejo de datos y evita la redundancia en los mismos. El funcionamiento se basa en dos tablas principales relacionadas por el `id_usuario`; datos personales con diecinueve campos e historias clínicas con dos campos que se une a tres tablas secundarias por medio de la clave primaria `id_historiaC`, y corresponden a patológicos, familiares y hábitos tóxicos con diez, ocho y seis campos, respectivamente.

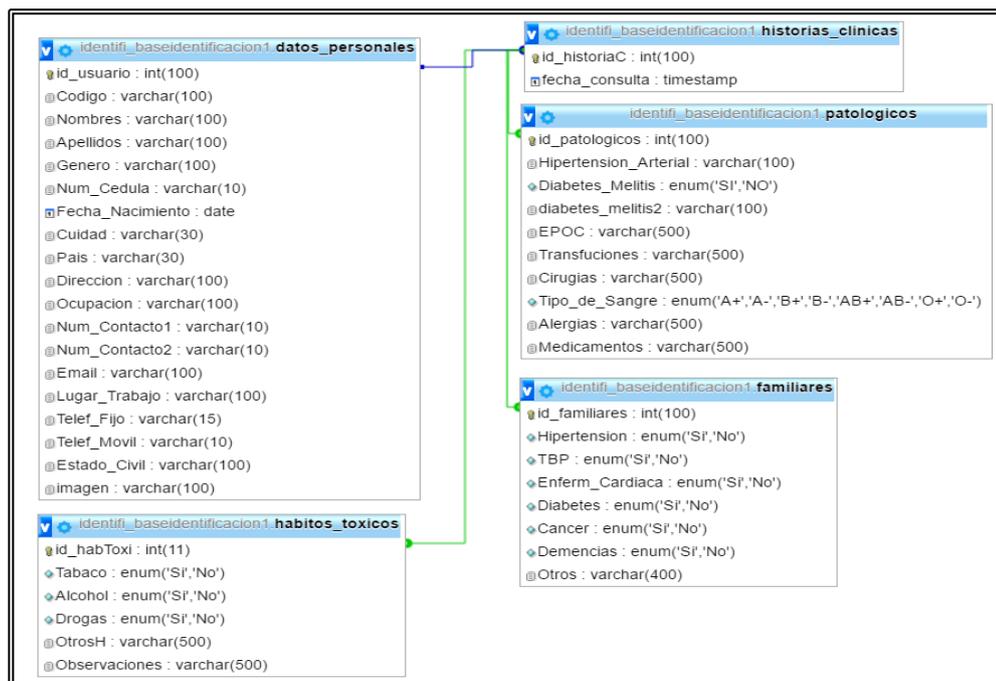


Figura 9-2: Diagrama de base de datos: Datos personales e Historias Clínicas.
Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

En la *Figura 10-2* podemos apreciar una tabla simple con siete campos realizada en phpMyAdmin que controla al personal (Policías, médicos, auxiliares y administrador), y brindará los permisos necesarios para el acceso a la información de los usuarios.

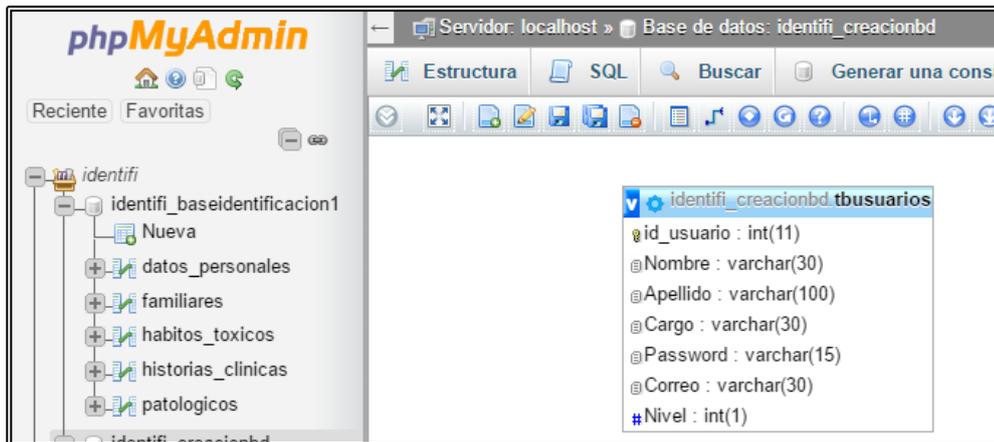


Figura 10-2: Control de personal en la base de datos.
Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

2.3.2. Dreamweaver

Es un editor de código HTML para el diseño y la administración de páginas Web. La utilización de este *software* permite trabajar con archivos HTML que incluyen un lenguaje basado en etiquetas y es el responsable de mostrar la página Web a través del navegador. (Adobe Systems Incorporated, 2010, p. 60). Para visualizar y manipular los datos almacenados se procedió a la programación de la aplicación web en Dreamweaver, entorno que se observa en la *Figura 11-2*, en donde se aprecia la declaración SQL para realizar una búsqueda

```

$lo_búsqueda3 = "-1";
if (isset($_GET['Num_Cedula'])) {
    $lo_búsqueda3 = $_GET['Num_Cedula'];
}
mysql_select_db($database_Identificacion, $Identificacion);
$query_búsqueda3 = sprintf("SELECT datos_personales.nombres,datos_personales.apellidos,
datos_personales.num_contacto1,datos_personales.imagen,historias_clinicas.fecha_consulta,
habitos_toxicos.*,familiares.*,patologicos.*
FROM `datos_personales` inner join historias_clinicas
on historias_clinicas.id_historiaC=datos_personales.id_usuario inner join habitos_toxicos
on habitos_toxicos.id_habToxi =historias_clinicas.id_historiaC inner join familiares
on familiares.id_familiares =historias_clinicas.id_historiaC inner join patologicos
on patologicos.id_patologicos =historias_clinicas.id_historiaC where datos_personales.Num_Cedula = %s",
GetSQLValueString($lo_búsqueda3, "text"));
$busqueda3 = mysql_query($query_búsqueda3, $Identificacion) or die(mysql_error());
$row_búsqueda3 = mysql_fetch_assoc($busqueda3);
$totalRows_búsqueda3 = mysql_num_rows($busqueda3);

$lo_búsqueda3 = "-1";
if (isset($_GET['busqueda3'])) {
    $lo_búsqueda3 = $_GET['busqueda3'];
}

```

Figura 11-2: Parte de la codificación – declaración SQL para una búsqueda.
Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

Página Web.- Para poder crear una página web necesitamos un Hosting Web, en donde se almacena la información, y se procesan los datos para poder ser visualizados. También necesitamos un dominio web, que es el URL por el que el usuario podrá acceder a la interfaz

gráfica. Existen dos tipos de modelos de Hosting Web, de pago y gratuito (EspacioPyme, 2004, <http://www.cambrasabadell.org/Att/files>).

Para el presente proyecto se ha contratado un hosting de pago debido a la disponibilidad del servicio, la confidencialidad y seguridad de la información. Se contrató los servicios de hosting con la URL: www.identificatecuador.com de la empresa Hosting Ecuador. Los archivos se subieron por medio del administrador de archivos en donde se cargan todos los documentos .php. Para poder acceder por medio del dominio al hosting, se utiliza el protocolo DNS como mecanismo de traducción del dominio a la dirección ip donde está alojado el mismo (Swhosting, 2014, <http://www.swhosting.com/blog>). Y el protocolo FTP, que opera en un entorno de cliente/servidor, logrando que su transferencia sea sencilla para el usuario (EspecialistasHosting, 2016, <http://www.especialistashosting.com/blog/index.php/2016/12/protocolo-ftp/>).

En la *Figura 12-2* se ingresa a la página web www.identificatecuador.com dando como primera alternativa el registro de personas el cual permitirá el ingreso de datos personales en la base de datos. En caso de no querer utilizar un registro nos muestra la opción Login, en la que puede acceder como Policía ingresando su nombre de usuario y contraseña correctos, y realizar la búsqueda de datos personales con el número de cédula de la persona que se requiera.

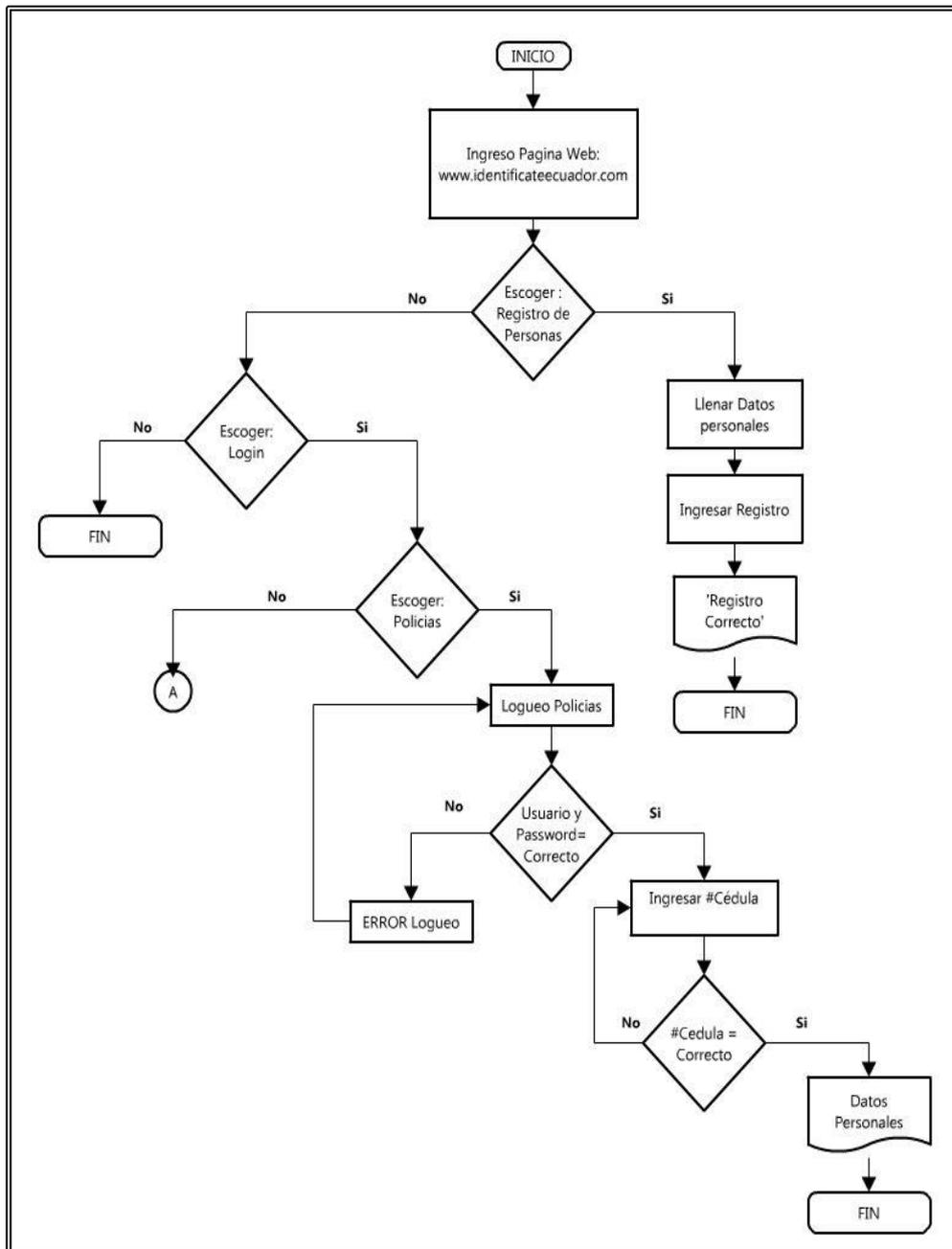


Figura 12-2: Diagrama de flujo página Web 1/3.

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

En la *Figura 13-2* muestra el ingreso como Auxiliar con su respectivo nombre de usuario y contraseña donde se permite buscar historias clínicas de pacientes por medio de su cédula. De la misma manera se escoge el acceso como Médicos permitiendo el ingreso a un menú que brinda las opciones de buscar, crear historias clínicas y ver el listado de pacientes existentes en la base de datos.

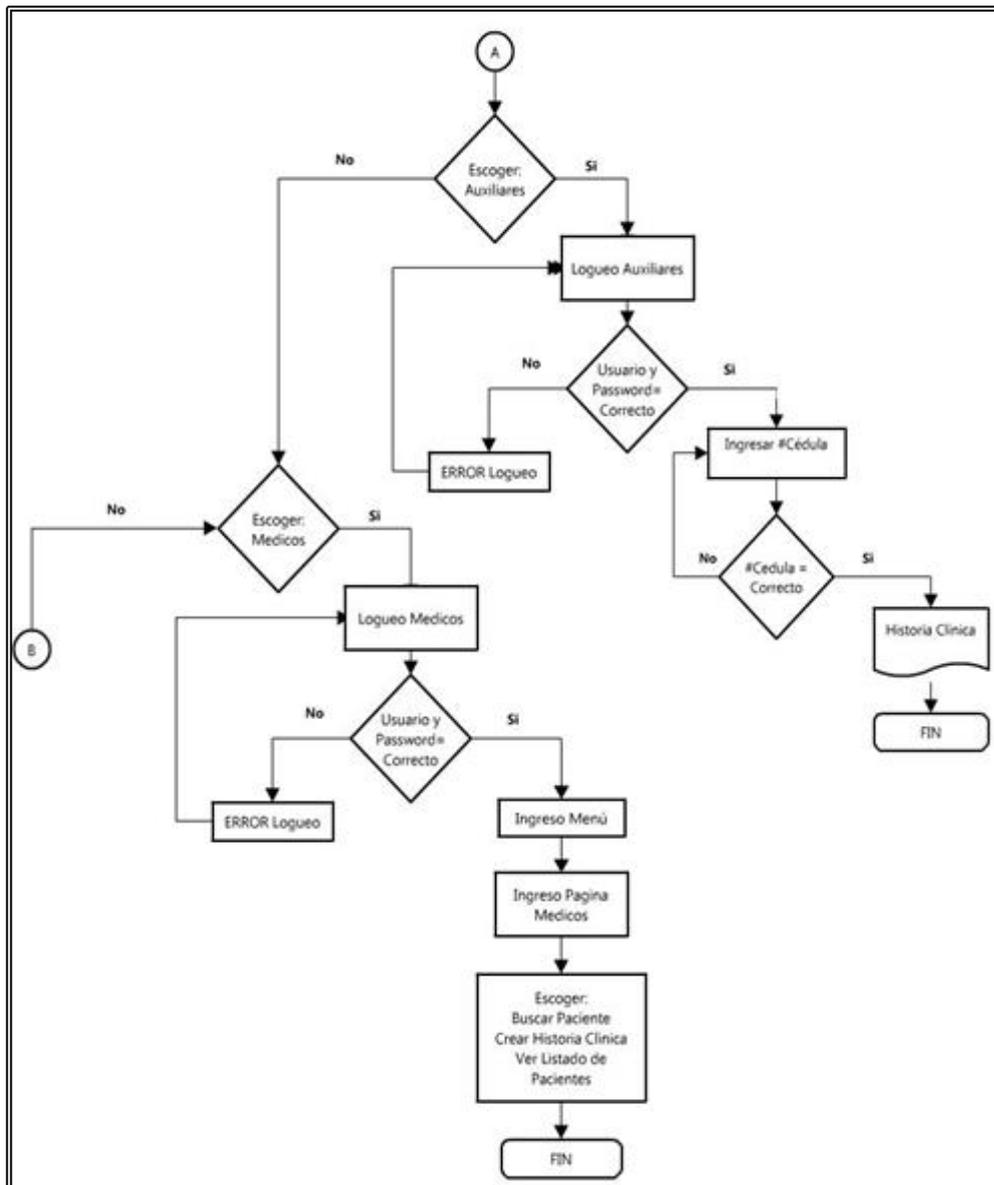


Figura 13-2: Diagrama de flujo página Web 2/3.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

Finalmente, en la *Figura 14-2* se muestra el ingreso como administrador permitiendo el acceso con su respectiva autenticación, el menú le brinda las opciones de usuario y personal, donde en la primera opción se puede buscar, eliminar y modificar datos personales y en la segunda se puede registrar, buscar, eliminar y modificar información del personal.

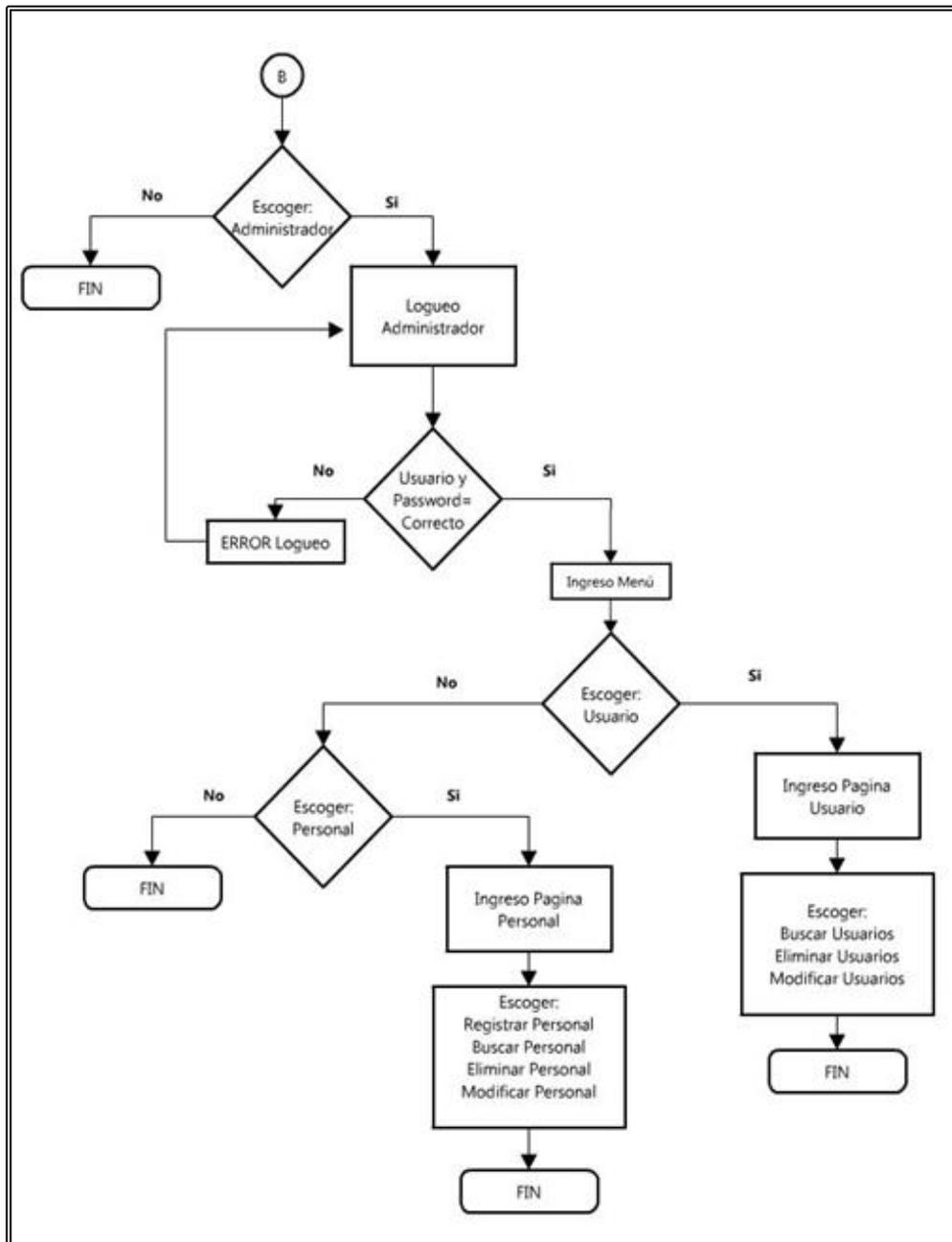


Figura 14-2: Diagrama de flujo página Web 3/3.

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

En la *Figura 15-2* se observa distribución gráfica de la página web.



Figura 15-2: Distribución gráfica de la página web.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

2.3.3. *Android Studio*

Es un entorno de desarrollo integrado (IDE) para el desarrollo de aplicaciones Android con un potente editor de códigos (Android Studio, 2016, <https://developer.android.com/studio/intro>). Utilizamos esta herramienta para crear una aplicación móvil que permita la conexión entre la información almacenada en la base de datos y el módulo lector, de manera que se pueda visualizar datos personales e historias clínicas de una manera segura. En la *Figura 16-2*, se observa un fragmento de código de una de las ventanas de la aplicación móvil en Android Studio, en la que se muestran los estilos de los botones e imágenes de la pantalla inicial.

```

<Button
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:text="Policias"
    android:id="@+id/btn1"
    android:layout_gravity="center"
    android:layout_marginRight="32dp"
    android:textColor="#2d63cf"
    android:textSize="14dp"
    android:layout_below="@+id/imageView5"
    android:layout_toStartOf="@+id/imageView4" />

<Button
    android:layout_width="140dp"
    android:layout_height="wrap_content"
    android:text="Auxiliares"
    android:id="@+id/btn2"
    android:layout_marginLeft="25dp"
    android:textColor="#2d63cf"
    android:textSize="14dp"
    android:layout_below="@+id/imageView4"
    android:layout_alignEnd="@+id/textView48" />

```

Figura 16-2: Fragmento de código programado – Configuración de botones.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

Aplicación Móvil.- Esta parte del sistema es la que brindará a policías y auxiliares el acceso a la información almacenada de en la bases de datos (datos personales e historias clínicas).

Se puede apreciar el funcionamiento de la aplicación móvil en las Figuras 17-2 Al inicializar la aplicación se escoge entre Policías o Auxiliares, los mismos que tendrán que ingresar su usuario y password para el ingreso a la plataforma, teniendo que activar el Bluetooth, el cual pedirá el permiso para ser activado en el dispositivo móvil, en caso de no poder activar el Bluetooth la aplicación se cerrará.

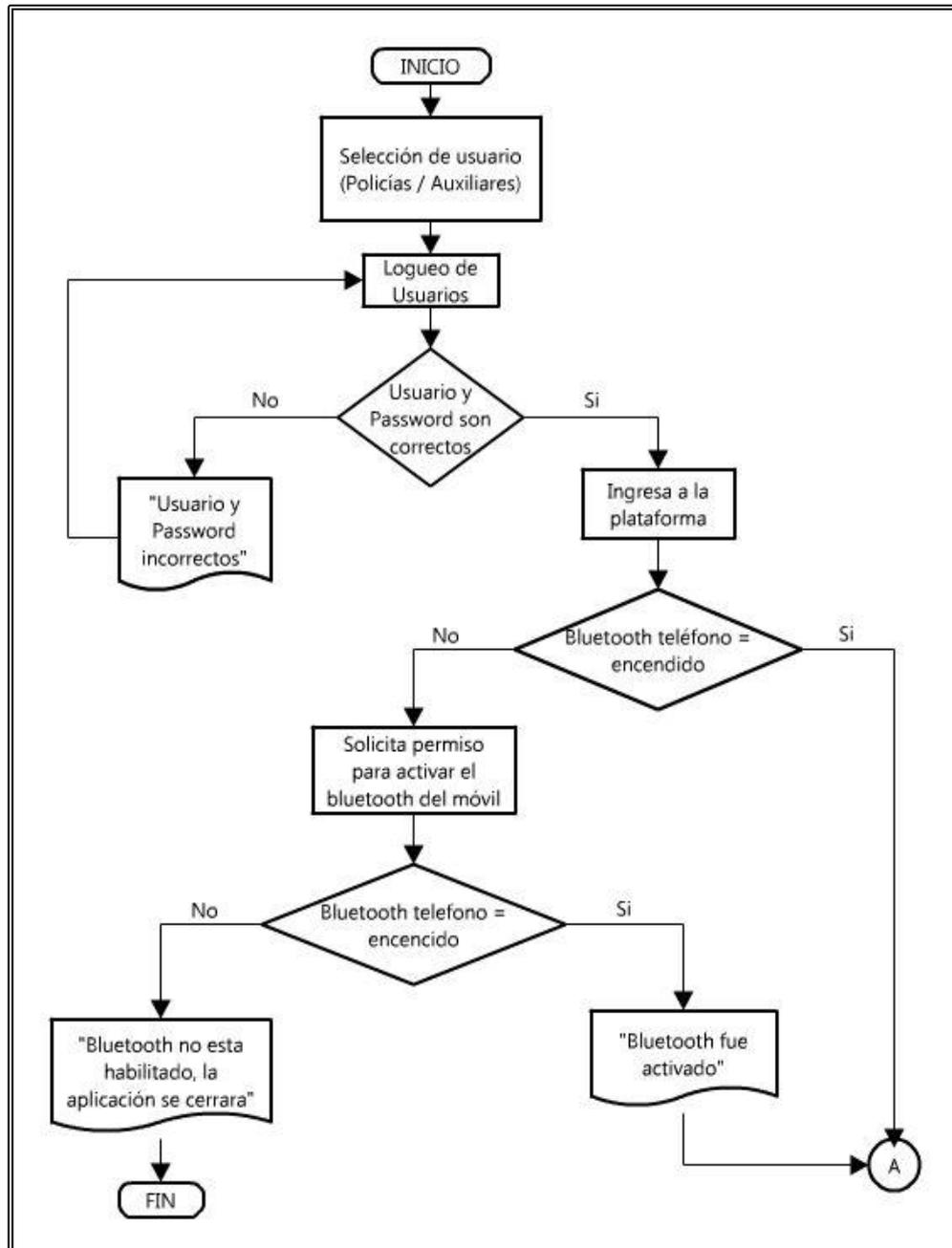


Figura 17-2: Diagrama de flujo Aplicación Móvil 1/2.

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

En la *Figura 18-2* se visualiza el proceso para escoger el Bluetooth HC-06, una vez conectado se realiza la lectura del código de la etiqueta, si el código existe se realiza la búsqueda de los datos dentro de la base de datos, permitiendo visualizar los datos personales e historias clínicas solicitados. En caso de no habilitar el Bluetooth la aplicación mostrará mensajes de error.

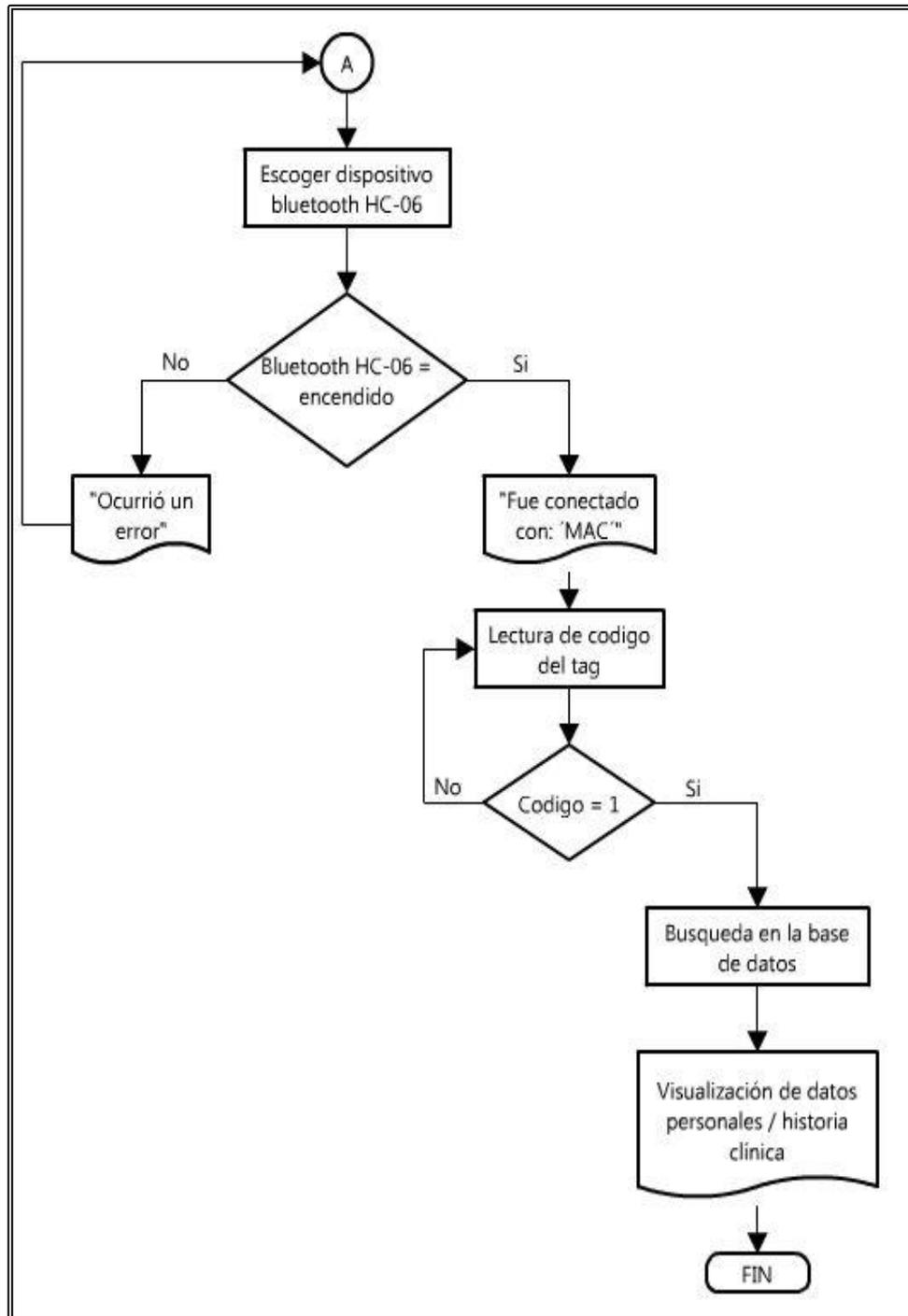


Figura 18-2: Diagrama de flujo Aplicación Móvil 2/2.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

En la *Figura 19-2* se observa la pantalla resultante de la parte de código mostrado anteriormente.



Figura 19-2: Pantalla resultante de la Aplicación Móvil.
Fuente: COBOS, Marlory & ORTIZ, Mayra 2017.

2.3.4. Arduino

Cuenta con un entorno de desarrollo integrado gratuito y disponible para las plataformas Windows, Linux y Mac. Es un lugar en el que se puede escribir la aplicación, ejecutarla y descargarla a la tarjeta de desarrollo, y de esta manera permite la interacción entre el *hardware* y el *software* (Doutel, 2015, <https://www.xataka.com/especiales>). El código Arduino utilizado en el módulo lector se puede apreciar en la *Figura 20-2*.

```

sketch_mar13a $
#include "SPI.h"
#include "MFRC522.h"

#define SS_PIN 10
#define RST_PIN 9
#define LED_PIN A0
#define LED_PIN2 A1

MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

void setup() {
  pinMode(2, OUTPUT);
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
  pinMode(LED_PIN, OUTPUT);
  pinMode(LED_PIN2, OUTPUT);
}

void loop() {
  if (!rfid.PICC_IsNewCardPresent()
  || !rfid.PICC_ReadCardSerial())
    return;

  MFRC522::PICC_Type piccType =
  rfid.PICC_GetType(rfid.uid.sak);

  if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
  piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
  piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    return;
  }

  String strID = "";
  for (byte i = 0; i < 4; i++) {
    strID +=
    (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
    String(rfid.uid.uidByte[i], HEX) +
    (i!=3 ? ":" : "");
    digitalWrite(2, HIGH);
    delay(100);
    digitalWrite(2, LOW);
  }
  strID.toUpperCase();
  Serial.println(strID);

  rfid.PICC_HaltA();
  rfid.PCD_StopCryptol();
}

```

Figura 20-2: Código utilizado en el módulo lector – Arduino Pro-mini.
Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

En la *Figura 21-2* se observa el diagrama de flujo del funcionamiento del programa que se carga en el módulo del Arduino Pro-Mini utilizado en el dispositivo lector. Primero se realiza la declaración de las variables de RFID y LED, si se lee una etiqueta se obtiene el código y se enciende el LED. En caso de no hacerlo finaliza.

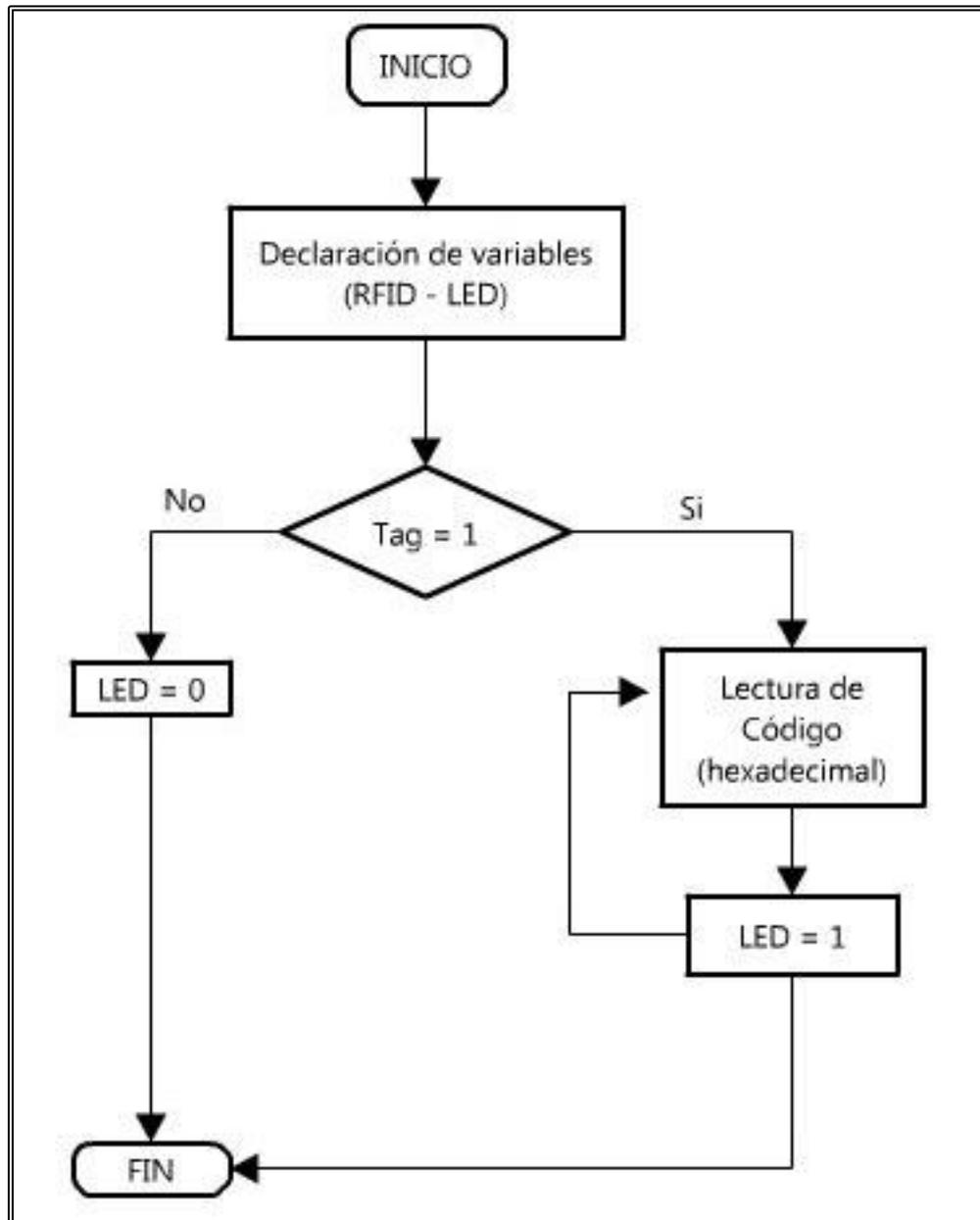


Figura 21-2: Diagrama de flujo Programación de arduino.
Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

CAPITULO III

3. PRUEBAS Y RESULTADOS DEL SISTEMA

El presente capítulo muestra los resultados obtenidos en las pruebas realizadas al prototipo de identificación, tanto en *hardware*, como en *software*. Pruebas *hardware* de funcionamiento tomando en cuenta tiempos de respuesta, distancia de operación y funcionamiento de la alimentación en el prototipo y pruebas en *software* del funcionamiento de la aplicación móvil y página Web. Se presenta también el análisis de costos de la implementación del prototipo.

3.1. Pruebas del hardware implementado.

Las dimensiones del módulo lector son de 9.1cm x 5.5cm x 6.2cm y la distancia de lectura del módulo lector RFID a las etiquetas es de 1.5 cm. En la *Figura 1-3* se puede apreciar el *hardware* implementado, en donde se observa la parte interna y externa del módulo lector y los distintos tipos de etiquetas utilizadas.

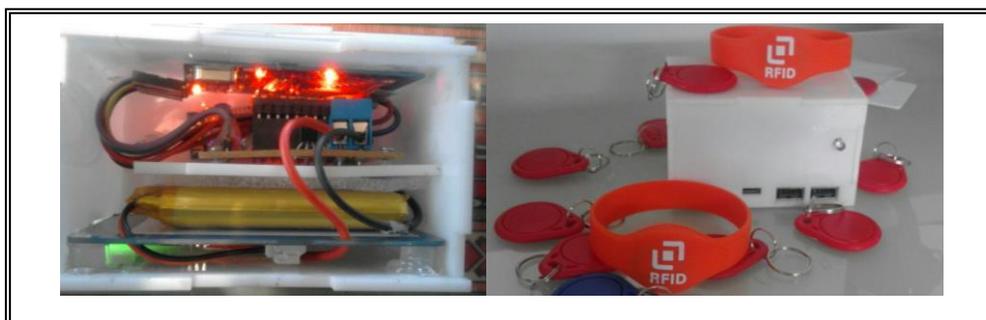


Figura 1-3: Parte interna y externa del módulo lector implementado.

Fuente: COBOS, Marlory & ORTIZ, Mayra 2017.

3.1.1. Tiempo de respuesta del dispositivo del lector RFID.

Las mediciones de tiempo en segundos se realizaron tomando en cuenta dos tipos de red inalámbrica, wi-fi y datos móviles. En la Tabla 1-3 se muestra las mediciones del tiempo de respuesta del lector RFID aplicado a dieciséis muestras entre llaveros, pulseras y tarjetas.

Tabla. 1-3: Mediciones tiempo de respuesta RFID en llaveros, pulseras y tarjetas.

No	Código	Tipo de etiqueta	Tiempo de Respuesta WI-Fi (segundos)	Tiempo de Respuesta Datos Móviles (Segundos)
1	BE:FA:B6:5E	Pulsera	2	10
2	7E:C9:B9:5E	Pulsera	2	2
3	9E:3E:B7:5E	Pulsera	1	3
4	F5:C8:36:D5	Tarjeta	2.5	4
5	F7:1E:C7:E3	Tarjeta	1	4
6	54:C6:6E:B8	Llavero	1	3
7	B4:FD:C1:A7	Llavero	3	2
8	14:53:95:B8	Llavero	1	2
9	B4:69:51:B8	Llavero	2	3
10	84:AA:64:B8	Llavero	1	2
11	94:06:6B:B8	Llavero	1	2
12	44:09:4E:B8	Llavero	1	2
13	D4:AC:C5:A7	Llavero	1	7
14	C4:61:C5:A7	Llavero	2	1
15	54:17:95:B8	Llavero	2	3
16	25:F6:6B:76	Llavero	2	6
PROMEDIO			1,4	3,5

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

Como se puede apreciar en la *Tabla 1-3*, el promedio de tiempo de respuesta del dispositivo dentro de una red Wi-Fi es de 1.4 segundos, mientras que en una red de datos móviles es de 3.5 segundos.

Considerando que el tiempo de respuesta en una red móvil es de 2.1 segundos mayor que de la red Wi-Fi, la diferencia de tiempos no es significativa, demostrando que en ambos casos el tiempo de respuesta del dispositivo es eficiente.

En el Ecuador la Cobertura de Servicio Móvil Avanzado (SMA) es de 96.98%, cubriendo el 99.88% de la población urbana y el 88.63% de la población rural del país (Ministerio de Telecomunicaciones y de la Sociedad de la Información, 2016 pág. <https://www.telecomunicaciones.gob.ec>). Garantizando la funcionalidad del prototipo tanto en la zona urbana como en la zona rural del país.

3.1.2. *Tiempo de respuesta de autenticación y búsqueda de datos.*

Para las mediciones de tiempo en segundos, se realizó la autenticación de policías y auxiliares, con datos en la red móvil. La Tabla 2-3 muestra las mediciones de tiempo de respuesta de autenticación y búsqueda de datos en cinco policías y cinco auxiliares.

Tabla 2-3: Mediciones de tiempo de respuesta de autenticación y búsqueda de datos.

POLICIAS (Datos Personales)			
No	Nombre de Usuario1.	Tiempo de Autenticación (segundos)	Tiempo de Búsqueda de Datos (segundos)
1	NadeshkaC	3	3
2	MartinL	3	2
3	CorinaM	1	3
4	SantiagoR	1	3
5	Mayra	1	4
PROMEDIO		1.8	3
AUXILIARES (Historia Clínica)			
1	CristianA	1	4
2	Andrea2	1	5
3	SilviaT	1	4
4	EstefaniaJ	2	3
5	AlfonsoP	1	3
PROMEDIO		1.2	3.8

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

Como se puede apreciar en la *Tabla 2-3*, el tiempo promedio de autenticación del personal policial y auxiliar fue de 1.8 y 1.2 segundos, respectivamente, comprobando la eficiencia al momento de la autenticación del personal. El tiempo de respuesta de la búsqueda de datos fue de 3 segundos para el caso del personal policial, mientras que para el personal auxiliar fue de 3.8 segundos; habiendo una diferencia de 0.8 segundos en el caso de solicitar la historia clínica.

3.1.3. *Distancia de Operación Bluetooth.*

Como se puede apreciar en la *Figura 2-3* la distancia de funcionamiento en espacio libre tiene un rango para el nivel de recepción de datos categorizados como: alto de 0 a 26 metros con una conexión óptima; regular de 27 a 40 metros, la calidad de la conexión no es buena debido a los tiempos de espera para establecer la comunicación y nulo cuando sobre pasa los 40 metros donde no se establece conexión alguna con el dispositivo.

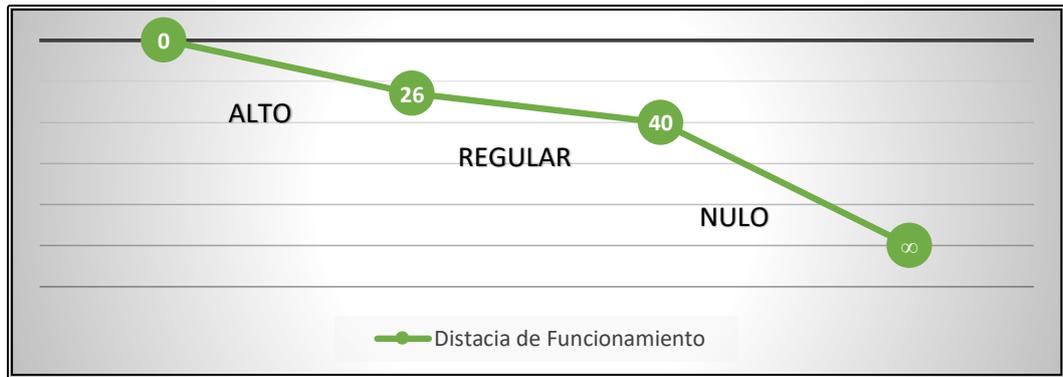


Figura 2-3: Rangos de funcionamiento respecto a la distancia en espacio libre.
Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

En entornos que presentan obstrucciones se observó, que la distancia de operación está acorde con lo establecido en el datasheet del Bluetooth HC-06 que funciona en un rango de 0 a 10 metros.

3.1.4. Consumo del prototipo

En base a las mediciones realizadas con el multímetro en cada elemento que conforma el módulo lector se consiguieron los valores que se visualizan en la *Tabla 3-3* en donde se pudo comprobar que los valores teóricos de consumo de voltaje y corriente se aproximan mucho a los valores reales. En cuanto al voltaje que ingresa en cada dispositivo tenemos que el Bluetooth consume 5.14 voltios y el módulo lector RFID 3,5 voltios. El arduino es alimentado por una batería de litio de 5.15 voltios con 300 miliamperios, como se observa en la *Tabla 3-3* el consumo total de corriente en el dispositivo es de 67.2 mA, por lo que se comprueba que el sistema tiene un bajo consumo de corriente lo que permitiría una larga duración de la batería.

Tabla 3-3: Consumo de corriente y voltaje en el módulo lector.

No.	Dispositivo	Consumo de Corriente (miliamperios)	Voltaje de entrada (voltios)
1	Bluetooth	38.2	5.14
2	Módulo RFID	29	3.5
Total		67.2	

Fuente: COBOS, Marlory & ORTIZ, Mayra 2017

Para verificar el funcionamiento de la batería se realizaron mediciones del tiempo de carga y descarga de donde se obtiene resultados de 4 horas y 18 minutos, y de 20 horas, respectivamente. Justificando así la eficiencia de la batería al cumplir con los requerimientos. En la *Figura 3-3* se muestra la medición del consumo de la batería Raspberry Pi PowerPack V1.0 utilizada en el módulo lector.

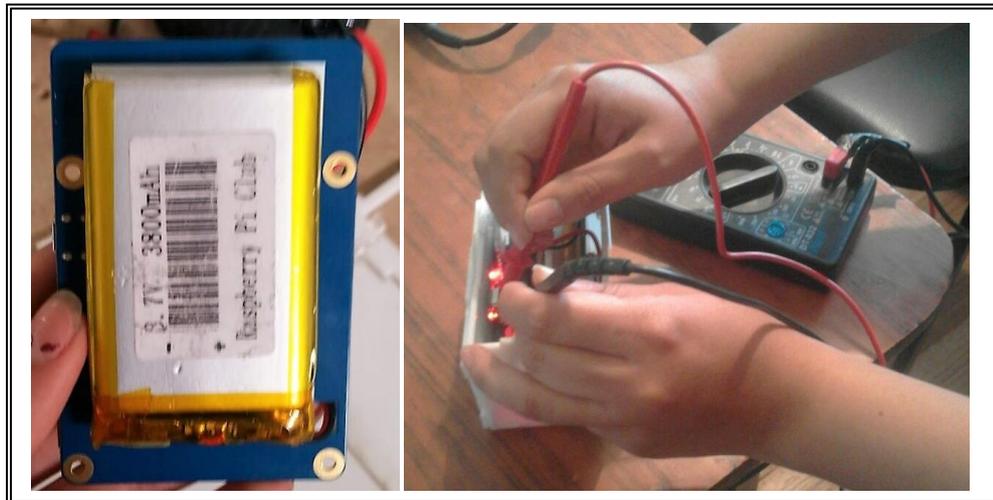


Figura 3-3: Medición de consumo Batería Raspberry Pi PowerPack V1.0.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

3.2. *Pruebas del software del prototipo implementado.*

3.2.1. *Aplicación Móvil*

En cuanto la interfaz gráfica y funcionamiento de la aplicación realizada en Android Studio 2.2.3, apreciar en la *Figura 4-3* el ícono del logo “Identificate Ecuador” la cual permite ejecutar la aplicación.

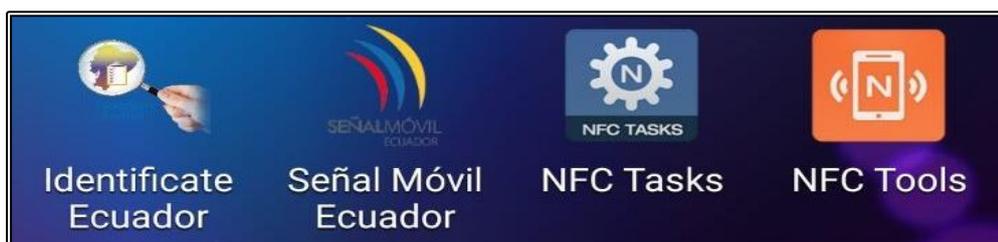


Figura 4-3: Icono de la aplicación móvil – Identificate Ecuador.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

Una vez seleccionado el tipo de usuario, la persona debe ingresar sus datos de autenticación, que serán registrados solamente por el administrador www.identificateecuador.com. En la *Figura 5-3* se observa la pantalla de autenticación.

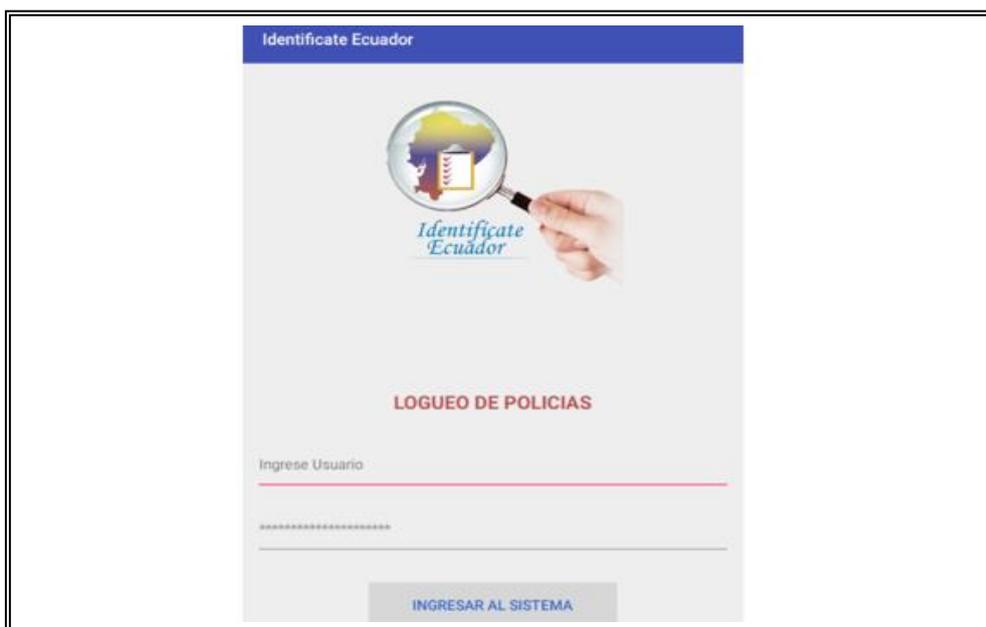


Figura 5-3: Pantalla de autenticación.

Fuente: COBOS, Marlory & ORTIZ, Mayra 2017.

Para el funcionamiento de la aplicación se requiere la activación del Bluetooth, para ello se presiona el botón *Conectar*, que despliega la lista de los dispositivos vinculados. Se comprobó la conexión entre el Bluetooth del dispositivo móvil y del módulo lector.

Para realizar la búsqueda en el servidor web, se presiona el botón buscar y se visualiza los datos personales o historia clínica correspondiente como se observa en la *Figura 6-3*. Mediante el botón “BUSCAR OTRA PERSONA”, se puede realizar nuevas búsquedas.

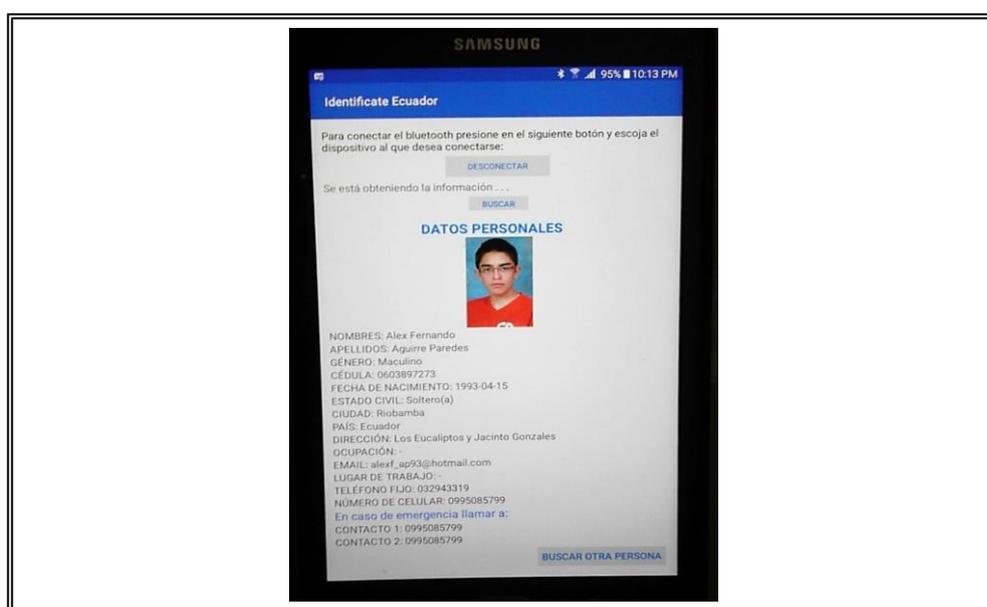


Figura 6-3: Búsqueda datos personales.

Fuente: COBOS, Marlory & ORTIZ, Mayra 2017.

En la *Figura 7-3* muestra una de las pruebas de funcionamiento de lectura de historias clínicas realizadas mediante la aplicación móvil, comprobando así el correcto funcionamiento de la misma.

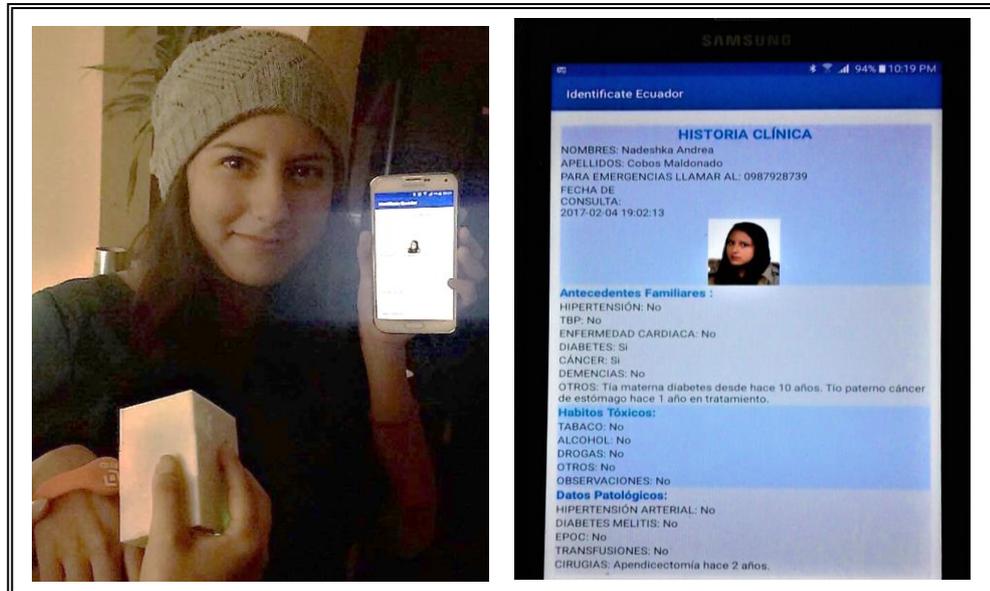


Figura 7-3: Lectura de historia clínica realizada en una persona.

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

3.2.2. *Página Web*

Las pruebas consistieron en acceder a la página www.identificateecuador.com, y comprobar el funcionamiento de los distintos formularios existentes en esta.

Registro.- Para comprobar el funcionamiento de los formularios de registro se ingresó a la sección de registro de personas, llenando el formulario que se despliega y comprobando que al ingresar los datos correctamente el registro es exitoso como se muestra en la *Figura 8-3*.

INSERTAR NUEVO REGISTRO DE PERSONAS.

Formulario de Registro:

Instrucciones: Los campos que estan con * son obligatorios. (El codigo debe solicitar al administrador de la página).

Código*: F8.MN:67:S3

Nombres*: Nadeshka Andrea

Apellidos*: Cobos Maldonado

Género*: Masculino
 Femenino

Numero de Cédula*: 0605987494

Fecha de Nacimiento(año-mes-día)*: 1998-06-30

Cuidad*: Riobamba

Pais*: Ecuador

Dirección*: Bolivia 17-11 y Colombia

Ocupación*: Estudiante

Número de Contacto*: 0987928739

Número de Contacto2*:

Email: Nadeshka0630@gmail.com

Lugar de Trabajo: Universidad

Teléfono Fijo*: 032393008

Teléfono Móvil*: 0969028143

Estado Civil*: Soltero(a) ▼

Foto*: descarga.jpg

Figura 8-3: Prueba de registro de usuarios.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

Autenticación.- En la *Figura 9-3* se aprecia el menú de autenticación de los usuarios,



Figura 9-3: Menú de autenticación dentro del sistema.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

La prueba realizada se puede apreciar en la *Figura 10-3* donde se autenticó al usuario administrador.

BIENVENIDO AL LOGIN DE ADMINISTRADOR

Introduce los datos requeridos:

Instrucciones: Coloca tu nombre de usuario y password para ingresar al menú solicitado, coloca los datos correctamente o de lo contrario no se te dará ingreso.

Nombre: CarlosG *

Password: **** *

Figura 10-3: Autenticación del administrador.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

Busqueda.- Dentro de la página web se encuentran varias búsquedas, para comprobar su funcionamiento se realizó dentro de la autenticación de Policías la búsqueda de datos personales los mismos que se obtuvieron mediante el número de cédula del usuario. En la *Figura 11-3* se muestra el resultado de la búsqueda realizada.

Datos Personales:



Nombres:	Diana Carolina
Apellidos:	Pazmiño Avalos
Género:	Femenino
Número de Cédula:	1720546538
Fecha de Nacimiento:	1992-12-03
Cuidad:	Riobamba
País:	Ecuador
Dirección:	Las Acacias
Ocupación:	Estudiante
Numeros de Contacto:	0987149910
	0995563431
Email:	caryto12.99@gmail.com
Lugar de Trabajo:	ESPOCH
Teléfono Fijo:	032300752
Teléfono Víctima	0984950388
Estado Civil:	Soltero(a)

Figura. 11-3: Búsqueda realizada por el policía – Datos personales.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

Registro de Historia Clínica.- Se procedió a realizar las pruebas de funcionamiento de la parte de médicos, realizado la creación de una historia clínica por medio del id del paciente, considerando que la historia clínica está dividida en tres tablas denominadas antecedentes familiares, hábitos tóxicos y datos patológicos se realizó el registro de los datos necesarios para cada tabla se considera que en la base de datos existe 30 historias clínicas ingresadas. En la *Figura 12-3* se visualizan los formularios de registro de historias clínicas.

Formulario de registro de hábitos tóxicos:

Instrucciones: Los campos que se encuentran con * son obligatorios.

Número Historia clínica*: 15

Tabaco*: Si
 No

Alcohol*: Si
 No

Drogas*: Si
 No

Otros*: No tiene otros hábitos tóxicos

Observaciones*: Bebedor social desde hace 5 años

Formulario de registro de antecedentes familiares:

Instrucciones: Los campos que se encuentran con * son obligatorios.

Número de historia clínica*: 15

Hipertensión*: Si
 No

TBP*: Si
 No

Enfermedad Cardíaca*: Si
 No

Diabetes*: Si
 No

Cáncer*: Si
 No

Demencias*: Si
 No

Otros*: Madre con diabetes tipo 2 hace 5 años

Formulario de registro de datos patológicos:

Instrucciones: Los campos que se encuentran con * son obligatorios.

Número de historia clínica*: 15

Hipertensión Arterial*: No

Diabetes Mellitus*: No

EPOC*: No

Transfusiones*: Transfusión realizada hace 3 años

Cirugías*: Apendicectomía hace 3 años

Tipo de Sangre*: AB+ ▾

Alergias*: Penicilina

Medicamentos*: acetaminofén una diaria durante un año

Insertar registro

Figura 12-3: Formularios de registro de historias clínicas.

Fuente: COBOS, Marllory & ORTIZ, Mayra 2017

Listados de datos.- Para verificar su funcionamiento se muestra en la *Figura 13-3* el listado de pacientes con historias clínicas al que tiene acceso el médico. Cada listado existente en la página permite modificar y eliminar los datos ingresados por medio de los botones existentes a su derecha, como resultado de las pruebas realizadas se obtuvo un total de 30 personas ingresadas a la base de datos con sus correspondientes historias clínicas y datos personales.

Identificate Ecuador

LISTA DE HISTORIAS CLÍNICAS INGRESADAS

[Regresar]

# Historia Clínica	Nombres	Apellidos	Fecha de consulta	Opciones
3	Mayra Estefania	Ortiz Jimenez	2017-02-01 16:02:03	Habitos tóxicos Antecedentes familiares Datos patológicos Habitos tóxicos
6	Vladimir Alejandro	Cobos Maldonado	2017-02-02 16:38:42	Antecedentes familiares Datos patológicos Habitos tóxicos
7	Galo Patricio	Hurtado Crespo	2017-02-01 21:15:26	Antecedentes familiares Datos patológicos

Figura 13-3: Listado de Historias Clínicas – Eliminación y modificación.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

Funcionamiento del prototipo

En la *Figura 14-3* se muestran las partes que conforman el prototipo implementado. Se probó todo el funcionamiento y se comprueba que el sistema funciona adecuadamente y cumple con los requerimientos establecidos para la implementación.

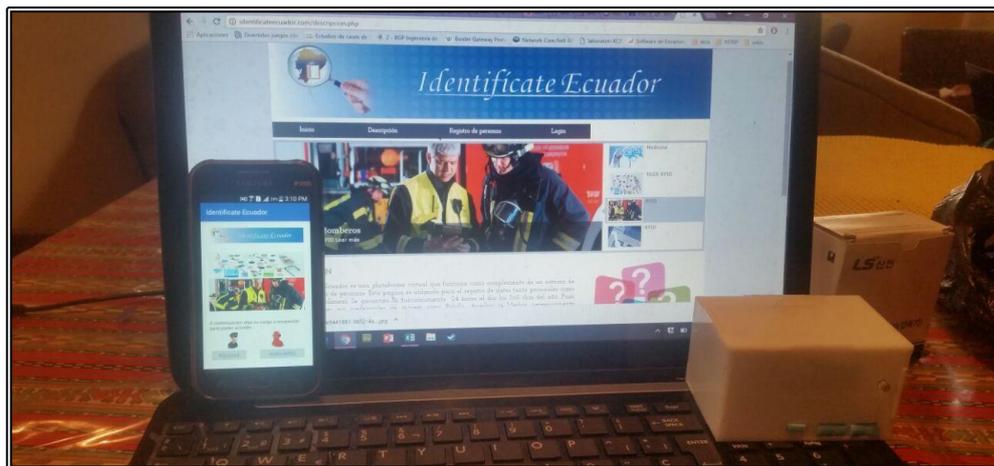


Figura 14-3: Partes del prototipo implementado.
Fuente: COBOS, Marllory & ORTIZ, Mayra 2017.

3.3. *Análisis económico del prototipo.*

En esta sección se presenta la parte económica, gastos y costos del prototipo implementado tanto en *hardware* como en *software*. En las *Tabla 4-3* se aprecia el presupuesto del prototipo implementado.

Tabla 4-3: Presupuesto del prototipo implementado.

	DISPOSITIVO	CANT.	COSTO UNITARIO	COSTO TOTAL
HARDWARE	Tarjeta Arduino Pro mini	1	7,00	7,00
	Módulo RFID	1	17,00	17,00
	tag RFID (llaveros)	11	0,50	5,50
	tag RFID (pulseras)	3	0,80	2,40
	tag RFID (tarjetas)	2	1,00	2,00
	Batería	1	25,00	25,00
	Led	1	0,10	0,10
	Bluetooth HC-06	1	10,50	10,50
	Regulador de Voltaje 3,3V	1	0,55	0,55
	Ensamblaje	1	20,00	20,00
SOFTWARE	Hosting - Dominio Web por un año	1	50,00	50,00
TOTAL				140,05

Fuente: COBOS, Marlory & ORTIZ, Mayra 201

De los valores obtenidos en la *Tabla 3-3*, se puede apreciar que la parte *hardware* del dispositivo tuvo un costo de con 90,05 dólares americanos (USD) , mientras que la parte *software* tiene un costo anual de 50,00 dólares americanos (USD), dando un total de 140,05 dólares americanos (USD). Del análisis realizado se puede determinar que el prototipo implementado es de bajo costo y cumple con los objetivos planteados en esta investigación.

CONCLUSIONES

1. Se implementó un prototipo de una red inalámbrica de sensores, cuya operatividad se basa en tres nodos denominados llave, lector, y de almacenamiento que trabaja en una base de datos ubicada en la web, que en conjunto permiten acceder a datos personales e historias clínicas. Su operatividad esta respaldada por el 96.98% de cobertura de servicio móvil presente en el país.
2. Después de la respectiva comparativa y análisis de las tecnologías existentes para la identificación, se eligió el sistema de identificación RFID que trabaja a una frecuencia de 13,56 MHz, debido a su corta distancia para la lectura, compatibilidad con los microprocesadores, bajo costo y alta durabilidad de las etiquetas, características que se ajustan a los requerimientos del prototipo.
3. Luego del análisis se determinó que la comunicación apropiada para el funcionamiento del prototipo es la convergencia entre Wifi, Bluetooth y Radio Frecuencia que cumplen con los requerimientos de conexión entre los nodos del prototipo.
4. Se seleccionó como elemento de procesamiento del módulo lector la tarjeta de desarrollo Arduino Pro-mini por su reducido tamaño, facilidad de manejo y bajo costo.
5. La plataforma web trabaja en un entorno gráfico amigable, sirve para la gestión de la información de usuarios e identificar personal de apoyo por medio del administrador. La plataforma móvil desarrollada es exclusiva de consulta para el personal de apoyo y permite el acceso a la información que se encuentra en la Web.
6. Las pruebas de comunicación del módulo lector implementado permitieron determinar que la conexión del Bluetooth HC-06 es hasta distancias de 26 metros con línea de vista directa sin pérdidas, valor que se acopla a los objetivos propuestos.
7. De las pruebas sobre el tiempo de respuesta del lector RFID, se hicieron tomando en cuenta dos tipos de red inalámbrica, conexión wifi y datos móviles, se determinó tiempos de respuesta es de 1.4 y 3.5 segundos, respectivamente, que constituyen tiempos de respuesta óptimos para esta aplicación.

8. Después de las mediciones realizadas para comprobar la obtención de datos dentro de la aplicación se comprobó que el tiempo de acceso a la información personal es de 3 segundos y 3.8 a datos clínicos con un número de 30 registros. Considerando que el tiempo es directamente proporcional a la cantidad de información que se solicita a la base de datos.
9. En lo que se refiere al consumo de energía del prototipo, al realizar las pruebas se determinó que requiere de 4 horas y 18 minutos para cargar su batería lo que garantiza 20 horas de funcionamiento. Valor que se ajusta al propósito por el cual fue creado el prototipo.

RECOMENDACIONES

1. Se recomienda que para implementaciones futuras se realice la selección de elementos electrónicos de manera que sea posible la miniaturización de módulo lector para una mejor manipulación del mismo.
2. Para mayor seguridad de la información se recomienda utilizar etiquetas a manera de chips subcutáneos para evitar pérdidas y proteger la integridad de la información de las personas.
3. Se recomienda utilizar nuevas tecnologías con paneles solares de la alimentación de voltaje para mejorar la eficiencia energética del dispositivo.
4. Ampliar las funcionalidades del prototipo con la finalidad de que el usuario tenga acceso a distinta información de una manera sencilla.

BIBLIOGRAFÍA

ADOBE SYSTEMS INCORPORATED. *Manual Uso de ADOBE DREAMWEAVER CS5.* California: Adobe Systems Incorporated, 2010, pp.60

AGENCIA NACIONAL DE TRÁNSITO. *Siniestros por provincia a nivel nacional.* [En línea] 21 de Mayo de 2016. [Consulta: 20 diciembre 2016]. Disponible en: <http://www.ant.gob.ec/index.php/descargable/file/3663-siniestros-mayo-2016>.

ALMONACID, Luis Antonio. *Estudio de factibilidad para Implementar Tecnología Rfid en Biblioteca Miraflores (Trabajo de titulación)(Ing.Electrónico)(Facultad de Ciencias de la Ingeniería).* Valdivia: Universidad Austral de Chile, 2007. pp.2-3. Disponible en: <http://cybertesis.uach.cl/tesis/uach/2007/bmfcia452e/doc/bmfcia452e.pdf>.

ALVARADO, Jorge Alberto. *Sistema de Control de Acceso con RFID (Tesis)(MS) (Departamento de Ingeniería Eléctrica Sección de Computación).* Mexico, 2008, pp. 1-2. Disponible en: <https://www.cs.cinvestav.mx/TesisGraduados/2008/tesisJorgeAlvarado.pdf>

ANDROID STUDIO. Conoce Android Studio. *Developer Android.* [En línea] 2016. [Consulta: 10 enero 2017]. Disponible en: <https://developer.android.com/studio/intro/index.html?hl=es-419>.

ARDUINO. Acordeon Arduino. [En línea] 2016 [Consulta: 15 enero 2017]. Disponible en: <https://www.arduino.cc/en/Reference/HomePage>.

ARDUINO PRODUCTS. *Arduino Pro Mini.* [En línea] 2017. [Consulta: 20 febrero 2017]. Disponible en: <https://www.arduino.cc/en/Main/ArduinoBoardProMini>.

BELLIDO, Francisco, et al, Técnica Industrial. *Comunicación Inalámbrica por Bluetooth.* [En línea] Octubre de 2004. [Consulta: 10 enero 2017]. Disponible en: <http://www.tecnicaindustrial.es/tifrontal/a-1434-Comunicacion-inalambrica-Bluetooth.aspx>.

BOT SCIENCE. Bot Science - Imagina, diseña, construye. *Módulo Bluetooth Slave (HC-06).* [En línea] 2015. [Consulta: 10 diciembre 2016]. Disponible en: http://botscience.net/store/index.php?route=product/product&product_id=69.

CARRERA, Isacc. Identificación de Cadáveres y Aspectos Forenses de los desastres. *Publicaciones de la Unidad de Investigación en Emergencias y Desastres.* [En línea] 2011. [Consulta: 23 enero 2017]. Disponible en:

https://www.administraciondejusticia.gob.es/paj/PA_WebApp_SGNTJ_NPAJ/descarga/Identificacion%20de%20cadaveres.pdf?idFile=a69bcd77-cfa7-4e9a-bdbf-b95e96c6c4f6.

CIUDAD, José María & SAMÁ, Eduard. *ESTUDIO, DISEÑO Y SIMULACIÓN DE UN SISTEMA DE RFID BASADO EN EPC*. Barcelona : Universidad Politécnica de Cataluña, 2005. pp.10

CÓDIGO DE BARRAS, España. Código de Barras España. *Imágenes de Muestra*. [En línea] 2017. [Consulta: 16 febrero 2017]. Disponible en: <https://codigos-de-barras.es/imagenes-de-muestra/>.

DIARIO EL TELÉGRAFO. *Ecuador busca pulir protocolos para la gestión de cadáveres en catástrofes*. [En línea] 09 de Diciembre de 2013. [Consulta: 10 enero 2017]. Disponible en: <http://www.eltelegrafo.com.ec/noticias/judicial/13/ecuador-busca-pulir-protocolos-para-afrontar-desastres-naturales>.

DOUTEL, Fernando. Guía del Arduinomaníaco: todo lo que necesitas saber sobre Arduino. *Xataka*. [En línea] 18 de Agosto de 2015. [Consulta: 22 enero 2017]. Disponible en: <https://www.xataka.com/especiales/guia-del-arduinomaniaco-todo-lo-que-necesitas-saber-sobre-arduino>.

ELECTRONICOS CALDAS. HC-06. *electronicoscaldas*. [En línea] 2016. [Consulta: 22 enero 2017]. Disponible en: <http://www.electronicoscaldas.com/modulos-rf/482-modulo-bluetooth-hc-06.html>.

ELECTRONILAB. Módulo RFID RC522 – 13.56Mhz. *ELECTRONICLAB Ingeniería y Diseño Electrónico*. [En línea] 2016. [Consulta: 10 enero 2017]. Disponible en: <https://electronilab.co/tienda/modulo-rfid-rc522-13-56mhz-rc522/>.

ESPACIOPYME. Cambrasa badel. *HOSTING*. [En línea] Julio de 2004. [Consulta: 01 enero 2017]. Disponible en: http://www.cambrasabadell.org/Att/doc804_1_27_04012005094829.pdf.

ESPECIALISTAS HOSTING. 2016. Especialistas Hosting. *Protocolo FTP*. [En línea] 16 de Diciembre de 2016. <http://www.especialistashosting.com/blog/index.php/2016/12/protocolo-ftp/>.

ESPINOSA, Virginia. Evaluación de Sistemas de Reconocimiento. *Escuela Universitaria Politécnica de Mataró*. [En línea] 2001. [Consulta: 05 enero 2017]. Disponible en: <http://www.jcee.upc.es/JCEE2001/PDFs%202000/13ESPINOSA.pdf>.

FÁBREGAS, Ferran. *Aprender Raspberry Pi con 100 ejercicios prácticos.* s.l. : Marcombo, 2015, pp.10-15.

FERNÁNDEZ, Roberto et al. *REDES INALÁMBRICAS DE SENSORES: teoría y aplicación práctica.* Rioja : UNIVERSIDAD DE LA RIOJA , 2009. pp.20-25.

FERNÁNDEZ, Sadot, MORCILLO, Carlos & MUÑOZ, José. RFID: La tecnología de identificación por radiofrecuencia. *Universidad Pontificia Comillas.* [En línea] 1 de enero de 2006. [Consulta: 22 diciembre 2016]. Disponible en:<https://repositorio.comillas.edu/xmlui/bitstream/handle/11531/5321/IIT-05-035A.pdf?sequence=1&isAllowed=y>.

FLORES, Esther. *Redes de Sensores Inalámbricas aplicado a la medicina (Trabajo de titulación)(MS)(Escuela de Técnica Superior de Ingenieros Industriales y de Telecomunicaciones).* Cantabria : Escuela Técnica Superior de Ingenieros Industriales y de Telecomunicación, 2012. pp.22-23 Disponible en: <https://repositorio.unican.es/xmlui/bitstream/handle/10902/1288/349251.pdf?sequence=1>

FUNDACIÓN TELEFÓNICA. *Las TIC en la Administración Local del futuro.* Madrid : Ariel y Fundación Telefónica, 2008. pp.135

GARCÍA, Carlos. *Integración de Redes Telemáticas IoT con Raspberry Pi.* : Universitat Oberta de Catalunya, 2015. pp.20

IENK. 2013. L78L33 DC 3.3V TO-92. *Ienk.* [En línea] 15 de Abril de 2013. [Consulta: 10 enero 2017]. Disponible en: http://www.ienk.com/document_news_info.html?products_id=3824.

POMEYROL, Jorge. Raspberry Pi 2: más potencia al precio de siempre. *eTech.* [En línea] muyLinux, 02 de Febrero de 2015. [Consulta: 06 febrero 2017]. Disponible en: <http://www.muylinux.com/2015/02/02/raspberry-pi-2>.

JARAMILLO, Mario et al. *Aplicación de Redes de Sensores Inalámbricos (Wsn) en un sistema de seguridad para los equipos móviles de la Universidad Eafit.(tesis)(Ing. Sistemas)(Departamento de Informática y Sistemas)* Medellín : UNIVERSIDAD EAFIT, facultad y escuela 2011. pp.15-16. Disponible en: https://repository.eafit.edu.co/bitstream/handle/10784/2397/JaramilloJaramillo_Mario_2011.pdf?sequence=1

JURADO, Daniel & SALAZAR, Byron. *Diseño y Construcción de un sistema prototipo para el Control de Vehículos por medio de dispositivos tag con identificación Rfid (Radio Frequency Identification) para la Dirección Nacional de Control del Tránsito y seguridad vial de la Policía*

Nacional (tesis)(Ing. Electrónica y Telecomunicaciones)(Facultad de Ingeniería Eléctrica y Electrónica). Quito : Escuela Politécnica Nacional, 2013. pp.45. Disponible en: <http://bibdigital.epn.edu.ec/bitstream/15000/5805/1/CD-4707.pdf>

MANZANO, Hector & TAPIA, Carlos. 2013. *Evaluación de la Plataforma Arduino e Implementación de un Sistema de Control de Posición Horizontal (tesis)(Ing. Electrónico mención Sistemas Computacionales)(Facultad de Ingenierías)*. Guayaquil : UNIVERSIDAD POLITÉCNICA SALESIANA Sede Guayaquil, 2013. pp.25-28. Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/5522/1/UPS-GT000511.pdf>

MAROTO, Sebastián. *Desarrollo de Aplicaciones basadas en WSN*. Valencia : Universidad Politécnica de Valencia, 2010. pp.35-37.

MATURANA, Jesus. Xataka. *Intel Galileo, placa de desarrollo fruto de la colaboración entre Intel y Arduino*. [En línea] 03 de Octubre de 2013. [Consulta: 23 febrero 2017]. Disponible en: <https://www.xataka.com/makers/intel-galileo-placa-de-desarrollo-fruto-de-la-colaboracion-entre-intel-y-arduino>.

MAYNÉ, Jordi. *Estado actual de las Comunicaciones por Radio Frecuencia*. s.l. : Silica, 2009, pp.12-19.

MINISTERIO DE SALUD PÚBLICA. *Ministerio de Salud Pública garantiza atención y cobertura de gastos a heridos tras accidente en Papallacta*. [En línea] 24 de Abril de 2016. [Consulta: 25 noviembre 2016]. Disponible en: <http://www.salud.gob.ec/ministerio-de-salud-publica-garantiza-atencion-y-cobertura-de-gastos-a-heridos-tras-accidente-en-papallacta/>.

MINISTERIO DE TELECOMUNICACIONES Y DE LA SOCIEDAD DE LA INFORMACIÓN . Ministerio de Telecomunicaciones y de la Sociedad de la Información . *Trabajamos para llevar más y mejores servicios móviles al país*. [En línea] 29 de Julio de 2016. [Consulta: 16 febrero 2017]. Disponible en: <https://www.telecomunicaciones.gob.ec/trabajamos-para-llevar-mas-y-mejores-servicios-moviles-al-pais/>.

MONZON, Ariel. Tecnología, ¿Qué es la RFID? *LA TECNOLOGÍA RFID Y SU APLICACIÓN EN LA CADENA DE SUMINISTRO*. [En línea] 16 de SEPTIEMBRE de 2014. [Consulta: 23 febrero 2017]. Disponible en: <http://logistica-sencilla.blogspot.com/2014/09/tecnologiaque-es-la-rfid.html>.

NATIONAL INSTRUMENTS. National Instruments. [En línea] 22 de Abril de 2009. [Consulta: 03 enero 2017]. Disponible en: <http://www.ni.com/white-paper/7142/es/>.

ORACLE. Oracle. *Oracle MySQL*. [En línea] 2017. [Consulta: 23 febrero 2017]. Disponible en: <https://www.oracle.com/lad/mysql/index.html>.

ORGANIZACIÓN MUNDIAL DE LA SALUD. Organización Mundial de la Salud. *Lesiones causadas por el tránsito*. [En línea] Mayo de 2016. [Consulta: 05 febrero 2017]. Disponible en: <http://www.who.int/mediacentre/factsheets/fs358/es/>.

PHP MY ADMIN. Bringing MySQL to the web. *Phpmyadmin*. [En línea] 2017. [Consulta: 15 diciembre 2016]. Disponible en: <https://www.phpmyadmin.net/>.

RAMÍREZ, Carlos Lino. *Diseño de una arquitectura para redes de sensores con soporte para aplicaciones de detección de eventos (tesis)(Doctorado)(Departamento de Informática de Sistemas y Computadores)*. Valencia : Universidad Politécnica de Valencia, 2012. pp.19. Disponibles en: <https://riunet.upv.es/bitstream/handle/10251/15152/tesisUPV3764.pdf>

ROLDÁN, Sandra. MEDICACIÓN SEGURA. *ORGANIZACIÓN MUNDIAL DE LA SALUD*. [En línea] 2001. [Consulta: 23 febrero 2017]. Disponible en: http://apps.who.int/patientsafety/patients_for_patient/pfps_webinar2_Sandra.pdf.

SECRETARÍA DE GESTIÓN DE RIESGOS. Secretaría de Gestión de Riesgos. *Informes de Situación: Terremoto 7.8 Pedernales / Réplicas Esmeraldas*. [En línea] 19 de Mayo de 2016. [Consulta: 23 noviembre 2016]. Disponible en: <http://www.gestionderiesgos.gob.ec/informes-de-situacion-actual-terremoto-magnitud-7-8/>.

SPARKFUN. Arduino Pro Mini 328 - 3.3V/8MHz. *Sparkfun*. [En línea] 2017. [Consulta: 03 febrero 2017]. Disponible en: <https://www.sparkfun.com/products/11114>.

SUNFOUNDER. 3800mAh 5V/1.8A Lithium Battery Power Pack Expansion Board for Raspberry Pi. *Sunfounder*. [En línea] 2017. [Consulta: 20 enero 2017]. Disponible en: <https://www.sunfounder.com/battery-power-pack.html>.

SWHOSTING. *¿Qué son las DNS y los registros de dominios?* [En línea] 28 de Noviembre de 2014. [Consulta: 06 diciembre 2016]. Disponible en: <http://www.swhosting.com/blog/que-son-las-dns-y-los-registros-de-dominios/>.

TESLABEM. LM7833, LD33VC Regulado de voltaje de 3.3V 800mA. *Teslabem*. [En línea] [Consulta: 23 febrero 2017]. Disponible en: 2016. <http://teslabem.com/lm7833-ld33vc-3-3v-4-7v-1a.html>.

YAAGOUBI, Mohammed El. 2012. *Acceso a Internet vía WiFiWiMax.* Leganés : Universidad Carlos III de Madrid, 2012. pp. 14-15

YÉPEZ, Juan & CAMPAÑA, Víctor. 2016. *Estudio de factibilidad de una red de sensores inalámbricas para la observación y el monitoreo de la calidad del agua (Proyecto de Titulación)(Ing. Networking y Telecomunicaciones)(Facultad de Ciencias Matemáticas y Físicas).* Guayaquil : UNIVERSIDAD DE GUAYAQUIL, 2016. p.30. Disponible en: <http://repositorio.ug.edu.ec/bitstream/redug/11917/1/MIGUEL.pdf>

ZAMUDIO, José Fabián. *La inteligencia de una tarjeta.* 2015, Kalysis Community - UNAM. pp. 20.

ANEXOS

ANEXO A CÓDIGO ARDUINO

```
#include "SPI.h"
#include "MFRC522.h"

#define SS_PIN 10
#define RST_PIN 9
#define LED_PIN A0
#define LED_PIN2 A1

MFRC522 rfid(SS_PIN, RST_PIN);

MFRC522::MIFARE_Key key;

void setup() {
  pinMode(2,OUTPUT);
  Serial.begin(9600);
  SPI.begin();
  rfid.PCD_Init();
  pinMode(LED_PIN, OUTPUT);
  pinMode(LED_PIN2, OUTPUT);
}

void loop() {
  if (!rfid.PICC_IsNewCardPresent() || !rfid.PICC_ReadCardSerial())
    return;

  // Serial.print(F("PICC type: "));
  MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);
  // Serial.println(rfid.PICC_GetTypeName(piccType));

  // Check is the PICC of Classic MIFARE type
  if (piccType != MFRC522::PICC_TYPE_MIFARE_MINI &&
      piccType != MFRC522::PICC_TYPE_MIFARE_1K &&
      piccType != MFRC522::PICC_TYPE_MIFARE_4K) {
    Serial.println(F("Your tag is not of type MIFARE Classic."));
    return;
  }

  String strID = "";
  for (byte i = 0; i < 4; i++) {
    strID +=
      (rfid.uid.uidByte[i] < 0x10 ? "0" : "") +
      String(rfid.uid.uidByte[i], HEX) +
      (i!=3 ? ":" : "");
    digitalWrite(2, HIGH);
    delay(100);
    digitalWrite(2, LOW);
  }
}
```

```

    strID.toUpperCase();
    Serial.println(strID);

// encendida de leds

    rfid.PICC_HaltA();
    rfid.PCD_StopCrypto1();
}

```

ANEXO B. CÓDIGO ANDROID STUDIO

MainActivity.java

```

package com.identificatecuador.tesis2;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

    Button siguiente1;
    Button siguiente2;
    ImageView imageView4 , imageView5;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        siguiente1 = (Button) findViewById(R.id.btn1);
        siguiente2 = (Button) findViewById(R.id.btn2);
        imageView4 = (ImageView) findViewById(R.id.imageView4);
        imageView5 = (ImageView) findViewById(R.id.imageView5);

        siguiente1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent siguiente1 = new Intent(MainActivity.this,
Main2Activity.class);
                startActivity(siguiente1);
            }
        });

        siguiente2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent siguiente2 = new Intent(MainActivity.this,
Main22Activity.class);
                startActivity(siguiente2);
            }
        });

        imageView4.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent imageView4 = new Intent(MainActivity.this,

```



```

        @Override
        public void run() {
            final String resultado =
enviarDatosGET(txtUsu.getText().toString(),
txtPas.getText().toString(),txtNivel.getText().toString());
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    int r = obtDatosJSON(resultado);
                    if (r > 0) {
                        Intent i = new
Intent(getApplicationContext(), registro1.class);
                        i.putExtra("cod",
txtUsu.getText().toString());
                        startActivity(i);
                    } else {
                        Toast.makeText(getApplicationContext(),
"Usuario o Pas Incorrectos", Toast.LENGTH_LONG).show();
                    }
                }
            });
        }
    };
    tr.start();
}

public String enviarDatosGET(String usu, String pas, String nivel
) {
    URL url = null;
    String linea = "";
    int respuesta = 0;
    StringBuilder resul = null;

    try {
        url = new
URL("http://www.identificatecuador.com/valida.php?usu=" + usu +
"&pas=" + pas + "&nivel="+nivel );
        HttpURLConnection conection = (HttpURLConnection)
url.openConnection();
        respuesta = conection.getResponseCode();

        resul = new StringBuilder();

        if (respuesta == HttpURLConnection.HTTP_OK) {
            InputStream in = new
BufferedInputStream(conection.getInputStream());
            BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

            while ((linea = reader.readLine()) != null) {
                resul.append(linea);
            }
        }
        catch (Exception e) {
        }
        return resul.toString();
    }

public int obtDatosJSON(String response) {
    int res = 0;
    try {

```

```

        JSONArray json = new JSONArray(response);
        if (json.length() > 0) {
            res = 1;
        }
    } catch (Exception e) {}
    return res;
}
}

```

Main22Activity.java

```

package com.identificateecuador.tesis2;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

import org.json.JSONArray;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;

import javax.net.ssl.HttpURLConnection;

public class Main22Activity extends AppCompatActivity implements
View.OnClickListener {

    Button btnIngresar;
    EditText txtUsu, txtPas, textNivel;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main22);

        txtUsu = (EditText) findViewById(R.id.txtusu);
        txtPas = (EditText) findViewById(R.id.txtpas);
        btnIngresar = (Button) findViewById(R.id.btnIngresar);
        btnIngresar.setOnClickListener(this);
        textNivel = (EditText) findViewById(R.id.textnivel);
    }

    @Override
    public void onClick(View v) {
        Thread tr = new Thread() {
            @Override
            public void run() {
                final String resultado =
enviarDatosGET(txtUsu.getText().toString(),
txtPas.getText().toString(), textNivel.getText().toString());
                runOnUiThread(new Runnable() {
                    @Override

```

```

        public void run() {
            int r = obtDatosJSON(resultado);
            if (r > 0) {
                Intent i = new
Intent(getApplicationContext(), registro2.class);
                i.putExtra("cod",
txtUsu.getText().toString());
                startActivity(i);
            } else {
                Toast.makeText(getApplicationContext(),
"Usuario o Pas Incorrectos", Toast.LENGTH_LONG).show();
            }
        }
    });
}
};
tr.start();
}

public String enviarDatosGET(String usu, String pas, String nivel
) {
    URL url = null;
    String linea = "";
    int respuesta = 0;
    StringBuilder resul = null;

    try {
        url = new
URL("http://www.identificatecuador.com/valida.php?usu=" + usu +
"&pas=" + pas + "&nivel="+nivel );
        HttpURLConnection conection = (HttpURLConnection)
url.openConnection();
        respuesta = conection.getResponseCode();

        resul = new StringBuilder();

        if (respuesta == HttpURLConnection.HTTP_OK) {
            InputStream in = new
BufferedInputStream(conection.getInputStream());
            BufferedReader reader = new BufferedReader(new
InputStreamReader(in));

            while ((linea = reader.readLine()) != null) {
                resul.append(linea);
            }
        }
    } catch (Exception e) {
    }

    return resul.toString();
}

public int obtDatosJSON(String response) {
    int res = 0;
    try {
        JSONArray json = new JSONArray(response);
        if (json.length() > 0) {
            res = 1;
        }
    } catch (Exception e) {}
    return res;
}

```

```

    }
}
ListaDispositivos.java
package com.identificatecuador.tesis2;

import android.app.ListActivity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.Set;

public class ListaDispositivos extends ListActivity {

    private BluetoothAdapter meuBluetoothAdapter2 = null;

    static String DIRECCION_MAC = null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        ArrayAdapter<String> ArrayBluetooth = new
        ArrayAdapter<String>(this, android.R.layout.simple_list_item_1);

        meuBluetoothAdapter2 = BluetoothAdapter.getDefaultAdapter();

        Set<BluetoothDevice> dispositivosPareados =
        meuBluetoothAdapter2.getBondedDevices();

        if(dispositivosPareados.size() > 0){
            for(BluetoothDevice dispositivo: dispositivosPareados){
                String nomeBt=dispositivo.getName();
                String macBT=dispositivo.getAddress();
                ArrayBluetooth.add(nomeBt + "\n" + macBT);
            }
        }
        setListAdapter(ArrayBluetooth);
    }

    @Override
    protected void onItemClick(ListView l, View v, int position,
    long id) {
        super.onItemClick(l, v, position, id);

        String InformacionGeral= ((TextView)v).getText().toString();

        //Toast.makeText(getApplicationContext(),"Info: " +
        InformacionGeral, Toast.LENGTH_LONG).show();

        String direccionmac =
        InformacionGeral.substring(InformacionGeral.length()-17);
        //Toast.makeText(getApplicationContext(),"mac: " +
        direccionmac, Toast.LENGTH_LONG).show();
    }
}

```

```

        Intent retornaMAC = new Intent();
        retornaMAC.putExtra(DIRECCION_MAC, direccionmac );
        setResult(RESULT_OK, retornaMAC);
        finish();
    }
}
registro1.java
package com.identificateecuador.tesis2;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.loopj.android.http.HttpGet;

import org.json.JSONArray;
import org.json.JSONObject;

import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.UUID;

import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.HttpClient;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.util.EntityUtils;

public class registrol extends AppCompatActivity {

    Button button, buscar, borrar ;
    ImageView foto;
    TextView info, textView5, textView6, textView7, textView52,
textView21, textView9, textView10, textView11, textView12, textView13,
textView14, textView15, textView16, textView17, textView18,
textView19;
    TextView textView8;

    private static final int SOLICITA_ACTIVACION = 1;
    private static final int SOLICITA_CONEXION = 2;
    private static final int SOLICITA_DESCONEXION = 3;
    private InputStream inputStream;

```



```

        meuBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

        if(meuBluetoothAdapter == null) {
            Toast.makeText(getApplicationContext(), "Su dispositivo no
posee bluetooth", Toast.LENGTH_LONG).show();
        }else if(!meuBluetoothAdapter.isEnabled()) {
            Intent ativaBluetooth = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult( ativaBluetooth ,
SOLICITA_ACTIVACION );
        }

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (conexionb) {

                    //desconectar

                    try {

                        conexionb = false;
                        meuBluetoothAdapter.disable();
                        meuSocket =
meuDevice.createRfcommSocketToServiceRecord(MI_UUID);
                        meuSocket.close();
                        button.setText("Conectar");
                        Toast.makeText(getApplicationContext(),
"Bluetooth desconectado", Toast.LENGTH_LONG).show();
                        meuBluetoothAdapter.enable();

                    } catch (IOException error) {
                        Toast.makeText(getApplicationContext(),
"OCURRIO UN ERROR:" + error, Toast.LENGTH_LONG).show();
                        error.printStackTrace();
                    }

                } else {
                    // conectar
                    Intent abreLista = new Intent(registro1.this,
ListaDispositivos.class);
                    startActivityForResult(abreLista,
SOLICITA_CONEXION );
                }
            }
        });

        //-----BORRAR -----

        public void onClear (View v){
            editText1.setText("");
            info.setText("Presione el boton BUSCAR para visualizar los
datos personales de la persona");

        }

        //-----
REALIZAR LA BUSQUEDA EN LA BASE DE DATOS-----

```

```

    private class ObtenerContacto extends AsyncTask<String, Void,
Void> {

        String oNames , oApellidos, oGenero, oCedula, oNacimiento,
oCiudad, oPais, oDireccion, oOcupacion,
                oContactouno, oContactodos, oEmail, oLugartrabajo,
oTelefonofijo, oTelefonomovil, oFoto , oEstadoCivil;

        Bitmap bitmap;

        @Override
        protected Void doInBackground(String... params) {

            Log.i("ConsultaContacto", "doInBackground");
            HttpClient httpClient = new DefaultHttpClient();

            String sCodigo = params[0];

            HttpGet get = new
HttpGet("http://www.identificatecuador.com/editar1.php?Codigo=" +
sCodigo);
            get.setHeader("Content-type", "application/json");

            try {
                HttpResponse resp= httpClient.execute(get);
                String respString =
EntityUtils.toString(resp.getEntity());

                JSONArray respJSON = new JSONArray(respString);

                for(int i =0; i< respJSON.length() ; i++) {
                    JSONObject jsonObject = respJSON.getJSONObject(i);

                    oNames = jsonObject.getString("Nombres");
                    oApellidos = jsonObject.getString("Apellidos");
                    oGenero = jsonObject.getString("Genero");
                    oCedula = jsonObject.getString("Num_Cedula");
                    oEstadoCivil =
jsonObject.getString("Estado_Civil");
                    oNacimiento =
jsonObject.getString("Fecha_Nacimiento");
                    oCiudad = jsonObject.getString("Cuidad");
                    oPais = jsonObject.getString("Pais");
                    oDireccion = jsonObject.getString("Direccion");
                    oOcupacion = jsonObject.getString("Ocupacion");
                    oContactouno =
jsonObject.getString("Num_Contacto1");
                    oContactodos =
jsonObject.getString("Num_Contacto2");
                    oEmail = jsonObject.getString("Email");
                    oLugartrabajo =
jsonObject.getString("Lugar_Trabajo");
                    oTelefonofijo = jsonObject.getString("Telef_Fijo")
;

                    oTelefonomovil =
jsonObject.getString("Telef_Movil");
                    oFoto = jsonObject.getString("imagen");

                    if (oFoto.equals("noimagen")){

                        }else

```

```

        {
            URL urlimagen= new
URL("http://www.identificatecuador.com/imagenes/personas/" + oFoto);
            HttpURLConnection conimagen =
(HttpURLConnection) urlimagen.openConnection();
            conimagen.connect();
            bitmap =
BitmapFactory.decodeStream(conimagen.getInputStream());
        }
    }

    } catch (Exception ex) {
        Log.e("ServicioRest", "Error!", ex);
        ex.printStackTrace();
    }

    return null;
}

private class DescargarImagen extends AsyncTask<ImageView,
Void, Bitmap>{
    ImageView foto;
    Bitmap bitm;
    @Override
    protected Bitmap doInBackground(ImageView... params) {
        foto= params[0];
        try {
            URL imageURL = new
URL("http://www.identificatecuador.com/imagen/personas/" + oFoto);
            HttpURLConnection conn= (HttpURLConnection)
imageURL.openConnection();
            conn.connect();
            bitm =
BitmapFactory.decodeStream(conn.getInputStream());

            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        return bitm;
    }
}

@Override
protected void onPostExecute(Void result){
    Log.i("ServicioRest", "onPostExecute");
    textView5.setText("NOMBRES: " + oNames);
    textView6.setText("APELLIDOS: " + oApellidos);
    textView7.setText("GÉNERO: " + oGenero);
    textView21.setText("CÉDULA: " + oCedula);
    textView52.setText("ESTADO CIVIL: " + oEstadoCivil);
    textView9.setText("FECHA DE NACIMIENTO: " + oNacimiento);
    textView10.setText("CIUDAD: " + oCiudad);
    textView11.setText("PAÍS: " + oPais);
    textView12.setText("DIRECCIÓN: " + oDireccion);
    textView13.setText("OCUPACIÓN: " + oOcupacion);
    textView14.setText("CONTACTO 1: " + oContactouno);
    textView15.setText("CONTACTO 2: " + oContactodos);
    textView16.setText("EMAIL: " + oEmail);
    textView17.setText("LUGAR DE TRABAJO: " + oLugartrabajo);
    textView18.setText("TELÉFONO FIJO: " + oTelefonofijo);
}

```

```

        textView19.setText("NÚMERO DE CELULAR: " +
oTelefonomovil);
        foto.setImageBitmap(bitmap);
    }

    @Override
    protected void onPreExecute() {
        Log.i("ServicioRest", "onPreExecute");
        info.setText("Se está obteniendo la información . . .");
    }
}

//-----
-----

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch (requestCode) {
            //-----ACTIVAR BLUETOOTH
AL INGRESAR A LA APLICACIÓN-----
            -----

            case SOLICITA_ACTIVACION:
                if(resultCode == Activity.RESULT_OK){
                    Toast.makeText(getApplicationContext(),"Bluetooth
fue activado",Toast.LENGTH_LONG ).show();
                }else {
                    Toast.makeText(getApplicationContext(),"Bluetooth
no esta habilitado, la aplicación se cerrará",Toast.LENGTH_LONG
).show();

                    finish();
                }
                break;
            //-----
            -----

            case SOLICITA_CONEXION:
                if(resultCode == Activity.RESULT_OK){

MAC=data.getExtras().getString(ListaDispositivos.DIRECCION_MAC);

                meuDevice =
meuBluetoothAdapter.getRemoteDevice(MAC);

                try {
                    meuSocket =
meuDevice.createRfcommSocketToServiceRecord(MI_UUID);
                    meuSocket.connect();
                    conexionb=true;
                    boton.setText("Desconectar");
                    Toast.makeText(getApplicationContext(),"Fue
conectado con:"+MAC,Toast.LENGTH_LONG ).show();
                    inputStream=meuSocket.getInputStream();
                    beginListenForData();

                }catch (IOException error){
                    //conexionb=false;

                Toast.makeText(getApplicationContext(),"OCURRIO UN

```

```

ERROR: "+error,Toast.LENGTH_LONG ).show();
    }

    }else{
        Toast.makeText(getApplicationContext(),"No se pudo
obtener la MAC",Toast.LENGTH_LONG ).show();

    }
}

//-----LEER CODIGO RFID -----
-----

void beginListenForData()
{
    final Handler handler = new Handler();
    stopThread = false;
    buffer = new byte[1024];
    Thread thread = new Thread(new Runnable()
    {
        public void run()
        {
            while(!Thread.currentThread().isInterrupted() &&
!stopThread)
            {
                try
                {
                    int byteCount = inputStream.available();
                    if(byteCount > 0)
                    {
                        byte[] rawBytes = new byte[byteCount];
                        inputStream.read(rawBytes);
                        final String Codigo=new
String(rawBytes,"UTF-8");
                        handler.post(new Runnable() {
                            public void run()
                            {
                                editText1.append(Codigo);
                            }
                        });
                    }
                }
                catch (IOException ex)
                {
                    stopThread = true;
                }
            }
        }
    });

    thread.start();
}

//-----
-----

registro2.java
package com.identificatecuador.tesis2;

import android.app.Activity;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothSocket;

```

```

import android.content.Intent;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

import com.loopj.android.http.HttpGet;

import org.json.JSONArray;
import org.json.JSONObject;

import java.io.IOException;
import java.io.InputStream;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.UUID;

import cz.msebera.android.httpclient.HttpResponse;
import cz.msebera.android.httpclient.client.HttpClient;
import cz.msebera.android.httpclient.impl.client.DefaultHttpClient;
import cz.msebera.android.httpclient.util.EntityUtils;

public class registro2 extends AppCompatActivity {

    Button button, buscar, borrar ;
    TextView info, textView42, textView51, textView53 , textView20,
textView21, textView22, textView23, textView24, textView25,
textView26, textView27, textView28, textView29, textView30, textView31, text
View32, textView33, textView34, textView35, textView36, textView37,
textView38, textView39, textView40, textView41;

    ImageView foto;
    private static final int SOLICITA_ACTIVACION = 1;
    private static final int SOLICITA_CONEXION = 2;
    private InputStream inputStream;
    private BluetoothSocket meuSocket;

    BluetoothAdapter meuBluetoothAdapter= null;
    BluetoothDevice meuDevice=null;

    boolean stopThread;
    byte buffer[];

    EditText editText1;
    private String urlLogin =
"http://identificatecuador.com/getEmp.php";

    boolean conexionb = false;

    private static String MAC= null;
    UUID MI_UUID = UUID.fromString("00001101-0000-1000-8000-

```

```
00805f9b34fb");
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_registro2);

        button= (Button) findViewById(R.id.button);
        buscar= (Button) findViewById(R.id.buscar);
        editText1 = (EditText) findViewById(R.id.editText1);
        borrar = (Button) findViewById(R.id.button3);

        textView20 = (TextView) findViewById(R.id.textView20);
        textView21 = (TextView) findViewById(R.id.textView21);
        textView22 = (TextView) findViewById(R.id.textView22);
        textView23 = (TextView) findViewById(R.id.textView23);
        textView24 = (TextView) findViewById(R.id.textView24);
        textView25 = (TextView) findViewById(R.id.textView25);
        textView26 = (TextView) findViewById(R.id.textView26);
        textView27 = (TextView) findViewById(R.id.textView27);
        textView28 = (TextView) findViewById(R.id.textView28);
        textView29 = (TextView) findViewById(R.id.textView29);
        textView30 = (TextView) findViewById(R.id.textView30);
        textView31 = (TextView) findViewById(R.id.textView31);
        textView32 = (TextView) findViewById(R.id.textView32);
        textView33 = (TextView) findViewById(R.id.textView33);
        textView34 = (TextView) findViewById(R.id.textView34);
        textView35 = (TextView) findViewById(R.id.textView35);
        textView37 = (TextView) findViewById(R.id.textView37);
        textView38 = (TextView) findViewById(R.id.textView38);
        textView39 = (TextView) findViewById(R.id.textView39);
        textView40 = (TextView) findViewById(R.id.textView40);
        textView41 = (TextView) findViewById(R.id.textView41);
        textView42 = (TextView) findViewById(R.id.textView42);
        textView51 = (TextView) findViewById(R.id.textView51);
        textView53 = (TextView) findViewById(R.id.textView53);

        foto=(ImageView) findViewById(R.id.foto);

        info= (TextView) findViewById(R.id.textView4);

        buscar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                ObtenerContacto tarea = new ObtenerContacto();
                tarea.execute(editText1.getText().toString());
            }
        });
//-----
        meuBluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

        if(meuBluetoothAdapter == null) {
            Toast.makeText(getApplicationContext(), "Su dispositivo no
posee bluetooth", Toast.LENGTH_LONG).show();
        }else if(!meuBluetoothAdapter.isEnabled()) {
            Intent ativaBluetooth = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(ativaBluetooth,
SOLICITA_ACTIVACION);
        }
    }
}
```

```

    }
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (conexionb) {
                //desconectar
                try {
                    conexionb = false;
                    meuBluetoothAdapter.disable();
                    meuSocket =
meuDevice.createRfcommSocketToServiceRecord(MI_UUID);
                    meuSocket.close();
                    button.setText("Conectar");
                    Toast.makeText(getApplicationContext(),
"Bluetooth desconectado", Toast.LENGTH_LONG).show();
                    meuBluetoothAdapter.enable();

                } catch (IOException error) {
                    Toast.makeText(getApplicationContext(),
"OCURRIO UN ERROR:" + error, Toast.LENGTH_LONG).show();
                    error.printStackTrace();
                }

            } else {
                // conectar
                Intent abreLista = new Intent(registro2.this,
ListaDispositivos.class);
                startActivityForResult(abreLista,
SOLICITA_CONEXION);
            }
        }
    });
}
//-----
BORRAR -----

    public void onClear (View v){
        editText1.setText("");
        info.setText("Presione el boton BUSCAR para visualizar los
datos personales de la persona");
    }

//-----
REALIZAR LA BUSQUEDA EN LA BASE DE DATOS-----
-----

    private class ObtenerContacto extends AsyncTask<String, Void,
Void> {

        String oFoto, oAlergias, oNames , oApellidos, oContactouno,
oOtrosF, oFechaConsulta ,oTabaco, oAlcohol, oDrogas, oOtros,
oObservaciones1 , oHipertension, oTBP, oEnfermedadcardiaca, oDiabetes,
oCancer, oDemencias, oHipertensionArterial, oDiabetesMelitis1,
oDiabetesMelitis2, oEPOC, oTransfusiones, oCirugias, oTipodesangre,
oMedicamentos;

        Bitmap bitmap;

        @Override
        protected Void doInBackground(String... params) {

```

```

        Log.i("ConsultaContacto", "doInBackground");
        HttpClient httpClient = new DefaultHttpClient();

        String sCodigo = params[0];

        HttpGet get = new
HttpGet("http://www.identificatecuador.com/busquedahistoriaclinica.php?Codigo=" + sCodigo );
        get.setHeader("Content-type", "application/json");

        try {
            HttpResponse resp= httpClient.execute(get);
            String respString =
EntityUtils.toString(resp.getEntity());

            JSONArray respJSON = new JSONArray(respString);

            for(int i =0; i< respJSON.length() ; i++) {
                JSONObject jsonObject = respJSON.getJSONObject(i);

                oNames = jsonObject.getString("nombres");
                oApellidos = jsonObject.getString("apellidos");
                oContactouno =
jsonObject.getString("num_contacto1");
                oFechaConsulta =
jsonObject.getString("fecha_consulta");
                oTabaco = jsonObject.getString("Tabaco");
                oAlcohol = jsonObject.getString("Alcohol");
                oDrogas = jsonObject.getString("Drogas");
                oOtros = jsonObject.getString("OtrosH");
                oObservaciones1 =
jsonObject.getString("Observaciones");
                oHipertension =
jsonObject.getString("Hipertension");
                oTBP = jsonObject.getString("TBP");
                oEnfermedadcardiaca =
jsonObject.getString("Enferm_Cardiaca");
                oDiabetes = jsonObject.getString("Diabetes");
                oCancer = jsonObject.getString("Cancer");
                oDemencias = jsonObject.getString("Demencias");
                oHipertensionArterial =
jsonObject.getString("Hipertension_Arterial");
                //oDiabetesMelitisis1 =
jsonObject.getString("Diabetes_Melitisis");
                oDiabetesMelitisis2 =
jsonObject.getString("diabetes_melitisis2");
                oEPOC = jsonObject.getString("EPOC");
                oTransfusiones =
jsonObject.getString("Transfusiones");
                oCirugias = jsonObject.getString("Cirugias");
                oTipodesangre =
jsonObject.getString("Tipo_de_Sangre");
                oMedicamentos =
jsonObject.getString("Medicamentos");
                oOtrosF = jsonObject.getString("Otros");
                oAlergias = jsonObject.getString("Alergias");

                oFoto = jsonObject.getString("imagen");

                if (oFoto.equals("noimagen")){

```

```

        }else
        {
            URL urlimagen= new
URL("http://www.identificateecuador.com/images/personas/" + oFoto);
            HttpURLConnection conimagen =
(HttpURLConnection) urlimagen.openConnection();
            conimagen.connect();
            bitmap =
BitmapFactory.decodeStream(conimagen.getInputStream());
        }

    }

    } catch (Exception ex) {
        Log.e("ServicioRest", "Error!", ex);
        ex.printStackTrace();
    }

    return null;
}

private class DescargarImagen extends AsyncTask<ImageView,
Void, Bitmap>{
    ImageView foto;
    Bitmap bitm;
    @Override
    protected Bitmap doInBackground(ImageView... params) {
        foto= params[0];
        try {
            URL imageURL = new
URL("http://www.identificateecuador.com/imagen/personas/" + oFoto);
            HttpURLConnection conn= (HttpURLConnection)
imageURL.openConnection();
            conn.connect();
            bitm =
BitmapFactory.decodeStream(conn.getInputStream());

            } catch (MalformedURLException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return bitm;
        }
    }

    @Override
    protected void onPostExecute(Void result){
        Log.i("ServicioRest", "onPostExecute");
        textView20.setText("NOMBRES: " + oNames);
        textView21.setText("APELLIDOS: " + oApellidos);
        textView22.setText("PARA EMERGENCIAS LLAMAR AL: " +
oContactouno);
        textView23.setText("FECHA DE CONSULTA: " +
oFechaConsulta);
        textView24.setText("TABACO: " + oTabaco);
        textView25.setText("ALCOHOL: " + oAlcohol);
        textView26.setText("DROGAS: " + oDrogas);
        textView27.setText("OTROS: " + oOtros);
        textView28.setText("OBSERVACIONES: " + oObservaciones1);
    }
}

```

```

        textView29.setText("HIPERTENSIÓN: " + oHipertension);
        textView30.setText("TBP: " + oTBP);
        textView31.setText("ENFERMEDAD CARDIACA: " +
oEnfermedadcardiaca);
        textView32.setText("DIABETES: " + oDiabetes);
        textView33.setText("CÁNCER: " + oCancer);
        textView34.setText("DEMENCIAS: " + oDemencias);
        textView35.setText("HIPERTENSIÓN ARTERIAL: " +
oHipertensionArterial);
        //textView36.setText("DIABETES MELITIS 1: " +
oDiabetesMelitis1);
        textView37.setText("DIABETES MELITIS: " +
oDiabetesMelitis2);
        textView38.setText("EPOC: " + oEPOC);
        textView39.setText("TRANSFUSIONES: " + oTransfusiones);
        textView40.setText("CIRUGIAS: " + oCirugias);
        textView41.setText("TIPO DE SANGRE: " + oTipodesangre);
        textView42.setText("MEDICAMENTOS:" + oMedicamentos );
        textView51.setText("OTROS: " + oOtrosF);
        textView53.setText("ALÉRGIAS: " + oAlergias);

        foto.setImageBitmap(bitmap);
    }

    @Override
    protected void onPreExecute() {
        Log.i("ServicioRest", "onPreExecute");
        info.setText("Se está obteniendo la información . . .");
    }
}

//-----
-----

    @Override
    protected void onActivityResult(int requestCode, int resultCode,
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        switch (requestCode) {

            //-----ACTIVAR BLUETOOTH
            AL INGRESAR A LA APLICACIÓN-----
            -----

            case SOLICITA_ACTIVACION:
                if(resultCode == Activity.RESULT_OK){
                    Toast.makeText(getApplicationContext(),"Bluetooth
fue activado",Toast.LENGTH_LONG ).show();
                }else {
                    Toast.makeText(getApplicationContext(),"Bluetooth
no esta habilitado, la aplicación se cerrará",Toast.LENGTH_LONG
).show();

                    finish();
                }
                break;

            //-----
            -----

            case SOLICITA_CONEXION:
                if(resultCode == Activity.RESULT_OK){

MAC=data.getExtras().getString(ListaDispositivos.DIRECCION_MAC);

```

```

        meuDevice =
meuBluetoothAdapter.getRemoteDevice(MAC);

        try {
            meuSocket =
meuDevice.createRfcommSocketToServiceRecord(MI_UUID);
            meuSocket.connect();
            conexionb=true;
            button.setText("Desconectar");
            Toast.makeText(getApplicationContext(),"Fue
conectado con:"+MAC,Toast.LENGTH_LONG ).show();
            inputStream=meuSocket.getInputStream();
            beginListenForData();

        }catch (IOException error){
            conexionb=false;

Toast.makeText(getApplicationContext(),"OCURRIO UN
ERROR:"+error,Toast.LENGTH_LONG ).show();
        }

        }else{
            Toast.makeText(getApplicationContext(),"No se pudo
obtener la MAC",Toast.LENGTH_LONG ).show();

        }
    }
}
//-----LEER CODIGO RFID -----
void beginListenForData()
{
    final Handler handler = new Handler();
    stopThread = false;
    buffer = new byte[1024];
    Thread thread = new Thread(new Runnable()
    {
        public void run()
        {
            while(!Thread.currentThread().isInterrupted() &&
!stopThread)
            {
                try
                {
                    int byteCount = inputStream.available();
                    if(byteCount > 0)
                    {
                        byte[] rawBytes = new byte[byteCount];
                        inputStream.read(rawBytes);
                        final StringCodigo=new
String(rawBytes,"UTF-8");
                        handler.post(new Runnable() {
                            public void run()
                            {
                                editText1.append(Codigo);
                            }
                        });
                    }
                }
            }
        }
    }
}

```

```

        catch (IOException ex)
        {
            stopThread = true;
        }
    }
});

thread.start();
}
//-----
}

```

ANEXO C. CÓDIGO PAGINA WEB.

Registro de personas

```

<?php require_once('Connections/Identificacion.php'); ?>
<?php
if (!function_exists("GetSQLValueString")) {
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue = "")
{
    if (PHP_VERSION < 6) {
        $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
    }

    $theValue = function_exists("mysql_real_escape_string") ?
mysql_real_escape_string($theValue) : mysql_escape_string($theValue);

    switch ($theType) {
        case "text":
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
            break;
        case "long":
        case "int":
            $theValue = ($theValue != "") ? intval($theValue) : "NULL";
            break;
        case "double":
            $theValue = ($theValue != "") ? doubleval($theValue) : "NULL";
            break;
        case "date":
            $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
            break;
        case "defined":
            $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
            break;
    }
    return $theValue;
}
}
}

```

```

// *** Redirect if username exists
$MM_flag="MM_insert";
if (isset($_POST[$MM_flag])) {
    $MM_dupKeyRedirect="error_registro.php";
    $loginUsername = $_POST['Codigo'];
    $LoginRS__query = sprintf("SELECT Codigo FROM datos_personales WHERE Codigo=%s",
    GetSQLValueString($loginUsername, "text"));
    mysql_select_db($database_Identificacion, $Identificacion);
    $LoginRS=mysql_query($LoginRS__query, $Identificacion) or die(mysql_error());
    $loginFoundUser = mysql_num_rows($LoginRS);

    //if there is a row in the database, the username was found - can not add the requested
    username
    if($loginFoundUser){
        $MM_qsChar = "?";
        //append the username to the redirect page
        if (substr_count($MM_dupKeyRedirect,"?") >=1) $MM_qsChar = "&";
        $MM_dupKeyRedirect = $MM_dupKeyRedirect . $MM_qsChar
        ."requername=".$_loginUsername;
        header ("Location: $MM_dupKeyRedirect");
        exit;
    }
}

$editFormAction = $_SERVER['PHP_SELF'];
if (isset($_SERVER['QUERY_STRING'])) {
    $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);
}

if ((isset($_POST["MM_insert"])) && ($_POST["MM_insert"] == "form1")) {
    $insertSQL = sprintf("INSERT INTO datos_personales (Codigo, Nombres, Apellidos, Genero,
    Num_Cedula, Fecha_Nacimiento, Ciudad, Pais, Direccion, Ocupacion, Num_Contacto1,
    Num_Contacto2, Email, Lugar_Trabajo, Telef_Fijo, Telef_Movil, Estado_Civil, imagen) VALUES
    (%s, %s, %s)",
        GetSQLValueString($_POST['Codigo'], "text"),
        GetSQLValueString($_POST['Nombres'], "text"),
        GetSQLValueString($_POST['Apellidos'], "text"),
        GetSQLValueString($_POST['Genero'], "text"),
        GetSQLValueString($_POST['Num_Cedula'], "int"),
        GetSQLValueString($_POST['Fecha_Nacimiento'], "date"),
        GetSQLValueString($_POST['Ciudad'], "text"),
        GetSQLValueString($_POST['Pais'], "text"),
        GetSQLValueString($_POST['Direccion'], "text"),
        GetSQLValueString($_POST['Ocupacion'], "text"),
        GetSQLValueString($_POST['Num_Contacto1'], "text"),
        GetSQLValueString($_POST['Num_Contacto2'], "text"),
        GetSQLValueString($_POST['Email'], "text"),
        GetSQLValueString($_POST['Lugar_Trabajo'], "text"),
        GetSQLValueString($_POST['Telef_Fijo'], "text"),
        GetSQLValueString($_POST['Telef_Movil'], "text"),
        GetSQLValueString($_POST['Estado_Civil'], "text"),

```

```

        GetSQLValueString($_POST['imagen'], "text"));

mysql_select_db($database_Identificacion, $Identificacion);
$result1 = mysql_query($insertSQL, $Identificacion) or die(mysql_error());

$insertGoTo = "Registro_ok.php";
if (isset($_SERVER['QUERY_STRING'])) {
    $insertGoTo .= (strpos($insertGoTo, '?')) ? "&" : "?";
    $insertGoTo .= $_SERVER['QUERY_STRING'];
}
header(sprintf("Location: %s", $insertGoTo));
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<style type="text/css">
.container .sidebar1 p strong {
    font-family: Verdana, Geneva, sans-serif;
}
.container .sidebar1 #form1 table {
}
.container .sidebar1 p strong {
    font-family: Verdana, Geneva, sans-serif;
}
.container .sidebar1 #form1 table tr td {
    font-family: Verdana, Geneva, sans-serif;
}
.container .sidebar1 p {
    font-family: Verdana, Geneva, sans-serif;
}
</style>
<!-- InstanceBegin template="/Templates/plantillabase.dwt.php"
codeOutsideHTMLOutsideLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Documento sin título</title>
<script src="SpryAssets/SpryValidationTextField.js" type="text/javascript"></script>
<link href="SpryAssets/SpryValidationTextField.css" rel="stylesheet" type="text/css" />
<!-- InstanceEndEditable -->

<link href="css/estiloprincipal.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="css/menu.css" type="text/css" />

<link href="https://fonts.googleapis.com/css?family=Cormorant+Garamond|Exo"
rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Josefin+Slab" rel="stylesheet">

<?php include("includes/google.php"); ?>
<style type="text/css">

```

```

body,td,th {
    font-family: "Josefin Slab", serif;
}
</style>
</head>

<body>

<div class="container">
  <div class="header">
    <?php include("includes/cabecera.php"); ?>
  <div class="clearfloat"></div>
  <?php include("includes/menu.php"); ?>
  <?php include("includes/slider.php"); ?>
</div>
<div class="sidebar1"><!-- InstanceBeginEditable name="Contenido izquierda" -->
  <p><strong>INSERTAR NUEVO REGISTRO DE PERSONAS.</strong></p>
  <p>Formulario de Registro:</p>
  <p><strong>Instrucciones: </strong>Los campos que estan con * son obligatorios. (El codigo
debe solicitar al administrador de la p&aacute;gina).</p>
  <p>&nbsp;</p>

  <script>
  function subirimagen(nombrecampo)
  {
    self.name = 'opener';
    remote = open('subirimagen.php?campo='+nombrecampo,
'remote','width=400,height=150,location=no,scrollbars=yes,menubars=no,toolbars=no,resizabl
e=yes,fullscreen=no, status=yes');
    remote.focus();
  }
  </script>

  <form action="<?php echo $editFormAction; ?>" method="post" name="form1"
id="form1">
  <table align="right">
    <tr valign="baseline">
      <td width="115" align="left" nowrap="nowrap"
bgcolor="#0099CC"><strong>C&oacute;digo*:</strong></td>
      <td width="412"><span id="sprytextfield1">
        <input type="text" name="Codigo" value="" size="32" />
        <span class="textfieldRequiredMsg">Ingrese su c&oacute;digo</span></span></td>
    </tr>
    <tr valign="baseline">
      <td align="left" nowrap="nowrap"
bgcolor="#0099CC"><strong>Nombres*:</strong></td>
      <td><span id="sprytextfield2">
        <input type="text" name="Nombres" value="" size="32" />
        <span class="textfieldRequiredMsg">Ingrese sus nombres</span></span></td>
    </tr>
  </table>

```

```

        <td align="left" nowrap="nowrap"
bgcolor="#0099CC"><strong>Apellidos*:</strong></td>
        <td><span id="sprytextfield3">
            <input type="text" name="Apellidos" value="" size="32" />
            <span class="textfieldRequiredMsg">Ingrese sus apellidos</span></span></td>
    </tr>
    <tr valign="baseline">
        <td align="left" nowrap="nowrap"
bgcolor="#0099CC"><strong>Género*:</strong></td>
        <td valign="baseline"><table>
            <tr>
                <td><input type="radio" name="Genero" value="Masculino" />
                Masculino</td>
            </tr>
            <tr>
                <td><input type="radio" name="Genero" value="Femenino" />
                Femenino</td>
            </tr>
        </table></td>
    </tr>
    <tr valign="baseline">
        <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Numero de
Cédula*:</strong></td>
        <td><span id="sprytextfield4">
            <input type="text" name="Num_Cedula" value="" size="32" />
            <span class="textfieldRequiredMsg">Ingrese su numero de cédula</span><span
class="textfieldMinCharsMsg">El número mínimo de caracteres es 10</span><span
class="textfieldMaxCharsMsg"> El número máximo de caracteres es 10</span></span></td>
    </tr>
    <tr valign="baseline">
        <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Fecha de
Nacimiento(a&ntilde;o-mes-d&iacute;a)*:</strong></td>
        <td><span id="sprytextfield5">
            <input type="text" name="Fecha_Nacimiento" value="" size="32" />
            <span class="textfieldRequiredMsg">Ingrese su fecha de nacimiento</span><span
class="textfieldInvalidFormatMsg">Formato no válido</span></span></td>
    </tr>
    <tr valign="baseline">
        <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Ciudad*:</strong></td>
        <td><span id="sprytextfield6">
            <input type="text" name="Ciudad" value="" size="32" />
            <span class="textfieldRequiredMsg">Ingrese su ciudad de
nacimiento</span></span></td>
    </tr>
    <tr valign="baseline">
        <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>País*:</strong></td>
        <td><span id="sprytextfield7">
            <input type="text" name="País" value="" size="32" />
            <span class="textfieldRequiredMsg">Ingrese su país de nacimiento</span></span></td>
    </tr>
    <tr valign="baseline">

```

```

        <td align="left" nowrap="nowrap"
bgcolor="#0099CC"><strong>Direcci&oacute;n*:</strong></td>
        <td><span id="sprytextfield8">
            <input type="text" name="Direccion" value="" size="32" />
            <span class="textfieldRequiredMsg">Ingrese su direcci&oacute;n</span></span></td>
        </tr>
        <tr valign="baseline">
            <td align="left" nowrap="nowrap"
bgcolor="#0099CC"><strong>Ocupaci&oacute;n:</strong></td>
            <td><span id="sprytextfield9">
                <input type="text" name="Ocupacion" value="" size="32" />
                <span class="textfieldRequiredMsg">Ingrese su ocupaci&oacute;n</span></span></td>
            </tr>
            <tr valign="baseline">
                <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>N&uacute;mero de
Contacto1*:</strong></td>
                <td><span id="sprytextfield10">
                    <input type="text" name="Num_Contacto1" value="" size="32" />
                    <span class="textfieldRequiredMsg">Ingrese un n&uacute;mero de contacto</span><span
class="textfieldMinCharsMsg">El n&uacute;mero m&iacute;nimo de caracteres es 10</span><span
class="textfieldMaxCharsMsg"> El n&uacute;mero m&iacute;nimo de caracteres es 1</span></span></td>
                </tr>
                <tr valign="baseline">
                    <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>N&uacute;mero de
Contacto2:</strong></td>
                    <td><span id="sprytextfield11">
                        <input type="text" name="Num_Contacto2" value="" size="32" />
                        <span class="textfieldMinCharsMsg">El n&uacute;mero m&iacute;nimo de caracteres es
10</span><span class="textfieldMaxCharsMsg">El n&uacute;mero m&iacute;ximo de caracteres es
1</span></span></td>
                    </tr>
                    <tr valign="baseline">
                        <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Email:</strong></td>
                        <td><span id="sprytextfield12">
                            <input type="text" name="Email" value="" size="32" />
                            <span class="textfieldInvalidFormatMsg">Formato no v&iacute;lido.</span></span></td>
                        </tr>
                        <tr valign="baseline">
                            <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Lugar de
Trabajo:</strong></td>
                            <td><span id="sprytextfield13">
                                <input type="text" name="Lugar_Trabajo" value="" size="32" />
                            </span></td>
                        </tr>
                        <tr valign="baseline">
                            <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Tel&eacute;fono
Fijo*:</strong></td>
                            <td><span id="sprytextfield14">
                                <input type="text" name="Telef_Fijo" value="" size="32" />
                                <span class="textfieldRequiredMsg">Ingrese su n&uacute;mero de tel&eacute;fono fijo</span><span
class="textfieldMinCharsMsg">El n&uacute;mero m&iacute;nimo de caracteres es 9</span><span
class="textfieldMaxCharsMsg">El n&uacute;mero m&iacute;ximo de caracteres e</span></span></td>

```

```

</tr>
<tr valign="baseline">
  <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Tel&eacute;fono
M&oacute;vil*:</strong></td>
  <td><span id="sprytextfield15">
    <input type="text" name="Telef_Movil" value="" size="32" />
    <span class="textfieldRequiredMsg">Ingrese su n&uacute;mero de tel&eacute;fono m&ouil</span><span
class="textfieldMinCharsMsg">El n&uacute;mero m&iacute;nimo de caracteres es 10</span><span
class="textfieldMaxCharsMsg">El n&uacute;mero m&iacute;ximo de caracteres es 10</span></span></td>
  </tr>
  <tr valign="baseline">
    <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Estado
Civil*:</strong></td>
    <td><select name="Estado_Civil">
      <option value="Soltero(a)" <?php if (!(strcmp(1, ""))) {echo "SELECTED";}
?>>Soltero(a)</option>
      <option value="Casado(a)" <?php if (!(strcmp(1, ""))) {echo "SELECTED";}
?>>Casado(a)</option>
      <option value="Divorciado(a)" <?php if (!(strcmp(1, ""))) {echo "SELECTED";}
?>>Divorciado(a)</option>
      <option value="Viudo(a)" <?php if (!(strcmp(1, ""))) {echo "SELECTED";}
?>>Viudo(a)</option>
      <option value="Union Libre" <?php if (!(strcmp(1, ""))) {echo "SELECTED";} ?>>Union
Libre</option>
    </select></td>
  </tr>
  <tr valign="baseline">
    <td align="left" nowrap="nowrap" bgcolor="#0099CC"><strong>Foto*:</strong></td>
    <td><input type="text" name="imagen" value="" size="32" />
    <input type="button" name="button" id="button" value="Ingresar imagen"
onclick="javascript:subirimagen('imagen');" /></td>
  </tr>
  <tr valign="baseline">
    <td align="right" nowrap="nowrap" bgcolor="#FFFFFF">&nbsp;</td>
    <td><input type="submit" value="Insertar registro" />
    <input type="reset" name="borrar" id="borrar" value="Restablecer" /></td>
  </tr>
</table>
<p>
  <input type="hidden" name="MM_insert" value="form1" />
</p>
</form>
<p>&nbsp;</p>
<p><strong><a href="_admin/usuarios_lista2.php">Regresar</a></strong></p>
<p>&nbsp;</p>
<!-- InstanceEndEditable -->
<!-- end .sidebar1 --></div>
<div class="content"><!-- InstanceBeginEditable name="Parte derecha" -->
  <h1>&nbsp;</h1>
  <script type="text/javascript">
var sprytextfield1 = new Spry.Widget.ValidationTextField("sprytextfield1", "none",
{validateOn:["blur"]});

```

```

var sprytextfield2 = new Spry.Widget.ValidationTextField("sprytextfield2", "none",
{validateOn:["blur"]});
var sprytextfield3 = new Spry.Widget.ValidationTextField("sprytextfield3", "none",
{validateOn:["blur"]});
var sprytextfield4 = new Spry.Widget.ValidationTextField("sprytextfield4", "none",
{validateOn:["blur"], minChars:10, maxChars:10});
var sprytextfield5 = new Spry.Widget.ValidationTextField("sprytextfield5", "date",
{validateOn:["blur"], format:"yyyy-mm-dd"});
var sprytextfield6 = new Spry.Widget.ValidationTextField("sprytextfield6", "none",
{validateOn:["blur"]});
var sprytextfield7 = new Spry.Widget.ValidationTextField("sprytextfield7", "none",
{validateOn:["blur"]});
var sprytextfield8 = new Spry.Widget.ValidationTextField("sprytextfield8", "none",
{validateOn:["blur"]});
var sprytextfield9 = new Spry.Widget.ValidationTextField("sprytextfield9", "none",
{validateOn:["blur"], isRequired:false});
var sprytextfield10 = new Spry.Widget.ValidationTextField("sprytextfield10", "none",
{validateOn:["blur"], minChars:10, maxChars:10});
var sprytextfield11 = new Spry.Widget.ValidationTextField("sprytextfield11", "none",
{validateOn:["blur"], isRequired:false, minChars:10, maxChars:10});
var sprytextfield12 = new Spry.Widget.ValidationTextField("sprytextfield12", "email",
{validateOn:["blur"], isRequired:false});
var sprytextfield13 = new Spry.Widget.ValidationTextField("sprytextfield13", "none",
{isRequired:false, validateOn:["blur"]});
var sprytextfield14 = new Spry.Widget.ValidationTextField("sprytextfield14", "none",
{validateOn:["blur"], minChars:9, maxChars:9});
var sprytextfield15 = new Spry.Widget.ValidationTextField("sprytextfield15", "none",
{validateOn:["blur"], minChars:10, maxChars:10});
</script>
<!-- InstanceEndEditable -->
<!-- end .content --></div>
<div class="footer">
<?php include("includes/pie.php"); ?></div>
<!-- end .container --></div>
</body>
<!-- InstanceEnd --></html>

```

Autenticación personal

```

<?php require_once('Connections/conexioncreacion.php'); ?>
<?php
if (!function_exists("GetSQLValueString")) {
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue = "")
{
if (PHP_VERSION < 6) {
    $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
}

    $theValue = function_exists("mysql_real_escape_string") ?
mysql_real_escape_string($theValue) : mysql_escape_string($theValue);

```

```

switch ($theType) {
    case "text":
        $theValue = ($theValue != "") ? "" . $theValue . "" : "NULL";
        break;
    case "long":
    case "int":
        $theValue = ($theValue != "") ? intval($theValue) : "NULL";
        break;
    case "double":
        $theValue = ($theValue != "") ? doubleval($theValue) : "NULL";
        break;
    case "date":
        $theValue = ($theValue != "") ? "" . $theValue . "" : "NULL";
        break;
    case "defined":
        $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
        break;
}
return $theValue;
}
}
?>
<?php
// *** Validate request to login to this site.
if (!isset($_SESSION)) {
    session_start();
}

$loginFormAction = $_SERVER['PHP_SELF'];
if (isset($_GET['accesscheck'])) {
    $_SESSION['PrevUrl'] = $_GET['accesscheck'];
}

if (isset($_POST['NOMBRE'])) {
    $loginUsername=$_POST['NOMBRE'];
    $password=$_POST['PASS'];
    $MM_fldUserAuthorization = "Nivel";
    $MM_redirectLoginSuccess = "menupolicias.php";
    $MM_redirectLoginFailed = "error.php";
    $MM_redirecttoReferrer = true;
    mysql_select_db($database_conexioncreacion, $conexioncreacion);

    $LoginRS__query=sprintf("SELECT Nombre, Password, Nivel FROM tbusuarios WHERE
Nombre=%s AND Password=%s",
    GetSQLValueString($loginUsername, "text"), GetSQLValueString($password, "text"));

    $LoginRS = mysql_query($LoginRS__query, $conexioncreacion) or die(mysql_error());
    $loginFoundUser = mysql_num_rows($LoginRS);
    if ($loginFoundUser) {

        $loginStrGroup = mysql_result($LoginRS,0,'Nivel');

```

```

        if (PHP_VERSION >= 5.1) {session_regenerate_id(true);} else {session_regenerate_id();}
//declare two session variables and assign them
$_SESSION['MM_Username'] = $loginUsername;
$_SESSION['MM_UserGroup'] = $loginStrGroup;

if (isset($_SESSION['PrevUrl']) && true) {
    $MM_redirectLoginSuccess = $_SESSION['PrevUrl'];
}
header("Location: " . $MM_redirectLoginSuccess );
}
else {
    header("Location: " . $MM_redirectLoginFailed );
}
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/plantillabase.dwt.php" codeOutsideHTMIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Documento sin título</title>
<!-- InstanceEndEditable -->

<link href="css/estiloprincipal.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="css/menu.css" type="text/css" />

<link href="https://fonts.googleapis.com/css?family=Cormorant+Garamond|Exo"
rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Josefin+Slab" rel="stylesheet">

<?php include("includes/google.php"); ?>
<style type="text/css">
body,td,th {
    font-family: "Josefin Slab", serif;
}
</style>
</head>

<body>

<div class="container">
    <div class="header">
        <?php include("includes/cabecera.php"); ?>
        <div class="clearfloat"></div>
        <?php include("includes/menu.php"); ?>
        <?php include("includes/slider.php"); ?>
        </div>
        <div class="sidebar1"><!-- InstanceBeginEditable name="Contenido izquierda" -->

```

```

<p><strong>BIENVENIDO AL LOGIN DE POLICIAS</strong></p>
<p><strong>Introduce los datos requeridos:</strong></p>
<p>Instrucciones: Coloca tu nombre de usuario y password para ingresar al men&uacute; solicitado, coloca los datos correctamente o de lo contrario no se te dar&aacute; ingreso.
</p>
<p>&nbsp;</p>
<center><form id="form1" name="form1" method="POST" action="<?php echo
$loginFormAction; ?>">
  <table width="315" border="0">
    <tr>
      <td width="103" align="right" valign="middle">Nombre:</td>
      <td width="196"><input name="NOMBRE" type="text" id="NOMBRE" maxlength="30" />
      <label for="NOMBRE">*</label></td>
    </tr>
    <tr>
      <td align="right" valign="middle">Password:</td>
      <td><input name="PASS" type="password" id="PASS" maxlength="15" />
      <label for="PASS">*</label></td>
    </tr>
    <tr>
      <td colspan="2"><center><input type="submit" name="button" id="button"
value="Ingresar" />
      <input type="reset" name="button2" id="button2" value="Restablecer"
/><center/></td>
    </tr>
  </table>
</form><center/>
<p>&nbsp;</p>
<p><strong>Nota: </strong>En caso de no recordar tus datos comuniquese con el
administrador del sitio.</p>
<!-- InstanceEndEditable -->
<!-- end .sidebar1 --></div>
<div class="content"><!-- InstanceBeginEditable name="Parte derecha" -->
  <h1>&nbsp;</h1>
<!-- InstanceEndEditable -->
<!-- end .content --></div>
<div class="footer">
<?php include("includes/pie.php"); ?></div>
<!-- end .container --></div>
</body>
<!-- InstanceEnd --></html>

```

Menú Policias y Auxiliares

```

<?php
//initialize the session
if (!isset($_SESSION)) {
  session_start();
}

// ** Logout the current user. **
$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true";

```

```

if ((isset($_SERVER['QUERY_STRING'])) && ($_SERVER['QUERY_STRING'] != "")){
    $logoutAction .= "&". htmlentities($_SERVER['QUERY_STRING']);
}

if ((isset($_GET['doLogout'])) && ($_GET['doLogout']=="true")){
    //to fully log out a visitor we need to clear the session variables
    $_SESSION['MM_Username'] = NULL;
    $_SESSION['MM_UserGroup'] = NULL;
    $_SESSION['PrevUrl'] = NULL;
    unset($_SESSION['MM_Username']);
    unset($_SESSION['MM_UserGroup']);
    unset($_SESSION['PrevUrl']);

    $logoutGoTo = "login_policias.php";
    if ($logoutGoTo) {
        header("Location: $logoutGoTo");
        exit;
    }
}
?>
<?php
if (isset($_SESSION)) {
    session_start();
}
$MM_authorizedUsers = "0";
$MM_donotCheckaccess = "false";

// *** Restrict Access To Page: Grant or deny access to this page
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {
    // For security, start by assuming the visitor is NOT authorized.
    $isValid = False;

    // When a visitor has logged into this site, the Session variable MM_Username set equal to
    their username.
    // Therefore, we know that a user is NOT logged in if that Session variable is blank.
    if (!empty($UserName)) {
        // Besides being logged in, you may restrict access to only certain users based on an ID
        established when they login.
        // Parse the strings into arrays.
        $arrUsers = Explode(",", $strUsers);
        $arrGroups = Explode(",", $strGroups);
        if (in_array($UserName, $arrUsers)) {
            $isValid = true;
        }
        // Or, you may restrict access to only certain users based on their username.
        if (in_array($UserGroup, $arrGroups)) {
            $isValid = true;
        }
        if (($strUsers == "") && false) {
            $isValid = true;
        }
    }
}

```

```

return $isValid;
}

$MM_restrictGoTo = "error.php";
if (!(isset($_SESSION['MM_Username'])) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup']))) {
    $MM_qsChar = "?";
    $MM_referrer = $_SERVER['PHP_SELF'];
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
    if (isset($_SERVER['QUERY_STRING']) && strlen($_SERVER['QUERY_STRING']) > 0)
    $MM_referrer .= "?" . $_SERVER['QUERY_STRING'];
    $MM_restrictGoTo = $MM_restrictGoTo . $MM_qsChar . "accesscheck=" .
urlencode($MM_referrer);
    header("Location: " . $MM_restrictGoTo);
    exit;
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/plantillabase.dwt.php" codeOutsideHTMlIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Documento sin título</title>
<!-- InstanceEndEditable -->

<link href="css/estiloprincipal.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="css/menu.css" type="text/css" />

<link href="https://fonts.googleapis.com/css?family=Cormorant+Garamond|Exo"
rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Josefin+Slab" rel="stylesheet">

<?php include("includes/google.php"); ?>
<style type="text/css">
body,td,th {
    font-family: "Josefin Slab", serif;
}
</style>
</head>

<body>

<div class="container">
<div class="header">
<?php include("includes/cabecera.php"); ?>
<div class="clearfloat"></div>
<?php include("includes/menu.php"); ?>
<?php include("includes/slider.php"); ?>

```

```

</div>
<div class="sidebar1"><!-- InstanceBeginEditable name="Contenido izquierda" -->
  <p><strong>BIENVENIDO A PÁGINA DE POLICÍAS</strong></p>
  <p>Bienvenido: <?php echo $_SESSION['MM_Username'];?> este es su menú principal de
  actividades.</p>
  <p>Ingrese el NÚMERO DE CÉDULA de la víctima:</p>
  <form id="form1" name="form1" method="get" action="_personas/busqueda2.php">
    <table width="506" border="0">
      <tr>
        <td width="423" align="center"><input name="busqueda2" type="text" id="busqueda2"
        size="50" maxlength="10" />
        <label for="busqueda2">*</label></td>
        <td width="73" align="center"><input type="submit" name="button" id="button"
        value="Buscar" /></td>
      </tr>
      <tr>
        <td>Ingrese el número de cédula.</td>
        <td>&nbsp;</td>
      </tr>
    </table>
  </form>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
  <p>&nbsp;</p>
  <p><a href="<?php echo $logoutAction ?>">Desconectar</a></p>
<!-- InstanceEndEditable -->
<!-- end .sidebar1 --></div>
<div class="content"><!-- InstanceBeginEditable name="Parte derecha" -->
  <h1>&nbsp;</h1>
<!-- InstanceEndEditable -->
<!-- end .content --></div>
<div class="footer">
  <?php include("includes/pie.php"); ?></div>
<!-- end .container --></div>
</body>
<!-- InstanceEnd --></html>

```

Menú Medicos.

```

<?php
//initialize the session
if (!isset($_SESSION)) {
  session_start();
}

// ** Logout the current user. **
$logoutAction = $_SERVER['PHP_SELF']."?doLogout=true";
if ((isset($_SERVER['QUERY_STRING'])) && ($_SERVER['QUERY_STRING'] != "")){
  $logoutAction .="&". htmlentities($_SERVER['QUERY_STRING']);
}

if ((isset($_GET['doLogout'])) && ($_GET['doLogout']=="true")){

```

```

//to fully log out a visitor we need to clear the session variables
$_SESSION['MM_Username'] = NULL;
$_SESSION['MM_UserGroup'] = NULL;
$_SESSION['PrevUrl'] = NULL;
unset($_SESSION['MM_Username']);
unset($_SESSION['MM_UserGroup']);
unset($_SESSION['PrevUrl']);

$logoutGoTo = "login_medicos.php";
if ($logoutGoTo) {
    header("Location: $logoutGoTo");
    exit;
}
}
?>
<?php
if (!isset($_SESSION)) {
    session_start();
}
$MM_authorizedUsers = "1";
$MM_donotCheckaccess = "false";

// *** Restrict Access To Page: Grant or deny access to this page
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {
    // For security, start by assuming the visitor is NOT authorized.
    $isValid = False;

    // When a visitor has logged into this site, the Session variable MM_Username set equal to
    their username.
    // Therefore, we know that a user is NOT logged in if that Session variable is blank.
    if (!empty($UserName)) {
        // Besides being logged in, you may restrict access to only certain users based on an ID
        established when they login.
        // Parse the strings into arrays.
        $arrUsers = Explode(",", $strUsers);
        $arrGroups = Explode(",", $strGroups);
        if (in_array($UserName, $arrUsers)) {
            $isValid = true;
        }
        // Or, you may restrict access to only certain users based on their username.
        if (in_array($UserGroup, $arrGroups)) {
            $isValid = true;
        }
        if (($strUsers == "") && false) {
            $isValid = true;
        }
    }
    return $isValid;
}

$MM_restrictGoTo = "error_medicos.php";

```

```

if (!(isset($_SESSION['MM_Username'])) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup']))) {
    $MM_qsChar = "?";
    $MM_referrer = $_SERVER['PHP_SELF'];
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
    if (isset($_SERVER['QUERY_STRING']) && strlen($_SERVER['QUERY_STRING']) > 0)
    $MM_referrer .= "?" . $_SERVER['QUERY_STRING'];
    $MM_restrictGoTo = $MM_restrictGoTo. $MM_qsChar . "accesscheck=" .
urlencode($MM_referrer);
    header("Location: " . $MM_restrictGoTo);
    exit;
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/plantillabase.dwt.php" codeOutsideHTMListsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Documento sin título</title>
<script src="SpryAssets/SpryMenuBar.js" type="text/javascript"></script>
<link href="SpryAssets/SpryMenuBarHorizontal.css" rel="stylesheet" type="text/css" />
<!-- InstanceEndEditable -->

<link href="css/estiloprincipal.css" rel="stylesheet" type="text/css" />
<link rel="stylesheet" href="css/menu.css" type="text/css" />

<link href="https://fonts.googleapis.com/css?family=Cormorant+Garamond|Exo"
rel="stylesheet">

<link href="https://fonts.googleapis.com/css?family=Josefin+Slab" rel="stylesheet">

<?php include("includes/google.php"); ?>
<style type="text/css">
body,td,th {
    font-family: "Josefin Slab", serif;
}
</style>
</head>

<body>

<div class="container">
    <div class="header">
        <?php include("includes/cabecera.php"); ?>
        <div class="clearfloat"></div>
        <?php include("includes/menu.php"); ?>
        <?php include("includes/slider.php"); ?>
        </div>
        <div class="sidebar1"><!-- InstanceBeginEditable name="Contenido izquierda" -->

```

```

<p><strong>BIENVENIDO A LA P&Aacute;GINA DE M&Eacute;DICOS</strong></p>
<p>Bienvenido: <?php echo $_SESSION['MM_Username'] ?> este es su men&uacute;
principal de actividades.</p>
<p>&nbsp;</p>
<ul id="MenuBar1" class="MenuBarHorizontal">
  <li><a href="_personas/busqueda1.php">Buscar Personas</a></li>
  <li><a href="_personas/registrohisto_clinica.php">Crear historia cl&iacute;nica</a></li>
  <li><a href="_personas/modificarhistoria.php">Lista de historias cl&iacute;nicas
ingresadas</a></li>
</ul>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p><a href="<?php echo $logoutAction ?>">Desconectar</a></p>
<p>&nbsp;</p>
<!-- InstanceEndEditable -->
<!-- end .sidebar1 --></div>
<div class="content"><!-- InstanceBeginEditable name="Parte derecha" -->
  <h1>&nbsp;</h1>
<!-- InstanceEndEditable -->
<!-- end .content --></div>
<div class="footer">
<?php include("includes/pie.php"); ?></div>
<!-- end .container --></div>
</body>
<!-- InstanceEnd --></html>

```

Búsqueda de información

```

<?php require_once('../Connections/Identificacion.php'); ?>
<?php
if (!function_exists("GetSQLValueString")) {
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue = "")
{
  if (PHP_VERSION < 6) {
    $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
  }

  $theValue = function_exists("mysql_real_escape_string") ?
mysql_real_escape_string($theValue) : mysql_escape_string($theValue);

  switch ($theType) {
    case "text":
      $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
      break;
    case "long":
    case "int":
      $theValue = ($theValue != "") ? intval($theValue) : "NULL";
      break;
    case "double":
      $theValue = ($theValue != "") ? doubleval($theValue) : "NULL";

```

```

        break;
    case "date":
        $theValue = ($theValue != "") ? "" . $theValue . "" : "NULL";
        break;
    case "defined":
        $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
        break;
    }
    return $theValue;
}
}

$colname_busqueda2 = "-1";
if (isset($_GET['Num_Cedula'])) {
    $colname_busqueda2 = $_GET['Num_Cedula'];
}
mysql_select_db($database_Identificacion, $Identificacion);
$query_busqueda2 = sprintf("SELECT * FROM datos_personales WHERE Num_Cedula = %s",
    GetSQLValueString($colname_busqueda2, "text"));
$busqueda2 = mysql_query($query_busqueda2, $Identificacion) or die(mysql_error());
$row_busqueda2 = mysql_fetch_assoc($busqueda2);
$totalRows_busqueda2 = mysql_num_rows($busqueda2);

$colname_busqueda2 = "-1";
if (isset($_GET['busqueda2'])) {
    $colname_busqueda2 = $_GET['busqueda2'];
}
mysql_select_db($database_Identificacion, $Identificacion);
$query_busqueda2 = sprintf("SELECT * FROM datos_personales WHERE Num_Cedula = %s",
    GetSQLValueString($colname_busqueda2, "text"));
$busqueda2 = mysql_query($query_busqueda2, $Identificacion) or die(mysql_error());
$row_busqueda2 = mysql_fetch_assoc($busqueda2);
?>
<!DOCTYPE HTML>
<html><!-- InstanceBegin template="/Templates/plantillaconsulta.dwt.php"
codeOutsideHTMLLsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<!-- InstanceBeginEditable name="doctitle" -->
<title>Busqueda Policia</title>
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" -->
<!-- InstanceEndEditable -->
<link href="../estiloconsultas.css" rel="stylesheet" type="text/css">
</head>

<body>

<div class="container">
<div class="header"><?php include("../includes/cabeceraconsultas.php"); ?></div>
<div class="content"><!-- InstanceBeginEditable name="partederecha" -->
<p><strong>Datos Personales de la Víctima:</strong></p>

```

```

<p> <strong>
  <MM_HIDDENREGION></MM_HIDDENREGION>
</strong></p>
<MM_HIDDENREGION></MM_HIDDENREGION>
<form name="form1" method="post" action="">
  <center>
    <p>&nbsp;</p>
    <table width="718" border="0" align="left">
      <tr>
        <td width="173" align="left" valign="middle"
bgcolor="#6699FF"><strong>Nombres:</strong></td>
        <td width="299" align="left" valign="middle" bgcolor="#C4C4FF"><strong><?php echo
$row_busqueda2['Nombres']; ?></strong></td>
      </tr>
      <tr>
        <td align="left" valign="middle" bgcolor="#6699FF"><strong>Apellidos:</strong></td>
        <td align="left" valign="middle" bgcolor="#C4C4FF"><strong><?php echo
$row_busqueda2['Apellidos']; ?></strong></td>
      </tr>
      <tr>
        <td align="left" valign="middle" bgcolor="#6699FF"><strong>Género:</strong></td>
        <td align="left" valign="middle" bgcolor="#C4C4FF"><strong><?php echo
$row_busqueda2['Genero']; ?></strong></td>
      </tr>
      <tr>
        <td align="left" valign="middle" bgcolor="#6699FF"><strong>Número de
Cédula:</strong></td>
        <td align="left" valign="middle" bgcolor="#C4C4FF"><strong><?php echo
$row_busqueda2['Num_Cedula']; ?></strong></td>
      </tr>
      <tr>
        <td align="middle" bgcolor="#6699FF"><strong>Fecha de Nacimiento:</strong></td>
        <td align="left" valign="middle" bgcolor="#C4C4FF"><strong><?php echo
$row_busqueda2['Fecha_Nacimiento']; ?></strong></td>
      </tr>
      <tr>
        <td align="middle" bgcolor="#6699FF"><strong>Ciudad:</strong></td>
        <td align="left" valign="middle" bgcolor="#C4C4FF"><strong><?php echo
$row_busqueda2['Ciudad']; ?></strong></td>
      </tr>
      <tr>
        <td align="middle" bgcolor="#6699FF"><strong>País:</strong></td>
        <td align="left" valign="middle" bgcolor="#C4C4FF"><strong><?php echo
$row_busqueda2['Pais']; ?></strong></td>
      </tr>
      <tr>
        <td align="middle" bgcolor="#6699FF"><strong>Dirección:</strong></td>
        <td align="left" valign="middle" bgcolor="#C4C4FF"><strong><?php echo
$row_busqueda2['Direccion']; ?></strong></td>
      </tr>
    </table>
  </center>
</form>

```



```

    <!-- end .content --></div>
<div class="footer">
    <?php include("../includes/pieconsultas.php"); ?></div>
<!-- end .container --></div>
</body>
<!-- InstanceEnd --></html>
<?php
mysql_free_result($busqueda2);
?>

```

Eliminación de Usuarios

```

<?php require_once('../Connections/conexioncreacion.php'); ?>
<?php
if (!isset($_SESSION)) {
    session_start();
}
$MM_authorizedUsers = "";
$MM_donotCheckaccess = "true";

// *** Restrict Access To Page: Grant or deny access to this page
function isAuthorized($strUsers, $strGroups, $UserName, $UserGroup) {
    // For security, start by assuming the visitor is NOT authorized.
    $isValid = False;

    // When a visitor has logged into this site, the Session variable MM_Username set equal to
    their username.
    // Therefore, we know that a user is NOT logged in if that Session variable is blank.
    if (empty($UserName)) {
        // Besides being logged in, you may restrict access to only certain users based on an ID
        established when they login.
        // Parse the strings into arrays.
        $arrUsers = Explode(",", $strUsers);
        $arrGroups = Explode(",", $strGroups);
        if (in_array($UserName, $arrUsers)) {
            $isValid = true;
        }
        // Or, you may restrict access to only certain users based on their username.
        if (in_array($UserGroup, $arrGroups)) {
            $isValid = true;
        }
        if (($strUsers == "") && true) {
            $isValid = true;
        }
    }
    return $isValid;
}

$MM_restrictGoTo = "../login_administrador.php";
if (!(isset($_SESSION['MM_Username']) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup'])))) {

```

```

$MM_qsChar = "?";
$MM_referrer = $_SERVER['PHP_SELF'];
if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
if (isset($_SERVER['QUERY_STRING']) && strlen($_SERVER['QUERY_STRING']) > 0)
$MM_referrer .= "?" . $_SERVER['QUERY_STRING'];
$MM_restrictGoTo = $MM_restrictGoTo . $MM_qsChar . "accesscheck=" .
urlencode($MM_referrer);
header("Location: " . $MM_restrictGoTo);
exit;
}
?>
<?php
if (!function_exists("GetSQLValueString")) {
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue = "")
{
if (PHP_VERSION < 6) {
    $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
}

    $theValue = function_exists("mysql_real_escape_string") ?
mysql_real_escape_string($theValue) : mysql_escape_string($theValue);

switch ($theType) {
case "text":
    $theValue = ($theValue != "") ? "" . $theValue . "" : "NULL";
    break;
case "long":
case "int":
    $theValue = ($theValue != "") ? intval($theValue) : "NULL";
    break;
case "double":
    $theValue = ($theValue != "") ? doubleval($theValue) : "NULL";
    break;
case "date":
    $theValue = ($theValue != "") ? "" . $theValue . "" : "NULL";
    break;
case "defined":
    $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
    break;
}
return $theValue;
}
}

if ((isset($_GET['id_usuario'])) && ($_GET['id_usuario'] != "")) {
    $deleteSQL = sprintf("DELETE FROM tbusuarios WHERE id_usuario=%s",
        GetSQLValueString($_GET['id_usuario'], "int"));

    mysql_select_db($database_conexioncreacion, $conexioncreacion);
    $Result1 = mysql_query($deleteSQL, $conexioncreacion) or die(mysql_error());
}

```

```

$deleteGoTo = "usuarios_lista.php";
if (isset($_SERVER['QUERY_STRING'])) {
    $deleteGoTo .= (strpos($deleteGoTo, '?') ? "&" : "?");
    $deleteGoTo .= $_SERVER['QUERY_STRING'];
}
header(sprintf("Location: %s", $deleteGoTo));
}

$colname_eliminar = "-1";
if (isset($_GET['id_usuario'])) {
    $colname_eliminar = $_GET['id_usuario'];
}
mysql_select_db($database_conexioncreacion, $conexioncreacion);
$query_eliminar = sprintf("SELECT * FROM tbusuarios WHERE id_usuario = %s",
GetSQLValueString($colname_eliminar, "int"));
$eliminar = mysql_query($query_eliminar, $conexioncreacion) or die(mysql_error());
$row_eliminar = mysql_fetch_assoc($eliminar);
$totalRows_eliminar = mysql_num_rows($eliminar);
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/plantillaadmin.dwt.php" codeOutsideHTMLOIsLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Administracion Identificate Ecuador</title>
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" -->
<!-- InstanceEndEditable -->
<link href="../css/estiloadmin.css" rel="stylesheet" type="text/css" />
</head>

<body>

<div class="container">
<div class="header">
<?php include("../includes/cabecera_admin.php"); ?></div>
<div class="sidebar1">
<?php include("../includes/menuizquierda_admin.php"); ?></div>
<!-- InstanceBeginEditable name="Admin2" -->
<p>&nbsp;</p>

<p>&nbsp;</p>
<!-- InstanceEndEditable -->

<!-- end .sidebar1 --></div>
<div class="content">

<!-- end .content --></div>
<div class="footer">
<?php include("../includes/pie_admin.php"); ?></div>

```

```

<!-- end .container --></div>
</body>
<!-- InstanceEnd --></html>
<?php
mysql_free_result($eliminar);
?>

```

Modificación de Usuarios

```

<?php require_once('../Connections/conexioncreacion.php'); ?>
<?php
if (!isset($_SESSION)) {
    session_start();
}
$MM_authorizedUsers = "";
$MM_donotCheckaccess = "true";

// When a visitor has logged into this site, the Session variable MM_Username set equal to
their username.
// Therefore, we know that a user is NOT logged in if that Session variable is blank.
if (!empty($UserName)) {
    // Besides being logged in, you may restrict access to only certain users based on an ID
established when they login.
    // Parse the strings into arrays.
    $arrUsers = Explode(",", $strUsers);
    $arrGroups = Explode(",", $strGroups);
    if (in_array($UserName, $arrUsers)) {
        $isValid = true;
    }
    // Or, you may restrict access to only certain users based on their username.
    if (in_array($UserGroup, $arrGroups)) {
        $isValid = true;
    }
    if (($strUsers == "") && true) {
        $isValid = true;
    }
}
return $isValid;
}

$MM_restrictGoTo = "../login_administrador.php";
if (!(isset($_SESSION['MM_Username']) && (isAuthorized("", $MM_authorizedUsers,
$_SESSION['MM_Username'], $_SESSION['MM_UserGroup'])))) {
    $MM_qsChar = "?";
    $MM_referrer = $_SERVER['PHP_SELF'];
    if (strpos($MM_restrictGoTo, "?") $MM_qsChar = "&";
    if (isset($_SERVER['QUERY_STRING']) && strlen($_SERVER['QUERY_STRING']) > 0)
    $MM_referrer .= "?" . $_SERVER['QUERY_STRING'];
    $MM_restrictGoTo = $MM_restrictGoTo. $MM_qsChar . "accesscheck=" .
urlencode($MM_referrer);

```

```

header("Location: ". $MM_restrictGoTo);
exit;
}
?>
<?php
if (!function_exists("GetSQLValueString")) {
function GetSQLValueString($theValue, $theType, $theDefinedValue = "",
$theNotDefinedValue = "")
{
if (PHP_VERSION < 6) {
    $theValue = get_magic_quotes_gpc() ? stripslashes($theValue) : $theValue;
}

    $theValue = function_exists("mysql_real_escape_string") ?
mysql_real_escape_string($theValue) : mysql_escape_string($theValue);

switch ($theType) {
case "text":
    $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
    break;
case "long":
case "int":
    $theValue = ($theValue != "") ? intval($theValue) : "NULL";
    break;
case "double":
    $theValue = ($theValue != "") ? doubleval($theValue) : "NULL";
    break;
case "date":
    $theValue = ($theValue != "") ? "'" . $theValue . "'" : "NULL";
    break;
case "defined":
    $theValue = ($theValue != "") ? $theDefinedValue : $theNotDefinedValue;
    break;
}
return $theValue;
}
}

$editFormAction = $_SERVER['PHP_SELF'];
if (isset($_SERVER['QUERY_STRING'])) {
    $editFormAction .= "?" . htmlentities($_SERVER['QUERY_STRING']);
}

if ((isset($_POST["MM_update"])) && ($_POST["MM_update"] == "form1")) {
    $updateSQL = sprintf("UPDATE tbusuarios SET Apellido=%s, Password=%s, Correo=%s,
Nivel=%s WHERE id_usuario=%s",
        GetSQLValueString($_POST['Apellido'], "text"),
        GetSQLValueString($_POST['Password'], "text"),
        GetSQLValueString($_POST['Correo'], "text"),
        GetSQLValueString($_POST['Nivel'], "int"),
        GetSQLValueString($_POST['id_usuario'], "int"));
}

```

```

mysql_select_db($database_conexioncreacion, $conexioncreacion);
$Result1 = mysql_query($updateSQL, $conexioncreacion) or die(mysql_error());

$updateGoTo = "usuarios_lista.php";
if (isset($_SERVER['QUERY_STRING'])) {
    $updateGoTo .= (strpos($updateGoTo, '?') ? "&" : "?");
    $updateGoTo .= $_SERVER['QUERY_STRING'];
}
header(sprintf("Location: %s", $updateGoTo));
}

$colname_modificar = "-1";
if (isset($_GET['id_usuario'])) {
    $colname_modificar = $_GET['id_usuario'];
}
mysql_select_db($database_conexioncreacion, $conexioncreacion);
$query_modificar = sprintf("SELECT * FROM tbusuarios WHERE id_usuario = %s",
GetSQLValueString($colname_modificar, "int"));
$modificar = mysql_query($query_modificar, $conexioncreacion) or die(mysql_error());
$row_modificar = mysql_fetch_assoc($modificar);
$totalRows_modificar = mysql_num_rows($modificar);
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/plantillaadmin.dwt.php" codeOutsideHTMLOutsideLocked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!-- InstanceBeginEditable name="doctitle" -->
<title>Administracion Identificate Ecuador</title>
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" -->
<!-- InstanceEndEditable -->
<link href="../css/estiloadmin.css" rel="stylesheet" type="text/css" />
</head>

<body>

<div class="container">
<div class="header">
<?php include("../includes/cabecera_admin.php"); ?></div>
<div class="sidebar1">
<?php include("../includes/menuizquierda_admin.php"); ?></div>
<!-- InstanceBeginEditable name="Admin2" -->
<p><strong>Modificacion de datos de Usuario.</strong></p>
<p>&nbsp;</p>
<form action="<?php echo $editFormAction; ?>" method="post" name="form1" id="form1">
<table align="center">
<tr valign="baseline">
<td nowrap="nowrap" align="right">Id_usuario:</td>
<td><?php echo $row_modificar['id_usuario']; ?></td>
</tr>

```

```

<tr valign="baseline">
  <td nowrap="nowrap" align="right">Nombre y Apellido:</td>
  <td><input type="text" name="Apellido" value="<?php echo
htmlentities($row_modificar['Apellido'], ENT_COMPAT, 'iso-8859-1'); ?>" size="32" /></td>
</tr>
<tr valign="baseline">
  <td nowrap="nowrap" align="right">Password:</td>
  <td><input type="text" name="Password" value="<?php echo
htmlentities($row_modificar['Password'], ENT_COMPAT, 'iso-8859-1'); ?>" size="32" /></td>
</tr>
<tr valign="baseline">
  <td nowrap="nowrap" align="right">Correo:</td>
  <td><input type="text" name="Correo" value="<?php echo
htmlentities($row_modificar['Correo'], ENT_COMPAT, 'iso-8859-1'); ?>" size="32" /></td>
</tr>
<tr valign="baseline">
  <td nowrap="nowrap" align="right">Nivel:</td>
  <td><select name="Nivel">
    <option value="0" <?php if (!(strcmp(0, htmlentities($row_modificar['Nivel'],
ENT_COMPAT, 'iso-8859-1')))) {echo "SELECTED";} ?>>Policias</option>
    <option value="1" <?php if (!(strcmp(1, htmlentities($row_modificar['Nivel'],
ENT_COMPAT, 'iso-8859-1')))) {echo "SELECTED";} ?>>Medicos</option>
    <option value="2" <?php if (!(strcmp(2, htmlentities($row_modificar['Nivel'],
ENT_COMPAT, 'iso-8859-1')))) {echo "SELECTED";} ?>>Auxiliares</option>
    <option value="3" <?php if (!(strcmp(3, htmlentities($row_modificar['Nivel'],
ENT_COMPAT, 'iso-8859-1')))) {echo "SELECTED";} ?>>Administrador</option>
  </select></td>
</tr>
<tr valign="baseline">
  <td nowrap="nowrap" align="right">&nbsp;</td>
  <td><input type="submit" value="Actualizar registro" /></td>
</tr>
</table>
<input type="hidden" name="MM_update" value="form1" />
<input type="hidden" name="id_usuario" value="<?php echo $row_modificar['id_usuario'];
?>" />
</form>
<p>&nbsp;</p>
<p><em><a href="usuarios_lista.php">Regresar</a></em></p>
<!-- InstanceEndEditable -->

<!-- end .sidebar1 --></div>
<div class="content">
<!-- end .content --></div>
<div class="footer">
  <?php include("../includes/pie_admin.php"); ?></div>
<!-- end .container --></div>
</body>
<!-- InstanceEnd --></html>
<?php
mysql_free_result($modificar);
?>

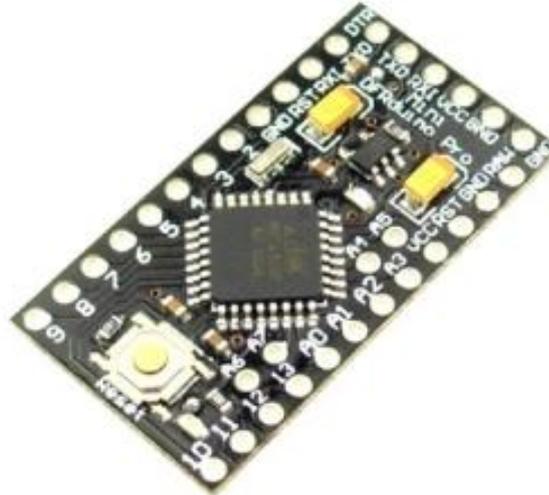
```

ANEXO D. DATASHEET ELEMENTOS ELECTRÓNICOS

DATASHEET – ARDUINO PRO MINI

RB-Dfr-403

DFRduino Pro Mini 3.3V Microcontroller



Introduction

The DFduino Pro Mini 8Mhz(Arduino Pro Mini Compatible) is a microcontroller board based on the ATmega328. It is a 3.3V DFduino running the 8MHz bootloader (select 'Arduino Duemilanove w/ 328' within the Arduino software).

It is an advance version of official Arduino Pro Mini. The DFRduino Pro mini has 8 analog inputs compared with official 6. The rest bits are exactly the same.

It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 8 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. A six pin header can be connected to an FTDI cable or DFRobot breakout board to provide USB power and communication to the board.

The DFRduino Pro Mini is intended for semi-permanent installation in objects or exhibitions. The board comes without pre-mounted headers, allowing the use of various types of connectors or direct soldering of wires. The pin layout is compatible with the Arduino Mini.

Specification

- DC input 3.3V up to 8V(power supply from RAW pin)
- DC input 3.3v(from VCC pin)
- Bootloader: Arduino Pro/Pro Mini (3.3V, 8Mhz) w/ Atmega328
- 3.3V regulator
- Max 150mA output
- ATmega328 running at 8MHz with external resonator (0.5% tolerance)
- Low-voltage board needs no interfacing circuitry to popular 3.3V devices and modules (GPS, accelerometers, sensors, etc)
- USB connection off board

- Max 150mA output
- ATmega328 running at 16MHz with external resonator (0.5% tolerance)
- Low-voltage board needs no interfacing circuitry for popular 3.3V devices and modules (GPS, accelerometers, sensors, etc)
- USB connection off board
- Supports auto-reset
- Over current protected
- Reverse polarity protected
- On board Power and Status LEDs
- Size:0.7x1.3" (18x33mm)

DATASHEET - MÓDULO RFID MF522-ED

Understanding RFID

RFID, or Radio Frequency Identification, is a system for transferring data over short distances (typically less than 6 inches). Often only one of the two devices needs to be powered, while the other is a passive device. This allows for easy use in such things as credit cards, key fobs, and pet collars as there is no need to worry about battery life. The downside is that the reader and the information holder (ie credit card) must be very close, and can only hold small amounts of data.

In this tutorial we will be using the MFRC522 13.56Mhz IC by MIFARE, as described at <https://www.addicore.com/product-p/126.htm>.

If you would like to purchase additional RFID cards or fobs they can be found at the following:

- [RFID Cards 13.56 MHz MF1ICS50](#)
- [RFID Key Fobs 13.56 MHz MF1ICS50](#)

Wiring

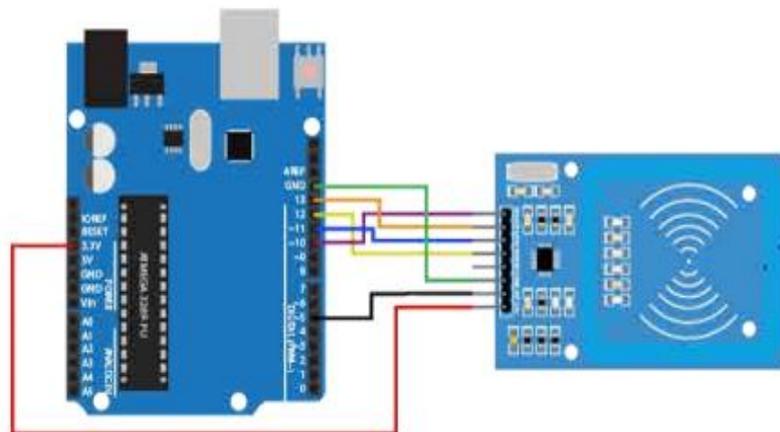
The following table shows the needed connections between the RFID and the Arduino Uno

Cautions:

*On the Arduino many of the pins are not swappable. Because this device uses the SPI bus, who's pins cannot be moved around, pins 11, 12, 13 must remain as shown. RST and IRQ are user specified.

*This device is NOT a 5 volt powered device. You MUST power it with 3.3 volts. If you do not, you risk overheating the RFID. Most Arduino boards include a 3.3V supply pin which can be used to power the RFID module. If 3.3 volts is not accessible, there are [LD33V regulators available at Addicore.com](#) that supply 3.3 volts.

RFID-RC522 Module	Arduino Uno
1 - SDA	Digital 10
2 - SCK	Digital 13
3 - MOSI	Digital 11
4 - MISO	Digital 12
5 - IRQ	--unconnected--
6 - GND	Gnd
7 - RST	Digital 5
8 - 3.3V	3.3v



Adding the Library

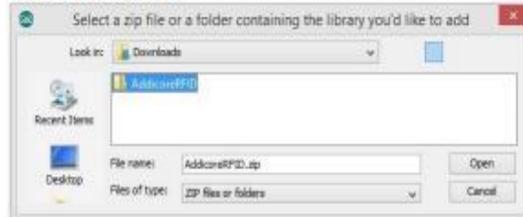
If you haven't already done so, the AddicoreRFID library needs to be added to your Arduino library depository.

1. Download the [AddicoreRFID library](#)

2. Open the Arduino IDE and navigate to *Sketch > Include Library*. A drop down menu will appear. Click the option to "Add .ZIP Library..."



3. A window will open. Navigate to the location where the AddicoreRFID library you downloaded above is currently located and open it.



After the library has been installed the bottom left side of the IDE should show the following message:

Library added to your libraries. Check "Include library" menu

4. You can confirm that the library has been installed by again navigating to *Sketch > Include Library*. The AddicoreRFID library should now show in the list of "Contributed libraries."

Program the Arduino

After installing the AddicoreRFID library in the steps above the library will be available to use in sketches but any example sketches included with the library may not be shown in *File > Examples* until after restarting the Arduino IDE.

1. Restart the Arduino IDE by closing all currently open Arduino IDE windows and then opening the Arduino IDE.
2. When the Arduino IDE opens navigate to *File > Examples > AddicoreRFID*. Select the "Addicore_RFID_Example" sketch. This will open a sketch which we will use with the RFID module that we wired to our Arduino earlier.
3. Now connect your Arduino to your computer and upload the code.

Reading RFID Tags

With your Arduino programmed with the "Addicore_RFID_Example" sketch and with it still plugged into your computer open the serial monitor by clicking the Serial Monitor icon or pressing *CTRL+SHIFT+M*.

If it isn't already selected set the serial baud rate to 9600 baud by clicking on the drop down selection in the lower right corner of the Serial Monitor window as shown below and selecting 9600 baud.



Now take one of the RFID cards or fobs that came in your RFID AddiKit and hold it near the white graphic printed on your RFID-RC522 module as shown below:



Once the module has read the RFID tag your Serial Monitor should show something similar to the following:



When the RFID reader senses a card, it starts by printing "RFID tag detected" on the Serial Monitor. The next line verifies the card type. In normal use, this would be unnecessary. The next two lines print out the data stored in the card, in this case an ID identifying the specific tag scanned.

Note the number 59 from the screenshot above. This is the first byte of the scanned tag's ID. The tag you scan will have a different ID than the one used in this example. In the case of this example I will write down 59 but you will need to write down the number preceding the first comma of the tag's ID that your Serial Monitor shows. Close the Serial Monitor and find in the Addicore_RFID_Example code the following lines of code:

```
// Should really check all pairs, but for now we'll just use the first
if(str[0] == 156) //You can change this to the first byte of your tag
{
  Serial.print("Hello Craig!\n");
} else if(str[0] == 244) { //You can change this to the first byte of your tag
  Serial.print("Hello Erin!\n");
}
```

Change the number in the following line of code to the number you wrote down above, in my case it is 59.

```
if(str[0] == 59) //You can change this to the first byte of your tag
```

You can change the text on the next full line of code to whatever you want to display when you scan the same RFID tag. I will leave mine to say "Hello Craig!"

Upload your changed code to your Arduino, reopen the Serial Monitor, and then scan the same RFID tag as before. Now when you scan the tag the Arduino will recognize the ID and will display the desired text. Below is an example of what mine looks like:



Electronic Brick of HC-06 Serial Port Bluetooth

Overview



What is an electronic brick? An electronic brick is an electronic module which can be assembled like Lego bricks simply by plugging in and pulling out. Compared to traditional universal boards and circuit modules assembled with various electronic components, electronic brick has standardized interfaces, plug and play, simplifying construction of prototype circuit on one' s own. There are many types of electronic bricks, and we provide more than twenty types with different functions including buttons, sensors, Bluetooth modules, etc, whose functions cover from sensor to motor drive, from Ethernet to wireless communication via Bluetooth, and so on. We will continue to add more types to meet the various needs of different projects.

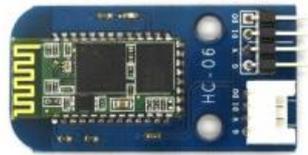
Electronic brick of HC-06 serial port Bluetooth can be connected to hardware UART or analog UART on the control board. With Bluetooth communication, it can achieve wireless transmission which can be applied in various kinds of remote communication occasions.

Features

1. Plug and play, easy to use. Compatible with the mainstream 2.54 buckled interfaces in the market.



2. With resetting button and status indicator



Specifications

PCB Size	39.5mm X 20.5mm X 1.6mm
Working voltage	3.3V DC
Operating voltage	3.3V DC
Compatible interfaces	2.544-pin buckled interface ⁽¹⁾
Communication protocols	Bluetooth 2.0
Communication range	<20m

Note 1: DO for data output (TX) port of UART, DI for data input (RX) port of UART, V and G for voltage at the common collector and ground respectively.

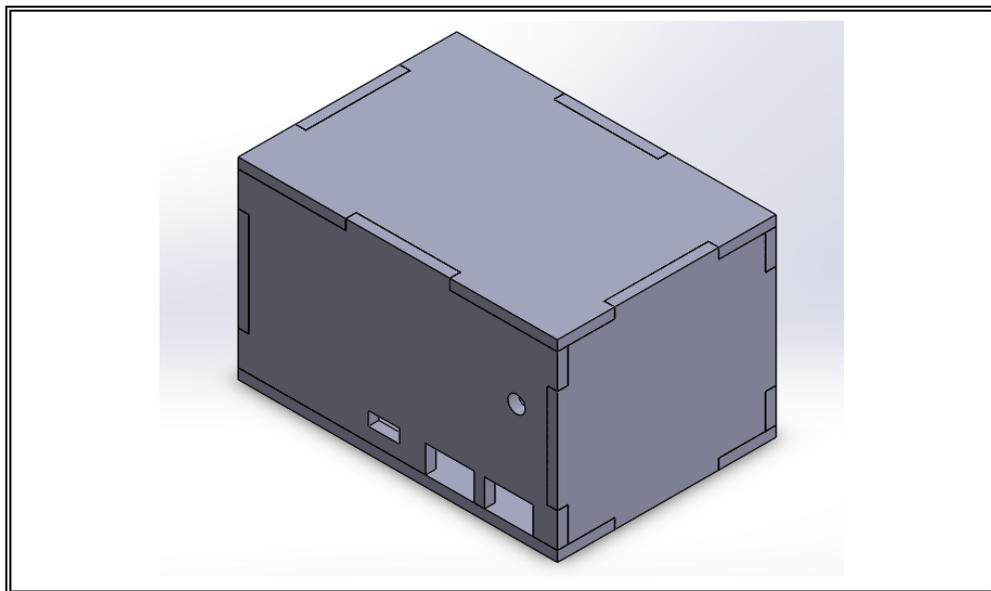
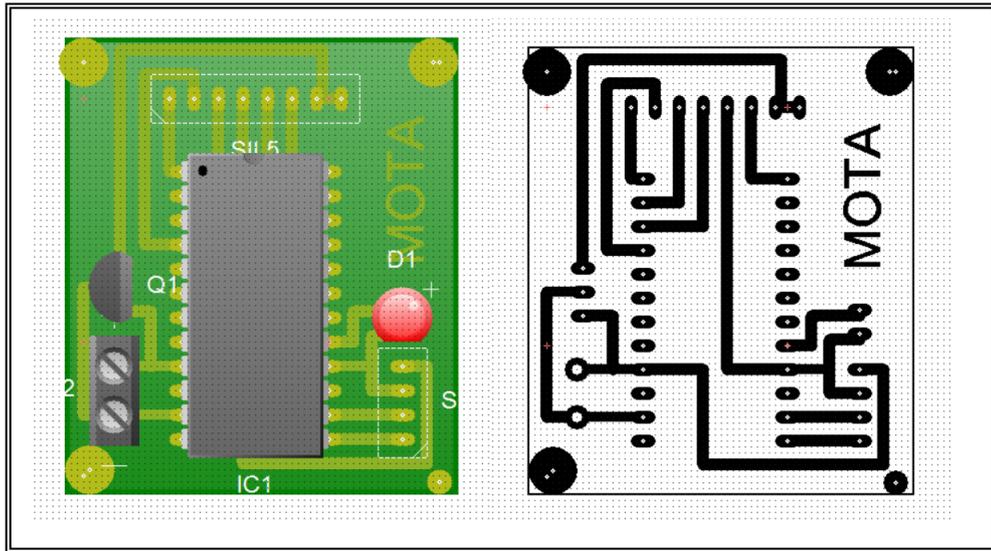
Electrical characteristics

Parameter	Min.	Typical	Max.	Unit
Working voltage	3.0	3.3	4.2	VDC
Digital output voltage (VCC=3.3V)	0	-	3.3	V
Working current (VCC=3.3V)	-	20	40	mA

Switch and Indicator

1. Resetting button
After pressing down resetting button, module will re-enter standby mode and the status lamp will flash.
2. Status lamp
When the module is under standby mode, status lamp will keep flashing; when the module is under connection mode, status lamp will keep being ON.

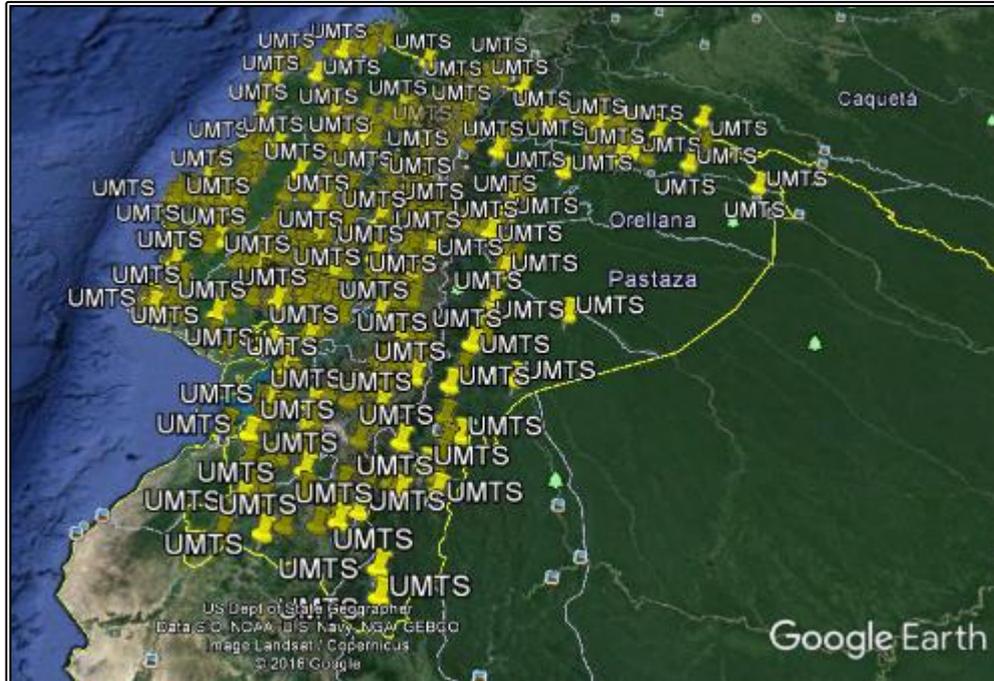
ANEXO E. DISEÑO DE LA PLACA Y ENSAMBLAJE DEL PROTOTIPO.



ANEXO F – COBERTURA – UBICACIÓN DE RADIO BASES

RADIOBASES 3G Y 4G CNT

3G UMTS



4G LTE



RADIOBASES 2G, 3G Y 4G CONETEL

2G GSM



3G UMTS

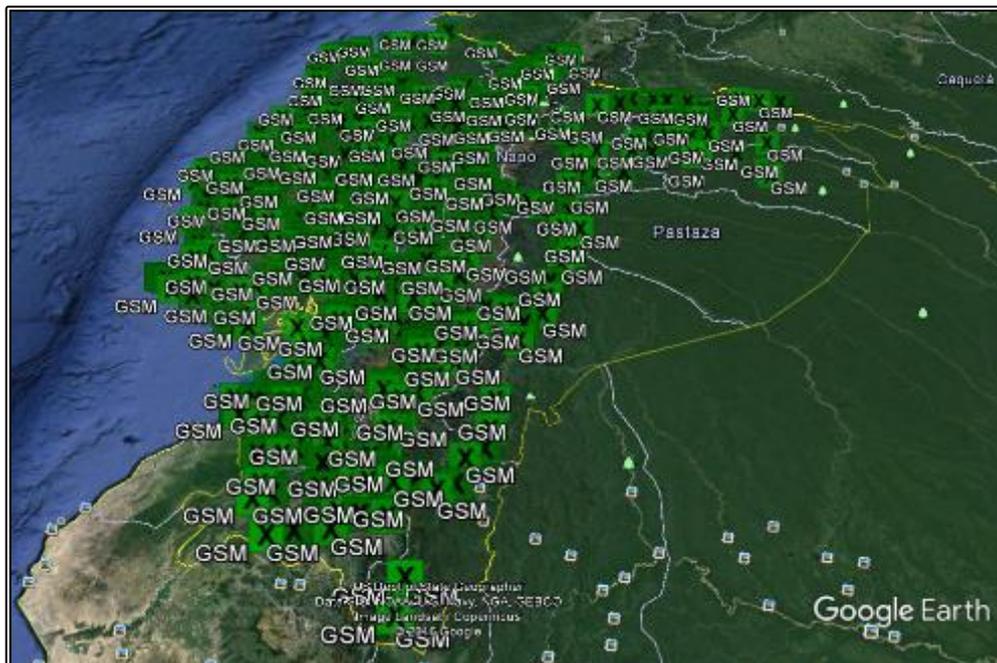


4G LTE

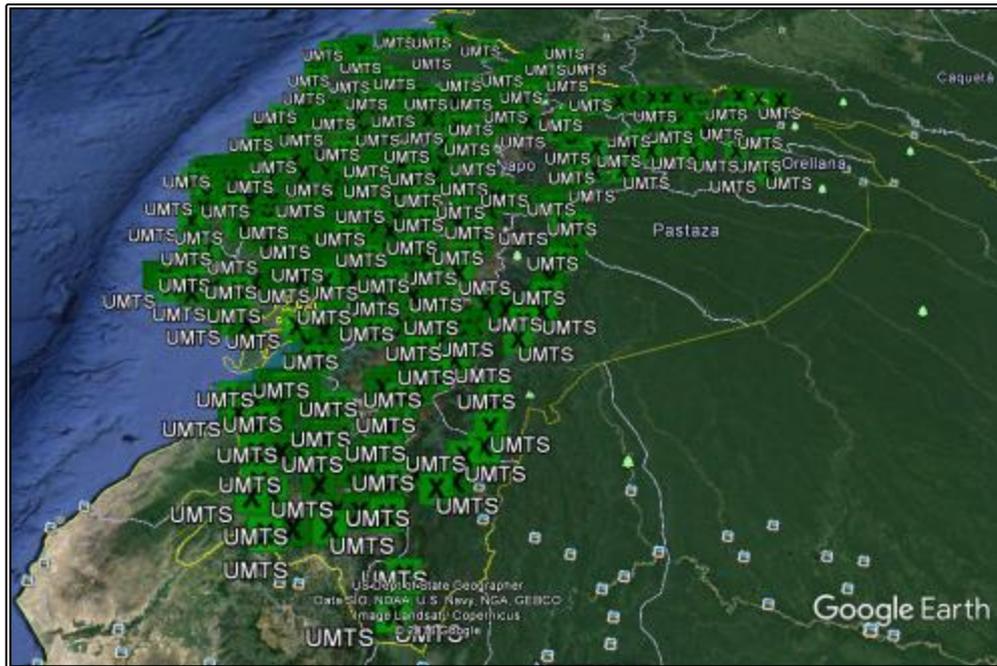


RADIOBASES 2G, 3G Y 4G OTELCEL

2G GMS



3G UMTS



4G LTE

