

**UNIVERSIDAD POLITÉCNICA SALESIANA**

**SEDE CUENCA**

**CARRERA: INGENIERÍA ELECTRÓNICA**

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE IDENTIFICACIÓN Y  
SEGUIMIENTO DE PACIENTES EN HOSPITALES**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:  
INGENIERO ELECTRÓNICO**

Autores:

Orellana Calle Juan Diego

Pesántez Pesántez Xavier Mauricio

Tutores:

Ing. Ángel Soto S. Mg. T

Dr. Ing. Jack Bravo T.

**CUENCA, NOVIEMBRE DE 2016**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros Juan Diego Orellana Calle con C.I. 01040519177 y Xavier Mauricio Pesántez Pesántez con C.I. 0105396972, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de grado intitulado: “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE IDENTIFICACIÓN Y SEGUIMIENTO DE PACIENTES EN HOSPITALES”, mismo que ha sido desarrollado para optar por el título de: Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 14 de noviembre de 2016



Juan Diego Orellana Calle

C.I.: 010405191-7



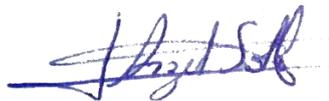
Xavier Mauricio Pesántez Pesántez

C.I.: 010539697-2

## CERTIFICACIÓN

En calidad de TUTOR DEL TRABAJO DE TITULACIÓN “Diseño e implementación de un sistema de identificación y localización de pacientes en hospitales”, elaborado por Orellana Calle Juan Diego y Xavier Mauricio Pesántez Pesántez, declaro y certifico la aprobación del presente trabajo de titulación basándose en la supervisión y revisión de su contenido.

Cuenca, 14 de noviembre de 2016



Ing. Ángel Fernando Soto Sarango Mg.T

TUTOR DEL TRABAJO DE TITULACIÓN

## DECLARATORIA DE RESPONSABILIDAD

Nosotros, Juan Diego Orellana Calle con C.I. 0104051917 y Xavier Mauricio Pesántez Pesántez con C.I. 0105396972, autores del trabajo de titulación “DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE IDENTIFICACIÓN Y SEGUIMIENTO DE PACIENTES EN HOSPITALES”, certificamos que el total contenido de este Proyecto Técnico es de nuestra exclusiva responsabilidad y auditoría.

Cuenca, 14 de noviembre de 2016



Juan Diego Orellana Calle

C.I.: 010405191-7



Xavier Mauricio Pesántez Pesántez

C.I.: 010539697-2

## **AGRADECIMIENTO**

Queremos expresar nuestro más sincero agradecimiento a nuestros tutores de proyecto técnico Ing. Angel Soto Mg.T y al Dr Ing. Jack Bravo, por ayudarnos en el desarrollo del presente trabajo de investigación, quienes con su sabiduría, experiencia y motivación, nos han enseñado a ejecutar adecuadamente los procesos de investigación y desarrollo de este trabajo.

Además queremos agradecer a todos los docentes y personal administrativo de la Universidad Politécnica Salesiana que nos han ayudado en esta etapa de formación y preparación para la vida profesional.

**Juan Diego - Xavier Mauricio**

## **DEDICATORIA**

A mi familia, en especial a mi madre y hermana que me demuestran constantemente que el trabajo duro es la única forma de lograr nuestras metas y que de una forma u otra me han motivado para seguir adelante aún en los momentos más difíciles de la vida. A todos mis compañeros y amigos de la Universidad que han sido parte de estos últimos años en mi vida, en especial Xavier que ha sido un gran amigo en esta etapa.

**Juan Diego**

A mis padres Lucia y Angel, y a mi hermano Fabian, por estar siempre presentes, apoyandome y motivandome a seguir siempre adelante, de ellos he aprendido que el esfuerzo es recompensado tarde o temprano, no hay que rendirse hasta conseguir lo que se quiere. Son lo mas importante en mi vida y gracias a ellos he logrado conseguir mi meta universitaria. Tambien a mi compañero y amigo Juan Diego, quien ha estado presente en este trayecto de estudiante universitario.

**Xavier Mauricio**

## ÍNDICE GENERAL

DEDICATORIA .....	VI
ÍNDICE GENERAL.....	VII
ÍNDICE DE FIGURAS.....	X
ÍNDICE DE TABLAS.....	XII
ABREVIATURAS.....	XIII
ANTECEDENTES .....	XIV
INTRODUCCIÓN.....	XV
<b>CAPÍTULO 1. ESTUDIO DE LAS TECNOLOGÍAS DE IDENTIFICACIÓN DE PACIENTES .....</b>	<b>1</b>
1.1 Seguridad de Pacientes en Hospitales .....	1
1.2 Identificación de Pacientes en el Ecuador.....	2
1.3 Sistemas de Identificación de Pacientes .....	3
1.3.1 Código de Barras .....	3
1.3.2 Tarjetas Magnéticas.....	4
1.3.3 Reconocimiento Óptico de Caracteres.....	5
1.3.4 Identificación por Radio Frecuencia .....	6
1.3.5 NFC.....	6
1.4 Identificación de Pacientes con Etiquetas Electrónicas y Radio Frecuencia 8	
1.5 RFID, Identificación por Radio Frecuencia .....	9
1.5.1 Etiquetas.....	10
1.5.2 Lectores .....	10
1.5.3 Antenas .....	11
1.5.4 Hardware de Procesamiento .....	11
1.6 Aplicaciones de RFID en Hospitales.....	12
<b>CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA .....</b>	<b>14</b>
2.1. Arquitectura del Sistema .....	14
2.2. Diseño e implementación del sistema. ....	17
2.2.1. Diseño de la red de sensores para el sistema de localización .....	17
2.2.2. Diseño, construcción y programación de dispositivo lector de pulseras RFID 18	

2.2.2.1.	Diseño y construcción de hardware .....	18
2.2.2.2.	Lector RDM630.....	18
2.2.2.3.	Manillas con Etiquetas RFID.....	19
2.2.2.4.	Microcontrolador .....	19
2.2.2.5.	Módulo Wi-Fi ESP8266.....	20
2.2.2.6.	Fuentes de Alimentación .....	21
2.2.2.7.	Estructura del Lector RFID.....	22
2.2.2.8.	Diseño e implementación de software del lector RFID .....	25
2.2.3.	Servidor Web.....	25
2.2.4.	Sistema de localización de Personal y Pacientes.....	26
2.2.4.1.	Algoritmo de Búsqueda de la ruta más corta.....	26
2.2.5.	Diseño e implementación de la aplicación móvil en Android.....	27
2.2.5.1.	Opción Paciente.....	29
2.2.5.2.	Opción Médico .....	31
2.2.5.3.	Opción Enfermera .....	32
2.2.5.4.	Opción Localización .....	32
2.2.5.5.	Opción Gestionar Personal .....	33
2.2.5.6.	Opción Cerrar Sesión .....	34
2.3.	Montaje del sistema.....	34
<b>CAPÍTULO 3. ANÁLISIS DE DATOS Y RESULTADOS .....</b>		<b>35</b>
3.1.	Pruebas de envío de Datos .....	35
3.1.1.	Ingreso del Paciente al Hospital.....	35
3.1.2.	Datos de Localización del Paciente .....	36
3.1.3.	Datos de Localización del Personal .....	36
3.2.	Pruebas de la Aplicación .....	37
3.2.1.	Paciente .....	37
3.2.2.	Médico y Enfermera .....	39
3.2.3.	Localización .....	40
3.2.4.	Historial.....	42
3.3.	Pruebas de RF .....	42
3.3.1.	Alcance y Atenuación Wi-Fi.....	42
3.3.1.1.	Con obstáculos.....	42
3.3.1.2.	Sin obstáculos .....	44
3.4.	Presupuesto .....	46

<b>CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>50</b>
<b>4.1 Conclusiones .....</b>	<b>50</b>
<b>4.2 Recomendaciones .....</b>	<b>51</b>
<b>ANEXOS .....</b>	<b>52</b>
<b>Anexo 1: Código de Arduino para los Lectores.....</b>	<b>53</b>
<b>Anexo 2: Scripts PHP.....</b>	<b>62</b>
<b>Anexo 3: Estructura de la Base de Datos .....</b>	<b>84</b>
<b>Anexo 4: Código de la Aplicación.....</b>	<b>88</b>
<b>Anexo 5: Código de la ruta más corta .....</b>	<b>114</b>
<b>BIBLIOGRAFÍA.....</b>	<b>121</b>

## ÍNDICE DE FIGURAS

<b>Figura 1:</b> Capas de la plataforma SAFER [26].....	XVI
<b>Figura 2:</b> Brazaletes para Identificación de Pacientes [6].....	2
<b>Figura 3:</b> Código de Barras del Sistema EAN [9]. .....	3
<b>Figura 4:</b> Código de Barras del Sistema EAN [9]. .....	4
<b>Figura 5:</b> Tarjeta Magnética según el estándar ISO/IEC 7811 [9].....	4
<b>Figura 6:</b> Tecnología OCR, a) Código de Barras, b) Empaque Final [12].....	5
<b>Figura 7:</b> Estructura General de un sistema RFID [14] .....	6
<b>Figura 8:</b> NFC modo de Emulación de Tarjeta [15].....	7
<b>Figura 9:</b> Etiqueta RFID u-Chip de la empresa Hitachi [15].....	10
<b>Figura 10:</b> Lectores RFID a) fijos, b) móviles [22].....	11
<b>Figura 11:</b> Antenas RFID de Motorola [23].....	11
<b>Figura 12:</b> Sistema RFID en el Hospital Universitario de Valencia [25] .....	13
<b>Figura 13:</b> Arquitectura del sistema SAFER [26]. .....	15
<b>Figura 14:</b> Arquitectura del sistema de identificación y seguimiento de pacientes. ....	15
<b>Figura 15:</b> Flujograma del sistema. ....	16
<b>Figura 16:</b> Sistema de Localización .....	18
<b>Figura 17:</b> Módulo Lector RFID RM630 [28] .....	19
<b>Figura 18:</b> Microcontrolador (MCU) Arduino MEGA [30]. .....	20
<b>Figura 19:</b> Módulo Wi-Fi ESP8266 [45].....	21
<b>Figura 20:</b> Fuente 5 VDC - 2A.....	22
<b>Figura 21:</b> Adaptador 5 VDC a 3.3 VDC.....	22
<b>Figura 22:</b> Esquema de conexión de los componentes .....	23
<b>Figura 23:</b> Placa para el acople de los módulos con el Arduino .....	24
<b>Figura 24:</b> Dispositivo montado en la carcasa.....	24
<b>Figura 25:</b> Pantalla de Ingreso al Sistema .....	28
<b>Figura 26:</b> Menú Principal de la Aplicación .....	28
<b>Figura 27:</b> Submenú Paciente .....	29
<b>Figura 28:</b> Opciones Internar/Ingresar, Dar Alta y Registrar .....	30
<b>Figura 29:</b> Consulta y datos del paciente .....	31
<b>Figura 30:</b> Submenú Médico y Opción Mis Datos .....	32
<b>Figura 31:</b> Localización, mapa y ruta más corta hacia el paciente .....	33
<b>Figura 32:</b> Submenú para gestión del personal.....	33
<b>Figura 33:</b> Prueba del Ingreso de un Paciente al Hospital .....	37
<b>Figura 34:</b> Prueba de la Alta Médica a un Paciente.....	38
<b>Figura 35:</b> Prueba Registro y Búsqueda de Pacientes .....	38

<b>Figura 36:</b> Prueba Registro y Búsqueda de Pacientes .....	39
<b>Figura 37:</b> Prueba Registro y Búsqueda de Pacientes .....	39
<b>Figura 38:</b> Localización de Paciente y Enfermera en el Hospital .....	40
<b>Figura 39:</b> Ruta más corta entre el paciente y la enfermera a cuidado del paciente.....	40
<b>Figura 40:</b> Localización de Pacientes y Personal en distintas áreas del hospital .....	41
<b>Figura 41:</b> Ruta más corta entre Pacientes y Personal en distintas áreas del hospital .....	41
<b>Figura 42:</b> Consulta de Historial Médico de Pacientes.....	42
<b>Figura 43:</b> Curva de cobertura entre Lector y Punto de Acceso con obstáculos .....	44
<b>Figura 44:</b> Curva de cobertura entre Lector y Punto de Acceso sin obstáculos .....	46

## ÍNDICE DE TABLAS

<b>Tabla 1:</b> Tabla comparativa entre RFID y NFC. ....	9
<b>Tabla 2:</b> Especificaciones técnicas Arduino MEGA [30]. ....	20
<b>Tabla 3:</b> Tabla comparativa Módulos Wi-Fi [31]. ....	21
<b>Tabla 4:</b> Conexión de Pines entre módulos. ....	23
<b>Tabla 5:</b> Usuarios y Contraseñas de prueba. ....	34
<b>Tabla 6:</b> Verificación de datos y tiempos ....	35
<b>Tabla 7:</b> Datos y tiempos de localización de pacientes. ....	36
<b>Tabla 8:</b> Datos y tiempos de localización de personal. ....	36
<b>Tabla 9:</b> Pruebas de cobertura entre Lector y Punto de Acceso con obstáculos ....	43
<b>Tabla 10:</b> Pruebas de cobertura entre Lector y Punto de Acceso sin obstáculos. ....	44
<b>Tabla 11:</b> Presupuesto para la construcción del prototipo del sistema. ....	46
<b>Tabla 12:</b> Presupuesto para la construcción del sistema de largo alcance ....	48

## ABREVIATURAS

TIC	Tecnologías de la Información y Comunicación
RF	Radiofrecuencia
RFID	Radio Frequency IDentification (Identificación por RF)
NFC	Near Fields Communication (Comunicación de Campo Cercano)
OCR	Optical Character Recognition (Reconocimiento Óptico de Carácteres)
PHP	Hypertext Preprocessor (Preprocesador de Hipertexto)
URL	Uniform Resource Locator (Localizador de Recursos Uniforme)
AT	Comando “Atención” del lenguaje para configuración de módems por Hayes Communications
INEC	Instituto Nacional de Estadística y Censos.
APP	Application
MCU	MicroController Unit

## ANTECEDENTES

La seguridad médica de pacientes en hospitales a nivel mundial representa un grave problema [1]. Estudios indican que alrededor de 100000 personas mueren cada año en Estados Unidos debido a errores médicos que pueden ser prevenidos [1]. A nivel nacional no existen estudios que evalúen cuantitativamente la mala práctica médica ni la seguridad de pacientes, ya que ni siquiera existe una ley que regule las mismas.

Para tratar de garantizar la seguridad de los pacientes, se utilizan sistemas de identificación. En Ecuador el sistema más utilizado es el de la manilla, en la cual se registran los datos primordiales del paciente; Sin embargo, este sistema no está unificado a nivel nacional y cada hospital maneja su propia norma [6,7]. Esta solución es muy limitada ya que no permite un monitoreo constante y por tanto no es posible garantizar la seguridad del paciente en todo momento.

A causa de esta problemática se ha decidido desarrollar un sistema que permita asistir la seguridad de los pacientes dentro de un centro médico. El sistema monitorea constantemente la localización del paciente, y lleva un registro de la misma y de su historial médico. De esta forma se puede mejorar la seguridad y eficiencia en el centro de salud.

Mediante este proyecto aspiramos obtener el título de ingeniero electrónico con mención en telecomunicaciones. Con la implementación de este sistema ponemos en práctica los conocimientos adquiridos a través de estos años de estudio en la universidad.

Este trabajo es una etapa en el desarrollo de la plataforma **SAFER** (Por sus siglas en inglés, Smart Health Assistant for Elderly Care) llevado a cabo por Grupo de Investigación GITEL de la Universidad Politécnica Salesiana.

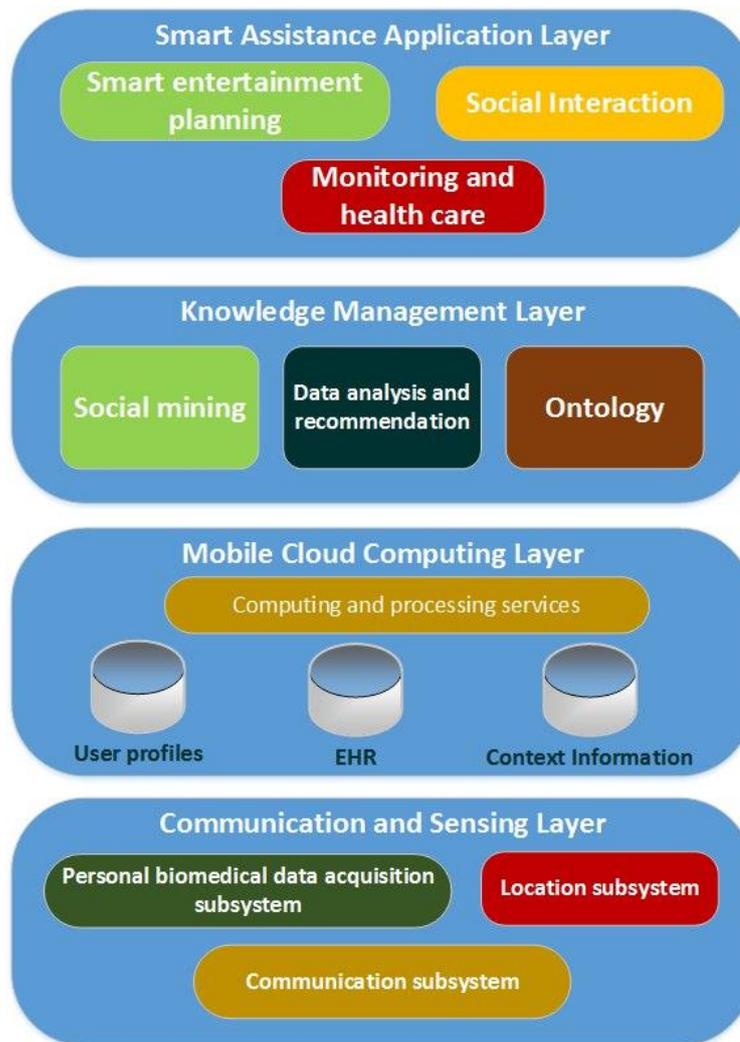
## INTRODUCCIÓN

Según variadas estadísticas [1], la seguridad de los pacientes en hospitales es un grave problema en la comunidad médica. Para tratar de minimizar los errores médicos se hacen uso de tecnologías de identificación. El gran impacto que han tenido las TIC, no puede ser despreciado y se han propuesto varias tecnologías candidatas para sustituir al tradicional brazalete de identificación médico. Entre las varias propuestas se tiene RFID (Por sus siglas en inglés, Radio Frequency IDentification) la cual gana principalmente en costes respecto a otras tecnologías [3,8].

La tecnología de Identificación por Radiofrecuencia tiene un notable desarrollo en los últimos años. Estos desarrollos tienen impacto en lo referente al costo de fabricación de la tecnología ya que se reduce en gran medida y se logra equiparlo al de otros métodos de identificación, como el código de barras [3,8].

A causa del entorno en el que se atienden a los pacientes y al uso de tecnologías desactualizadas se decide desarrollar e implementar un prototipo de un sistema de identificación y seguimiento de pacientes. Este trabajo se realiza bajo los lineamientos de SAFER; el cual es un proyecto del grupo de investigación en telecomunicaciones y telemática (GITEL) de la Universidad Politécnica Salesiana, sede Cuenca bajo la supervisión del Dr. Jack Fernando Bravo Torres.

La plataforma SAFER, véase figura 1, se constituye de 4 capas: i) Capa de Comunicaciones, se encarga de recolectar información sobre el estado de salud y localización de las personas usando interfaces inalámbricas (Wi-Fi, RFID, ZigBee, redes celulares etc.), ii) Capa de Nube, analiza y procesa la información, iii) Capa de Manejo de conocimiento, en la que se gestionan bases de datos y perfiles de usuario de cada uno de los sistemas, y iv) Capa de Aplicaciones, agrupa programas inteligentes para la notificación de caídas, localización de personal y pacientes, entre otras [26].



**Figura 1:** Capas de la plataforma SAFER [26].

La finalidad de este sistema es que este se adecue al usuario, de esta forma se evitan incomodidades y se reduce la curva de aprendizaje para que el beneficiario pueda utilizar todas las funciones previstas.

Debido al gran alcance del proyecto SAFER, este se compone de varias fases. En nuestro caso aportaremos con la etapa denominada “Diseño e implementación de un sistema de identificación y seguimiento de pacientes en Hospitales”. El mencionado podrá identificar al paciente utilizando una pulsera con un chip. Además de detectar la localización del mismo en base a coordenadas fijas asignadas a cada lector. Los lectores señalan un ambiente definido del hospital y constantemente monitorean las etiquetas que se encuentran en las pulseras antes mencionadas.

Cuando un paciente llega al hospital se le ingresa en el sistema mediante una manilla única la cual contiene información inalterable por medios tradicionales. De esta forma se sustituye las pulseras con información escrita en ellas como medio de identificación.

La información del hospital, personal y pacientes es gestionada mediante una base de datos y una aplicación desarrollada para el sistema operativo Android. De esta forma se gana versatilidad al permitir que el personal pueda consultar la información clínica de los pacientes incluso editarla si se trata de médicos o administradores. Un administrador por su parte tiene la ventaja de poder gestionar a los doctores o enfermeras, agregando o eliminando a estos del sistema.

El sistema también permite la localización del personal y pacientes en el centro de salud, mediante una ayuda visual en la aplicación. Al mismo tiempo se puede trazar la ruta del médico hacia el paciente en el mapa mostrado.

El desarrollo de este sistema está enfocado en alcanzar los siguientes objetivos: diseñar e implementar el hardware y software necesarios para el funcionamiento del sistema, realizar pruebas que validen el funcionamiento de los productos generados en el apartado anterior, en especial pruebas de cobertura y de la aplicación móvil.

Este trabajo consta de 4 capítulos. En el primero se estudia la problemática, presentando estadísticas sobre errores médicos y también los sistemas que se están implementando actualmente para la identificación de pacientes, haciendo énfasis en la tecnología RFID como principal candidata. En el segundo capítulo se presentan los detalles sobre el diseño y construcción del sistema, además de dar descripciones del funcionamiento del mismo. En el tercer capítulo se presentan las pruebas y resultados obtenidos al poner en funcionamiento el sistema, para finalmente en el cuarto capítulo presentar las conclusiones, así como algunas recomendaciones para el desarrollo futuro del sistema.

## **CAPÍTULO 1. ESTUDIO DE LAS TECNOLOGÍAS DE IDENTIFICACIÓN DE PACIENTES**

En este capítulo se muestran las distintas tecnologías de identificación de pacientes que existen en la actualidad y la tendencia que se está adoptando tanto a nivel mundial, como en el país. Se definen a las tecnologías de identificación como tal y se muestra la integración de las TIC en las mismas, además se proporcionan argumentos para la elección de RFID como tecnología para nuestro sistema. Al final se hace una comparación de los sistemas de identificación y localización que existen en otros países en los que se hace empleo de RFID.

### **1.1 Seguridad de Pacientes en Hospitales**

Según estadísticas del Instituto de Medicina de Estados Unidos, la seguridad de los pacientes en un centro de salud es un problema grave en la comunidad médica. Alrededor de 44000 a 98000 personas mueren cada año resultado de errores médicos, los cuales pueden ser prevenidos [1]. Se han observado altas tasas de errores en salas de cuidados intensivos y en el área de cirugía [1].

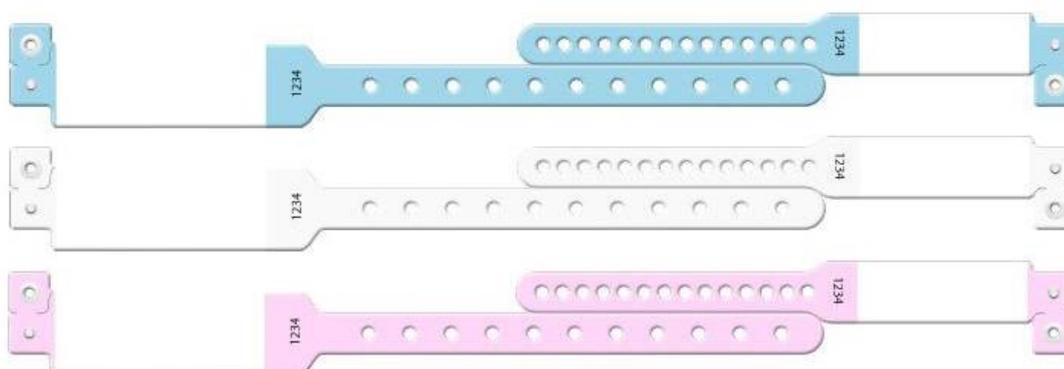
Cabe destacar que los fallos se ocasionaron debido a que el personal no sigue el protocolo o se presentan condiciones, procesos y sistemas defectuosos que inducen a los funcionarios al error [1, 2]. Estudios recientes demuestran que la administración de drogas equivocadas al paciente es responsable entre 1 y 10% de casos de hospitalización por mala práctica médica [2,3].

Las TIC surgen como una oportunidad para minimizar las oportunidades del error humano. Actualmente varias tecnologías de identificación de pacientes y equipamiento médico (manillas, tarjetas electrónicas, dispositivos adheribles, etc.) que han sido desarrolladas para este fin [4].

## 1.2 Identificación de Pacientes en el Ecuador

Según la OMS (Organización Mundial de Salud), la Identificación de Pacientes es de gran importancia para la seguridad del paciente a fin de evitar daños y salvar vidas [5]. De acuerdo a información del Hospital Provincial General Docente de la ciudad de Riobamba, en el año 2013, se firmó un convenio entre el Ministerio de Salud Pública del Ecuador y la ONG “Accreditation Canada International” para acreditar 44 hospitales nacionales y obtener un mejor nivel de calidad. Entre los objetivos a cumplir del convenio firmado se encuentra la norma de Identificación de Pacientes, la cual garantiza la calidad asistencial y previene errores que puedan poner en peligro la vida del paciente [6].

Actualmente, en el Ecuador, para la identificación de pacientes se cuenta con el método del brazalete, véase figura 2, en el cual solo se escribe información básica perteneciente al paciente [6].



**Figura 2:** Brazaletes para Identificación de Pacientes [6].

Según información proporcionada por el Hospital Vicente Corral Moscoso de la ciudad de Cuenca, se menciona que, el brazalete debe cumplir con ciertas características según el tipo de paciente que lo vaya a utilizar, ya sean adultos, niños o recién nacidos, además de que el color también identifica el estado de cada paciente, siendo los más críticos aquellos que son etiquetados con el color naranja como pacientes de alto riesgo [7].

En el sistema de salud ecuatoriano este método no es reciente, sin embargo, anteriormente cada centro de salud contaba con su propia normativa para la

identificación de pacientes, ahora lo que se busca es la implementación de un sistema de identificación único (SIU) para todos los pacientes que reciben atención médica en los establecimientos hospitalarios del país [7].

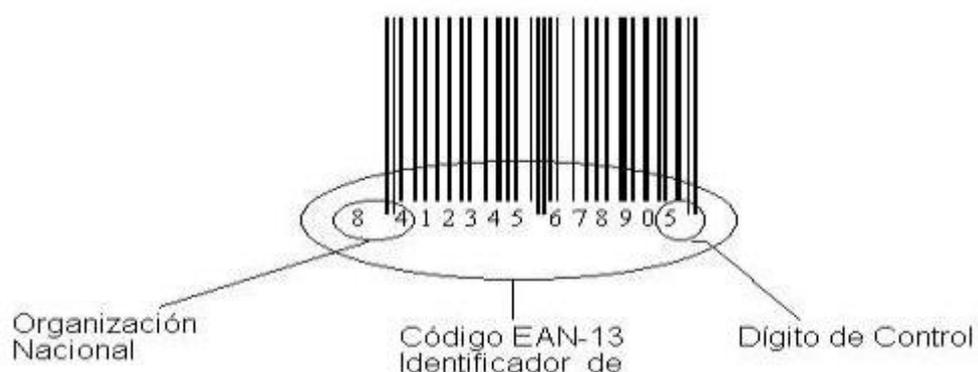
### 1.3 Sistemas de Identificación de Pacientes

En la actualidad, existen diversas tecnologías de identificación de pacientes, tales como código de barras, tarjetas magnéticas, OCR (del inglés, Optical Character Recognition), RFID, NFC (del inglés, Near Field Communication), entre otras [8]. A continuación, se explican las tecnologías más relevantes incluido RFID, para centrarse en la siguiente sección en esta última.

#### 1.3.1 Código de Barras

Consiste en una serie de barras de color negro y espacios en blanco, de diferentes anchos, los cuales están impresos sobre una etiqueta, estas barras a blanco y negro son leídas mediante un escáner, el cual mide la luz reflejada y lo transforma en números y letras.

En Europa, en el año 1977 se desarrolla el sistema EAN (véase figura 3) y después se generalizó internacionalmente mediante el sistema UPC (Universal Product Code). Generalmente se utilizaba para productos como latas, botellas, libros, muebles, cajas, etc. [9].



**Figura 3:** Código de Barras del Sistema EAN [9].

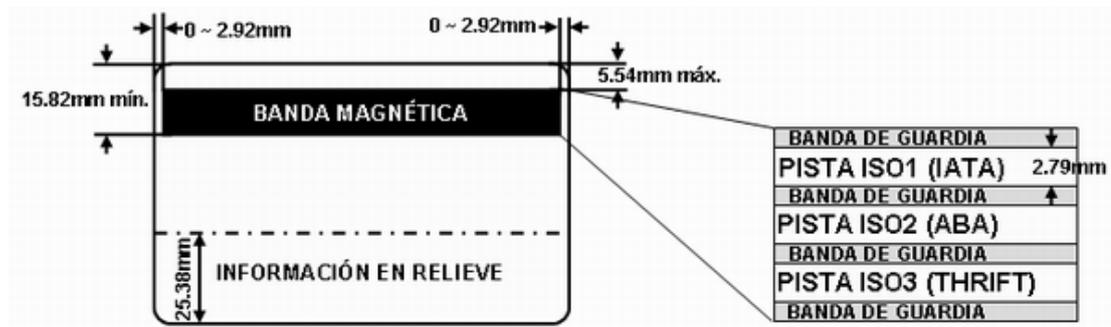
En la actualidad se utiliza el código de barras en varios hospitales en EEUU, la Unión Europea y otros países de tercer mundo, para almacenar la información de un paciente, controlar la medicación y el inventario, a modo de ejemplo véase la figura 4 [9].



**Figura 4:** Código de Barras del Sistema EAN [9].

### 1.3.2 Tarjetas Magnéticas

Estas tarjetas constan de una banda magnética hecha de finas partículas magnéticas mezcladas con una resina e impresas sobre una tarjeta de plástico [10,11], cómo se observa en la figura 5. Este sistema de información es pasivo, ya que requiere otros dispositivos para leer y grabar la información. Esta tecnología utiliza cualquier campo magnético externo cercano a la banda, por lo que se puede afectar el contenido de la información. También se tienen problemas de deterioro de la banda magnética debido al uso. Estas tarjetas son utilizadas como Tarjetas de crédito, Tarjetas de débito, Cerraduras Electrónicas, Empresas, Hospitales, etc. [10,11].



**Figura 5:** Tarjeta Magnética según el estándar ISO/IEC 7811 [9].

### 1.3.3 Reconocimiento Óptico de Caracteres

Reconocimiento óptico de caracteres (OCR, por sus siglas en inglés Optical Character Recognition) es una tecnología que transforma los documentos físicos en digitales a través de procesos de reconocimiento de imágenes y caracteres [12,13]. Las imágenes son captadas mediante una cámara digital y enviadas a un computador o un dispositivo con un software que permita OCR, es una tecnología muy empleada en smartphones como puede verse en la figura 6 [12,13].

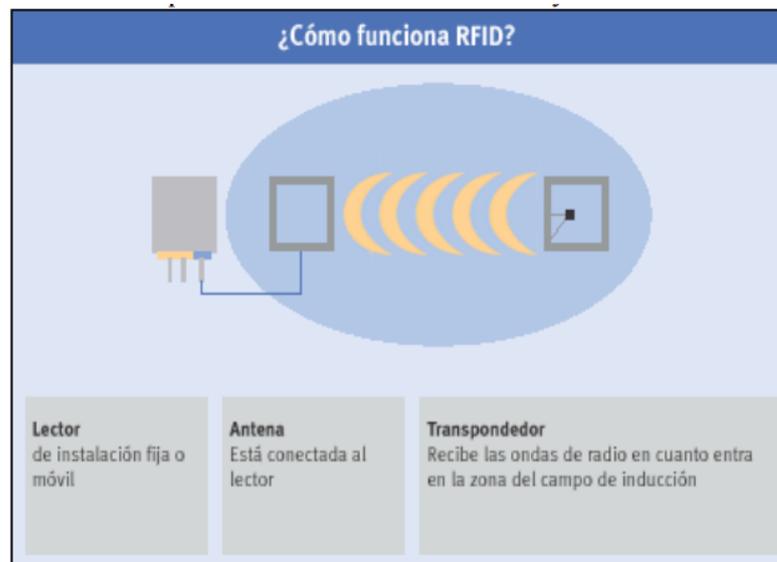


**Figura 6:** Tecnología OCR, a) Código de Barras, b) Empaque Final [12].

### 1.3.4 Identificación por Radio Frecuencia

El Sistema de Identificación por radio frecuencia (RFID, por sus siglas en ingles Radio Frequency Identification) es una tecnología usada en la identificación de objetos a distancia, sin necesidad de contacto directo, ni visual [14].

Básicamente un sistema RFID, véase figura 7, consta de: la etiqueta que consiste en un chip de silicio en conjunto con una antena, encapsulados en un solo cuerpo; un lector con su respectiva antena, el cual es capaz de leer y modificar los datos de la etiqueta. Las bandas de frecuencia que trabaja RFID son LF, HF y UHF. Generalmente es utilizado en inventario y seguimiento de bienes y productos en Centros Comerciales, Hospitales, Empresas, etc. [14].



**Figura 7:** Estructura General de un sistema RFID [14]

### 1.3.5 NFC

La Comunicación de campo cercano (NFC por sus siglas en ingles Near Field Communications) es una tecnología que opera en la banda de 13.56 MHz, la cual permite intercambiar información a corta distancia entre dos dispositivos, esta tecnología se basa en el estándar ISO/IEC 14443 [15].

NFC opera a potencias bajas, por lo que la distancia entre los dos dispositivos que deseen identificarse deberá ser muy corta, por debajo de los 10 cm. Esta tecnología es muy utilizada en los teléfonos móviles [15]. Existen 3 modos de operación de NFC:

- Emulación de Tarjeta: En este modo de operación el dispositivo NFC puede emular el comportamiento de una tarjeta inteligente sin contacto (Estándar ISO/IEC 14443), lo que permite al dispositivo ser utilizado en aplicaciones de pago y dinero virtual [15], se muestran ejemplos en la figura 8.
- Peer to Peer: Este modo es para comunicación bidireccional entre dos dispositivos NFC para intercambiar información entre ellos. [15].
- Lectura/Escritura: En este modo el dispositivo NFC puede leer y escribir el contenido de las etiquetas NFC, este modo es utilizado en sistemas de localización geográfica, información de productos en supermercados, entre otros, en donde se ubican tarjetas en diferentes puntos, por lo cual el usuario solo tendrá que acercarse su dispositivo NFC y obtener la información de una manera rápida y sencilla [15].



**Figura 8:** NFC modo de Emulación de Tarjeta [15]

#### **1.4 Identificación de Pacientes con Etiquetas Electrónicas y Radio Frecuencia**

Una manera de minimizar los errores médicos es la implementación de sistemas de identificación de pacientes. Estas permiten obtener una información sobre el historial médico del paciente y otros datos clínicos relevantes a la movilidad del paciente en el centro médico [16]. Un sistema de identificación con radiofrecuencia debería permitir realizar las siguientes acciones:

- Gestión en Pacientes: Con el uso de un PDA, lector o un dispositivo móvil y una etiqueta electrónica se pretende tener datos médicos del paciente en tiempos muy cortos [16].
- Identificación y Seguridad: Se debería permitir la identificación en tiempo real de las acciones de los pacientes, especialmente en áreas críticas como las salas de cuidados intensivos [16].
- Inventario Instantáneo: Se debería permitir el manejo de la información de inventario con el uso de la etiqueta, ya que con el registro de cada paciente se actualiza el inventario en el hospital, de esta manera se obtienen fechas y lotes al inventariar [16].

Con el uso de estas tecnologías se pretende asegurar la administración correcta de medicamentos a pacientes, además de eliminar el tiempo de búsqueda de historiales médicos, el tiempo dedicado a la toma de apuntes manuales y la automatización de muchas otras tareas. A la final el resultado deseado es gestionar los datos de una manera más eficaz, reducir tiempos y ahorrar costos [16].

RFID y NFC poseen ventajas frente a otros sistemas de identificación debido al uso de etiquetas electrónicas en las cuáles la información puede ser manipulada, a comparación de los sistemas de códigos de barras, brazaletes en los cuales la información es inmutable o en las tarjetas magnéticas, las cuales tienen problemas de alteración de información debido al desgaste. Al utilizar RFID o NFC se logran optimizar los tiempos de acceso y fiabilidad de la información [17].

En la Tabla 1, se presentan las principales características de las tecnologías RFID y NFC a modo de cuadro comparativo.

**Tabla 1:** *Tabla comparativa entre RFID y NFC.*

<b>Tecnología</b>	<b>RFID</b>	<b>NFC</b>
<b>Modos de Operación</b>	Bidireccionales o Half Duplex	Bidireccional
<b>Intercambio de Información</b>	Código pregrabado definido por el usuario	Depende del modo de aplicación
<b>Rango de Frecuencias</b>	125 KHz, 134 KHz, 13.56 MHz, 433 MHz, 860-960 MHz, 2.45 GHz	13.56 MHz
<b>Terminales</b>	Específicos para cada rango de frecuencia y fabricante	Teléfono móvil con hardware NFC
<b>Estándares</b>	ISO, solo interfaz de transmisión.  EPN, etiquetas e interfaz de transmisión.	ISO, interfaz y protocolo de transmisión, protocolos de prueba e interfaz
<b>Costo</b>	Medio-Alto	Medio-Bajo
<b>Velocidad de Transmisión</b>	80 kbps, 160kbps, 320 kbps, 650 kbps	Entre 106 kbps, y 464 kbps

RFID tiene ventajas en lo que respecta a modos de operación y radio de alcance de los lectores, sin embargo, NFC destaca en lo que refiere a costos y la sencillez de configuración [18, 19].

### **1.5 RFID, Identificación por Radio Frecuencia**

RFID (Radio Frequency IDentification) es una tecnología usada para automatizar los datos e identificar productos, artículos, componentes, personas e incluso animales con el uso de dispositivos conocidos como etiquetas o tags [8].

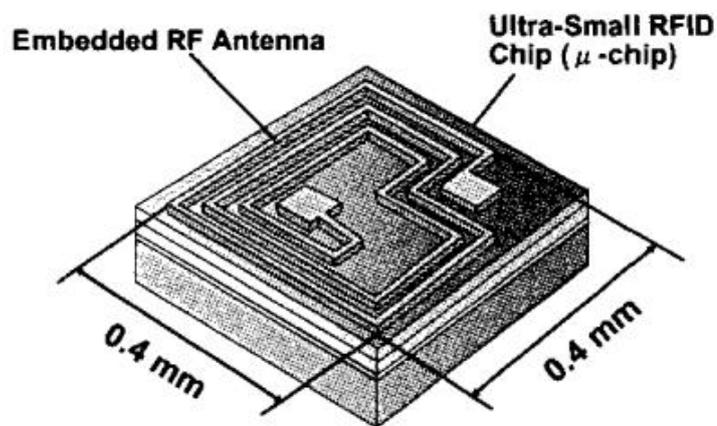
La tecnología tiene sus orígenes en los años 40 [8], sin embargo, los primeros sistemas con una cobertura de varios metros para RFID aparecieron en los años 70. Las frecuencias a las que opera esta tecnología se encuentran entre los 860 MHz y los 960 MHz, aunque existen variaciones según el fabricante y las regulaciones del país donde se implementan [20].

En un sistema RFID existen cuatro componentes, etiquetas, lectores, antenas y hardware de procesamiento. A continuación, se explican brevemente:

### 1.5.1 Etiquetas

Son dispositivos que tienen una antena, un chip de silicio, un receptor y transmisor de RF, un modulador. Además del número de identificación único las etiquetas [8,21].

Existen dos tipos de etiquetas, pasivas y activas. Las primeras absorben energía del campo electromagnético generado por los lectores, a modo de ejemplo véase la figura 9, ya que no incorporan una batería. Las segundas cuentan con su alimentación propia para la comunicación con los lectores [8].



**Figura 9:** Etiqueta RFID u-Chip de la empresa Hitachi [15]

### 1.5.2 Lectores

Estos transmiten pulsos de energía mediante RF, los mismos son recibidos por las etiquetas, las cuales al detectar el pulso emiten una respuesta que es detectada por el mismo lector. Existen dos tipos de lectores, los fijos que se ubican en puntos estratégicos del lugar y lectores móviles los cuales puede transportar una persona [8], para una referencia física véase la figura 10.



**Figura 10:** Lectores RFID a) fijos, b) móviles [22]

### 1.5.3 Antenas

Se utilizan para transferir información entre los lectores y las etiquetas, el diseño afecta en gran medida el rendimiento del sistema [8,21], los tipos de antenas usados varían según la aplicación, véase la figura 11 para una referencia de las mismas.



**Figura 11:** Antenas RFID de Motorola [23]

### 1.5.4 Hardware de Procesamiento

Se trata de una estructura de datos que se utiliza para procesar la información que obtienen los lectores. Se suele utilizar servidores de bases de datos para almacenar y manejar la información, la cual es finalmente visualizada en un computador, tablet, smartphone, etc [8].

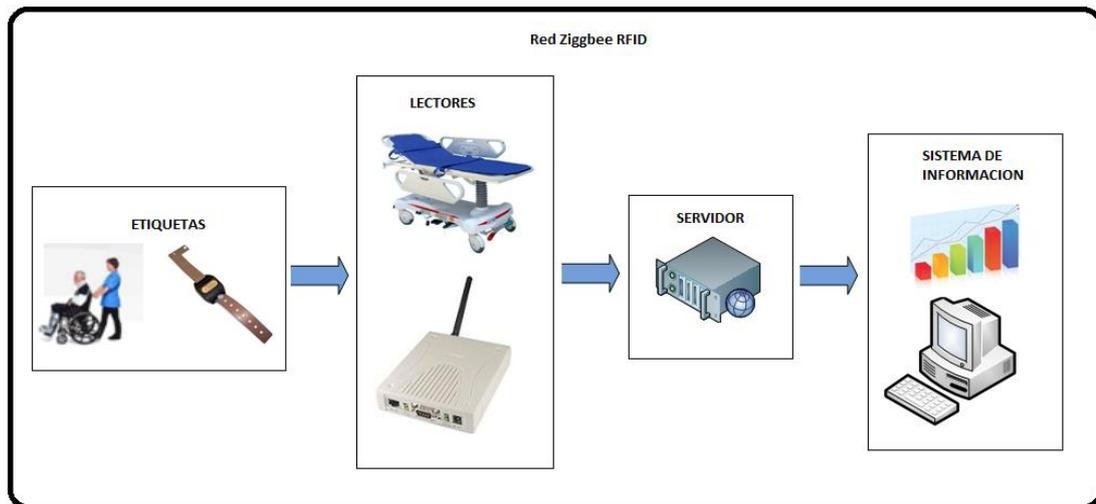
## 1.6 Aplicaciones de RFID en Hospitales

En los hospitales de Estados Unidos, se estima que el coste por robo de equipamiento y bienes es de US \$ 4000.00 anuales por cada cama, lo que implica pérdidas de 3,9 miles de millones de dólares anuales [25]. También se destaca cifras asociadas con elementos olvidados en el cuerpo de un paciente después de una operación, un total de 1500 objetos al año, a continuación, se muestran algunos casos hospitalarios en donde se utiliza la tecnología RFID de diversas maneras para solucionar estos y otros inconvenientes [25].

En el Hospital Costa del Sol de Marbella, España, se utiliza un sistema de identificación de pacientes RFID, en el cual se cuenta con etiquetas RFID de alta frecuencia dentro de pulseras para almacenar la información de cada paciente. El personal del hospital utiliza lectores RFID para identificar a cada paciente y asegurar que se ha suministrado la medicación correcta y a tiempo [24].

En el Hospital Universitario de Valencia (España) se utiliza un sistema de localización en tiempo real (RTLS, por sus siglas en inglés Real Time Locating System) para el seguimiento de pacientes y de los bienes (sillas de ruedas, ecógrafos portátiles, desfibriladores, etc.) en las instalaciones del hospital [25]. Las etiquetas RFID están colocadas en las pulseras de los pacientes y en cada uno de los bienes del hospital. Cada etiqueta es activa y transmite un número de identificación único cada dos segundos a través de una señal de 2.4GHz [25].

Los números de identificación de las etiquetas están vinculados a través del software SphericaHospital que reside en un servidor local y luego es enviado al sistema de información hospitalaria. Se utiliza en combinación de la tecnología ZigBee, la cual no interfiere con la red Wi-Fi del hospital. El sistema incluye 82 lectores montados en los carros del personal de enfermería, con un puerto USB conectado a una computadora, por lo que la información del paciente cercano se muestra en la pantalla (véase figura 12). Este hospital controla 1500 pacientes a la vez mediante el sistema RTLS [25].



**Figura 12:** Sistema RFID en el Hospital Universitario de Valencia [25]

Otro caso de aplicación RFID hospitalaria se da en el hospital Mercy Medical (Iowa), en el cual se realiza seguimiento de sus bienes (balones de oxígeno, cables de filtro, etc.). Este sistema utiliza etiquetas pasivas de HF. Se cuenta con lectores fijos ubicados en puntos estratégicos y 16 lectores móviles llamados botiquines, los cuales realizan recuentos automáticos de los inventarios, además de un software WaveMark para monitorizar y gestionar inventarios [24].

## **CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DEL SISTEMA**

En este capítulo se presentan los puntos más relevantes sobre el diseño e implementación del sistema. Este trabajo es la continuación del desarrollo de la plataforma SAFER que es llevada a cabo por parte del Grupo de Investigación en Telecomunicaciones (GITEL) de la Universidad Politécnica Salesiana Sede Cuenca.

Este capítulo se encuentra dividido en tres secciones: en la primera se muestra toda la arquitectura bajo la cual está construido el sistema; En la segunda se indica el diseño e implementación del sistema, explicando los elementos de hardware usados para la construcción de los prototipos y las principales pautas seguidas en el desarrollo de la aplicación para dispositivos Android. En la tercera sección se muestran las pautas principales que se deben tener en cuenta para el montaje del sistema en un centro de salud.

### **2.1. Arquitectura del Sistema**

Debido a que este trabajo es la continuación del desarrollo del sistema SAFER la arquitectura del sistema ya se ha presentado anteriormente [26]; por lo que a continuación se presentará la misma de manera general haciendo énfasis en la parte de identificación y localización de pacientes, véase figura 13.

El sistema cinco tiene bloques independientes [26], que se mencionan a continuación:

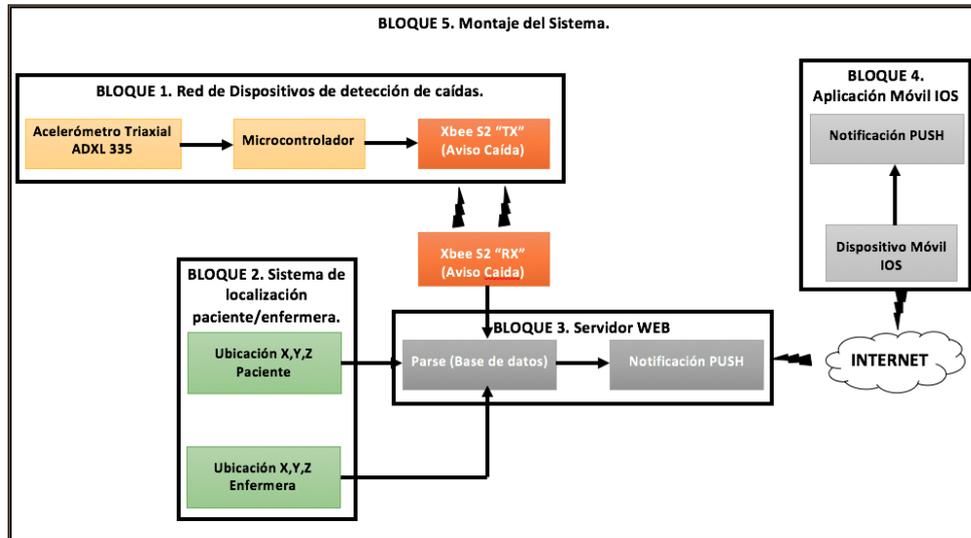
**Bloque 1:** Red de dispositivos de detección de caídas.

**Bloque 2:** Sistema de localización de pacientes/enfermeras.

**Bloque 3:** Servidor Web.

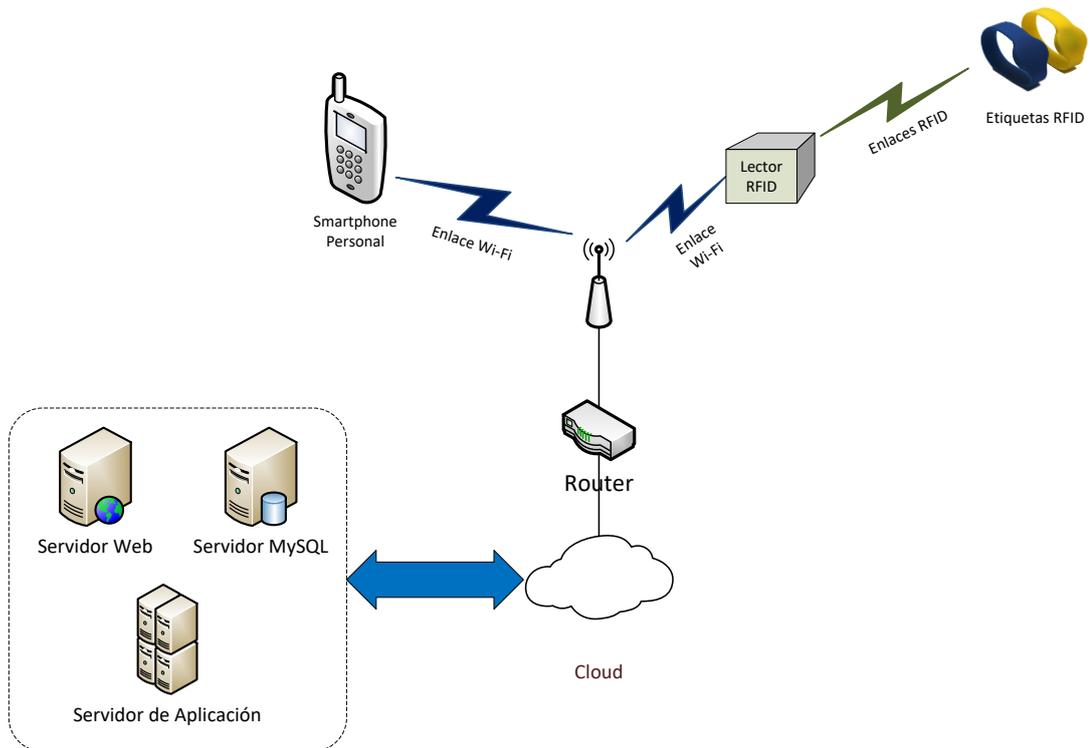
**Bloque 4:** Aplicación móvil.

**Bloque 5:** Montaje del Sistema.



**Figura 13:** Arquitectura del sistema SAFER [26].

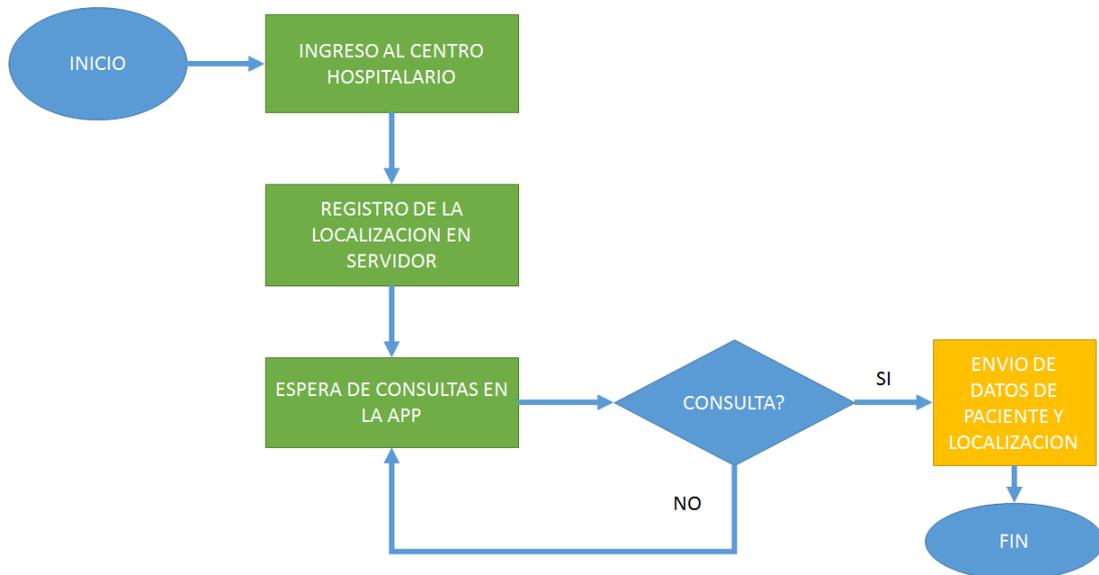
Nuestro sistema de identificación de pacientes en concreto funciona con la arquitectura mostrada en la figura 14, en la que tanto los lectores como los smartphones con la aplicación comparten una misma red Wi-Fi dentro del hospital.



**Figura 14:** Arquitectura del sistema de identificación y seguimiento de pacientes.

**Fuente:** Los autores.

El sistema de identificación y localización de pacientes sigue el diagrama de flujo mostrado en la figura 15.



**Figura 15:** *Flujograma del sistema.*

**Fuente:** *Los autores.*

El software que se utilizará para el desarrollo del sistema y las funciones mostradas en las figuras 14 y 15 se describe brevemente a continuación:

**Arduino**, para la programación de los módulos WiFi y RFID que en su conjunto se encargan de enviar la identificación y localización de los pacientes hacia un servidor web.

**Android Studio**, software para la programación de la aplicación en el sistema operativo Android, basado en Java y XML.

**MySQL**, gestor de base de datos para la gestión de la información de pacientes, personal médico y del sistema de localización.

**PHP**, lenguaje de programación de páginas web que permite la comunicación de la aplicación con el sistema gestor de base de datos.

## **2.2. Diseño e implementación del sistema.**

El diseño del sistema se hace en cuatro etapas. Se hace énfasis en la explicación de las tres primeras etapas que corresponden al desarrollo de nuestro proyecto.

- Diseño de la red de sensores para el sistema de localización.
- Diseño, construcción y programación de dispositivo lector de pulseras RFID.
- Diseño y programación de la aplicación móvil en Android.
- Programación de scripts PHP para Servidor Web.

A continuación, se presentarán los materiales requeridos para la implementación de cada uno de los bloques, así como una pequeña descripción de la función de cada uno en el sistema.

### **2.2.1. Diseño de la red de sensores para el sistema de localización**

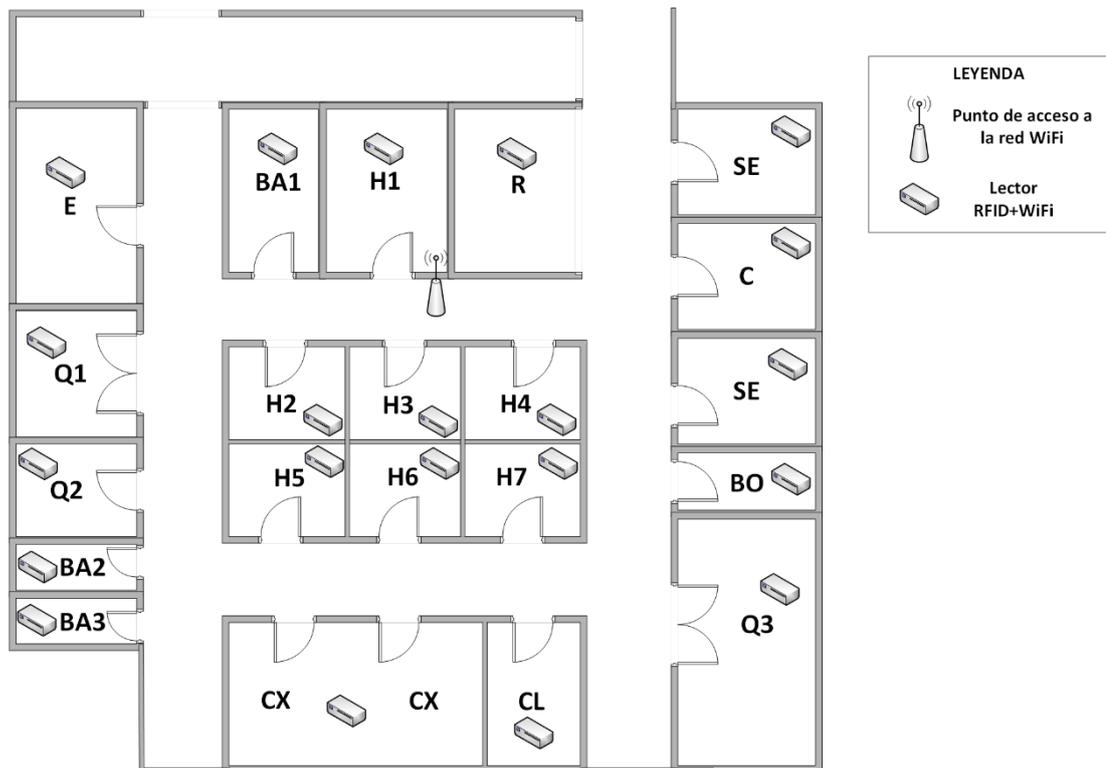
Para alcanzar nuestros objetivos se ha visto necesario el diseñar una red de sensores, la cual está compuesta por varios lectores que a su vez poseen varias tecnologías incorporadas en los mismos.

Se debe usar la tecnología RFID para la identificación de pacientes, debido a su bajo coste y consumo de energía comparada con otras. Además, es necesario utilizar el protocolo Wi-Fi ya que es de uso habitual en hospitales y que no representa riesgos adicionales en su adopción, además de su bajo precio. Gracias a esto se puede cambiar la posición de los lectores sin necesidad de cambios mayores en la infraestructura de la red local.

Todos los sensores distribuidos se conectan a un nodo principal por piso. Cada lector cubre un área específica. El sistema transmite continuamente los datos de localización del paciente y personal a la base de datos.

Cada hospital posee una base de datos que permite controlar la información de los pacientes y el personal, así como la localización en el mismo. De esta manera se logra optimizar tiempos de acceso y mayor fiabilidad de la información.

Por ejemplo, en un piso cualquiera de un hospital los sensores tendrían la disposición mostrada en la figura 16.



**Figura 16:** Sistema de Localización

Fuente: Los autores.

## 2.2.2. Diseño, construcción y programación de dispositivo lector de pulseras RFID

### 2.2.2.1. Diseño y construcción de hardware

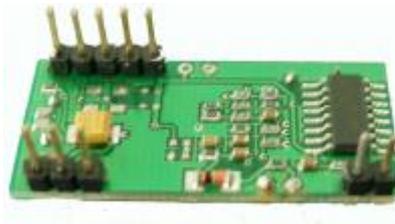
Este dispositivo se ha desarrollado con la función de leer las etiquetas RFID, identificando al paciente al asignarle una pulsera con una etiqueta durante su estancia en el hospital. Además, debido a que los lectores son fijos y cubren una determinada área del centro de salud se logrará tener localizado al paciente. Estos dispositivos leen las etiquetas continuamente y actualizan la posición de los mismos en la base de datos. En las siguientes secciones se detallan los dispositivos utilizados para la construcción del prototipo aquí mencionado.

### 2.2.2.2. Lector RDM630

Para la elección del módulo lector RFID se consideró como punto principal el coste, por lo que se decidió que las etiquetas utilizadas para el sistema sean del tipo pasivo, ya que los precios de las activas pueden multiplicar el costo de 5 a 10 veces por etiqueta.

El lector utilizado en nuestro caso es el RDM630, se seleccionó debido a los costes y a que la implementación es un prototipo, sin embargo, se pretende que para implementaciones futuras del sistema se utilicen lectores más complejos que posean un mayor alcance.

El lector elegido es de corto alcance, trabaja en el rango de LF a 125 kHz con un rango de lectura de 1 a 5 centímetros [28]. El módulo puede ser manejado mediante una conexión serial RS232 la cual trabaja a 9600 baudios [28]. La tensión de alimentación es de 5 VDC con un consumo de corriente de 50 mA [28], véase la figura 17 para una referencia gráfica del mismo.



**Figura 17:** *Módulo Lector RFID RM630 [28]*

**Fuente:** *Los autores.*

Como sugerencia para futuras implementaciones se podría utilizar lectores como los de la compañía LinkSprite los cuáles son trabajan en UHF 860 – 915 MHz y además incorporan una antena tipo panel que en conjunto logran rangos de lectura de 8 a 15 metros [29].

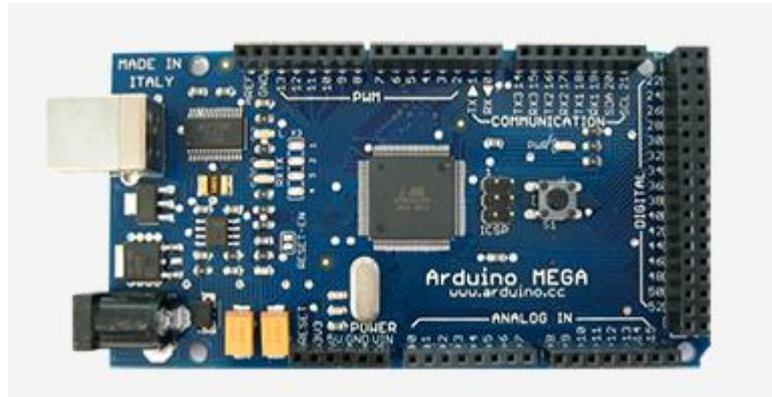
### **2.2.2.3. Manillas con Etiquetas RFID**

Las manillas escogidas trabajan a la frecuencia de 125 kHz, estas además están fabricadas de silicona hipo-alérgica y son resistentes a agua y sudor. Además, en la totalidad de normas hospitalarias consultadas estas se utilizan en la muñeca por lo tanto no se requieren manillas ajustables, aunque el sistema funcionaría de igual manera si se utilizara cualquier otro tipo de manilla en tanto se respete la frecuencia de funcionamiento.

### **2.2.2.4. Microcontrolador**

Se ha seleccionado un Arduino Mega 2560, véase figura 18, como microcontrolador para coordinar la operación de los módulos anteriormente mencionados, este

básicamente se encarga de dar las ordenes de lectura de etiquetas RFID y de comunicar los resultados de la lectura al módulo Wi-Fi para el envío de los datos al servidor Web.



**Figura 18:** Microcontrolador (MCU) Arduino MEGA [30].

**Tabla 2.** Especificaciones técnicas Arduino MEGA [30].

Procesador	Operación/ Voltaje de ingreso	Veloci- dad CPU	Analogicas In/Out	Digiales IO/PW M	EEPRO M [KB]	SRAM [KB]	USB	UART
ATmega 1280	5V / 7-12V	8MHz	16/0	54/15	4	8	Micr o	4

#### 2.2.2.5. Módulo Wi-Fi ESP8266

Se ha optado por elegir como medio de comunicación inalámbrica debido a que esta tecnología está implementada en la mayor parte de los hospitales, además de que los costos son muy accesibles ya sea para implementarla o actualizarla. Se plantearon dos candidatos para escoger como módulo Wi-Fi el primero de ellos el NRF24L01, el cual es capaz de programarse para ser compatible con los estándares IEEE 802.11xx y el segundo el módulo ESP8266 el cual es directamente compatible con los estándares mencionados. Además, este último puede ser operado mediante comandos AT enviados por una interfaz serial en el módulo. A modo de comparación se muestra en la tabla 5 las características de los dos módulos mencionados [31,33].

**Tabla 3:** Tabla comparativa Módulos Wi-Fi [31,33].

Producto	Protocolos	Frecuencia	Descripción	Velocidad máxima de transmisión	Voltaje/Corriente de Operación	Interfaz
NRF24L01	RF, IEEE 802.11b	2.4 GHz	Módulo transceptor de bajo consumo compatible con ISM	2Mbps	1.9 – 3.6 V 14 mA	SPI
ESP8266	IEEE 802.11b/g/n	2.4 GHz	Módulo de transmisión de RF	250Kbps	3.3V 1000 mA	RS232

**Fuente:** Los autores.

Con motivo de disminuir el costo y de hacer el dispositivo compatible con las tecnologías existentes se escogió el módulo ESP8266. El dispositivo se muestra en la figura 19.



**Figura 19:** Módulo Wi-Fi ESP8266 [32].

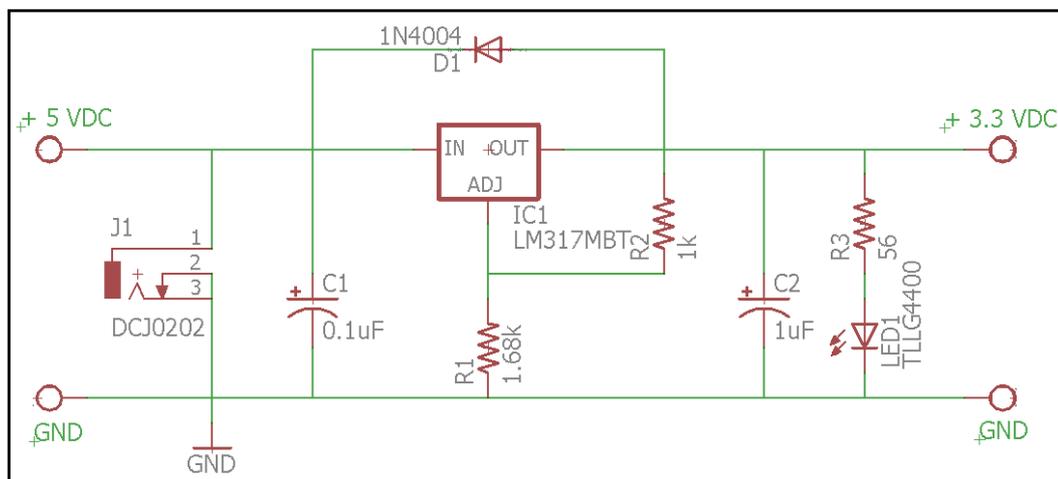
#### 2.2.2.6. Fuentes de Alimentación

La fuente de alimentación principal es externa, con los requerimientos de corriente del Arduino y los módulos, se utilizó una fuente de 5VDC con una corriente de salida de 2 A (véase figura 20); La cual se conecta a un tomacorriente para alimentar a todo el dispositivo lector. Para la parte de la alimentación del módulo Wi-Fi se diseñó un adaptador de 5 a 3.3 VDC, usando con un LM317. En la figura 21 se presenta el esquema del adaptador.



**Figura 20:** Fuente 5 VDC - 2A.

**Fuente:** Los autores.



**Figura 21:** Adaptador 5 VDC a 3.3 VDC.

**Fuente:** Los autores.

### 2.2.2.7. Estructura del Lector RFID

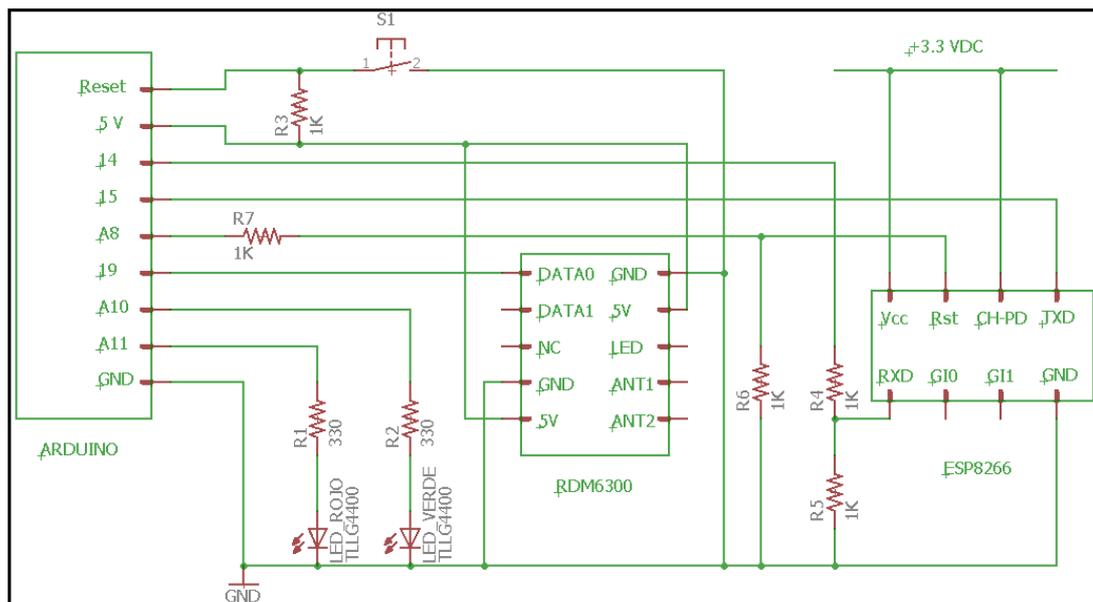
Para facilitar el uso y protección del dispositivo se ha visto necesario hacer uso de una caja comercial para proyectos y adaptarla según el tamaño del dispositivo. El dispositivo permite además el desacoplo de los módulos que se encuentran montados sobre el Arduino haciendo uso de una PCB diseñado para este propósito.

Con el hardware seleccionado según los argumentos expuestos se procede a conectar los componentes según se muestra en el esquema mostrado en la figura 22, además se agregó dos leds que muestran el estado del dispositivo y un botón de Reset para reiniciar todo el programa en el Arduino. Las conexiones de los pines de los módulos utilizados se presentan en la Tabla 4.

**Tabla 4:** Conexión de Pines entre módulos.

ARDUINO MEGA	MÓDULO ESP8266	MODULO RFID - RMD 6300
14 - TX3	RXD	
15 - RX3	TXD	
19 - RX1		DATA 0

**Fuente:** Los autores.



**Figura 22:** Esquema de conexión de los componentes

**Fuente:** Los autores.

Se ha diseñado un PCB que permita acoplar los módulos con el Arduino MEGA a partir del esquemático mostrado anteriormente, este actúa a modo de shield sobre el Arduino y permite que los módulos se desacoplen fácilmente en caso de requerir mantenimiento (véase figura 23).



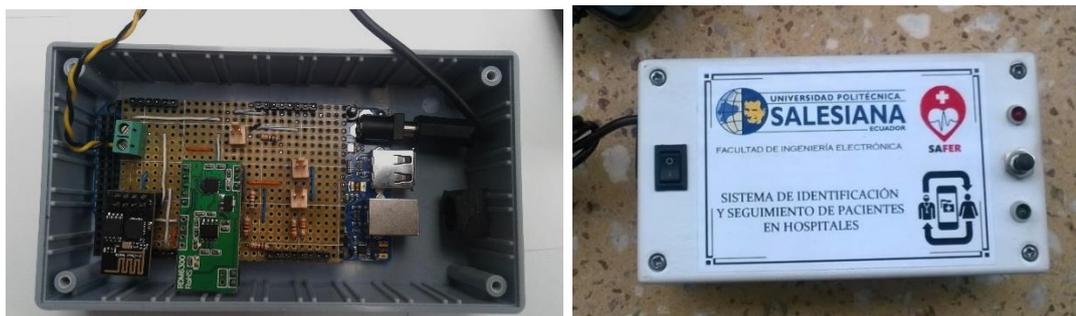
**Figura 23:** Placa para el acople de los módulos con el Arduino

**Fuente:** Los autores.

Con todos los componentes elaborados se procede a adaptarlos a la carcasa. La fuente de alimentación será externa al dispositivo requiriendo esta ser de 5VDC – 2A.

En la tapa de la carcasa se montan el interruptor, pulsante y los leds requeridos para la activación y funcionamiento del circuito, además en la parte posterior se monta el adaptador de 5 VDC a 3.3 VDC, el cual suministrará la tensión requerida por el módulo WiFi. También se monta en la tapa la antena del lector RFID en esta pieza.

En la caja en si se montan mediante tornillos el Arduino Mega, la PCB, los módulos WiFi y RFID. Con todo esto se tiene el dispositivo como se muestra en la figura 24.



**Figura 24:** Dispositivo montado en la carcasa

**Fuente:** Los autores.

### **2.2.2.8. Diseño e implementación de software del lector RFID**

Los softwares de los lectores RFID fueron realizados en Arduino v1.6. Se desarrolló la programación para 3 lectores RFID, un lector para el ingreso del Paciente y dos lectores para registro de la localización. Los lectores se encargan de leer los códigos de las manillas RFID que se encuentren en su radio de cobertura y estos códigos son enviados hacia el servidor.

Al encender el Lector RFID, este se conecta automáticamente a la Red Wifi pre-establecida, usando el nombre de red y la contraseña. Si se logra conectar a la red se encenderá el led color verde, caso contrario se encenderá el led color rojo.

Una vez conectados los lectores RFID a la Red, una intermitencia en el led de color verde indicará que los lectores están listos para leer cualquier pulsera RFID.

Cuando la pulsera se encuentre en el radio de lectura, el lector detectará si es una manilla registrada en inventario del hospital, si lo es, se enviará el código de la manilla al servidor, y el led verde se encenderá, confirmando un envío exitoso, caso contrario si la manilla no está registrada, o no se envió correctamente los datos se encenderá el led rojo, indicando algún error.

Después de que se envié el código al servidor, el led verde vuelve a su estado de intermitente, y el lector regresa a su modo de lectura para cualquier pulsera.

Los códigos de programación de Arduino pueden ser visualizados en el **ANEXO 1**.

### **2.2.3. Servidor Web**

El servidor web se implementa utilizando tres programas distintos:

- Apache: Para la gestión del acceso web que será la base para la comunicación entre la aplicación y el servidor, utilizando el protocolo HTTP [16].
- MySQL, que en este caso se encarga de la gestión de la base de datos, la cual contiene la información del paciente y del personal médico de la institución.
- PHP, dicho lenguaje servirá para el acceso a la base de datos mediante el uso de scripts que consultan o escriben información determinada de la base de datos y permite además su ejecución sobre un protocolo HTTP.

Para la gestión de la información se utiliza un arreglo del tipo JSON el cual es una estructura de datos propia de Java, y se puede implementar con ciertos comandos usando PHP, el cual se encarga del acceso y consulta a la base de datos, para posteriormente transmitir la información mediante el protocolo HTTP a la aplicación.

El código de los scripts PHP para las distintas tareas ejecutadas sobre la base de datos puede ser visto en el **ANEXO 2**.

Para el acceso a la base de datos se puede utilizar la línea de comandos de MySQL o simplemente utilizar PHPmyadmin el cual permite el acceso a la información con el usuario **root** y la contraseña **RFID4682**.

Para realizar el diseño de la base de datos, con el fin tener un correcto almacenamiento y distribución de la información hospitalaria, se utilizó el modelo Entidad-Relación, propuesto por P. Cheng (1976), el cual se basa en una asociación entre entidades [35]. La estructura de la base de datos se presenta en el **ANEXO 3**.

#### **2.2.4. Sistema de localización de Personal y Pacientes**

El sistema de localización de personal y pacientes trabaja de la siguiente manera: a cada paciente o miembro del personal del hospital se le asigna un brazalete, el mismo que debe portar en todo momento. Cuando el dispositivo detecta uno de estos brazaletes, registra en la base de datos la información correspondiente asociando el número de serie de la manilla con el número del lector. De esta manera se obtiene la ubicación del paciente, debido a que los lectores son fijos y bastará realizar una consulta sobre la ubicación del lector y el área de cobertura del mismo para determinar la posición del paciente.

##### **2.2.4.1. Algoritmo de Búsqueda de la ruta más corta**

Nuestro sistema además de ubicar al paciente, también muestra la ruta más corta para que el usuario con el teléfono móvil pueda encontrar de la forma más rápida al paciente en caso de ser necesario.

Para la implementación en este caso se ha decidido tratar al centro hospitalario como si se tratase de un laberinto. En este contexto se toma como unidad de medida mínima

un cuadrado de 0.5x0.5 m, siendo esta la resolución mínima del mapa precargado en el dispositivo móvil. Adoptando esta medida se toma todas las paredes del hospital como si se tratasen de muros del laberinto, determinando la ruta más corta a través de pasillos u otros ambientes del centro médico.

El algoritmo utilizado para resolver el laberinto resultante es el BFS, por sus siglas en inglés Breadth First Search, que es utilizado para buscar elementos en un grafo. Este es un algoritmo de búsqueda sin información, es decir no utiliza ninguna estrategia heurística para resolver el problema. Se empieza de un nodo raíz y se exploran sus vecinos, luego se exploran los vecinos adyacentes de este último y así sucesivamente hasta que se recorre todo el árbol y se encuentra una solución [34].

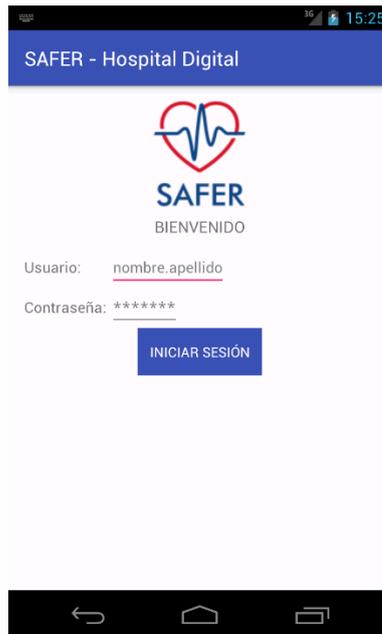
El código utilizado en Android Studio para implementar el algoritmo de localización para encontrar la ruta más corta en el plano se puede observar en el **ANEXO 5**.

#### **2.2.5. Diseño e implementación de la aplicación móvil en Android**

La aplicación se desarrolló utilizando Android Studio 2.0. El requerimiento para correr la aplicación es poseer un dispositivo con el sistema operativo Android 4.0.3 o versiones superiores del mismo.

El código implementado en Android Studio para la Aplicación a ser instalada en el Dispositivo Android se presenta en el **ANEXO 4**.

Al entrar en la aplicación se presenta la pantalla de Log In, obsérvese figura 25, en la cual el usuario podrá ingresar su usuario y contraseña para autenticarse y obtener acceso a todas las funciones del sistema.



**Figura 25:** Pantalla de Ingreso al Sistema

**Fuente:** Los autores.

Si la autenticación fue exitosa el usuario ingresa a un menú en el cual se le dan 7 opciones entre las que se encuentran: Paciente, Médico, Enfermera, Historial, Localización, Gestión de Personal y Cerrar Sesión, como puede observarse en la figura 26.



**Figura 26:** Menú Principal de la Aplicación

**Fuente:** Los autores.

Además, al ingresar se le indicará al usuario en una ventana flotante los permisos que este posee en el sistema, estando estos restringidos a tres niveles:

Nivel 3.- Administrador, quien puede gestionar información relativa al ingreso de nuevos médicos y enfermeras.

Nivel 2.-Médico, permite editar y visualizar la información del historial médico del paciente, así como el ingreso al sistema de localización y la opción “Médico” del menú principal.

Nivel 1.- Enfermera, permite visualizar la información, del paciente, da acceso al sistema de localización y la opción “Enfermera” del menú principal.

A continuación, se presenta en detalle cada una de las opciones del menú principal indicando las funciones a realizarse en cada una de las pantallas desplegadas al presionar sobre la opción.

### 2.2.5.1. Opción Paciente

Al ingresar en esta opción se le ofrecen al usuario las cuatro posibilidades, mostradas en la figura 27: Internar, Dar Alta, Nuevo y Buscar.



**Figura 27:** *Submenú Paciente*

Fuente: Los autores.

En “Internar” se pide que se acerque una manilla al dispositivo ubicado en la recepción, al realizar esto se actualiza el número de serie y se muestra en pantalla al usuario que está realizando el ingreso del paciente, véase figura 27, también se debe ingresar el número de cédula del paciente, si se encuentra en la base de datos (el paciente ha ingresado anteriormente al hospital) se habilita el botón de asignar para realizar el ingreso del paciente, caso contrario se despliega un cuadro de diálogo en el que se indica que se trata de un nuevo paciente y se pide que se realice el registro de todos los datos del mismo. Esta última acción equivale a presionar la opción “Registrar” del menú de paciente.

En “Dar Alta” se asigna el alta médica al paciente requiriendo el número de cédula, si este es válido y se presiona Buscar, muestra al paciente encontrado en la base de datos y se habilita el botón “Dar Alta” (consulte la figura 28 para una referencia gráfica). Si se presiona el botón mencionado aparece un cuadro de diálogo en el que se pide se confirme si desea dar de alta al paciente con las opciones Si y Cancelar.

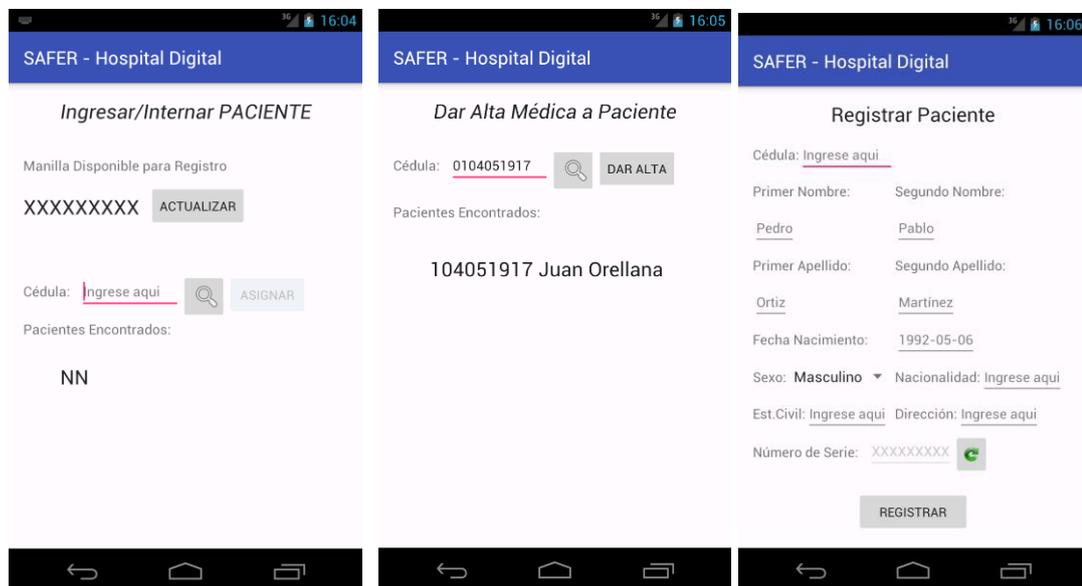
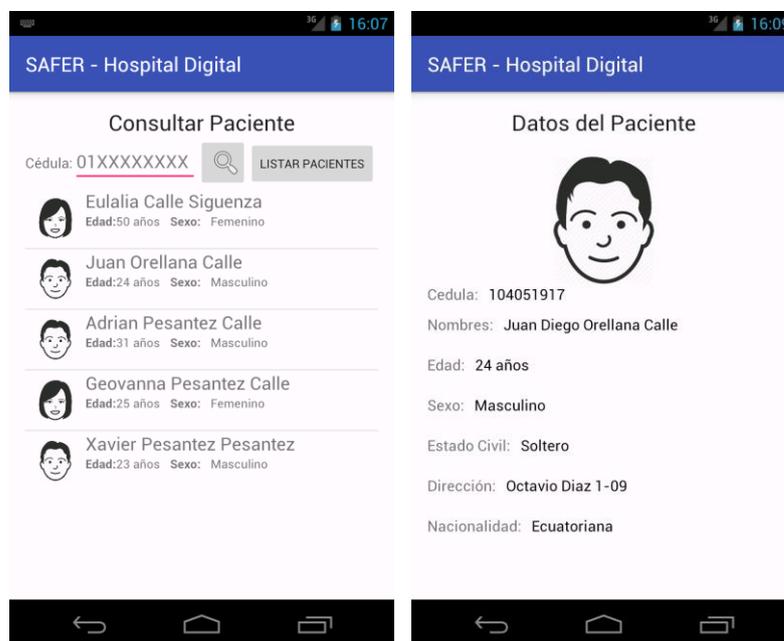


Figura 28: Opciones Internar/Ingresar, Dar Alta y Registrar

Fuente: Los autores.

En “Consultar” se puede observar los datos personales del paciente. El paciente se encuentra digitando el número de cédula y presionando el botón buscar o con el botón

de listar pacientes en el cual se mostrarán todos los pacientes ingresados en el centro médico, véase figura 29. Al realizar clic en el paciente deseado se despliegan los datos personales del mismo en una nueva pantalla.

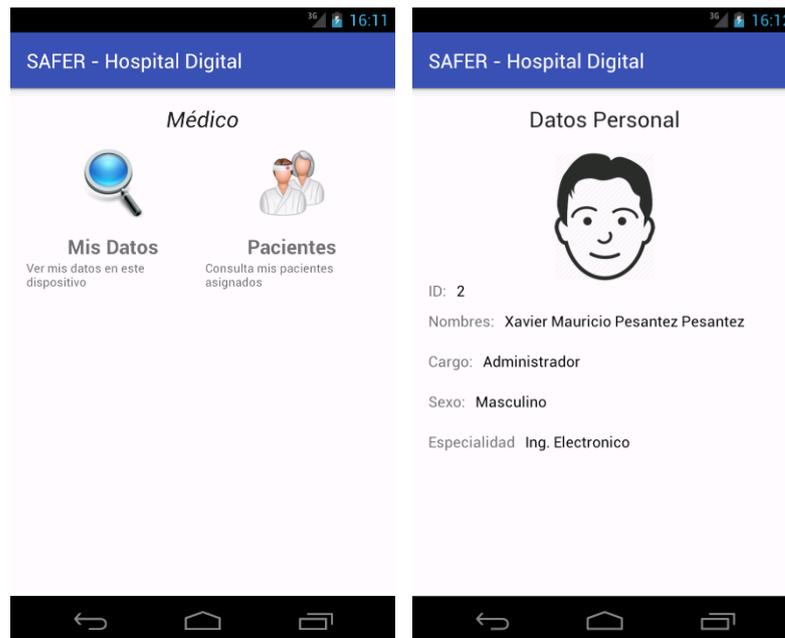


**Figura 29:** Consulta y datos del paciente

**Fuente:** Los autores.

### 2.2.5.2. Opción Médico

Cuando se ingresa a esta opción se despliega un submenú, el cual presenta dos opciones “Mis Datos” en el cual se pueden visualizar los datos del médico en la base de datos del hospital, véase figura 30. También se tiene la opción de pacientes en la cual se muestran los pacientes asignados al médico que realiza la consulta, al hacer clic en esta opción se muestra el menú de búsqueda de la figura 27.



**Figura 30:** Submenú Médico y Opción Mis Datos

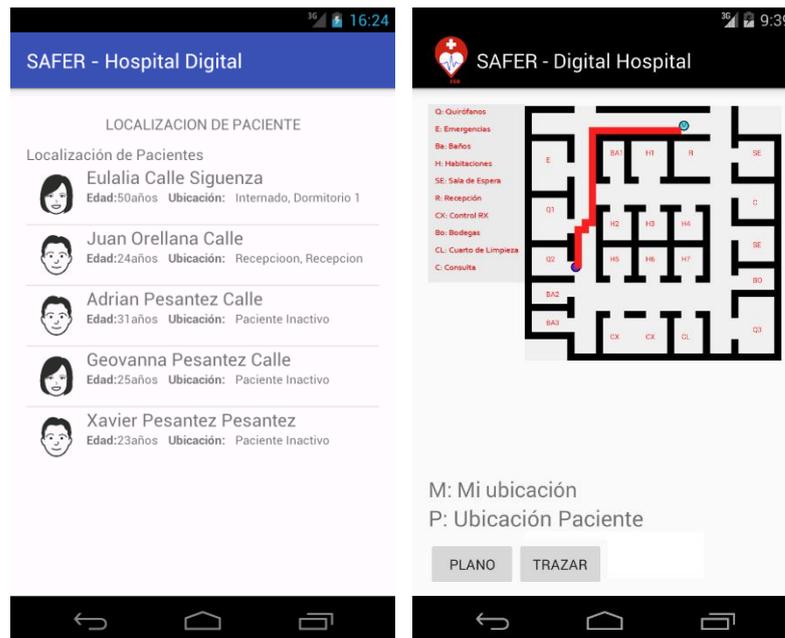
**Fuente:** Los autores.

### 2.2.5.3. Opción Enfermera

La opción “Enfermera” muestra el mismo submenú que el menú médico, a diferencia que este solo es accesible para personal médico con nivel de permisos 1.

### 2.2.5.4. Opción Localización

Cuando se ingresa a esta opción se despliega una lista de los pacientes que se encuentran en el centro médico. Se puede realizar clic sobre cada uno de los elementos de esta lista con lo que se llevará inmediatamente a una nueva pantalla en la que se muestran: las ubicaciones del médico/enfermera y el paciente, así como la ruta más corta entre los implicados en la consulta, esto se puede ver más claramente en la figura 31.

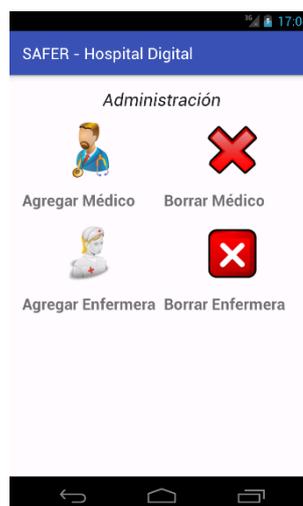


**Figura 31:** Localización, mapa y ruta más corta hacia el paciente

**Fuente:** Los autores.

### 2.2.5.5. Opción Gestionar Personal

Este submenú permite registrar médicos y enfermeras de la misma manera en la que se registran los pacientes. Esta opción es solamente accesible para usuarios con nivel de permisos 1, es decir administradores del sistema. Se permiten también las opciones de borrar de la base de datos al personal médico en caso de ser necesario, el submenú correspondiente a esta opción se muestra en la figura 32.



**Figura 32:** Submenú para gestión del personal

**Fuente:** Los autores.

### 2.2.5.6. Opción Cerrar Sesión

Con esta opción se cierra sesión en el dispositivo y se vuelve de nuevo a la pantalla de inicio, para proceder a insertar las credenciales nuevamente.

## 2.3. Montaje del sistema

Para el montaje del sistema es necesario establecer una red Wi-Fi que es sobre la cual trabajará el sistema o en su caso utilizar la red existente en el hospital en la cual se desee implementar. El direccionamiento de la red puede ser el que se desee ya sea con un servidor web en la nube o utilizando un servidor local y una IP estática, de cualquiera de las dos maneras esto debe configurarse en la aplicación mediante las opciones en la pantalla de inicio.

Como usuarios de prueba están registrados las personas mostradas en la tabla 5, se detallan los permisos y las contraseñas de los mismos.

**Tabla 5:** *Usuarios y Contraseñas de prueba*

<b>Tipo de Usuario</b>	<b>Usuario</b>	<b>Contraseña</b>
Administrador	juan.orellana	loko3345
	xavier.pesantez	noujir3010
Enfermera	diana.castillo	diacas789
	paula.calle	paucal321
Médico	maria.pesantez	maferp452
	fanny.orellana	fannylu478

**Fuente:** *Los autores.*

Configurados los componentes y conectados como se muestra en la figura 13, el sistema queda listo para realizar las lecturas de las etiquetas y realizar consultas sobre la información o ubicación del paciente dentro del centro médico.

## CAPÍTULO 3. ANÁLISIS DE DATOS Y RESULTADOS

En el presente capítulo se presentan los datos y su respectivo análisis. Las pruebas del sistema se realizaron en un escenario controlado de 150 m<sup>2</sup> con tres lectores de prueba distribuidos en este espacio. Se contó con tres personas que simularon ser pacientes, así como otras dos que ejercieron las funciones de doctor y enfermera.

En total se realizaron 21 pruebas de la aplicación y 14 mediciones para verificar el envío de datos y la cobertura de los lectores.

Finalmente, todos estos datos se agrupan en tres secciones que se muestran a continuación, dividiéndolos en pruebas de envío de datos, de la aplicación y mediciones de radiofrecuencia.

### 3.1. Pruebas de envío de Datos

Las pruebas consisten en el envío de la información de la manilla hacia la base de datos. Estas se dividen en tres secciones: datos de la lectura de la manilla al ingreso del hospital, localización de paciente y localización de personal, las cuáles se presentan a continuación:

#### 3.1.1. Ingreso del Paciente al Hospital

Esta sección simula la llegada de un paciente al hospital, en el mismo se toman los datos y se procede a identificar al paciente con la manilla. Si el paciente ya tenía visitas previas al centro de salud el proceso se agiliza ya que solo debe proporcionar su número de cédula. Se verificó el envío de datos hacia el servidor, así como el tiempo que tarda en desempeñarse esta actividad.

**Tabla 6:** Verificación de datos y tiempos

Lector	Num.Serie Manilla	Tiempo	Integridad de Datos
Recepción	497049653	12.34 s	Si
	505369693	13.594 s	Si
	655553533	14.038 s	Si

**Fuente:** Los autores.

### 3.1.2. Datos de Localización del Paciente

En este apartado se prueba el envío de datos de la ubicación del paciente en el hospital cuando uno de los lectores realiza alguna detección. Debido a que se trata de un prototipo se cuentan solamente con dos lectores. Los resultados se observan en la tabla 7.

**Tabla 7:** *Datos y tiempos de localización de pacientes*

Lector	Num.Serie Manilla	Tiempo	Integridad de Datos
Habitación 6	497049653	14.058 s	Si
	505369693	14.203 s	Si
Quirófano 2	497049653	12.915 s	Si
	505369693	13.274 s	Si

**Fuente:** *Los autores.*

### 3.1.3. Datos de Localización del Personal

De manera análoga a la anterior se realizó pruebas estimando los tiempos y la integridad de datos, usando esta vez manillas asignadas al personal, véase tabla 8.

**Tabla 8:** *Datos y tiempos de localización de personal*

Lector	Num.Serie Manilla	Tiempo	Integridad de Datos
Habitación 6	655553533	13.77 s	Si
	526666523	13.695 s	Si
Quirófano 2	655553533	13.594 s	Si
	526666523	13.818 s	Si

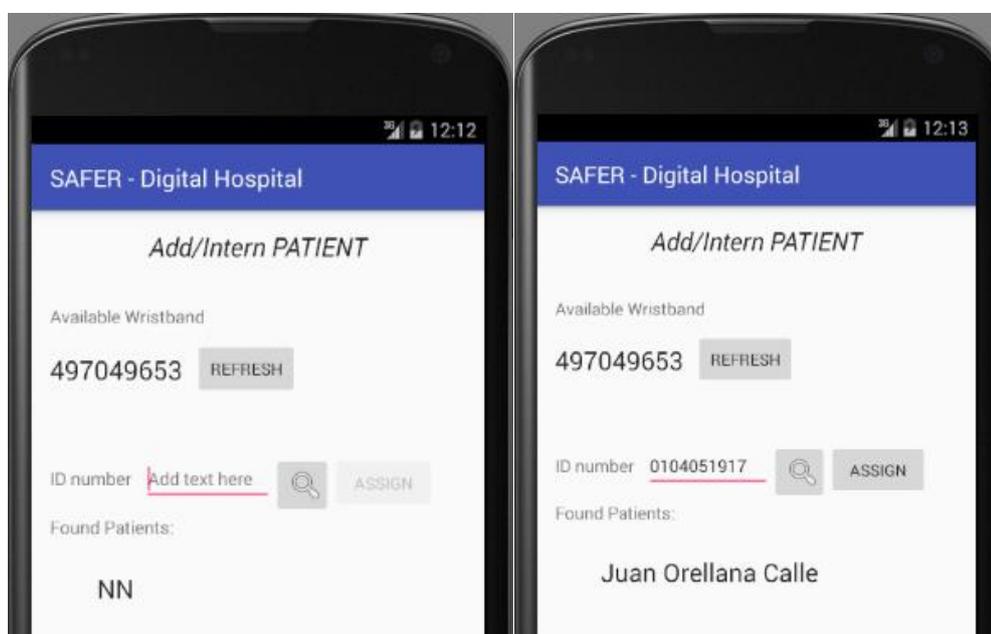
**Fuente:** *Los autores.*

## 3.2. Pruebas de la Aplicación

En esta sección se muestran capturas de la aplicación correspondientes a las acciones realizadas en cada una de las pruebas, como el registro de pacientes, la edición del historial médico, etc. Además, se comprueba que la aplicación está disponible en español e inglés.

### 3.2.1. Paciente

En esta sección se prueba los apartados correspondientes a la sección de paciente, tales son el registro de un nuevo paciente mediante el uso del lector ubicado en la recepción en conjunto con la aplicación, los resultados pueden verse en la figura 33.



**Figura 33:** Prueba del Ingreso de un Paciente al Hospital

**Fuente:** Los autores.

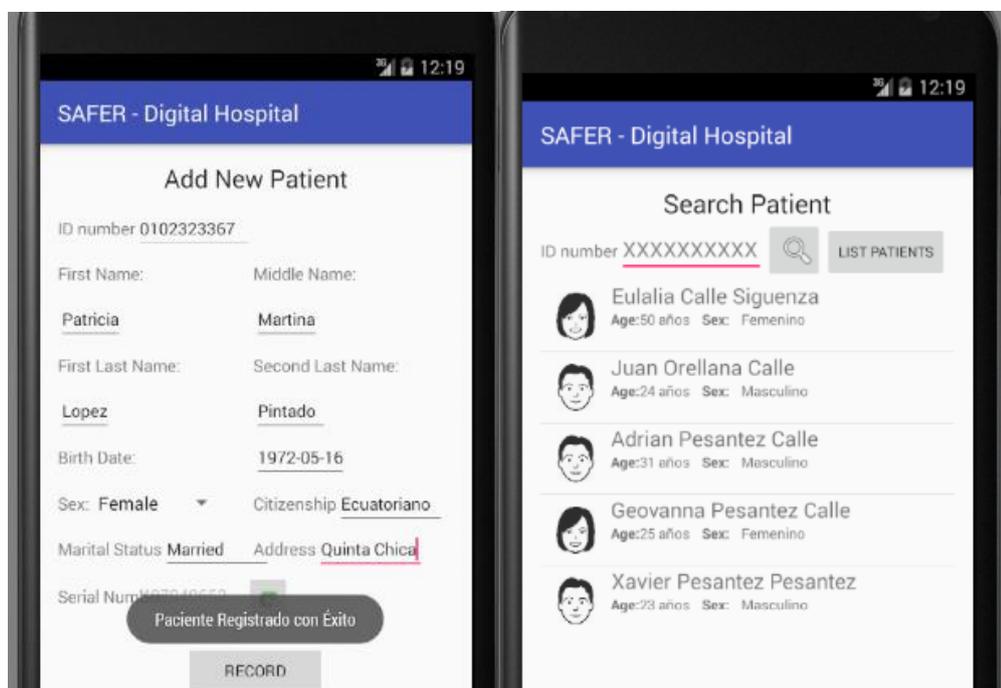
A continuación, se puede realizar la prueba de la alta médica del paciente en la cual se recibe la manilla asignada al mismo en recepción y se procede a pedir su número de cédula para desvincularla del sistema, los resultados se pueden ver en la figura 34.



**Figura 34:** Prueba de la Alta Médica a un Paciente

**Fuente:** Los autores.

Finalmente se realizan las pruebas para el registro de un nuevo paciente con todos sus datos y la consulta de los datos de un paciente mediante el menú de buscar y listar pacientes, los resultados se muestran en la figura 35.

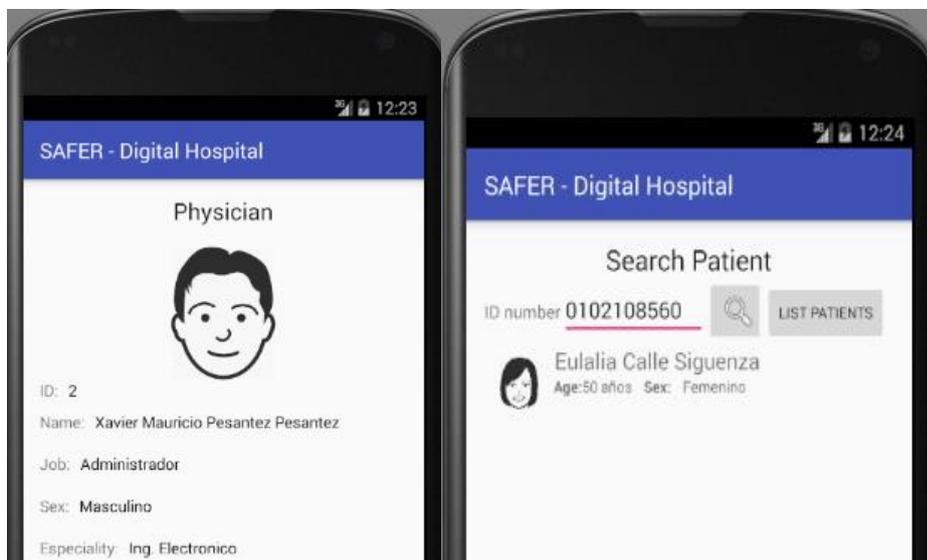


**Figura 35:** Prueba Registro y Búsqueda de Pacientes

**Fuente:** Los autores.

### 3.2.2. Médico y Enfermera

En las opciones del médico y enfermera se procede a probar las opciones de consultar mis datos y la búsqueda de pacientes, tal como se muestra en la figura 36.



**Figura 36:** Prueba Registro y Búsqueda de Pacientes

**Fuente:** Los autores.

De forma análoga se procede a probar las opciones para la enfermera, que son las mismas que para el médico con la diferencia de la gestión de los permisos, como se ve en la figura 37.



**Figura 37:** Prueba Registro y Búsqueda de Pacientes

**Fuente:** Los autores.

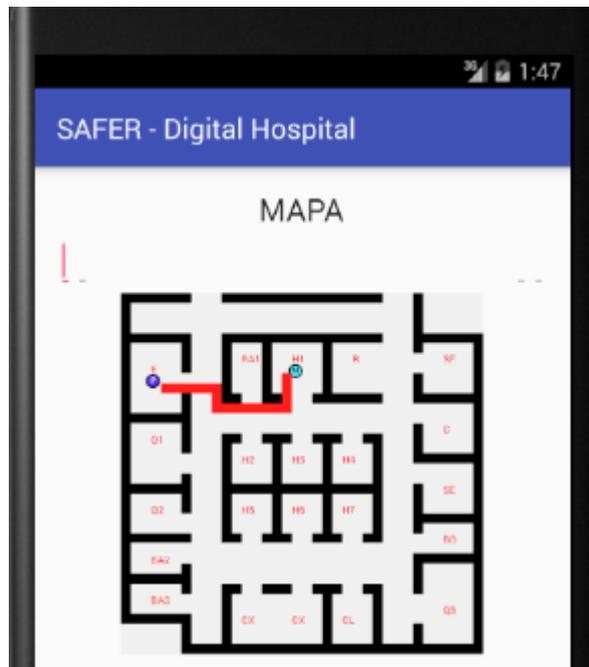
### 3.2.3. Localización

En el apartado de la localización se proceden a realizar varias pruebas moviendo a los pacientes y personal de un ambiente a otro y haciendo que los lectores registren estos cambios. Los resultados se pueden ver en las figuras 38 a 41.



**Figura 38:** Localización de Paciente y Enfermera en el Hospital

Fuente: Los autores.



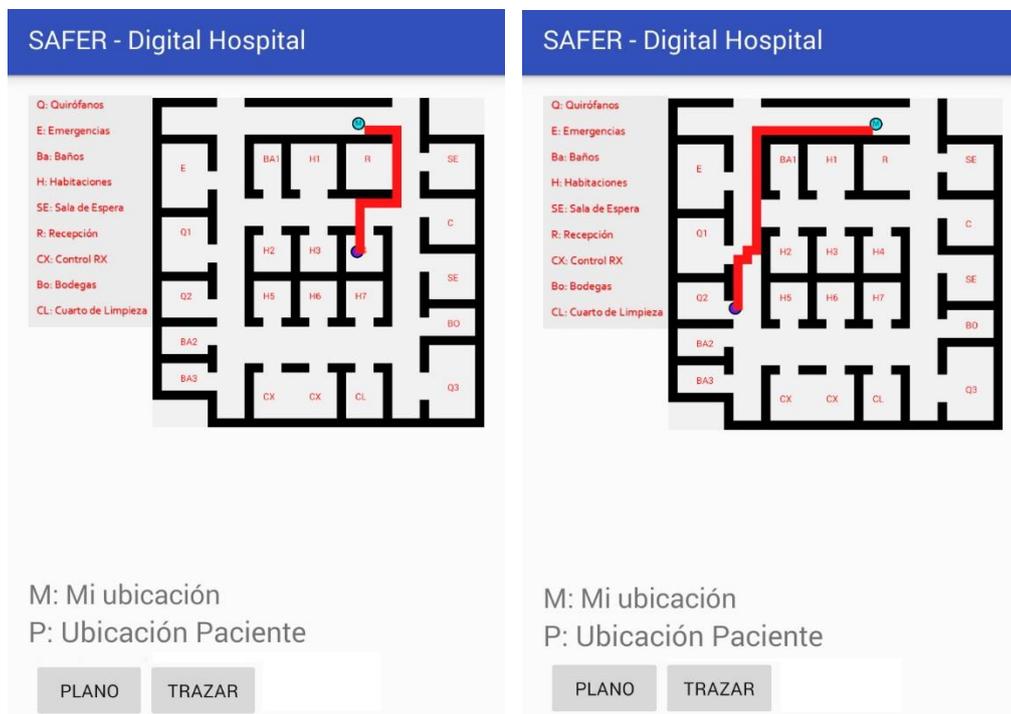
**Figura 39:** Ruta más corta entre el paciente y la enfermera a cuidado del paciente

Fuente: Los autores.



**Figura 40:** Localización de Pacientes y Personal en distintas áreas del hospital

**Fuente:** Los autores.

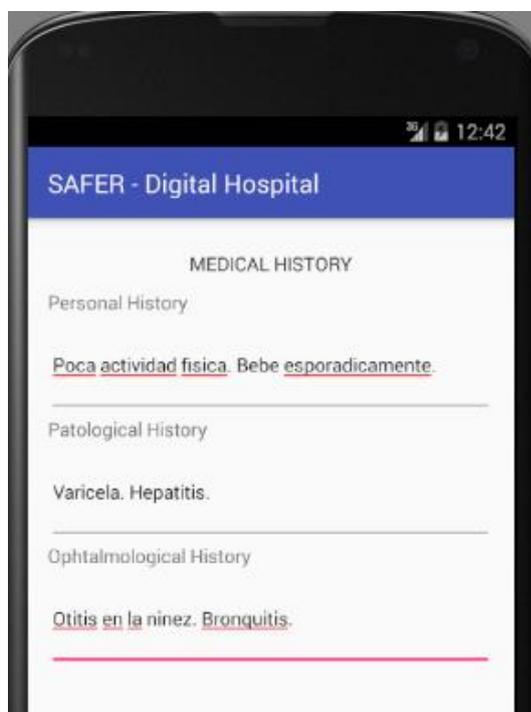


**Figura 41:** Ruta más corta entre Pacientes y Personal en distintas áreas del hospital

**Fuente:** Los autores.

### 3.2.4. Historial

Se puede observar una consulta del historial médico del paciente en la figura 42.



**Figura 42:** Consulta de Historial Médico de Pacientes

**Fuente:** Los autores.

## 3.3. Pruebas de RF

Se realizaron mediciones de la potencia de recepción del módulo Wi-Fi, desde el punto de acceso hasta la ubicación del lector. Estas se explican con detalle en las secciones siguientes.

### 3.3.1. Alcance y Atenuación Wi-Fi

Las pruebas se dividen en dos categorías, la primera en un espacio con obstáculos y la segunda en un ambiente totalmente vacío.

#### 3.3.1.1. Con obstáculos

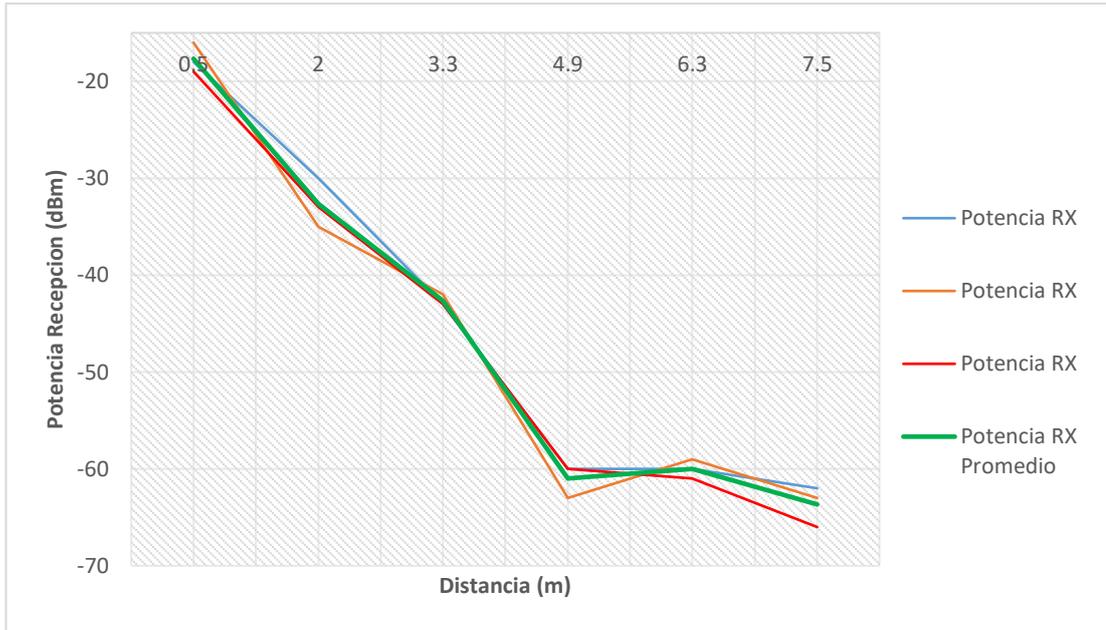
Las mediciones con obstáculos se basaron en la potencia recibida por el lector, además se estimó la distancia en línea recta hacia el punto de acceso. Entre los obstáculos se contó con una pared y varios muebles de madera y metálicos. En la tabla 9 y en la figura 43, se presentan los datos obtenidos.

**Tabla 9:** Pruebas de cobertura entre Lector y Punto de Acceso con obstáculos

Medición	Potencia RX	Distancia	Integridad de Datos
1	-18 dBm	0.2 m	Si
2	-16 dBm		Si
3	-19 dBm		Si
4	-30 dBm	2.15 m	Si
5	-35 dBm		Si
6	-33 dBm		Si
7	-43 dBm	4.85 m	Si
8	-42 dBm		Si
9	-44 dBm		Si
10	-60 dBm	6.50 m	Si
11	-63 dBm		Si
12	-60 dBm		Si
13	-60 dBm	9 m	Si
14	-59 dBm		Si
15	-61 dBm		Si
16	-62 dBm	11.8 m	No
17	-63 dBm		No
18	-66 dBm		No

**Fuente:** Los autores.

Como se observa en la tabla 9, existe una correcta integridad de datos por arriba de los -60dBm.



**Figura 43:** Curva de cobertura entre Lector y Punto de Acceso con obstáculos

**Fuente:** Los autores.

### 3.3.1.2. Sin obstáculos

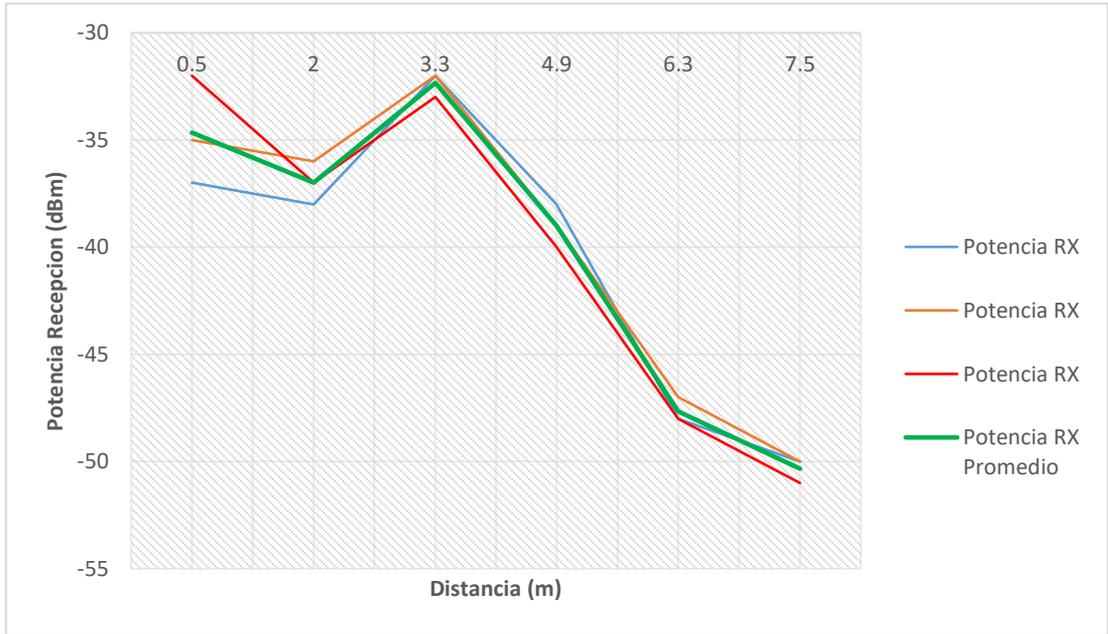
Se despejo el ambiente anterior de todo obstáculo y se colocó el punto de acceso de manera que exista línea de vista al lector. La potencia ideal de salida del punto de acceso utilizado es de 20 dBm, sin embargo, mediciones comprueban que la potencia real de salida es de entre 0 y -2 dBm. A continuación, en la tabla 10 y en la figura 44, se presentan las mediciones.

**Tabla 10:** Pruebas de cobertura entre Lector y Punto de Acceso sin obstáculos

Medición	Potencia RX	Distancia	Integridad de Datos
1	-37 dBm	0.5 m	Si
2	-35 dBm		Si
3	-32 dBm		Si
4	-38 dBm	2. m	Si
5	-36 dBm		Si

6	-37 dBm		Si
7	-32 dBm	3.3 m	Si
8	-32 dBm		Si
9	-33 dBm		Si
10	-38 dBm	4.9 m	Si
11	-39 dBm		Si
12	-40 dBm		Si
13	-48 dBm	6.3 m	Si
14	-47 dBm		Si
15	-48 dBm		Si
16	-50 dBm	7.5 m	Si
17	-50 dBm		Si
18	-51 dBm		Si
16	-44 dBm	9.5 m	Si
17	-43 dBm		Si
18	-44 dBm		Si

**Fuente:** *Los autores.*



**Figura 44:** Curva de cobertura entre Lector y Punto de Acceso sin obstáculos

**Fuente:** Los autores.

### 3.4. Presupuesto

A continuación, en la tabla 11, se presentan los elementos utilizados para la construcción de los prototipos y su valor comercial. Es de destacar que este presupuesto es de la implementación.

**Tabla 11:** Presupuesto para la construcción del prototipo del sistema

Descripción	Cantidad	Valor unitario	Valor total
Arduino MEGA	3	\$22.00	\$66.00
Lector RFID RDM6300 + Antena	3	\$12.00	\$36.00
Manillas RFID 125 kHz	3	\$2.50	\$7.50

Módulo Wi-Fi ESP8266	3	\$10.00	\$30.00
Router D-Link DIR-610	1	\$26.00	\$26.00
Adaptador 5V 2A	3	\$7.50	\$22.50
Caja plástica para proyectos	3	\$5.75	\$17.25
Materiales para elaboración de PCBs	1	\$42.66	\$42.66
Diseño y Elaboración Placa	15 horas	\$5	\$75.00
Programación Aplicación y Scripts servidor	250 horas	\$6	\$1500
Programación del Lector (Arduino, Módulos)	60 horas	\$6	\$360
Diseño e Implementación de Base de Datos	24 horas	\$6	\$144
		<b>Subtotal</b>	<b>\$2326.91</b>
		<b>I.V.A.</b>	<b>\$279.23</b>
		<b>Total</b>	<b>\$2606.14</b>

**Fuente:** *Los autores.*

Dejando de lado los costos adicionales del sistema como los routers o la programación de la aplicación y base de datos, el costo por lector con la tecnología mostrada es de \$71.47. A continuación, en la tabla 12, se muestra el costo de construcción utilizando un sistema de largo alcance como es el de la compañía LinkSprite.

**Tabla 12:** Presupuesto para la construcción del sistema de largo alcance

<b>Descripción</b>	<b>Cantidad</b>	<b>Valor unitario</b>	<b>Valor total</b>
Arduino MEGA	1	\$22.00	\$22.00
Lector RFID Cottonwood Long Range	1	\$187.00	\$187.00
Antena LinkSprite UHF RFID Reader Antenna	1	\$79.00	\$79.00
Módulo Wi-Fi ESP8266	1	\$10.00	\$10.00
Adaptador 5V 3A	1	\$12.20	\$12.20
Caja plástica para proyectos	1	\$9.00	\$9.00
Materiales para elaboración de PCBs	1	\$20.00	\$20.00
Diseño y Elaboración Placa	15 horas	\$5	\$75.00
Programación del Lector (Arduino, Módulos)	40 horas	\$5	\$200
<b>Subtotal</b>			<b>\$614.2</b>
<b>I.V.A.</b>			<b>\$73.71</b>
<b>Total</b>			<b>\$687.91</b>

**Fuente:** Los autores.

De manera análoga a la anterior el costo únicamente de construcción del lector es de \$339.2 lo cual es 4.75 veces el costo del prototipo general sin embargo al tener un mayor alcance se necesitarán muchos menos por planta por lo cual el costo no se incrementa demasiado.

## **CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES**

### **4.1 Conclusiones**

Los errores médicos son un gran problema en la sociedad debido al pobre manejo de la información que se da en los hospitales. Los sistemas de identificación de pacientes que existen en nuestro medio están desactualizados en relación a la actualidad; en que las TIC están tan implantadas en la sociedad, por eso se ha presentado esta alternativa que implementa varias tecnologías para lograr brindar los servicios antes mencionados.

El sistema presentado es capaz de gestionar la información de una forma más eficaz y eficiente, manteniendo la disponibilidad de la misma en todo momento, ayudando a reducir errores que pueden costar vidas.

Se utilizó el sistema operativo Android, debido su gran porcentaje de uso en comparación con otros, además del bajo costo de los dispositivos que implementan este S.O. Asimismo, se manejó la plataforma Arduino por la sencillez, rapidez y a que no se necesitó una gran capacidad de procesamiento.

La plataforma utiliza protocolos web como su principal método de intercambio de información, lo cual ofrece mayor accesibilidad y da versatilidad de que los servicios se implementen en distintas plataformas independientes del lenguaje de programación utilizado.

Se ha comprobado el correcto funcionamiento del sistema de localización con los prototipos construidos, así como de el algoritmo para trazar la ruta más corta hacia el paciente. De esta manera ayudándose de la aplicación el personal puede reducir los tiempos de atención en el hospital.

Finalmente se deja abierta la posibilidad de que los datos de localización se puedan utilizar para crear perfiles de usuario o reportes de eficiencia, lo cual es el siguiente escalón en el desarrollo de la plataforma SAFER.

## 4.2 Recomendaciones

Para futuros desarrollos de la plataforma SAFER se recomienda, usar módulos RFID de mayor alcance, para obtener una mayor cobertura y reducir la cantidad de lectores, para cubrir un área determinada.

Se podrían mejorar los tiempos de actualización de la ubicación de los pacientes y personal médico, mediante la implementación de puntos de acceso con la tecnología IEEE 802.11n.

Para aumentar aún más la seguridad y confidencialidad se podrían recurrir a técnicas de cifrado de datos, ya que la información es hospitalaria.

Se recomienda también realizar investigaciones para el desarrollo de algoritmos que permitan una mejor distribución de lectores, maximizando de esta forma la eficiencia en la cobertura en cada una de las plantas del hospital, reduciendo el número de lectores y por ende el costo del sistema.

En cuanto a la aplicación esta podría mejorarse implementando un sistema de mensajería entre médicos y enfermeras, para evitar traslados innecesarios de personal en el hospital.

Nótese también que el sistema es fácilmente modificable para llevar un control de personal en empresas o fábricas.

Todas estas recomendaciones se hacen para ir formando un sistema versátil que sea adaptable según el ambiente y a las necesidades de los usuarios, mediante dispositivos que faciliten el estilo de vida, garanticen la fiabilidad de la información, y poder llevar un control de las actividades, el cual es el objetivo principal del proyecto SAFER.

## **ANEXOS**

## Anexo 1: Código de Arduino para los Lectores

```

//*****
//Sistema de Identificacion y Seguimiento de Pacientes en Hospitales
//Universidad Politecnica Salesiana
//Ingenieria Electronica, Telecomunicaciones
//Juan Orellana - Xavier Pesantez

//Codigo de Lectores de Manillas RFID
//Lectura de codigos de Manillas RFID y envio de codigos al Servidor

//*****

//Declaracion de variables para la conexión a la red WIFI

#define SSID "RFID" // Nombre de la Red
#define PASS "RFID4682" // Clave de la Red
#define DST_IP "192.168.3.2" // Direccion IP del Servidor

//Declaración de variables y constantes

int ResetWIFI = 8; //pin de Reset del Arduino
int BIEN = 11; // pin indicador led verde
int MAL = 12; // pin indicador led rojo

int newtag[14] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0}; // variable usada para comparar tags
String TAG ; //variable de cogido de la manilla RFID
String N_Lector ; //variable de num de lector
int correr = 0 ; //variable de esperando manilla

boolean connected=false;
String content = "";
char character;
int valor = 0;
int lectura = 0;

//Declaracion de los codigos de cada manilla RFID

int tag3[14] = {2,49,68,48,48,51,49,68,69,65,55,53,53,3}; //manilla negra 1
String T3 = "65553533";

int tag4[14] = {2,49,68,48,48,51,50,50,65,49,70,49,65,3}; //manilla negra 2
String T4 = "497049653";

int tag5[14] = {2,48,48,48,48,51,57,67,54,52,66,66,52,31}; //manilla roja (EJEMPLO, NO
REGISTRADA)
String T5 = "526666523";

int tag6[14] = {2,48,49,48,48,53,65,57,48,50,53,69,69,3}; //manilla amarilla
String T6 = "505369693";

//////////SETUP//////////

void setup(){
```

```

// Reset inicial del Modulo WIFI
pinMode(ResetWIFI, OUTPUT);
digitalWrite(ResetWIFI,LOW);
delay(500);
digitalWrite(ResetWIFI,HIGH);
delay(500);

//Leds Indicadores
pinMode(MAL, OUTPUT);
pinMode(BIEN, OUTPUT);

// Inicio del Serial (Visualizar en PC)
Serial.begin(9600);
Serial.setTimeout(5000);

//Inicio del Serial 3 (Usado para TX/RX del modulo WIFI)
Serial3.begin(9600);
Serial3.setTimeout(5000);
//Inicio del Serial 1 (Usado para RX del lector RFID)
Serial1.begin(9600);

// Verificacion del modulo WIFI
Serial.print("=====");
Serial.println("");
Serial.print("Verificando conexion WIFI");
Serial.println("");
Serial3.println("AT");
delay(300);

if(Serial3.find("OK"))
{
  Serial.println("Preparando conexion WIFI");
  // Aviso de BIEN
  B_ON();
  M_OFF();
}
else
{
  Serial.println("Modulo sin respuesta, Reset o Reinicio");
  // Aviso MAL
  M_ON();
  B_OFF();
}

while(1); // detiene el programa en este punto, si no hubo respuesta del modulo wifi
}

//Reintento de conexion con la red WIFI (5 veces)
for(int i=0;i<5;i++)
{
  if(connectWiFi())
  {
    connected = true;
    break;
  }
}

```

```

}
if (!connected){
  Serial.print("NO SE PUDO CONECTAR A LA RED, Revise las conexiones de RED");
  Serial.println("");
  Serial.print("=====");
  Serial.println("");
  while(1);
}
delay(5000);
Serial3.println("AT+CIPMUX=0");
}

//////////////////LOOP//////////////////
void loop(){

//*****ESPERA GENERAL DE MANILLAS RFID*****

  Inicio_Loop:

    //ESPERANDO MANILLA RFID
    esperando_manilla(); // espera general, despues de la espera inicial

//*****DETECCION DE CODIGO Y ALMACENAMIENTO*****

  Serial.print("Manilla DETECTADA");
    //Avisos
    M_OFF();
    B_ON();

  // Serial.println("");
  //Serial.print("Codigo Manilla: ");
  //delay(100);

  //Leer las manillas RFID
  for (int z = 0 ; z < 14 ; z++) // bucle para leer cada byte
  {
    lectura = Serial1.read(); //lectura del serial 1
    // delay(50);
    newtag[z] = lectura; // almacenamiento de la lectura en un array de 14
  // Serial.print(newtag[z]);
  }
  delay(400);
  Serial1.flush(); // vacia el buffer, evita multiples lecturas
  delay(400);

//*****VERIFICACION DE CODIGOS*****

  checkmytags(); //comparar las tags
  Serial.println("");

//*****ASIGNACION DE CODIGOS PARA ENVIAR AL
SERVIDOR*****

```

*// N\_Lector varía según el numero asignado al lector, en nuestro caso se utilizó 3 lectores.*

```
N_Lector="1"; //para el LECTOR NUMERO 1 (Ingreso)
N_Lector="2"; //para el LECTOR NUMERO 2 (Habitación)
N_Lector="3"; //para el LECTOR NUMERO 3 (Quirófano)
```

```
// accion que realiza al comparar las tags segun "valor"
```

```
//-----
if (valor == 3)
{
TAG = T3; //Asignacion del codigo de la manilla rfid a enviar
//Serial.print("Manilla 3 (Negro 1) Reconocida: "); //imprime la manilla reconocida
Serial.print("Manilla Reconocida: ");
imprimir(); //imprime y carga el codigo (TAG) de la manilla a enviar y reinicia "valor"
goto ConServ;
}

//-----
if (valor == 4)
{
TAG = T4;
// Serial.print("Manilla 4 (Negro 2) Reconocida: ");
Serial.print("Manilla Reconocida: ");
imprimir(); //imprime y carga el codigo (TAG) de la manilla a enviar y reinicia "valor"
goto ConServ;

}

//-----
if (valor == 5)
{
TAG = T5;
// Serial.print("Manilla 5 (Rojo) Reconocida: ");
Serial.print("Manilla Reconocida: ");
imprimir(); //imprime y carga el codigo (TAG) de la manilla a enviar y reinicia "valor"
goto ConServ;

}

//-----
if (valor == 6)
{
TAG = T6;
// Serial.print("Manilla 6 (Amarillo) Reconocida: ");
Serial.print("Manilla Reconocida: ");
imprimir(); //imprime y carga el codigo (TAG) de la manilla a enviar y reinicia "valor"
goto ConServ;
}
```

```

//*****ACCION QUE REALIZA SI LA MANILLA NO ESTA
REGISTRADA*****

    //if (valor == 0)
    // {
    Serial.print("MANILLA NO REGISTRADA");
    Serial.println("");
    delay(100);
    Serial.print("=====");
    Serial.println("");
    Serial.print("ESPERANDO UNA MANILLA RFID...");
    Serial.println("");

    delay(200); //(1)
    Serial1.end(); //(2) //Fin serial y continua a espera general
    delay(200); //(3) 1,2,3 - Evitan que continúe en el error, si se da error de lectura

    goto Inicio_Loop; //Se dirige al inicio del loop a espera general
    // goto dos;
    // }

//*****SI LA MANILLA ESTA REGISTRADA, ENVIO AL
SERVIDOR*****

//Conexion al Servidor
    ConServ:

    Serial.print("Conectando y Enviando datos al Servidor...");
    Serial.println("");
    String cmd = "AT+CIPSTART=\"TCP\",\",";
    cmd += DST_IP;
    cmd += "\",80";
    Serial3.println(cmd);
    Serial.println(cmd);

    // Forzar conexion con el servidor
    if(Serial3.find("Error")) return;

    // Get requests de envio al servidor
    // Se utilizó 1 get request diferente por cada lector:

//Para el Lector 1:
    cmd = "GET /listaEspera.php?id=1&num_serie="+TAG+" HTTP/1.0\r\n\r\n";

//Para el Lector 2:

    cmd = "GET
/regarLocalizacion.php?codigo_lector="+N_Lector+"&num_serie="+TAG+"
HTTP/1.0\r\n\r\n";

//Para el Lector 3:

```

```

    cmd = "GET
/registrarLayoutLocalizacion.php?codigo_lector="+N_Lector+"&num_serie="+TAG+"
HTTP/1.0\r\n\r\n";

Serial.print(cmd); //Texto enviado al servidor para registrar el codigo de la manilla RFID

// Envio del texto en "cmd"
Serial3.print("AT+CIPSEND=");
Serial3.println(cmd.length());
if(Serial3.find(">"))
{
    Serial.print(">");
    Serial.print("Enviando...");
        //Avisos
        B_OFF();
        B_ON();
} // se digire a imprimir SEND OK, 200 OK
else
{
    Serial3.println("AT+CIPCLOSE"); //Cerrar la conexion con el servidor
    Serial.println("Conexion Fallida.Acerque manilla RFID o Revise la conexion del Servidor
a la RED");
    delay(100);
    Serial.println("=====");
    M_ON();
    delay(2000);
    M_OFF();
    delay(100);
    Serial.print("ESPERANDO UNA MANILLA RFID...");
    Serial.println("");

    goto fin_serial_esperando; //Fallo envio, se salta a espera general
// return; //Reintenta el envio al servidor hasta que se envie
}

Serial3.print(cmd);
delay(1700);
while (Serial3.available())
{
    char c = Serial3.read();
    Serial.write(c);
    if(c=='\r') Serial.print('\n');
}

Serial.println("=====");
Serial.println("ENVIO EXITOSO");
Serial.println("=====");
Serial.print("ESPERANDO UNA MANILLA RFID...");
Serial.println("");
delay(500);

    fin_serial_esperando:

```

```

delay(200); //(1)
Serial1.end(); //(2)
delay(200); //(3) Evitan multiples lecturas de la misma tag rfid despues del envio exitoso

} // fin void loop...

```

```

//////////Funciones//////////

```

```

// Funcion para la Conexion WIFI
boolean connectWiFi()
{
  String cmd="AT+CWJAP=\"";
  cmd+="SSID";
  cmd+="\", \"";
  cmd+="PASS";
  cmd+="\"";
  Serial.print("Conectando a la Red ");
  Serial.print("");
  Serial.print(SSID);
  Serial.print("");
  Serial.print("...");
  Serial.println("");
  Serial3.println(cmd);
  delay(1500);

  if(Serial3.find("OK"))
  {
    Serial.println("Conexion Realizada a la Red");
    //Avisos
    M_OFF();
    B_OFF();
    B_ON();
    Serial3.println("AT+CIFSR");
    Serial3.flush();
    delay(1500);
    while(Serial3.available() {
      character = Serial3.read();
      if(character=='.' || character>='0' && character<='9')
        content.concat(character);
    }

    if(content != ""){
      //Serial.print("Direccion IP de ESP8266: ");
      // Serial.println(content);
      Serial.print("Lector Listo");
      Serial.println("");
      Serial.print("ESPERANDO UNA MANILLA RFID...");
      Serial.println("");

      //ESPERANDO MANILLA RFID
      esperando_manilla(); //espera inicial, al arrancar el lector
    }
  }
}

```

```

Serial.print("Preparando primera lectura...");
Serial.println("");
return true;
}

else
{
Serial.println("No se puede conectar a la Red WiFi, Reintentando...");
// Aviso MAL
M_ON();
B_OFF();
return false;
}
}

// Funcion para comparar dos Tags o manillas RFID
boolean comparetag(int aa[14], int bb[14]) //para comparar las tags
{
boolean ff = false;
int fg = 0;
for (int cc = 0 ; cc < 14 ; cc++)
{
if (aa[cc] == bb[cc])
{
fg++;
}
}
if (fg == 14)
{
ff = true;
}
return ff;
}

//Funcion para comparar cada manilla RFID
void checkmytags() // compara cada tag usando "comparetag"
{

if (comparetag(newtag, tag3) == true)
{
valor=3;
}

if (comparetag(newtag, tag4) == true)
{
valor=4;
}
if (comparetag(newtag, tag5) == true)
{
valor=5;
}
if (comparetag(newtag, tag6) == true)
{
valor=6;
}
}

```

```

}
//Funcion de Aviso de todo correcto
void B_ON()
{
    delay(250);
    digitalWrite(BIEN,HIGH);
    delay(250);
}
void B_OFF()
{
    delay(250);
    digitalWrite(BIEN,LOW);
    delay(250);
}

//Fuccion de Aviso de algun error
void M_ON()
{
    delay(250);
    digitalWrite(MAL,HIGH);
    delay(250);
}
void M_OFF()
{
    delay(250);
    digitalWrite(MAL,LOW);
    delay(250);
}
//Funcion de Aviso de esperando manilla RFID
void esperando_manilla()
{
    correr = 1;
    Serial1.begin(9600);//////// aqui inicio y ya no inicio al inciar el void loop (REVISAR)
    delay(300);
    while(true)
    {
        B_OFF();
        B_ON();
        if (Serial1.available(>0) // Cuando se acerca una manilla RFID...
        {
            delay(300); //Retardo inicial para evotar errores de lectura;
            correr=0;
            break;
        }
    }
}
//Funcion para imprimir los datos de la manilla rfid
void imprimir()
{
    Serial.print(TAG);
    Serial.println("");
    delay(400);
    valor=0;
}

```

## Anexo 2: Scripts PHP

```
//-----Asignar Brazalete-----  
Se usa para emparejar al paciente con el brazaletе en la base de datos cuando este llega al  
hospital
```

```
//-----  
<?php  
  
$cedula=$_GET['cedula'];  
$numero_serie=$_GET['numero_serie'];  
  
$query = "INSERT INTO `braz_paciente` (`cedula`, `num_serie`) VALUES  
('$cedula', '$numero_serie')";  
  
//echo $query;  
$server="localhost";  
$user="root";  
$password="RFID4682";  
$database="hospital_digital_v2";  
  
$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede  
conectar con la Base de Datos \n");  
//@mysql_select_db($database) or die("No se puede alcanzar la base de datos");  
  
mysqli_set_charset($conexion,'utf8');  
mysqli_query($conexion,$query);  
mysqli_close();  
  
?>
```

```
//-----Asignar Brazalete 2 -----  
Se usa para emparejar al personal con el brazaletе en la base de datos cuando este se registra  
en la base de datos desde la aplicacion  
//-----
```

```
<?php  
  
$usuario=$_GET['user'];  
$numero_serie=$_GET['num_serie'];  
  
$query = "SELECT id_personal FROM `personal` WHERE username='$usuario'";  
  
//echo $query;  
$server="localhost";  
$user="root";  
$password="RFID4682";  
$database="hospital_digital_v2";  
  
$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede  
conectar con la Base de Datos \n");  
//@mysql_select_db($database) or die("No se puede alcanzar la base de datos");  
mysqli_set_charset($conexion,'utf8');
```

```

$resultado = mysqli_query($conexion,$query);

if (! $resultado)
{
    echo "La consulta SQL contiene errores.".mysql_error();
    exit();
}
else
{

    //obtenemos los datos resultado de la consulta
    while ($row = mysqli_fetch_array($resultado))
    {
        $id = current($row);
    }

}

$date = date('Y-m-d');

$query2 = "INSERT INTO `braz_personal` (`id_personal`, `num_serie`,`fecha`)
VALUES ('$id', '$numero_serie','$fecha')";
mysqli_query($conexion,$query2);

$query4 = "INSERT INTO `loc_personal` (`codigo_lector`, `num_serie`) VALUES
(1, '$numero_serie')";
mysqli_query($conexion,$query4);

mysqli_close();

```

?>

```

//-----Borrar Espera -----
Se utiliza para eliminar el registro creado en la asignacion de la manilla al usuario
//-----

```

<?php

```

$numero_serie=$_GET['numero_serie'];

$query = "DELETE FROM `hospital_digital_v2`.`lista_espera` WHERE
`lista_espera`.`num_serie` = $numero_serie";

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$databse) or die("No se puede
conectar con la Base de Datos \n");
//@mysqli_select_db($databse) or die("No se puede alcanzar la base de datos");

mysqli_set_charset($conexion,'utf8');

```

```
mysqli_query($conexion,$query);
mysqli_close();
```

?>

```
//-----Buscar Brazaletes-----
Consulta el numero de serie del brazaletes enviado por el lector hacia la base de datos
//-----
```

<?php

```
header("Content-Type: text/html;charset=utf-8");

$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$databse) or die("No se puede
conectar con la Base de Datos \n");
mysqli_query($conexion,"SET NAMES 'utf8'");
//@mysqli_select_db($databse) or die("No se puede alcanzar la base de datos");

mysqli_set_charset($conexion,'utf8');

$query = "SELECT * FROM `lista_espera`";

$resultado = mysqli_query($conexion,$query);

if (!$resultado)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta

    while ($row = mysqli_fetch_row($resultado))
    {
        echo json_encode($row);
    }
}

mysqli_close();
```

?>

```
//-----Consultar Historial-----
Consulta el resumen historial médico del paciente
//-----
```

```

<?php

header("Content-Type: text/html;charset=utf-8");

$num_serie=$_GET['num_serie'];

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$databse) or die("No se puede
conectar con la Base de Datos \n");
mysqli_query($conexion,"SET NAMES 'utf8'");
@mysqli_select_db($databse) or die("No se puede alcanzar la base de datos");

mysqli_set_charset($conexion,'utf8');

$query = "SELECT `cedula` FROM `brazalete` WHERE num_serie='$num_serie'";

$resultado = mysqli_query($conexion,$query);

if (!$resultado)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta

    while ($row = mysqli_fetch_array($resultado))
    {
        $cedula = current($row);
    }
}
// echo $cedula;

$query2 = "SELECT `ant_personal`,`ant_patologicos`,`ant_ofthalmologico` FROM
`historial_medico` WHERE cedula='$cedula'";

$resultado2 = mysqli_query($conexion,$query2);

if (!$resultado2)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}

```

```

else
{
    //obtenemos los datos resultado de la consulta

    while ($row2 = mysqli_fetch_row($resultado2))
    {
        echo json_encode($row2);
    }
}

mysqli_close();

?>

//-----Consultar Localizacion-----
Consulta la localizacion del paciente dentro de la base de datos
//-----

<?php

$cedula=$_GET['cedula'];

$query = "SELECT `num_serie` FROM `braz_paciente` WHERE
cedula='$cedula'";

$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$databse) or die("No se puede
conectar con la Base de Datos \n");
//@mysqli_select_db($databse) or die("No se puede alcanzar la base de datos");

mysqli_set_charset($conexion,'utf8');
$resultado = mysqli_query($conexion,$query);

if (! $resultado)
{
    echo "La consulta SQL contiene errores." .mysql_error();
    exit();
}
else
{

    //obtenemos los datos resultado de la consulta
    while ($row = mysqli_fetch_array($resultado))
    {

```

```

        $num_serie = current($row);
    }
}

$query2 = "SELECT `fecha`,`codigo_lector` FROM `loc_paciente` WHERE
num_serie='$num_serie'";

$resultado2 = mysqli_query($conexion,$query2);

if (!$resultado2)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta

    while ($row2 = mysqli_fetch_array($resultado2))
    {
        $valor=$row2[1];
    }
    $codigo_lector=$valor;
}

$query3 = "SELECT `departamento`,`area` FROM `lector` WHERE
codigo_lector='$codigo_lector'";

$resultado3 = mysqli_query($conexion,$query3);

if (!$resultado3)
{
    echo "La consulta SQL contiene errores.".mysql_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta

    while ($row3 = mysqli_fetch_row($resultado3))
    {
        echo json_encode($row3);
    }
}
mysqli_close();

```

?>

```
//-----Consultar Paciente-----  
Consulta los datos basicos del paciente dentro de la base de datos  
//-----
```

<?php

```
$cedula=$_GET['cedula'];  
  
//;"  
$query = "SELECT * FROM `paciente_historial` WHERE cedula='$cedula';";  
  
//echo $query;  
$server="localhost";  
$user="root";  
$password="RFID4682";  
$database="hospital_digital_v2";  
  
$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede  
conectar con la Base de Datos \n");  
//@mysqli_select_db($database) or die("No se puede alcanzar la base de datos");  
  
mysqli_set_charset($conexion,'utf8');  
$resultado = mysqli_query($conexion,$query);  
  
if (! $resultado)  
{  
    echo "La consulta SQL contiene errores.".mysqli_error();  
    exit();  
}  
else  
{  
  
    //obtenemos los datos resultado de la consulta  
    while ($row = mysqli_fetch_row($resultado))  
    {  
        echo json_encode($row);  
    }  
}  
  
mysqli_close();
```

?>

```
//-----Consultar Personal-----  
Consulta los datos basicos del personal dentro de la base de datos  
//-----
```

<?php

```

$username=$_GET['username'];

//;"
$query = "SELECT * FROM `personal` WHERE username='$username'";

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$databse) or die("No se puede
conectar con la Base de Datos \n");
//@mysqli_select_db($databse) or die("No se puede alcanzar la base de datos");

mysqli_set_charset($conexion,'utf8');
$resultado = mysqli_query($conexion,$query);

if (! $resultado)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta
    while ($row = mysqli_fetch_row($resultado))
    {
        echo json_encode($row);
    }
}

mysqli_close();

?>

//-----Dar Alta-----
Desempareja la manilla de la base de datos cuando el paciente abandona el hospital
//-----

<?php

$cedula=$_GET['cedula'];

$query = "DELETE FROM `braz_paciente` WHERE `cedula` = $cedula";

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

```

```

$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede
conectar con la Base de Datos \n");
//@mysqli_select_db($database) or die("No se puede alcanzar la base de datos");

mysqli_set_charset('utf8');
mysqli_query($conexion,$query);
mysqli_close();

?>

//-----Eliminar Personal-----
Elimina un registro de la tabla de personal
//-----

<?php

$id=$_GET['id'];

$query = "DELETE FROM `personal` WHERE id_personal='$id'";

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$database="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede
conectar con la Base de Datos \n");
//@mysqli_select_db($database) or die("No se puede alcanzar la base de datos");

mysqli_set_charset('utf8');
$resultado = mysqli_query($conexion,$query);

if (! $resultado)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta
    while ($row = mysqli_fetch_row($resultado))
    {
        echo json_encode($row);
    }
}

mysqli_close();

```

?>

```
//-----Lista Espera-----  
Script usado por el lector de recepcion para registrar los datos del brazalete a asignarse  
//-----
```

<?php

```
$num_serie=$_GET['num_serie'];  
$id=$_GET['id'];
```

```
$query = "INSERT INTO `lista_espera` (`id`, `fecha`, `num_serie`) VALUES (1,  
CURRENT_TIMESTAMP, $num_serie) ON DUPLICATE KEY UPDATE  
num_serie=$num_serie, fecha=CURRENT_TIMESTAMP;";
```

```
//echo $query;  
$server="localhost";  
$user="root";  
$password="RFID4682";  
$database="hospital_digital_v2";
```

```
$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede  
conectar con la Base de Datos \n");  
//@mysql_select_db($database) or die("No se puede alcanzar la base de datos");
```

```
mysqli_set_charset('utf8');  
mysqli_query($conexion,$query);
```

```
mysqli_close();
```

?>

```
//-----Listar Pacientes-----  
Hace una lista de todos los pacientes que se encuentran en el hospital  
//-----
```

<?php

```
$query = "SELECT `cedula`, `primer_nombre`, `primer_apellido`,  
`segundo_apellido`, `fecha_nacimiento`, `sexo` FROM `paciente_historial`";
```

```
//echo $query;  
$server="localhost";  
$user="root";  
$password="RFID4682";  
$database="hospital_digital_v2";
```

```
$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede  
conectar con la Base de Datos \n");
```

```
mysqli_set_charset($conexion,'utf8');  
$resultado = mysqli_query($conexion,$query);
```

```

        if (!$resultado)
        {
            echo "La consulta SQL contiene errores.".mysql_error();
            exit();
        }
        else
        {

            //obtenemos los datos resultado de la consulta
            while ($row = mysqli_fetch_row($resultado))
            {
                echo json_encode($row);
            }

        }

        mysqli_close();

?>

//-----Listar Personal-----
Hace una lista del personal en el hospital
//-----
<?php

    $query = "SELECT
`id_personal`,`primer_nombre`,`primer_apellido`,`segundo_apellido`,`sexo`,`cargo`,`especialidad` FROM `personal`";

    //echo $query;
    $server="localhost";
    $user="root";
    $password="RFID4682";
    $database="hospital_digital_v2";

    $conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede
conectar con la Base de Datos \n");

    mysqli_set_charset($conexion,'utf8');
    $resultado = mysqli_query($conexion,$query);

    if (!$resultado)
    {
        echo "La consulta SQL contiene errores.".mysql_error();
        exit();
    }
    else
    {

        //obtenemos los datos resultado de la consulta

```

```

        while ($row = mysqli_fetch_row($resultado))
        {
            echo json_encode($row);
        }
    }

    mysqli_close();

?>

//-----Listar Pacientes por Cedula-----
Busca un determinado paciente en la bd mediante su cedula
//-----

<?php

    $cedula=$_GET['cedula'];

    $query = "SELECT
`cedula`,`primer_nombre`,`primer_apellido`,`segundo_apellido`,`fecha_nacimiento`,`sexo`
FROM `paciente_historial` WHERE cedula='$cedula'";

    //echo $query;
    $server="localhost";
    $user="root";
    $password="RFID4682";
    $database="hospital_digital_v2";

    $conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede
conectar con la Base de Datos \n");
    //@mysqli_select_db($database) or die("No se puede alcanzar la base de datos");

    mysqli_set_charset('utf8');
    $resultado = mysqli_query($conexion,$query);
    if (! $resultado)
    {
        echo "La consulta SQL contiene errores.".mysqli_error();
        exit();
    }
    else
    {
        //obtenemos los datos resultado de la consulta
        while ($row = mysqli_fetch_row($resultado))
        {
            echo json_encode($row);
        }
    }

    mysqli_close();

```

?>

//-----Listar Personal por Cedula-----

Busca un determinado medico/enfermera en la bd mediante su cedula

//-----

<?php

```
$id=$_GET['id'];
```

```
$query ="SELECT `id_personal`, `primer_nombre`, `primer_apellido`,  
`segundo_apellido`, `sexo`, `cargo`, `especialidad` FROM `personal` WHERE  
id_personal='$id';
```

```
//echo $query;  
$server="localhost";  
$user="root";  
$password="RFID4682";  
$database="hospital_digital_v2";
```

```
$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede  
conectar con la Base de Datos \n");
```

```
//@mysqli_select_db($database) or die("No se puede alcanzar la base de datos");
```

```
mysqli_set_charset('utf8');
```

```
$resultado = mysqli_query($conexion,$query);
```

```
if (! $resultado)
```

```
{  
    echo "La consulta SQL contiene errores.".mysqli_error();  
    exit();  
}
```

```
else
```

```
{
```

```
    //obtenemos los datos resultado de la consulta
```

```
    while ($row = mysqli_fetch_row($resultado))
```

```
    {  
        echo json_encode($row);  
    }
```

```
}
```

```
mysqli_close();
```

?>

//-----Log In-----

Verifica los datos del usuario para el sistema

//-----

```
<?php
```

```
header("Content-Type: text/html;charset=utf-8");

$usuario=$_GET['usuario'];
$contrasena=$_GET['contrasena'];

$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$databse) or die("No se puede
conectar con la Base de Datos \n");

mysqli_query($conexion,"SET NAMES 'utf8'");

mysqli_set_charset('utf8');

$query = "SELECT `upassword` FROM `personal` WHERE username='$usuario'";

$resultado = mysqli_query($conexion,$query);

if (!$resultado)
{
    echo "La consulta SQL contiene errores.".mysql_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta

    while ($row = mysqli_fetch_array($resultado))
    {
        $pass_bd = current($row);

    }
    //Selecciona la contraseña del usuario, guarda en variable
}

if(strcmp($pass_bd,$contrasena)==0)
{
    echo 1;
}
else
{
    echo 0;
}

echo ",";
```

```

$query = "SELECT `permissions` FROM `personal` WHERE username='$usuario'";

$resultado = mysqli_query($conexion,$query);

while ($row = mysqli_fetch_row($resultado))
{
    echo $row[0];
}

mysqli_close();

?>

//-----Registrar Cirugia-----
Registra una cirugia del paciente en la BD
//-----

<?php

$tipo=$_GET['tipo'];
$descripcion=$_GET['descripcion'];
$cedula=$_GET['cedula'];
$id=$_GET['id'];

$query = "INSERT INTO `cirugias_previas` (`tipo`, `descripcion`, `cedula`,
`id_personal`) VALUES ('$tipo', '$descripcion','$cedula', '$id')";

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$databse) or die("No se puede
conectar con la Base de Datos \n");
//@mysql_select_db($databse) or die("No se puede alcanzar la base de datos");

mysqli_set_charset('utf8');
mysqli_query($conexion,$query);
mysqli_close();

?>

//-----Registrar Consulta-----
Registra una consulta medica del paciente en la BD
//-----

```

```

<?php

$sintomas=$_GET['sintomas'];
$diagnostico=$_GET['diagnostico'];
$medicacion=$_GET['medicacion'];
$num_historial=$_GET['num_historial'];
$reg_senescyt=$_GET['reg_senescyt'];

$query = "INSERT INTO `consulta` (`sintomas`,`diagnostico`,`medicacion`,`num_historial`,`reg_senescyt`) VALUES ('$sintomas','$diagnostico','$medicacion','$num_historial','$reg_senescyt')";

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysql_connect($server,$user,$password,$databse) or die("No se puede conectar con la Base de Datos \n");
//@mysql_select_db($databse) or die("No se puede alcanzar la base de datos");

mysqli_set_charset('utf8');
mysqli_query($conexion,$query);
mysqli_close();

?>

```

```

//-----Registrar Localizacion-----
Script ejecutado por el lector para registrar una ubicacion en la base de datos
//-----

```

```

<?php

$codigo_lector=$_GET['codigo_lector'];
$num_serie=$_GET['num_serie'];

$query = "INSERT INTO `loc_paciente` (`codigo_lector`,`num_serie`) VALUES ('$codigo_lector','$num_serie')";
$query2 = "INSERT INTO `loc_personal` (`codigo_lector`,`num_serie`) VALUES ('$codigo_lector','$num_serie')";

//echo $query;

$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

```

```

$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede
conectar con la Base de Datos \n");
//@mysql_select_db($database) or die("No se puede alcanzar la base de datos");

mysqli_set_charset('utf8');
mysqli_query($conexion,$query);
mysqli_query($conexion,$query2);
mysqli_close();

?>

//-----Registrar Paciente-----
Registra un nuevo paciente en la base de datos
//-----

<?php

$cedula=$_GET['cedula'];
$primer_nombre=$_GET['primer_nombre'];
$segundo_nombre=$_GET['segundo_nombre'];
$primer_apellido=$_GET['primer_apellido'];
$segundo_apellido=$_GET['segundo_apellido'];
$sexo=$_GET['sexo'];
$fecha_nacimiento=$_GET['fecha_nacimiento'];
$nacionalidad=$_GET['nacionalidad'];
$direccion=$_GET['direccion'];
$estado_civil=$_GET['estado_civil'];

$query = "INSERT INTO `paciente_historial` (`cedula`, `primer_nombre`,
`segundo_nombre`, `primer_apellido`, `segundo_apellido`, `sexo`,
`fecha_nacimiento`, `nacionalidad`, `direccion`, `estado_civil`) VALUES ('$cedula',
'$primer_nombre', '$segundo_nombre',
'$primer_apellido', '$segundo_apellido','$sexo', '$fecha_nacimiento', '$nacionalidad',
'$direccion', '$estado_civil')";

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$database="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede
conectar con la Base de Datos \n");
//@mysql_select_db($database) or die("No se puede alcanzar la base de datos");

mysqli_set_charset('utf8');
mysqli_query($conexion,$query);
mysqli_close();

?>

//-----Registrar Personal-----
Registra un nuevo medico/enfermera en el la BD
//-----

```

<?php

```
$primer_nombre=$_GET['pri_nombre'];
$segundo_nombre=$_GET['seg_nombre'];
$primer_apellido=$_GET['pri_apellido'];
$segundo_apellido=$_GET['seg_apellido'];
$sexo=$_GET['sexo'];
$cargo=$_GET['cargo'];
$especialidad=$_GET['esp'];
$username=$_GET['user'];
$password=$_GET['pass'];
$permissions=$_GET['level'];
$codigo_hospital=$_GET['cod_hosp'];

$query = "INSERT INTO `personal` (`primer_nombre`, `segundo_nombre`,
`primer_apellido`, `segundo_apellido`, `sexo`,
`cargo`, `especialidad`, `username`, `password`, `permissions`, `codigo_hospital`)
VALUES ('$primer_nombre', '$segundo_nombre',
'$primer_apellido', '$segundo_apellido', '$sexo', '$cargo', '$especialidad',
'$username', '$password', '$permissions', '$codigo_hospital')";

//echo $query;
$server="localhost";
$user="root";
$password="RFID4682";
$database="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede
conectar con la Base de Datos \n");
//@mysql_select_db($database) or die("No se puede alcanzar la base de datos");

mysqli_set_charset('utf8');
mysqli_query($conexion,$query);
mysqli_close();
```

?>

```
//-----Buscar Final-----
Busca las coordenadas del paciente en la base de datos
//-----
```

<?php

```
$cedula=$_GET['cedula'];

$query = "SELECT `num_serie` FROM `braz_paciente` WHERE
cedula='$cedula'";

//echo $query;

$server="localhost";
$user="root";
```

```

$password="RFID4682";
$database="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$database) or die("No se puede
conectar con la Base de Datos \n");
//@mysql_select_db($database) or die("No se puede alcanzar la base de datos");

mysqli_set_charset($conexion,'utf8');
$resultado = mysqli_query($conexion,$query);

if (! $resultado)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{

    //obtenemos los datos resultado de la consulta
    while ($row = mysqli_fetch_array($resultado))
    {
        $num_serie = current($row);
    }

}

$query3 = "SELECT `fecha`,`codigo_lector` FROM `loc_paciente` WHERE
num_serie='$num_serie'";

$resultado3 = mysqli_query($conexion,$query3);

if (! $resultado3)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{

    //obtenemos los datos resultado de la consulta

    while ($row3 = mysqli_fetch_array($resultado3))
    {
        $valor = $row3[1] ;
    }

    $cod_lector = $valor;

}

$query4 = "SELECT `coor_x`,`coor_y` FROM `lector` WHERE
codigo_lector='$cod_lector'";

$resultado4 = mysqli_query($conexion,$query4);

```

```

if (! $resultado4)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta

    while ($row4 = mysqli_fetch_row($resultado4))
    {
        echo json_encode($row4);
    }
}

mysqli_close();

?>

//-----Buscar Inicio-----
Busca las coordenadas del personal en la base de datos
//-----

<?php

$username=$_GET['username'];

$query = "SELECT `id_personal` FROM `personal` WHERE
username='$username'";

//echo $query;

$server="localhost";
$user="root";
$password="RFID4682";
$databse="hospital_digital_v2";

$conexion=mysqli_connect($server,$user,$password,$databse) or die("No se puede
conectar con la Base de Datos \n");
//@mysqli_select_db($databse) or die("No se puede alcanzar la base de datos");

mysqli_set_charset($conexion,'utf8');
$resultado = mysqli_query($conexion,$query);

if (! $resultado)
{
    echo "La consulta SQL contiene errores.".mysqli_error();
    exit();
}
else
{

    //obtenemos los datos resultado de la consulta
    while ($row = mysqli_fetch_array($resultado))

```

```

        {
            $id_personal = current($row);
        }
    }

    $query2 = "SELECT `num_serie` FROM `braz_personal` WHERE
id_personal=$id_personal";

    $resultado2 = mysqli_query($conexion,$query2);

    if (! $resultado2)
    {
        echo "La consulta SQL contiene errores.".mysqli_error();
        exit();
    }
    else
    {
        //obtenemos los datos resultado de la consulta

        while ($row2 = mysqli_fetch_array($resultado2))
        {
            $num_serie = current($row2);
        }
    }

    $query3 = "SELECT `fecha`,`codigo_lector` FROM `loc_personal` WHERE
num_serie='$num_serie'";

    $resultado3 = mysqli_query($conexion,$query3);

    if (! $resultado3)
    {
        echo "La consulta SQL contiene errores.".mysqli_error();
        exit();
    }
    else
    {
        //obtenemos los datos resultado de la consulta

        while ($row3 = mysqli_fetch_array($resultado3))
        {
            $valor = $row3[1] ;
        }

        $cod_lector = $valor;
    }

    $query4 = "SELECT `coor_x`,`coor_y` FROM `lector` WHERE
codigo_lector='$cod_lector'";

    $resultado4 = mysqli_query($conexion,$query4);

    if (! $resultado4)

```

```
{
    echo "La consulta SQL contiene errores.".mysql_error();
    exit();
}
else
{
    //obtenemos los datos resultado de la consulta

    while ($row4 = mysqli_fetch_row($resultado4))
    {
        echo json_encode($row4);
    }
}
mysqli_close();
```

?>

## Anexo 3: Estructura de la Base de Datos

### DICCIONARIO DE METADATOS

Entidad: **PACIENTE\_HISTORIAL**

Descripción: Tabla que contendrá la información básica del paciente, así como un resumen de su historial médico

<b>Campo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
cedula (PK)	int	10	Clave única de Registro de Población
primer_nombre	varchar	15	Primer Nombre del Paciente
segundo_nombre	varchar	15	Segundo Nombre del Paciente
primer_apellido	varchar	15	Primer Apellido del Paciente
segundo_apellido	varchar	15	Segundo Apellido del Paciente
sexo	varchar	10	Género del Paciente
fecha_nacimiento	date		Fecha de Nacimiento dd/mm/yy del paciente
nacionalidad	varchar	15	Pais de Nacimiento del Paciente
direccion	varchar	40	Dirección actual del Paciente
estado_civil	varchar	15	Describe el estado civil actual del paciente
ant_personal	varchar	300	Antecedentes personal físicos y psicológicos del paciente
ant_patologicos	varchar	300	Antecedentes patológicos, sobre enfermedades previas
ant_ofthalmologico	varchar	300	Antecedentes sobre enfermedades de garganta del paciente
tipo_sangre	varchar	5	Tipo de sangre del paciente

Entidad: **PERSONAL**

Descripción: Tabla que contendrá la información básica del personal del hospital

<b>Campo</b>	<b>Tipo</b>	<b>Longitud</b>	<b>Descripción</b>
id_personal	bigint	20	Número único del título habilitante del médico
primer_nombre	varchar	20	Primer Nombre del Médico
segundo_nombre	varchar	20	Segundo Nombre del Médico
primer_apellido	varchar	20	Primer Apellido del Médico
segundo_apellido	varchar	20	Segundo Apellido del Médico
sexo	varchar	10	Género del Médico
cargo	varchar	20	Cargo del paciente en el hospital
especialidad	varchar	20	Especialidad del médico, en caso de no existir indicar general
username	varchar	50	Nombre de usuario dentro del sistema
upassword	varchar	30	Contraseña del usuario dentro del sistema
permissions	bigint	20	Nivel de permisos del usuario en el sistema
codigo_hospital(FK)	int	10	Clave Foranea

Entidad: **BRAZ\_PACIENTE**

Descripción: Entidad que contiene información sobre la pulsera RFID que se le asigna a cada paciente

Campo	Tipo	Longitud	Descripción
num_serie (PK)	int	10	Número único de serie de la tarjeta RFID en la pulsera
fecha	date		Fecha de Adquisición del Brazalete
cedula (FK)	int	10	Clave Foránea

Entidad: **BRAZ\_PERSONAL**

Descripción: Entidad que contiene información sobre la pulsera RFID que se le asigna a cada paciente

Campo	Tipo	Longitud	Descripción
num_serie (PK)	int	10	Número único de serie de la tarjeta RFID en la pulsera
fecha	date		Fecha de Adquisición del Brazalete
id_personal (FK)	bigint	20	Clave Foránea

Entidad: **HOSPITAL**

Descripción: Entidad que posee información sobre el hospital en el cual se encuentra el paciente

Campo	Tipo	Longitud	Descripción
codigo_hospital (PK)	int	10	Número único de identificación del hospital en el país
nombre	varchar	40	Nombre del Hospital
provincia	varchar	40	Provincia en la cual se encuentra el hospital
ciudad	varchar	10	Ciudad en la que se encuentra el hospital
dirección	varchar	40	Dirección de la casa de salud

Entidad: **LECTOR**

Descripción: Lector RFID situado en un determinado departamento o área

Campo	Tipo	Longitud	Descripción
codigo_lector (PK)	int	10	Número único de identificación del lector en el hospital
departamento	varchar	20	Departamento del hospital en el cual se encuentra ubicado el lector
area	varchar	20	Área dentro del departamento en la que se ubica el lector
coor_x	int	10	Coordenada x en el plano del hospital
coor_y	int	10	Coordenada y en el plano del hospital
planta	varchar	20	Planta del hospital en donde se encuentra ubicado el lector
codigo_hospital (FK)	int	10	Clave Foránea

Entidad: **OPERACION\_MEDICA**

Descripción: Contiene información acerca de las consultas que el paciente ha tenido en el hospital

Campo	Tipo	Longitud	Descripción
id_operacion (PK)	int	10	Identificador único de la consulta en la BD
tipo	varchar	50	Tipo de operación: Consulta o Cirugía realizada en el paciente
fecha	timestamp		Fecha de registro automático en el sistema de la operación
sintomas	varchar	300	Sintomatología que llevó al paciente a la consulta médica
diagnostico	varchar	300	Diagnóstico del médico sobre la condición del paciente
medicacion	varchar	300	Registro de medicamentos utilizados para el tratamiento
cedula (FK)	int	10	Cédula del paciente
id_personal (FK)	bigint	20	Identificador único del personal en la base de datos

Entidad: **LOC\_PACIENTE**

Descripción: Contiene información acerca del historial de localización del paciente

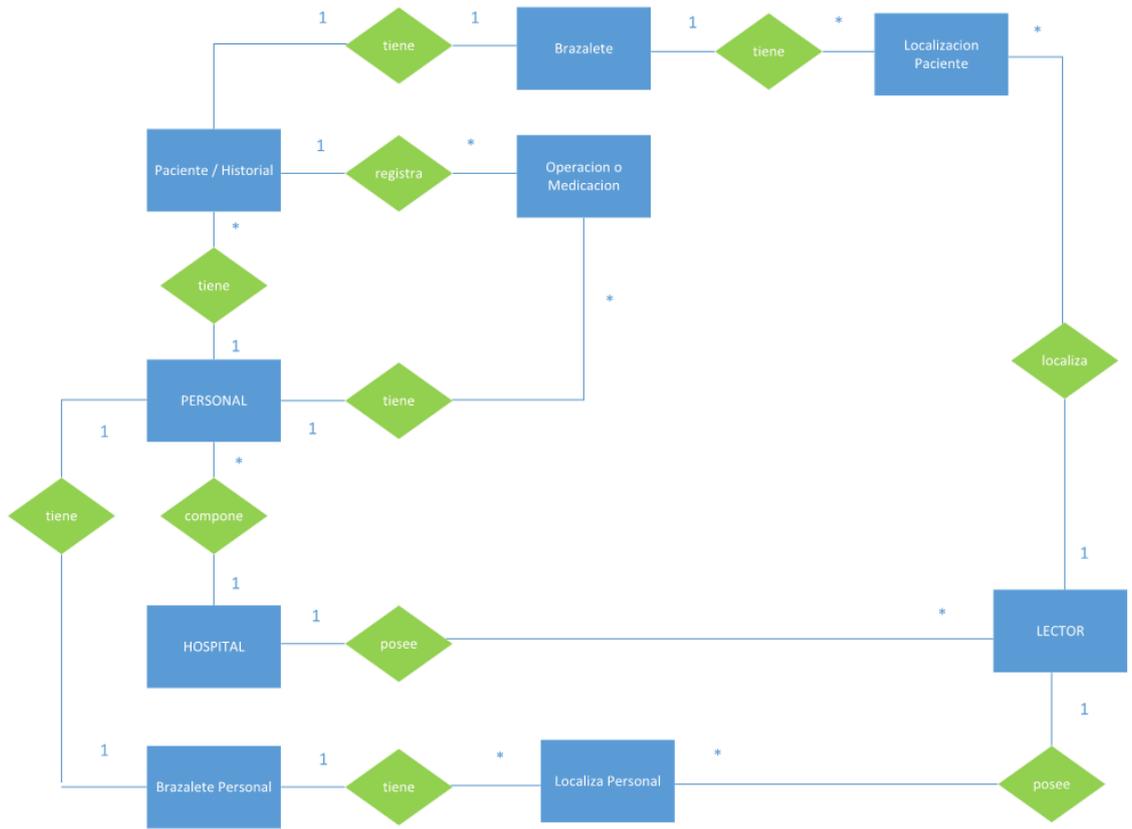
Campo	Tipo	Longitud	Descripción
num_localizacion (PK)	int	10	Identificador único de la localización
fecha	timestamp		Fecha de la Localización del Paciente
codigo_lector (FK)	int	10	Clave Foránea
num_serie (FK)	int	10	Clave Foránea

Entidad: **LOC\_PERSONAL**

Descripción: Contiene información acerca del historial de localización del personal

Campo	Tipo	Longitud	Descripción
num_localizacion (PK)	int	10	Identificador único de la localización
fecha	timestamp		Fecha de la Localización del Paciente
codigo_lector (FK)	int	10	Clave Foránea
num_serie (FK)	int	10	Clave Foránea

#### DIAGRAMA ENTIDAD – RELACION



## Anexo 4: Código de la Aplicación

A continuación, se incluyen las clases más significativas de la aplicación para referencias de los archivos Java y XML usados para programar toda la aplicación consulte la información proporcionada en el disco

```
//-----Clase LogIn-----
Gestiona el ingreso de los usuarios en la aplicación mediante el uso de usuario y contraseña
//-----

package app.juanorellana.com.hospitaldigital2;

import android.content.Intent;
import android.os.AsyncTask;
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.ProgressBar;
import android.widget.Toast;

import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;
import java.io.UnsupportedEncodingException;
import java.net.HttpURLConnection;
import java.net.URL;
import java.util.Random;

public class LoginActivity extends AppCompatActivity implements View.OnClickListener
{

    Button btnLogIn;
    EditText txtUsername, txtPassword;
    ProgressBar barraLog;
    String todo;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        getSupportActionBar().setIcon(R.mipmap.ic_launcher);

        btnLogIn = (Button) findViewById(R.id.btnLogIn);
        btnLogIn.setOnClickListener(this);

        txtUsername = (EditText) findViewById(R.id.txtUsuario);
        txtPassword = (EditText) findViewById(R.id.txtContrasena);
    }
}
```

```

txtUsername.setText("");
txtPassword.setText("");

barraLog = (ProgressBar) findViewById(R.id.barLogIn);
barraLog.setVisibility(View.GONE);

}

@Override
public void onClick(View v) {
    switch (v.getId())
    {
        case R.id.btnLogIn:

            String usuario = txtUsername.getText().toString();
            String contrasena = txtPassword.getText().toString();
            comprobarLog(usuario, contrasena);
            break;
        }
    }

public void comprobarLog(String user, String password)
{
    String login_url = URLs_Servidor.getDns_server() + "login.php?usuario=" + user +
"&contrasena=" + password;
    new attemptLogIn().execute(login_url);
}

private class attemptLogIn extends AsyncTask<String, Integer, Boolean>
{
    @Override
    protected void onPreExecute()
    {
        super.onPreExecute();
        btnLogIn.setEnabled(false);
        barraLog.setMax(100);
        barraLog.setProgress(0);
        barraLog.setEnabled(true);
    }
    @Override
    protected Boolean doInBackground(String... params)
    {
        Random rnd = new Random();
        Integer progress = rnd.nextInt(100);

        try
        {
            publishProgress(progress);

            //Se obtiene la URL
            String direccion = params[0];
            ConectividadHTTP discharge = new ConectividadHTTP();
            todo = discharge.peticonGET(direccion);
        }
    }
}

```

```

String verificacion = todo.substring(0,1);

if (verificacion.equals("1"))
{
    return true;
}
else
{
    return false;
}

}
catch (Exception e)
{
    return false;
} //fin del try catch
}

@Override
protected void onPostExecute(Boolean s)
{
    //super.onPostExecute(s);
    btnLogIn.setEnabled(true);
    barraLog.setVisibility(View.GONE);

    if(s)
    {
        Intent intent = new Intent(LoginActivity.this,MainMenuActivity.class);
        intent.putExtra("usuario",txtUsername.getText().toString());
        intent.putExtra("permiso",todo.substring(2,3));
        startActivity(intent);
    }
    else
    {
        Toast.makeText(getBaseContext(),"Usuario o contraseña
invalido",Toast.LENGTH_SHORT).show();
    }
}

@Override
protected void onProgressUpdate(Integer... values)
{
    super.onProgressUpdate(values);
    barraLog.setVisibility(View.VISIBLE);
    barraLog.setProgress(values[0]);
}

@Override
protected void onCancelled()
{
    super.onCancelled();
}
} //fin de la clase attemptLogIn
}

```

```
//-----Clase Consultar Historial-----  
Realiza la consulta del resumen del historial médico del paciente con los métodos HTTP  
//-----
```

```
package app.juanorellana.com.hospitaldigital2;
```

```
import android.content.Intent;  
import android.os.AsyncTask;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.AdapterView;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.ImageView;  
import android.widget.ListView;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.io.Reader;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.util.ArrayList;
```

```
public class ConsultarHistorial extends AppCompatActivity implements  
View.OnClickListener  
{
```

```
    ListView listaPacientes;  
    EditText cedulaConsultar;  
    Button btnConsultar, btnListarPacientes;  
    ProgressBar barProgressCons;
```

```
    String[][] datosPacientes;
```

```
    int indexExtra=0;  
    Integer permiso;
```

```
    ArrayList<Patient_List_Input> datos = new ArrayList<Patient_List_Input>();
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState)  
    {
```

```
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_consultar_historial);  
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
        getSupportActionBar().setIcon(R.mipmap.ic_launcher);
```

```
        datos.add(new Patient_List_Input(R.drawable.avatar_man, "Nombre del Paciente",  
"XX", "Masculino"));  
        datos.add(new Patient_List_Input(R.drawable.avatar_woman, "Nombre de la
```

```

Paciente", "XX", "Femenino"));

cedulaConsultar = (EditText) findViewById(R.id.txtCedulaHistorial);

listaPacientes = (ListView) findViewById(R.id.listViewPacientes);
btnConsultar = (Button) findViewById(R.id.btnConsultarHistorial);
btnConsultar.setOnClickListener(this);

btnListarPacientes = (Button) findViewById(R.id.btnListPatHisto);
btnListarPacientes.setOnClickListener(this);

barProgressCons = (ProgressBar) findViewById(R.id.barraConsultarPacientes);
barProgressCons.setVisibility(View.GONE);

listaPacientes.setAdapter(new List_Adapter(this,R.layout.lista_vista_paciente,datos)
{
    @Override
    public void onEntrada(Object entrada, View view)
    {
        TextView nombrePaciente = (TextView)
view.findViewById(R.id.txtPatientNameView);
        nombrePaciente.setText(((Patient_List_Input) entrada).getPatientName());

        TextView edadPaciente = (TextView) view.findViewById(R.id.txtAgeView);
        edadPaciente.setText(((Patient_List_Input) entrada).getPatientAge());

        TextView sexoPaciente = (TextView) view.findViewById(R.id.txtGenreView);
        sexoPaciente.setText(((Patient_List_Input) entrada).getPatientSex());

        ImageView fotoPaciente = (ImageView) view.findViewById(R.id.imgPhoto);
        fotoPaciente.setImageResource(((Patient_List_Input)entrada).getPatientPhoto());
    }
});

listaPacientes.setOnClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id)
    {
        Intent consHistorial = new Intent(ConsultarHistorial.this, ConsultaActivity.class);
        consHistorial.putExtra("cedula",datosPacientes[position][0]);
        consHistorial.putExtra("permiso",permiso);
        startActivity(consHistorial);
    }
});

int indice = recogerExtras();

switch (indice)
{
    case 0:
        Toast.makeText(ConsultarHistorial.this, "Ingrese el número de cedula del
paciente del historial a consultar", Toast.LENGTH_SHORT).show();
        break;

```

```

        case 1:
            Toast.makeText(ConsultarHistorial.this,"Ingrese el número de cedula de la
            cirugía",Toast.LENGTH_SHORT).show();
            break;

        case 2:
            Toast.makeText(ConsultarHistorial.this,"Ingrese el número de cedula del
            paciente al que se realizó la consulta",Toast.LENGTH_SHORT).show();
            break;

        case 3:Toast.makeText(ConsultarHistorial.this,"Ingrese el número de del paciente
            que desea ver las imagenes médicas",Toast.LENGTH_SHORT).show();
            break;

        default:
            Toast.makeText(ConsultarHistorial.this,"Error
            Desconocido",Toast.LENGTH_SHORT).show();
            break;
    }

}

public int recogerExtras()
{
    try
    {
        //Aquí recogemos y tratamos los parámetros
        Bundle extras = getIntent().getExtras();
        int s = extras.getInt("indicador");
        permiso = extras.getInt("permiso");
        return s;
    }
    catch(Exception e)
    {
        Toast.makeText(getBaseContext(),"Error al Iniciar,
        regrese",Toast.LENGTH_SHORT).show();
        return 0;
    }
}

@Override
public void onClick(View v)
{
    switch (v.getId())
    {
        case R.id.btnConsultarHistorial:

            Funciones_Cadenas prima = new Funciones_Cadenas();
            String cedula = cedulaConsultar.getText().toString();

            boolean verificador = prima.verificarCedula(cedula);

```

```

        if(verificador)
        {
            Toast.makeText(this,"Cedula Correcta,
Consultando...",Toast.LENGTH_SHORT).show();
            new PatientRequestHTTP().execute(cedula);
        }
        else
        {
            Toast.makeText(this,"Cedula Incorrecta, Verifique el número de
cédula",Toast.LENGTH_SHORT).show();
        }

        break;

    case R.id.btnListPatHisto:

        Toast.makeText(this,"Consultando.
Espere...",Toast.LENGTH_SHORT).show();
        new ListPatientHTTP().execute();

    default:

        break;
    }
}
//fin del metodo onClick

```

```

private class PatientRequestHTTP extends AsyncTask<String,Integer,String>
{
    @Override
    protected void onPreExecute()
    {
        btnConsultar.setEnabled(false);
        btnListarPacientes.setEnabled(false);

        listaPacientes.setEnabled(false);
        listaPacientes.setVisibility(View.GONE);

        barProgressCons.setProgress(0);
        barProgressCons.setMax(100);
        barProgressCons.setVisibility(View.VISIBLE);
    }

    @Override
    protected String doInBackground(String... params)
    {
        try
        {
            //Construir la url utilizando el nuevo metodo

            String url = URLs_Servidor.getListPatients();
            String query = "cedula=" +params[0];
            ConectividadHTTP obj = new ConectividadHTTP();
            String resp = obj.peticionGET(url+query);

```



```

listaPacientes.setEnabled(false);
listaPacientes.setVisibility(View.GONE);

barProgressCons.setProgress(0);
barProgressCons.setMax(100);
barProgressCons.setVisibility(View.VISIBLE);

}

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

    try
    {
        //Se pasa directamente la URL para listar los pacientes

        String url = URLs_Servidor.getListAllPatients();

        URL link_cons = new URL(url);

        HttpURLConnection conexion = (HttpURLConnection)
link_cons.openConnection();
        conexion.setRequestProperty("Accept-Encoding", "identity");
        conexion.setRequestMethod("GET");
        conexion.setDoInput(true);
        conexion.connect();

        InputStream entrada = conexion.getInputStream();

        Reader reader = new InputStreamReader(entrada, "UTF-8");

        int length = conexion.getContentLength();

        publishProgress(length);

        char[] buffer = new char[length];
        reader.read(buffer);
        return new String(buffer);

    }
    catch(Exception e)
    {
        Toast.makeText(getBaseContext(), "Se ha producido un error",
Toast.LENGTH_SHORT);
    }

    return null;
}

@Override
protected void onProgressUpdate(Integer... values)
{
    super.onProgressUpdate(values);
}

```

```

}

@Override
protected void onPostExecute(String valores)
{
    barProgressCons.setVisibility(View.GONE);

    btnConsultar.setEnabled(true);
    btnListarPacientes.setEnabled(true);

    listaPacientes.setEnabled(true);
    listaPacientes.setVisibility(View.VISIBLE);

    StringDecode separador = new StringDecode();

    String [][] listaPacientes = separador.separaPalabras(valores);

    llenarLista(listaPacientes);
}

@Override
protected void onCancelled()
{
    super.onCancelled();
}
} //fin de la clase ListPatientHTTP

private void llenarLista(String[][] datosLista)
{
    datos.clear();
    datosPacientes = datosLista;

    //Estructura de datosLista
    //datos[][0]:Cedula
    //datos[][1]:Primer Nombre, datos[][2]:Primer Apellido, datos[][3]:Segundo Apellido
    //datos[][4]:FechaDeNacimiento, datos[][5]:Sexo

    Funciones_Cadenas calc = new Funciones_Cadenas();

    for(int i=0; i<datosLista.length;i++)
    {
        String nombre = datosLista[i][1] + " " + datosLista[i][2] + " " + datosLista[i][3];
        String fNacimiento = datosLista[i][4];
        String sexo = datosLista[i][5];

        String edad = String.valueOf(calc.calcularEdad(fNacimiento));

        if(sexo.equals("Femenino"))
        {
            datos.add(new Patient_List_Input(R.drawable.avatar_woman, nombre, edad,
sexo));
        }
        else

```



```

getSupportActionBar().setIcon(R.mipmap.ic_launcher);

listaLocali = (ListView) findViewById(R.id.lstLocalizacion);

datosLocalizacion.add(new Patient_List_Input(R.drawable.avatar_man, "Nombre del
Paciente", "XX", "Cirugia"));
datosLocalizacion.add(new Patient_List_Input(R.drawable.avatar_woman, "Nombre
de la Paciente", "XX", "Cirugia"));

new ListPatientsHTTP().execute();

listaLocali.setAdapter(new List_Adapter(this, R.layout.lista_vista_localizacion,
datosLocalizacion) {
    @Override
    public void onEntrada(Object entrada, View view) {
        TextView nombrePaciente = (TextView)
view.findViewById(R.id.txtPatientNameViewLocation);
        nombrePaciente.setText(((Patient_List_Input) entrada).getPatientName());

        TextView edadPaciente = (TextView)
view.findViewById(R.id.txtAgeViewLocation);
        edadPaciente.setText(((Patient_List_Input) entrada).getPatientAge());

        TextView sexoPaciente = (TextView) view.findViewById(R.id.txtLocationView);
        sexoPaciente.setText(((Patient_List_Input) entrada).getPatientSex());

        ImageView fotoPaciente = (ImageView)
view.findViewById(R.id.imgPhotoLocation);
        fotoPaciente.setImageResource(((Patient_List_Input) entrada).getPatientPhoto());
    }
});

listaLocali.setOnItemClickListener(new AdapterView.OnItemClickListener() {

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id)
    {

        if(datosPacientes[position][2].equals("Paciente Inactivo"))
        {
            Toast.makeText(LocationActivity.this, "El paciente no se encuentra
internado. Imposible determinar ubicación.", Toast.LENGTH_SHORT).show();

        }
        else
        {
            Intent mapa = new Intent(LocationActivity.this, MapaActivity.class);
            mapa.putExtra("cedula", listaPacientes[position][0]);
            mapa.putExtra("usuario", usuario);
            startActivity(mapa);

        }
    }
});

```

```

    });

    usuario = recogerExtras();
}

public String recogerExtras()
{
    try
    {
        //Aquí recogemos y tratamos los parámetros
        Bundle extras = getIntent().getExtras();
        String s = extras.getString("usuario");
        return s;
    }
    catch(Exception e)
    {
        return null;
    }
}

private class ListPatientsHTTP extends AsyncTask<Void,Integer,Void>
{
    @Override
    protected void onPreExecute()
    {

    }

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

        try
        {
            String url = URLs_Servidor.getListAllPatients();

            ConectividadHTTP consultar = new ConectividadHTTP();
            String response = consultar.peticionGET(url);

            StringDecode cadena = new StringDecode();

            listaPacientes = cadena.separaPalabras(response);

            datosPacientes = new String[listaPacientes.length][4];

            Funciones_Cadenas calc = new Funciones_Cadenas();

            for (int i=0; i<listaPacientes.length; i++)
            {
                //Nombre del Paciente

```

```
        datosPacientes[i][0] = listaPacientes[i][1] + " " + listaPacientes[i][2] + " " +  
listaPacientes[i][3];
```

```
        //Edad del Paciente
```

```
        datosPacientes[i][1] = String.valueOf(calc.calcularEdad(listaPacientes[i][4]));
```

```
        //Localizacion del Paciente
```

```
        datosPacientes[i][2] = searchLocationHTTP(listaPacientes[i][0]);
```

```
        //Genero del Paciente
```

```
        datosPacientes[i][3] = listaPacientes[i][5];
```

```
    }  
}
```

```
catch(Exception e)  
{
```

```
}
```

```
return null;
```

```
}
```

```
@Override
```

```
protected void onProgressUpdate(Integer... values)
```

```
{
```

```
    super.onProgressUpdate(values);
```

```
}
```

```
@Override
```

```
protected void onPostExecute(Void valores)
```

```
{
```

```
    llenarList(datosPacientes);
```

```
}
```

```
@Override
```

```
protected void onCancelled()
```

```
{
```

```
    super.onCancelled();
```

```
}
```

```
}//fin de la clase ListPatientHTTP
```

```
private String searchLocationHTTP(String cedula)
```

```
{
```

```
    try
```

```
    {
```

```
        String url = URLs_Servidor.getLocatePatient();
```

```
        String query = "cedula=" + cedula;
```

```
        ConectividadHTTP consultar = new ConectividadHTTP();
```

```
        String response = consultar.peticionGET(url+query);
```

```
        StringDecode sep = new StringDecode();
```

```

        String localizacion[][] = sep.separaPalabras(response);
        String direccion = localizacion[0][0] + ", " + localizacion[0][1];
        return direccion;
    }
    catch(Exception e)
    {
        return "Paciente Inactivo";
    }
}

//fin del metodo SearchLocationHTTP

private void llenarList(String[][] datosPacientes)
{
    datosLocalizacion.clear();
    for(int i=0;i<datosPacientes.length;i++)
    {
        if(datosPacientes[i][3].equals("Masculino"))
        {
            datosLocalizacion.add(new Patient_List_Input(R.drawable.avatar_man,
datosPacientes[i][0], datosPacientes[i][1], datosPacientes[i][2]));
        }
        else
        {
            datosLocalizacion.add(new Patient_List_Input(R.drawable.avatar_woman,
datosPacientes[i][0], datosPacientes[i][1], datosPacientes[i][2]));
        }
    }
    listaLocali.invalidateViews();
}
}

//-----Clase Mapa -----
Dibuja el mapa del hospital y las rutas hacia los pacientes
//-----

package app.juanorellana.com.hospitaldigital2;

import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Typeface;
import android.os.AsyncTask;
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.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class MapaActivity extends AppCompatActivity implements View.OnClickListener
{

    Button Regresar; //boton Regresar
    Button Cargar; // boton Cargar Plano b/n
    Button Trazar; //trazar ruta mas corta

    EditText INICIO_0;
    EditText INICIO_1;
    EditText FIN_0;
    EditText FIN_1;
    TextView OrientacionM;
    int[] InicioLab = new int[2];
    int[] FinLab = new int[2];
    String usuario;
    String cedula;

    //VARIABLES DE GRAFICACION, PLANO Y RUTA

    // Dimensiones del bitmap
    int escala=20; //espesor de pixeles
    int ancho=36; // dimension x del fondo (desde 0,1,2,3,...)
    int alto=36; // dimension y del fondo (desde 0,1,2,3,...)

    int anchob = ancho*escala; //dimension x para dibujar
    int altob = alto*escala; //dinension y para dibujar

    int Laberinto2d[][] = new int[alto][ancho]; //laberinto en dos dimensiones alto x ancho

    //Creacion del Bitmap "b"
    Bitmap b = Bitmap.createBitmap(anchob, altob, Bitmap.Config.ARGB_8888); //crea un
fondo (bitmap)

    ///Creacion del canvas para dibujar los pixeles
    Canvas c = new Canvas(b);
    Paint paint = new Paint();

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

```

```

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setIcon(R.mipmap.ic_launcher);
recogerExtras();
new SearchStartHTTP().execute(usuario);
new SearchEndHTTP().execute(cedula);

//Asignar id a los Botones
Cargar = (Button) findViewById(R.id.btnPlano);
Trazar = (Button) findViewById(R.id.btnTrazar);
Cargar.setOnClickListener(this);
Trazar.setOnClickListener(this);
INICIO_0 = (EditText) findViewById(R.id.etI0);
INICIO_1 = (EditText) findViewById(R.id.etI1);
FIN_0 = (EditText) findViewById(R.id.etF0);
FIN_1 = (EditText) findViewById(R.id.etF1);

// ImageView imageview2 =(ImageView)findViewById(R.id.iv2); //asignacion del
imageview2
// imageview2.getResources().getDrawable(R.drawable.leyenda);

INICIO_0.setVisibility(View.GONE);
INICIO_1.setVisibility(View.GONE);
FIN_0.setVisibility(View.GONE);
FIN_1.setVisibility(View.GONE);
//Toast.makeText(getBaseContext(),"Partida: " +InicioLab[0]+ "Destino: "+
cedula,Toast.LENGTH_SHORT).show();
}

public void recogerExtras()
{
    try
    {
        //Aquí recogemos y tratamos los parámetros
        Bundle extras = getIntent().getExtras();
        cedula = extras.getString("cedula");
        usuario = extras.getString("usuario");
    }
    catch(Exception e)
    {
        Toast.makeText(getBaseContext(),"NULL",Toast.LENGTH_SHORT).show();
    }
}

private void InicioFin()
{
    //Método para leer las cajas de texto del Punto Inicial y Final
    //y actualiza a las variables de clase

    int I0 = Integer.parseInt(INICIO_0.getText().toString());
    int I1 = Integer.parseInt(INICIO_1.getText().toString());
    int F0 = Integer.parseInt(FIN_0.getText().toString());
    int F1 = Integer.parseInt(FIN_1.getText().toString());

    InicioLab[0]= I0;
}

```

```

InicioLab[1]= I1;
FinLab[0]= F0;
FinLab[1]= F1;
}

```

```

/// Funcion para cargar los datos del Plano y dibujarlo
public void cargar_plano() throws IOException
{

    //Asignacion del color del bitmap
    b.eraseColor(Color.rgb(60, 240, 240)); //asignacion del color del fondo CELESTE

    //El bitmap sera visualizado en un ImageView
    ImageView imageview=(ImageView)findViewById(R.id.ivI); //asignacion del
imageview
    imageview.setImageBitmap(b); //asignacion del fondo a un ImageView

    //Configuracion características de paint
    paint.setAntiAlias(true); //antialiasing (trazo suave)
    paint.setStrokeWidth(20F); //grosor de la linea
    paint.setStyle(Paint.Style.FILL); //rellenar con el color asignado

    ///// Cargar los Datos para el Mapa

    List<String> listado = new ArrayList<>(); //crear una lista de Strings para los datos
    String linea; //variable para guardar los datos

    InputStream is = this.getResources().openRawResource(R.raw.hospital); //cargar la
direccion y el nombre del archivo txt de los datos en "is"
    BufferedReader reader = new BufferedReader(new InputStreamReader(is));
    //almacenar los datos en un buffer "reader"

    //bucle para almacenar los datos en "listado"
    if (is!=null)
    {
        while((linea=reader.readLine())!=null)
        {
            listado.add(linea); //lee todos los datos
        }
    }
    is.close(); //cerrar el archivo txt

    //Pasar la lista "listado "a un String "listado2"
    String listado2 = new String(listado.toString());

    //Eliminar los corchetes inicial y final que se crean en "listado"
    String listado3 = new String(listado2.substring(1,listado2.length()-1));

    //System.out.println("listado3"+listado); //muestra los datos

```

```

// String Array de los datos, señalando que estan separados por coma
String Array1[] = listado3.split(",");
//System.out.println("listado3 size: "+ Array1.length); //longitud de los datos en Array1

// Array de enteros para transformar el String Array a Int Array
int[] numbers = new int[Array1.length];

//////////Bucle para Dibujar el Mapa//////////

// Contador para los datos del mapa
int contador = 0;

//Ubicacion inicial y dimensiones de los rectangulos (pixeles)
int izq, dere, sup, infe; // ubicaciones de los pixeles del mapa

izq=0*escala; //lado izquierdo
sup=0*escala; //lado superior
dere=2*escala; //lado derecho
infe=2*escala; //lado inferior

/////Bucle para DIBUJAR el Mapa/////

int x=0;
int y=0;
int[] grafico = new int[alto*ancho];

for (x=0;x<altob;x=x+20)
{
    int supx = sup +x; // cambio de filas
    int infex = infe +x;

    //System.out.println("fila: "+y);

    for (y = 0; y < anchob; y=y+ 20)
    {
        int izqx = izq +y;
        int derex = dere +y;

        //Pasar el Array de String del Mapa a un Array de enteros
        numbers[contador] = Integer.parseInt(Array1[contador]);

        //Transformar a un Array de numbers de 1 y 0 a 2255 y 0 para graficar (N:0:255,
B:1:0)

        if(numbers[contador]==0)
        {
            grafico[contador]=240;
        }
        else
        {
            grafico[contador] = 0;
        }

        //Dibuja el Mapa con el Array de Enteros (Datos)

```

```

        paint.setColor(Color.rgb(grafico[contador], grafico[contador],grafico[contador]));
//asignar los datos del Mapa (Array de Enteros)
        c.drawRect(izqx, supx, derex, infex, paint); //dibuja cada uno de los lados del
Mapa

        contador = contador+1; //contador del array enteros del mapa

    }
}
contador=0;

```

*////GRAFICA DE PUNTO MI UBICACION ///*

```

//InicioFin(); //Metodo que asigna los valores de Inicio y Fin del Laberinto
int SupI= InicioLab[0]*escala;
// System.out.println("SupI: "+ SupI);
int IzqI= InicioLab[1]*escala;
// System.out.println("IzqI: "+ IzqI);

```

```

int InfI= SupI+escala;
// System.out.println("InfI: "+ InfI);
int DerI= IzqI+escala;
// System.out.println("DerI: "+ DerI);

```

```

//Dibujar el circulo de M
paint.setColor(Color.rgb(0,0,0)); //asignar color negro
c.drawCircle(IzqI+7,SupI-4,14,paint);

```

```

paint.setColor(Color.rgb(80,240,255)); //asignar color azul
c.drawCircle(IzqI+7,SupI-4,11,paint);

```

```

//Asignar un Texto
paint.setColor(Color.rgb(0,0,0)); //asignar color
paint.setTextSize(16); //tamaño de letra
paint.setTypeface(Typeface.SANS_SERIF); //tipo de letra
c.drawText("M", InicioLab[1]*escala, InicioLab[0]*escala, paint);

```

*////GRAFICA DE PUNTO UBICACION PACIENTE ///*

```

int SupF= FinLab[0]*escala;
int IzqF= FinLab[1]*escala;

```

```

int DerF= IzqF + escala;
int InfF= SupF + escala;

```

```

//Dibujar el circulo de P

```

```

paint.setColor(Color.rgb(0,0,0)); //asignar color negro

```

```

c.drawCircle(IzqF+4,SupF-4,14,paint);

paint.setColor(Color.rgb(80,40,255)); //asignar color azul
c.drawCircle(IzqF+4,SupF-4,11,paint);

//Asignar un Texto
paint.setColor(Color.rgb(255,255,255)); //asignar color
paint.setTextSize(16); //tamaño de letra
paint.setTypeface(Typeface.DEFAULT_BOLD); //tipo de letra
c.drawText("P",FinLab[1]*escala, FinLab[0]*escala, paint);

//////////Recorrer el Mapa//////////

//Array de b/n 'numbers' en Array 1D 1x225
System.out.println("numbers: "+ Arrays.toString(numbers));
//System.out.println("numbers: "+numbers[1]);

//Guardar datos b/n de un array 1D en un Array2D alto x ancho
//int Laberinto2d[][] = new int[alto][ancho];
int count=0;
for(int i=0; i<alto;i++)
{
    for (int j = 0; j < ancho; j++)
    {
        if(count==numbers.length) break;
        // array2d[i][j] = numbers[(j * alto) + i];
        Laberinto2d[i][j] = numbers[count];
        count++;
    }
}

// System.out.println("Laberinto2d: "+Arrays.deepToString(Laberinto2d));

//Asignar Nombres de los Departamentos
paint.setColor(Color.rgb(255,0,0)); //asignar color
paint.setTextSize(20); //tamaño de letra
paint.setTypeface(Typeface.SANS_SERIF); //tipo de letra

c.drawText("E", 3*escala, 8*escala, paint); //X,Y
c.drawText("Q1", 3*escala, 15*escala, paint); //X,Y
c.drawText("Q2", 3*escala, 22*escala, paint); //X,Y
c.drawText("BA2", 3*escala, 27*escala, paint); //X,Y
c.drawText("BA3", 3*escala, 31*escala, paint); //X,Y

c.drawText("BA1", 12*escala, 7*escala, paint); //X,Y
c.drawText("H2", 12*escala, 17*escala, paint); //X,Y
c.drawText("H5", 12*escala, 22*escala, paint); //X,Y
c.drawText("CX", 12*escala, 33*escala, paint); //X,Y

c.drawText("H1", 17*escala, 7*escala, paint); //X,Y
c.drawText("H3", 17*escala, 17*escala, paint); //X,Y
c.drawText("H6", 17*escala, 22*escala, paint); //X,Y
c.drawText("CX", 17*escala, 33*escala, paint); //X,Y

```

```

c.drawText("R", 23*escala, 7*escala, paint); //X,Y
c.drawText("H4", 22*escala, 17*escala, paint); //X,Y
c.drawText("H7", 22*escala, 22*escala, paint); //X,Y
c.drawText("CL", 22*escala, 33*escala, paint); //X,Y

```

```

c.drawText("SE", 32*escala, 7*escala, paint); //X,Y
c.drawText("C", 32*escala, 14*escala, paint); //X,Y
c.drawText("SE", 32*escala, 20*escala, paint); //X,Y
c.drawText("BO", 32*escala, 25*escala, paint); //X,Y
c.drawText("Q3", 32*escala, 32*escala, paint); //X,Y

```

```

System.out.println("FIN DEL TRAZADO DEL PLANO");

```

```

}

```

```

public void trazar_ruta() throws IOException
{

```

```

    //La ruta sera visualizada en el El bitmap "b" mediante el mismo ImageView usado
    para graficar el plano

```

```

    //CustomImageView imageview = (CustomImageView)
    findViewById(R.id.customImageView1); //asignacion del imageview

```

```

    ImageView imageview = (ImageView) findViewById(R.id.iv1); //asignacion del
    imageview

```

```

    imageview.setImageBitmap(b); //asignacion del fondo a un ImageView

```

```

    //Resolver Laberinto y obtener la Ruta mas corta

```

```

    RutaLaberinto Resolver = new RutaLaberinto();

```

```

    int[][] rutaLab = Resolver.RutaLaberinto(Laberinto2d,InicioLab,FinLab);

```

```

    //System.out.println("RUTA LAB: " + Arrays.deepToString(rutaLab));

```

```

    //System.out.println("FIN Ruta encontrada");

```

```

    //Bucle para DIBUJAR LA RUTA MAS CORTA DEL MAPA//

```

```

    //variables escaladas de coordenadas de la trayectoria

```

```

    int SUPt, IZQt, INFt, DERt = 0;

```

```

    //Variables usadas para graficar la ruta

```

```

    int col_0 = 0 , col_1= 0;

```

```

    DMap:

```

```

    for (int xx=0; xx < rutaLab.length-1 ;xx++)

```

```

    {

```

```

        for (int yy = 0; yy < rutaLab[0].length ; yy++)

```

```

        {

```

```

            if(col_1 == rutaLab.length-1)

```

```

            {

```

```

                break DMap;

```

```

            }

```

```

//System.out.println("Pos A:" + col_1 + "," + col_0);
SUPt = escala*rutaLab[col_1][col_0];
col_0++;

//System.out.println("Pos B:" + col_1 + "," + col_1);
IZQt = escala*rutaLab[col_1][col_0];
col_1++;

INFt=SUPt+escala;
DERt=IZQt+escala;

//System.out.println("supx: " + SUPt);
//System.out.println("izqx: " + IZQt);
//System.out.println("infx: " + INFt);
//System.out.println("derx: " + DERt);

//Dibuja la trayectoria con el color rgb
paint.setColor(Color.rgb(255,30,30)); //asignar color
c.drawRect( IZQt, SUPt, DERt, INFt, paint); //dibuja cada uno de los cuadros

col_0 = 0;
}
}

System.out.println("FIN DEL TRAZADO DE LA RUTA");

}

@Override
public void onClick(View v)
{
    switch(v.getId())
    {
        case R.id.btnPlano:
            try
            {
                cargar_plano();
            }
            catch(Exception e)
            {
                Toast.makeText(MapaActivity.this,"Error: " +
e,Toast.LENGTH_SHORT).show();
            }
            break;

        case R.id.btnTrazar:
            try
            {
                if(InicioLab[0]==FinLab[0] && InicioLab[1]==FinLab[1])
                {

```

```
        Toast.makeText(MapaActivity.this, "El paciente se encuentra en la misma habitacion", Toast.LENGTH_LONG).show();
```

```
    }  
    else  
    {  
        trazar_ruta();  
    }  
} catch (Exception e)  
{  
    Toast.makeText(MapaActivity.this, "Error: " +  
e, Toast.LENGTH_SHORT).show();  
  
    }  
    break;  
}  
}
```

```
private class SearchStartHTTP extends AsyncTask<String,Integer,String>
```

```
{  
    @Override  
    protected void onPreExecute()  
    {  
  
    }  
  
    @Override  
    protected String doInBackground(String... params)  
    {  
  
        try  
        {  
            String url = URLs_Servidor.getSearchStart();  
            String query = "username=" + params[0];  
            System.out.println(url+query);  
            ConectividadHTTP consultar = new ConectividadHTTP();  
            String response = consultar.peticionGET(url+query);  
            return response;  
  
        }  
        catch(Exception e)  
        {  
            return null;  
        }  
  
    }  
  
    @Override  
    protected void onProgressUpdate(Integer... values)  
    {  
        super.onProgressUpdate(values);  
    }  
}
```

```

@Override
protected void onPostExecute(String valores)
{
    StringDecode divisor = new StringDecode();
    String[][] coor_ini = divisor.separaPalabras(valores);
    System.out.println("COORD. INICIO:");
    System.out.println(valores);
    InicioLab[0] = Integer.valueOf(coor_ini[0][0]);
    InicioLab[1] = Integer.valueOf(coor_ini[0][1]);
}

@Override
protected void onCancelled()
{
    super.onCancelled();
}
} //fin de la clase SearchStartHTTP

private class SearchEndHTTP extends AsyncTask<String,Integer,String>
{
    @Override
    protected void onPreExecute()
    {
    }

    @Override
    protected String doInBackground(String... params)
    {
        try
        {
            String url = URLs_Servidor.getSearchEnd();
            String query = "cedula=" + params[0];
            System.out.println(url+query);
            ConectividadHTTP consultar = new ConectividadHTTP();
            String response = consultar.peticionGET(url+query);
            return response;
        }
        catch(Exception e)
        {
            return null;
        }
    }

    @Override
    protected void onProgressUpdate(Integer... values)
    {
        super.onProgressUpdate(values);
    }

    @Override
    protected void onPostExecute(String valores)

```

```

    {
        StringDecode divisor = new StringDecode();
        String[][] coor_fin = divisor.separaPalabras(valores);
        System.out.println("COORD. FIN:");
        System.out.println(valores);
        FinLab[0] = Integer.valueOf(coor_fin[0][0]);
        FinLab[1] = Integer.valueOf(coor_fin[0][1]);
    }

    @Override
    protected void onCancelled()
    {
        super.onCancelled();
    }
} //fin de la clase SearchStartHTTP
}

```

## Anexo 5: Código de la ruta más corta

```
/**
 * Clase RutaLaberinto, encuentra la ruta mas corta entre dos puntos implementando el
 algoritmo de busqueda en anchura BTS
 */

package app.juanorellana.com.hospitaldigital2;

import java.io.IOException;
import java.util.Arrays;

/**
 * Created by Xavier Mauricio and Juan Orellana on 28/06/2016.
 */
public class RutaLaberinto
{

    public int[][] RutaLaberinto(int[][] Laberinto, int[]Partida, int[]Final) throws
    IOException
    {

        int[][] lab=Laberinto;

        int[] inicio = Partida;
        int[] fin = Final;

        int[][] lab2 = new int[lab.length][lab[0].length];
        int[][] niveles = new int[lab.length][lab[0].length];

        int nivelnodo=0;

        for(int i=0;i<lab.length;i++)
        {
            for(int j=0;j<lab[0].length;j++)
            {
                if(lab[i][j]==1)
                {
                    {
                        lab2[i][j]=-1;//pared
                        niveles[i][j]=-1;
                    }
                }
                else
                {
                    {
                        lab2[i][j]= 0;//vacio
                        niveles[i][j]=0;
                    }
                }
            }
        }

        niveles[inicio[0]] [inicio[1]] = 1;
```

```

lab2[inicio[0]] [inicio[1]] = 1;

//System.out.println("lab2"+ Arrays.deepToString(lab2));
//System.out.println("niveles"+ Arrays.deepToString(niveles));

System.out.println("fin 0: "+fin[0]);
System.out.println("fin 1: "+fin[1]);

nivelnodo=1;

int contNiv=0;

while (lab2[fin[0]] [fin[1]] == 0)
{
    contNiv++;

    for(int i=0;i<lab.length;i++)
    {
        for(int j=0;j<lab[0].length;j++)
        {
            if(niveles[i][j]==contNiv)
            {
                try
                {
                    //arriba
                    if (lab2[i - 1][j] == 0)
                    {
                        niveles[i - 1][j] = contNiv + 1;
                        lab2[i - 1][j] = 1;

                    }
                }
                catch (Exception e)
                {
                    System.out.print("error: "+e+" , ");
                }
                try
                {
                    //abajo
                    if (lab2[i + 1][j] == 0)
                    {
                        niveles[i + 1][j] = contNiv + 1;
                        lab2[i + 1][j] = 1;

                    }
                }
                catch (Exception e)
                {
                    System.out.print("error: "+e+" , ");
                }
                try
                {

```

```

        //izquierda
        if (lab2[i][j - 1] == 0)
        {
            niveles[i][j - 1] = contNiv + 1;
            lab2[i][j - 1] = 1;
        }
    }
    catch (Exception e)
    {
        System.out.print("error: "+e+" , ");
    }
    try
    {
        //derecha
        if (lab2[i][j + 1] == 0)
        {
            niveles[i][j + 1] = contNiv + 1;
            lab2[i][j + 1] = 1;
        }
    }
    catch (Exception e)
    {
        System.out.print("error: "+e+" , ");
    }
}
}
}

//System.out.println("matriz niveles"+ Arrays.deepToString(niveles));

int[][] Ruta = new int[contNiv][2];
int filaR = fin[0], columnaR = fin[1];

for(int i=0;i<contNiv;i++)
{

    double[] distancias = new double[4];

    try
    {
        //arriba
        if(lab2[filaR - 1][columnaR] != -1)
        {
            if (niveles[filaR - 1][columnaR] == contNiv - i)
            {
                double dx = Math.pow(((filaR - 1) - inicio[0]), 2);
                double dy = Math.pow(((columnaR) - inicio[1]), 2);
                distancias[0] = Math.sqrt(dx + dy);
            }
        }
    }
}

```

```

        else
        {
            distancias[0]=-1;
        }
    }
    else
    {
        distancias[0]=-1;
    }
}
catch (Exception e)
{
    //System.out.print("error: "+e+" , ");
    distancias[0]=-1;
}
try
{
    //abajo
    if(lab2[filaR + 1][columnaR] != -1)
    {
        if (niveles[filaR + 1][columnaR] == contNiv - i)
        {
            double dx = Math.pow(((filaR + 1) - inicio[0]), 2);
            double dy = Math.pow(((columnaR) - inicio[1]), 2);
            distancias[1] = Math.sqrt(dx + dy);
        }
        else
        {
            distancias[1]=-1;
        }
    }
    else
    {
        distancias[1]=-1;
    }
}
catch (Exception e)
{
    //System.out.print("error: "+e+" , ");
    distancias[1]=-1;
}
try
{
    //izquierda
    if(lab2[filaR][columnaR - 1] != -1)
    {
        if (niveles[filaR][columnaR - 1] == contNiv - i)
        {
            double dx = Math.pow(((filaR) - inicio[0]), 2);
            double dy = Math.pow(((columnaR - 1) - inicio[1]), 2);
            distancias[2] = Math.sqrt(dx + dy);

```

```

    }
    else
    {
        distancias[2]=-1;
    }
}
else
{
    distancias[2]=-1;
}
}
catch (Exception e)
{
    distancias[2]=-1;
}

try
{
    //derecha
    if(lab2[filasR][columnaR + 1] != -1)
    {
        if (niveles[filasR][columnaR + 1] == contNiv - i)
        {
            double dx = Math.pow(((filasR) - inicio[0]), 2);
            double dy = Math.pow(((columnaR + 1) - inicio[1]), 2);
            distancias[3] = Math.sqrt(dx + dy);
        }
        else
        {
            distancias[3]=-1;
        }
    }
    else
    {
        distancias[3]=-1;
    }
}
catch (Exception e)
{
    //System.out.print("error: "+e+" , ");
    distancias[3]=-1;
}

int index = indexOfMin(distancias);
//System.out.println("index: "+index);
//System.out.println(Arrays.toString(distancias));

switch (index)
{
    case 0:

```

```

        filaR = filaR - 1;
        Ruta[i][0] = filaR;
        Ruta[i][1] = columnaR;
        break;

    case 1:
        filaR = filaR + 1;
        Ruta[i][0] = filaR;
        Ruta[i][1] = columnaR;
        break;

    case 2:
        columnaR = columnaR - 1;
        Ruta[i][0] = filaR;
        Ruta[i][1] = columnaR;
        break;

    case 3:
        columnaR = columnaR + 1;
        Ruta[i][0] = filaR;
        Ruta[i][1] = columnaR;
        break;
    }
}

Ruta[contNiv-1][0] = inicio[0];
Ruta[contNiv-1][1] = inicio[1];

return Ruta;
}

private static int indexOfMin(double[] a)
{
    //Método para encontrar la posición del menor número en el vector
    //Trigonométricamente el punto más cercano, cuando existe un caso de decisión
    multiple
    int loc = 0;
    double min = 0;
    int cnt = a.length;
    int j = 0;
    for (int i = 0; i < cnt; i++)
    {
        if(a[i] > 0)
        {
            if(j==0)
            {
                j++;
                min=a[i];
                loc=i;
            }
        }
    }
}

```

```
    }  
    else  
    {  
        if (a[i] < min)  
        {  
            min = a[i];  
            loc = i;  
        }  
    }  
}  
}  
return loc;  
//fin del metodo indexofMin  
}
```

# BIBLIOGRAFÍA

- [1] B. H. Jeong, C. Y. Cheng, V. Prabhu, and B. J. Yu, “An RFID Application Model For Surgery Patient Identification,” in *Advanced Management of Information for Globalized Enterprises*, 2008. AMIGE 2008. IEEE Symposium on, Sept 2008, pp. 1 – 3
- [2] E. V. Cantillo, “Seguridad de los pacientes. Un compromiso de todos para un cuidado de calidad,” *Salud uninorte*, vol. 23, no. 1, 2007, pp. 112 –119.
- [3] A. Aguilar, W. van der Putten, and F. Kirrane, “Positive patient identification using RFID and wireless networks,” in *HISI 11th Annual Conference and Scientific Symposium*. Citeseer, 2006.
- [4] J. C. W. Chan, R. W. Chu, B. W. Y. Young, F. Chan, C. C. Chow, W. C. Pang, C. Chan, S. H. Yeung, P. K. Chow, J. Lau et al., “Use of an electronic barcode system for patient identification during blood transfusion: 3-Year experience in a regional hospital,” *Hong Kong Medical Journal*, vol. 10, no. 3, 2004, pp. 166 – 171.
- [5] Organización Mundial de la Salud, OMS, “Patient Identification,” 2007, [Online], Disponible en: <http://www.who.int/patientsafety/solutions/patientsafety/PS-Solution2.pdf>
- [6] M. Robalino, “Práctica Organizacional: Norma de Capacitación en seguridad de Pacientes,” [Online], 2014, Disponible en: <http://hospitalriobamba.gob.ec/lotaip/Norma6.pdf>
- [7] Hospital Vicente Corral Moscoso, “Norma Técnica de Seguridad del Paciente,” *Coordinación Zonal de Salud 6*, 2015, pp. 30-34.
- [8] J. I. San José, J. Blanco, J. José de Dios, R. Zangróniz y J. Pastor “La identificación por radiofrecuencia (RFID) y sus aplicaciones,” [Online], 2015, Disponible en: <https://www.epc.uclm.es/ep/wp-content/uploads/2015/09/18.pdf>
- [9] Dr. Boris Gala Lopez. Msc., “TECNOLOGIAS INFORMATICAS EN POS DE LA SEGURIDAD DEL PACIENTE: EL CODIGO DE BARRAS EN SANIDAD,” [Online], 2009, Disponible en: [http://www.rcim.sld.cu/revista\\_7/articulo\\_hm/codigbarras.htm](http://www.rcim.sld.cu/revista_7/articulo_hm/codigbarras.htm)

- [10] Sandor Inopia, “Problemas con las tarjetas de banda magnética,” [Online], 2009, Disponible en: <http://www.planetainopia.com/2009-11-03/problemas-con-las-tarjetas-de-banda-magn-tica.html>
- [11] Clockcard Identification, “Tarjetas de Banda Magnetica,” [Online], 2015. Disponible en: <http://www.larconsia.com/banda%20magnetica.asp>
- [12] G. Diodati, A. Gómez, M. Martinez, S. Benitez, D. Luna, F. Gonzalez, B de Quiros, “MHealth: Reconocimiento Óptico de Caracteres para la obtención de la medicación consumida por el paciente,” 2014.
- [13] ABBYY FineReader OCR, “Reconocimiento óptico de caracteres (OCR),” [Online], 2015, Disponible en: <https://www.abbyy.com/es-la/finereader/about-ocr/what-is-ocr/>
- [14] Luis Miguel Blazquez del Toro, “SISTEMAS DE IDENTIFICACION POR RADIOFRECUENCIA,” [Online], 2006. Disponible en: <http://www.it.uc3m.es/~jmb/RFID/rfid.pdf>
- [15] J. P. Puma and M. Huerta and R. Alvizu and R. Clotet, “Mobile Identification: NFC in the Healthcare Sector,” 2012, pp. 39 - 42.
- [16] D. I. Tapia and J. R. Cueli and Ó. García and J. M. Corchado and J. Bajo and A. Saavedra, “Identificación por Radiofrecuencia: Fundamentos y Aplicaciones,” Proceedings de las primeras Jornadas Científicas sobre RFID. Ciudad Real, Spain, 2007, pp. 1 - 5.
- [17] H. Kostinger and M. Gobber and T. Grechenig and B. Tappeiner and W. Schramm, “Developing a NFC based patient identification and ward round system for mobile devices using the android platform,” 2013, pp. 176 - 179.
- [18] A. Marcus and G. Davidzon and D. Law and N. Verma and R. Fletcher and A. Khan and L. Sarmenta, “Using NFC-Enabled Mobile Phones for Public Health in Developing Countries,” 2009, pp. 30 - 35.
- [19] L. Bhadrachalam and S. Chalasani and R. V. Boppana, “Impact of RFID technology on economic order quantity models,” 2009, pp. 327 - 332.

- [20] P. V. Nikitin and K. V. S. Rao, "Performance limitations of passive UHF RFID systems," 2006, pp. 1011 - 1014.
- [21] H. Mora-Mora and V. Gilart-Iglesias and D. Gil and A. Sirvent-Llamas, "A Computational Architecture Based on RFID Sensors for Traceability in Smart Cities," 2015, pp. 13591 - 13626.
- [22] Dipole, "Lectores RFID," [Online], 2014, Disponible en: <http://www.dipolerfid.es/Productos/Lectores-RFID/Default.aspx>
- [23] Dipole, "Antenas RFID," [Online], 2014, Disponible en: <http://www.dipolerfid.es/Productos/Lectores-RFID/Default.aspx>
- [24] José de la Sota Rius, "Tecnología de identificación por radiofrecuencia (RFID): aplicaciones en el ámbito de la salud," [Online], 2012. Disponible en: <http://www.madrid.org/bvirtual/BVCM001808.pdf>
- [25] RFID Point, "Gran hospital español adopta RFID para la gestión de pacientes y bienes," [Online], 2012. Disponible en: <http://www.rfidpoint.com/casos-de-exito/gran-hospital-espanol-adopta-rfid-para-la-gestion-de-pacientes-y-bienes/>
- [26] H. Redrován, J. Andrade, A. Soto, "Diseño e Implementación de un Sistema de Detección y Notificación de Caídas en Personas de la Tercera Edad", Universidad Politécnica Salesiana, 2016.
- [27] H. Zaw, E. Simon y C. Kuanf, "A Monocular View invariant Fall Detection System for the Elderly in Assisted Home Environments", Seventh International Conference on Intelligent Environments, 2011.
- [28] Upverter, "Seedstudio RFR101A1M (RDM630)," [Online], 2015, Disponible en: <https://upverter.com/upn/bfb4ec33820ba990/>
- [29] LinkSprite, "Cottonwood: Long Range UHF RFID Reader UART (ISO180000-6C EPC G2)," [Online], 2016, Disponible en: <http://store.linksprite.com/cottonwood-long-range-uhf-rfid-reader-uart-iso18000-6c-epc-g2/>
- [30] Arduino, "Arduino MEGA", [Online], 2016, Disponible en: <https://www.arduino.cc/en/Main/arduinoBoardMega>

- [31] nRF24L01 Single Chip 2.4GHz Transceiver, Product Especification, [Online], 2007, Disponible en: <http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01>
- [32] R. Sánchez, “ESP8266 con NodeMCU”, [Online], 2016, Disponible en: <http://rubensm.com/esp8266-con-nodemcu/>
- [33] SparkFun Electronics, “WiFi Module - ESP8266”, [Online], 2016, Disponible en: <https://www.sparkfun.com/products/13678>
- [34] Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. "Introduction to algorithms second edition." (2001), 531 – 539.
- [35] A. Vicente, “Bases de Datos y Sistemas de Información, Diseño Conceptual,” [Online], 2012, Disponible en: [http://www.pedeciba.edu.uy/bioinformatica/bdsi/Diseno\\_conceptual.pdf](http://www.pedeciba.edu.uy/bioinformatica/bdsi/Diseno_conceptual.pdf)