



UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
TELECOMUNICACIONES  
DEPARTAMENTO DE SEÑALES Y SISTEMAS



**DESARROLLO DE UN SISTEMA INFORMATIVO PARA CENTROS  
MÉDICOS UTILIZANDO BLUETOOTH LE Y RASPBERRY PI**

JIMÉNEZ JOSELIN  
ORTUÑO JOSÉ

Bárbula, 13 de Julio del 2016



UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
TELECOMUNICACIONES  
DEPARTAMENTO DE SEÑALES Y SISTEMAS



**DESARROLLO DE UN SISTEMA INFORMATIVO PARA CENTROS  
MÉDICOS UTILIZANDO BLUETOOTH LE Y RASPBERRY PI**

TRABAJO ESPECIAL DE GRADO PRESENTADO ANTE LA ILUSTRE UNIVERSIDAD DE  
CARABOBO PARA OPTAR AL TÍTULO DE INGENIERO DE TELECOMUNICACIONES

JIMÉNEZ JOSELIN  
ORTUÑO JOSÉ

Bárbula, 13 de Julio del 2016



UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
TELECOMUNICACIONES  
DEPARTAMENTO DE SEÑALES Y SISTEMAS



### **AVAL DEL TUTOR**

Quien suscribe BILL STEVE TORRES, titular de la cédula de identidad 13.548.024, en mi carácter de TUTOR del Trabajo Especial de Grado titulado:

**DESARROLLO DE UN SISTEMA INFORMATIVO PARA CENTROS MÉDICOS  
UTILIZANDO BLUETOOTH LE Y RASPBERRY PI**

Y presentado por los bachilleres JIMÉNEZ JOSELIN, cédula de identidad 20.162.185, ORTUÑO JOSÉ, cédula de identidad 19.701.593, para optar al Título de Ingeniero de Telecomunicaciones, hago constar que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se le designe

**Firma**

Prof. BILL STEVE TORRES

CI: 13.548.024

TUTOR

Bárbula, 13 de Julio del 2016



UNIVERSIDAD DE CARABOBO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA EN  
TELECOMUNICACIONES  
DEPARTAMENTO DE SEÑALES Y SISTEMAS



### CERTIFICADO DE APROBACIÓN

Los abajo firmantes miembros del jurado asignado para evaluar el trabajo especial de grado titulado “DESARROLLO DE UN SISTEMA INFORMATIVO PARA CENTROS MÉDICOS UTILIZANDO BLUETOOTH LE Y RASPBERRY PI”, realizado por los bachilleres JIMÉNEZ JOSELIN, cédula de identidad 20.162.185, ORTUÑO JOSÉ, cédula de identidad 19.701.593, hacemos constar que hemos revisado y aprobado dicho trabajo.

**Firma**

Prof. BILL STEVE TORRES

TUTOR

**Firma**

Prof. OSMAN AHMAD

JURADO

**Firma**

Prof. APONTE CARLOS

JURADO

Bárbula, 13 de Julio del 2016

# Dedicatoria

*A Dios, que me dio la fortaleza para culminar esta etapa tan importante.*

*A mi madre, por luchar siempre para llevarme hasta donde estoy.*

*A mis tios, por ser como unos padres para mí y apoyarme en todo momento.*

*Al resto de mi familia, el pilar fundamental de mi vida, los que han estado a mi lado en todo momento y me han hecho llegar hasta donde estoy.*

*A mis compañeros, todos y cada uno de los que compartieron estos años conmigo haciéndome crecer como persona.*

**JIMÉNEZ JOSELIN**

*Dedicado a Dios, a mis padres, a mí hermano, y a toda mi familia por darme el amor, la ayuda y el apoyo necesario para poder alcanzar esta meta.*

**ORTUÑO JOSÉ**

# Agradecimientos

A Dios, por permitirme alcanzar esta meta llena de tantas bendiciones y superar todos los obstáculos que se me presentaron.

A mi familia, por apoyarme incondicionalmente en todo momento, formándome en un hogar amoroso y siempre empujándome a ser mejor persona y cumplir mis metas. Gracias a ellos estoy aquí y soy lo que soy.

A mi compañero de tesis, José Miguel Ortuño, la primera persona que conocí en la Universidad y que me acompañó en todo momento en este camino que fue mi carrera Universitaria, la persona que siempre me apoyó y ayudó cuando lo necesitaba y que se convirtió en un hermano para mí.

A mi tutor, Bill Steve Torres, por ser una pieza muy importante en el desarrollo de este proyecto y ayudarnos en cuanto necesitábamos.

A todos mis compañeros de clase, que de alguna u otra forma me impulsaron a alcanzar mis metas, por estar siempre ahí y ser más que compañeros hermanos de vida, con los que conviví tanto tiempo y que me enseñaron lo que significa una amistad verdadera.

A todas las personas que sacaron un poco de su tiempo para ayudarme a culminar esta etapa.

**JIMÉNEZ, JOSELIN**

Le agradezco a Dios por haberme llenado de fuerzas, salud, sabiduría y determinación en este camino que apenas empieza.

Gracias a mi padre José Félix Ortuño, a mi madre María Esther Valera y a mi hermano Luis Gerardo Ortuño por haber estado siempre a mi lado apoyándome, dándome ánimos y fuerzas para lograr esta meta, gracias por creer en mí.

A mis abuelos que están en la tierra como en el cielo, a mi Tía María, Tío Abraham, Tía Nery, Tía Alejandra, Tía Ana, Tío Omar y a todos los integrantes

de mi gran familia, con todo el corazón gracias porque siempre me dieron una o las dos manos cuando lo necesité, y siempre estuvieron pendientes de mi trayectoria a través de estos años.

Le agradezco inmensamente a Joselin Jiménez por su amistad incondicional, paciencia, comprensión, por el apoyo que junto a su familia me han dado durante todos estos años, por haberme aguantado y por haberme acompañado de principio a fin durante toda la carrera, incluyendo la elaboración de este proyecto, Muchísimas Gracias.

A Melissa Mata que junto a Joselin Jiménez compartieron conmigo desde el primer día de la carrera, a Jesús Alvarado quien junto a su familia me abrieron las puertas de su hogar, Verónica Hernández, Ángel Vargas, Rupleidys Nathaly Flores Reyes, Esteffani Parada, Anthonny Tavera, José Malpica, Juan Baldan, Nelson Landinez, Luis Sanchez, Francisco Gonzalez, Alejandro Rojas, Gustavo Rojas, Yudith Ojeda, Edylin Somoza, César Camaripano y a todos los demás compañeros y amigos que dejo de nombrar pero que siempre estuvieron compartiendo conocimientos y momentos inolvidables conmigo a lo largo de estos años.

A mi Alma Mater La Universidad de Carabobo, por regalarme experiencias inolvidables y un ambiente que siempre consideraré mí segundo hogar.

Al profesor Bill Steve por brindarme su apoyo como tutor de este proyecto y amigo, guiándome en la culminación de mi carrera.

A la señora del comedor que nunca me aprendí su nombre pero siempre me tendió su mano sosteniendo un delicioso plato de arroz con pollo, Muchísimas Gracias.

A la señora de ragazzi por tener siempre las empanadas más baratas y grandes del boulevard de la facultad de ingeniería.

Y a cada uno de los profesores que ayudaron en mi formación académica a lo largo de mi vida.

**ORTUÑO, JOSÉ**

# Índice general

<b>Índice de Figuras</b>	<b>IX</b>
<b>Índice de Tablas</b>	<b>XI</b>
<b>Acrónimos</b>	<b>XIII</b>
<b>Resumen</b>	<b>XV</b>
<b>I. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	4
1.2.1. Objetivo general . . . . .	4
1.2.2. Objetivos específicos . . . . .	4
1.3. Alcances . . . . .	5
<b>II. Marco Conceptual</b>	<b>7</b>
2.1. Bluetooth LE . . . . .	7
2.2. Raspberry Pi . . . . .	7
2.3. Android . . . . .	8
2.4. Java . . . . .	8
2.5. Protocolo HTTP . . . . .	9
2.6. Servidores Web . . . . .	9
2.7. Base de datos . . . . .	10
2.8. Python . . . . .	10
2.9. Protocolos de correo electrónico . . . . .	11
2.9.1. Protocolo SMTP . . . . .	11
2.9.2. Protocolo POP . . . . .	11
2.9.3. Protocolo IMAP . . . . .	13
<b>III. Procedimientos de la Investigación</b>	<b>19</b>
3.1. Revisión Teórica . . . . .	19
3.2. Diseño . . . . .	20
3.3. Desarrollo . . . . .	28

3.3.1. Configuración de la Raspberry: . . . . .	29
3.3.2. Instalación del transmisor Bluetooth LE: . . . . .	30
3.3.3. ScanBT: . . . . .	32
3.3.4. Instalación del servidor Web: . . . . .	33
3.3.5. Base de datos: . . . . .	34
3.3.6. Tabla del listado de doctores: . . . . .	34
3.3.7. Tabla de registro de especialidades: . . . . .	35
3.3.8. Tabla de administradores: . . . . .	36
3.3.9. Tablas de registro de pacientes: . . . . .	37
3.3.10. Interfaz administrativa: . . . . .	38
3.3.11. Interfaz del usuario: . . . . .	40
3.3.12. Script de Python: . . . . .	41
3.3.13. Incorporar el sistema a la red: . . . . .	43
3.3.14. Aplicación Android: . . . . .	45
3.3.14.1. MainActivity.java: . . . . .	45
3.3.14.2. SelecciónActivity.java: . . . . .	46
3.3.14.3. Listamedicos.java: . . . . .	46
3.3.14.4. InsertarPaciente.java: . . . . .	47
3.3.14.5. Confirm_Activity.java: . . . . .	47
3.3.15. Envío del código de cita del paciente: . . . . .	48
<b>IV. Análisis, interpretación y presentación de los resultados</b>	<b>51</b>
4.1. Transmisión y recepción de la señal Bluetooth . . . . .	52
4.2. Base de datos . . . . .	56
4.3. Aplicación Android . . . . .	58
4.4. Interfaces Web . . . . .	62
<b>V. Conclusiones y recomendaciones</b>	<b>67</b>
5.1. Conclusiones . . . . .	67
5.2. Recomendaciones . . . . .	68
<b>A. Códigos del sistema</b>	<b>71</b>
<b>Referencias Bibliográficas</b>	<b>73</b>

# Índice de figuras

3.1. Elementos del sistema. . . . .	21
3.2. Estructura de la Aplicación. . . . .	21
3.3. Elementos presentes en la Raspberry Pi. . . . .	22
3.4. Funcionamiento del Correo Electrónico . . . . .	23
3.5. Diagrama de funcionamiento de la Base de datos. . . . .	24
3.6. Diagrama de funcionamiento de la Aplicación. . . . .	24
3.7. Diagrama de funcionamiento del correo electrónico del Centro Médico. . . . .	25
3.8. Diagrama de funcionamiento de la interfaz Web administrativa. . . . .	26
3.9. Diagrama de funcionamiento de la interfaz Web del usuario. . . . .	26
3.10. Diagrama de funcionamiento del Script en Python. . . . .	27
3.11. Diagrama del sistema completo. . . . .	27
3.12. Diseño de la arquitectura física del sistema. . . . .	28
3.13. Integración de la Raspberry Pi con el sistema. . . . .	29
3.14. Configuración de la cantidad de conexiones simultáneas permitidas. . . . .	34
3.15. Estructura Tabla 1. . . . .	35
3.16. Estructura Tabla 2. . . . .	36
3.17. Estructura Tabla 3. . . . .	37
3.18. Estructura Tabla 4. . . . .	38
3.19. Creación de la aplicación Android. . . . .	45
4.1. Escenarios de funcionamiento del sistema completo. . . . .	51
4.2. Plano Dimetel sede Bárbula con mediciones del RSSI en diversos puntos. . . . .	53
4.3. Límites de la zona de cobertura del transmisor. . . . .	54
4.4. Medición del RSSI a 1 m del transmisor. . . . .	54
4.5. Medición del RSSI a 18 m del transmisor. . . . .	55
4.6. Pruebas base de datos. Tabla de especialidades. . . . .	56
4.7. Pruebas base de datos. Tabla de especialistas. . . . .	57
4.8. Pruebas base de datos. Tabla de pacientes. . . . .	57
4.9. Pruebas base de datos. Tabla de administradores. . . . .	57
4.10. Pruebas aplicación en Android Studio. Especialidades. . . . .	58
4.11. Pruebas aplicación en Android Studio. Especialistas. . . . .	58
4.12. Pruebas aplicación en Android Studio. Mensaje previa cita. . . . .	59

---

4.13. Pruebas aplicación en Android Studio. Mensaje orden de llegada. . . . .	59
4.14. Pruebas aplicación en Android Studio. Formulario de registro. . . . .	60
4.15. Pruebas aplicación en Android Studio. Mensaje final. . . . .	60
4.16. Pruebas aplicación en Android Studio. Confirmar asistencia. . . . .	61
4.17. Pruebas aplicación en Android Studio. Paciente confirmado. . . . .	62
4.18. Pruebas interfaz administrativa. Listado de especialidades. . . . .	63
4.19. Pruebas interfaz administrativa. Listado de especialistas. . . . .	63
4.20. Pruebas interfaz administrativa. Tipo de consulta. . . . .	64
4.21. Pruebas interfaz administrativa. Tabla de registros. . . . .	64
4.22. Pruebas interfaz administrativa. Iniciando sesión. . . . .	65
4.23. Pruebas interfaz administrativa. Eliminando. . . . .	65
4.24. Pruebas interfaz usuario. Lista de especialidades. . . . .	66
4.25. Pruebas interfaz usuario. Listado de especialistas. . . . .	66

# Indice de tablas

2.1. Tabla comparativa de modelos de tarjetas Raspberry Pi . . . . .	14
--	----

# Acrónimos

<b>DHTML</b>	<b>D</b> ynamic <b>H</b> yper <b>T</b> ext <b>M</b> arkup <b>L</b> anguage
<b>FTP</b>	<b>F</b> ile <b>T</b> ransfer <b>P</b> rotocol
<b>HTML</b>	<b>H</b> yper <b>T</b> ext <b>M</b> arkup <b>L</b> anguage
<b>HTTP</b>	<b>H</b> yper <b>T</b> ext <b>T</b> ransfer <b>P</b> rotocol
<b>IMAP</b>	<b>I</b> nternet <b>M</b> essage <b>A</b> ccess <b>P</b> rotocol
<b>IP</b>	<b>I</b> nternet <b>P</b> rotocol
<b>JRE</b>	<b>J</b> ava <b>R</b> untime <b>E</b> nvironment
<b>LE</b>	<b>L</b> ow <b>E</b> nergy
<b>PC</b>	<b>P</b> ersonal <b>C</b> omputer
<b>POP</b>	<b>P</b> ost <b>O</b> ffice <b>P</b> rotocol
<b>RSSI</b>	<b>R</b> eceived <b>S</b> ignal <b>S</b> trength <b>I</b> ndication
<b>SD</b>	<b>S</b> ecure <b>D</b> igital
<b>SMTP</b>	<b>S</b> imple <b>M</b> ail <b>T</b> ransfer <b>P</b> rotocol
<b>UC</b>	<b>U</b> niversidad de <b>C</b> arabobo
<b>URL</b>	<b>U</b> niform <b>R</b> esource <b>L</b> ocator
<b>UUID</b>	<b>U</b> niversally <b>U</b> nique <b>I</b> Dentifier

# **DESARROLLO DE UN SISTEMA INFORMATIVO PARA CENTROS MÉDICOS UTILIZANDO BLUETOOTH LE Y RASPBERRY PI**

por

JIMÉNEZ JOSELIN y ORTUÑO JOSÉ

Presentado en el Departamento de Señales y Sistemas  
de la Escuela de Ingeniería en Telecomunicaciones  
el 13 de Julio del 2016 para optar al Título de  
Ingeniero de Telecomunicaciones

## **RESUMEN**

El sistema de información que se implementa actualmente en la mayoría de las clínicas y hospitales de Venezuela para mantener a la comunidad informada sobre cada uno de los especialistas que se encuentran activos dentro del centro médico presenta deficiencias en cuanto a sus altos costos, su carencia de información y sus requerimientos de tiempo y personal para hacer que los mismos se encuentren actualizados día a día. El desarrollo de un sistema informativo que integre las tecnologías Bluetooth LE, Raspberry Pi y las aplicaciones Android establece una manera eficaz, rápida e inmediata de mantener dicha información al día y a disposición de los usuarios de la institución, así como también una disminución en los gastos monetarios que conllevan los actuales sistemas de información utilizados. Para lograr

el funcionamiento se elaborará una aplicación para dispositivos Android y en ella se podrá seleccionar la búsqueda deseada, eligiendo a partir de una lista el tipo de especialista requerido; dicha búsqueda dará como resultado la información de los doctores disponibles en la especialidad escogida, en la que se incluyen nombre del doctor, número de consultorio y la información referente a las consultas y disponibilidad. La aplicación deberá reconocer la señal proveniente del dispositivo elaborado con la tarjeta de desarrollo Raspberry Pi para poder realizar la confirmación de las citas, además dicha Raspberry contendrá un servidor web y la base de datos con la información de todos los especialistas disponibles en el centro médico, que podrá ser editada y visualizada mediante interfaces web previamente establecidas.

Palabras Claves: Bluetooth, Android, Raspberry

Tutor: BILL STEVE TORRES

Profesor del Departamento de Señales y Sistemas

Escuela de Telecomunicaciones. Facultad de Ingeniería

# Capítulo I

## Introducción

### 1.1. Motivación

Las personas continuamente presentan una necesidad de sentirse informados, tal como lo indica Juan José Calva González en su libro titulado “Las necesidades de Información” [1] en el que hace referencia al hecho de que el ser humano además de poseer necesidades primarias básicas como el hambre, la sed y el sueño, posee necesidades de información que se presentan de forma recurrente en sus actividades, y que tal motivo los lleva a acudir a algún elemento informativo que le permita satisfacer esa necesidad, y al mismo tiempo indica pautas que permiten identificar los elementos necesarios para ayudar a satisfacer estas necesidades en las comunidades, lo cual permite afirmar que los usuarios de los centros médicos poseen estas mismas necesidades informativas, y que en el momento que llegan a estos la primera iniciativa que tienen es dirigirse a un punto donde le ofrezcan información, el cual debe contar con un sistema que sea capaz de brindarla de manera rápida y concreta, que sea relevante para cada una de las personas que acuden a este establecimiento.

Con la rápida evolución de las tecnologías y la automatización de los sistemas de información, se van abriendo nuevas fronteras a la hora de crear soluciones a problemas que pueden ser tanto sencillos como muy complicados; es por ello que

cada día surgen nuevas herramientas que hacen más fácil la vida de las personas y permiten definir una trayectoria hacia inventos que definan la siguiente evolución tecnológica.

La gran demanda de las herramientas tecnológicas lleva a la humanidad a plantear una infinidad de sistemas automatizados de fácil acceso, que permitan solventar rápidamente las necesidades de los usuarios, teniendo como requisito fundamental un sistema de fácil manejo y modificación para que puedan ser actualizados con eficacia y rapidez.

El actual sistema de información que se implementa en la mayoría de las clínicas y hospitales de Venezuela para mantener a la comunidad informada sobre cada uno de los especialistas que se encuentran activos dentro del centro médico se basa en una cartelera donde se coloca el listado de los mismos; dicha cartelera requiere de una inversión de material para realizar cada una de las actualizaciones que sean requeridas, material el cual muchas veces genera elevados costos, y que además no permite brindar información suficiente para los pacientes. También se requiere de un personal encargado del mantenimiento y la actualización de la cartelera, que necesita constante atención para poder mantener al día la información que allí se muestra, lo que se hace especialmente difícil en centros médicos donde hay gran cantidad de especialistas que día a día pueden necesitar cambios en su información, por lo que muchas veces por carencia o descuido del personal se ve comprometido el servicio prestado por ésta.

Debido a la deficiencia en estos casos de la cartelera informativa, muchas veces el personal del centro médico se ve en la necesidad de dejar sus labores para prestarle atención a las personas que llegan en busca de información, lo cual representa un problema para la institución ya que cuentan con un personal establecido para áreas específicas, pero que a fin de cuentas terminan realizando tareas que no les competen.

Existen tecnologías, como el Bluetooth, que hacen posible la transferencia de información entre dispositivos y que hoy en día han logrado grandes avances en cuanto a su funcionamiento y ahorro de energía [2].

Aunque los dispositivos con conexión Bluetooth han estado en el mercado desde hace varios años, actualmente se ha desarrollado una nueva tecnología llamada Bluetooth LE (Bluetooth v4.0), que viene dispuesta en los últimos smartphones del mercado, y que viene a abrir las puertas a un entorno mucho más inteligente con multitud de dispositivos conectados [2].

Bluetooth LE se fundamenta en la reducción del consumo y, por tanto, en minimizar la potencia de transmisión de la señal utilizada y el aumento del radio de cobertura [2].

La importante reducción del consumo de energía y la extensión de la autonomía también son importantes para el desarrollo de sensores inteligentes, debido a que estos podrían estar encendidos durante meses sin necesidad de reemplazar la batería. Un claro ejemplo de este caso sería el proyecto de fin de carrera de Daniel Gómez Fernández [3], en el que hace uso de esta nueva tecnología de Bluetooth LE para la creación de sensores que permitan a las personas con discapacidad visual monitorear el paso de vehículos al cruzar la calle mediante una aplicación para dispositivos móviles que reconoce cuando el usuario se encuentra en el radio aceptado de cobertura y comienza la comunicación para asistirlo en esta tarea, lo que representa un enfoque similar al planteado en el presente proyecto y del cual se adoptará el uso de la tecnología Bluetooth LE para la detección de dispositivos. Estos avances están siendo implementados también en diversas áreas del comercio e incluso en los hogares bajo un concepto llamado “Internet de las cosas” [4] (o también llamado por su acrónimo en inglés “IoT”) lo que permite facilitar las tareas cotidianas de las personas, y el control de los dispositivos que se encuentran en los alrededores.

Gracias a la creación de esta tecnología se llegaron a establecer los dispositivos denominados “beacons”. Este término, que de a poco ha ido ganando gran popularidad, hace referencia a un pequeño dispositivo que emite señales de onda corta por medio de la tecnología Bluetooth LE [5].

El sistema de localización de especialistas tomará como base el principio de funcionamiento de estas nuevas tendencias tecnológicas y estará compuesto por una

aplicación para dispositivos Android capaz de permitirle a los usuarios mantenerse informados sobre los especialistas que laboran en el centro médico.

Considerando los problemas de costos, el personal necesario para actualizar las carteleras, y además el hecho de que la información allí mostrada no es suficiente para mantener a los pacientes informados, surge la idea de diseñar un sistema informativo para centros médicos utilizando Bluetooth LE y Raspberry Pi, el cual con pocos requerimientos de administración y mantenimiento se convertirá en una herramienta que cuente con la capacidad necesaria para solucionar los problemas anteriormente mencionados.

## **1.2. Objetivos**

### **1.2.1. Objetivo general**

Desarrollar un sistema informativo para centros médicos utilizando Bluetooth LE y Raspberry Pi.

### **1.2.2. Objetivos específicos**

- Integrar la tarjeta de desarrollo Raspberry Pi 2 modelo B con la tecnología Bluetooth LE para la elaboración del transmisor.
- Desarrollar una aplicación para dispositivos Android capaz de trabajar con la tecnología Bluetooth LE utilizando la herramienta de programación Android Studio.
- Diseñar la estructura de la base de datos a la cual se comunicará la aplicación Android.
- Implementar un sistema de control de acceso a la base de datos integrando las tecnologías Bluetooth LE, Android y la tarjeta de desarrollo Raspberry Pi 2 modelo B.

### **1.3. Alcances**

El sistema transmisor estará conformado por una tarjeta de desarrollo Raspberry Pi 2 modelo B a la que se le configurará un adaptador de Bluetooth versión 4.0, también conocido como Bluetooth LE, y una aplicación para dispositivos Android en la que los usuarios podrán obtener información de los especialistas que laboran en la institución e incorporarse en la lista de espera de los doctores que trabajen con esa modalidad; dicha aplicación utilizará el área de cobertura del transmisor y el UUID para permitirle al usuario confirmar su asistencia con un código previamente asignado al momento de solicitar la consulta. La aplicación elaborada en la herramienta de programación Android Studio junto con la tecnología de Bluetooth LE solo permitirá la confirmación al comprobar que el usuario se encuentra dentro del área de cobertura del transmisor. Se elaborará también una interfaz gráfica en la que el personal designado por la institución modificará y mantendrá al día la información almacenada en la base de datos.

## Capítulo II

# Marco Conceptual

### 2.1. Bluetooth LE

También llamado en algunas ocasiones “Bluetooth Smart” (Bluetooth Inteligente) es un derivado del clásico Bluetooth que consume solo una fracción de la potencia y que fue introducido en el Bluetooth versión 4.0. Es considerado la manera más fácil de diseñar algo que puede comunicarse con cualquier otro dispositivo moderno de cualquier plataforma (iOS, Android, etc).[6] Se puede decir que existen dos roles fundamentales de los dispositivos que trabajan con esta tecnología: Dispositivos centrales, como teléfonos o tablets en los que se requiere mucha más potencia de procesamiento, y dispositivos periféricos, que son pequeños y de baja potencia, capaces de conectarse con los dispositivos centrales, como monitores cardíacos, etiquetas de proximidad, etc.[7]

### 2.2. Raspberry Pi

Las Raspberry Pi son una serie de computadoras del tamaño de una tarjeta de crédito desarrolladas en Reino Unido por la fundación “Raspberry Pi”. Pueden ejecutarse con una gran variedad de Sistemas Operativos y ser conectadas a un televisor, un teclado o un mouse para ser utilizadas de manera muy sencilla. Pueden

usarse para muchas de las funciones que se realizan con una PC de escritorio, como para manejar procesadores de texto, hojas de cálculo, juegos, etc, pero requiriendo de una tarjeta SD para almacenamiento.[8]

### **2.3. Android**

Es un sistema operativo basado en Linux que permite desarrollar aplicaciones en el entorno de trabajo de Java. Lo que diferencia Android de otros sistemas operativos, es que cualquier persona que sepa programar puede crear nuevas aplicaciones, widgets, o incluso modificar el propio sistema operativo haciéndolo compatible con una gran cantidad de dispositivos móviles como teléfonos, tablets, net-books, reproductores de música e incluso algunos computadores de escritorio. Dado que está creado bajo la denominación de software libre, llega a ser muy fácil comenzar a programar en esta plataforma, posicionando a ésta como una de las plataformas que ha evolucionado más rápido en los últimos años.[9][10]

### **2.4. Java**

Java es un lenguaje de programación orientado a objetos desarrollado y presentado por James Gosling y Bill Joy en 1995, cuya sintaxis es muy similar a la de C y C++. Es un lenguaje de programación muy popular debido a que permite programar una aplicación capaz de ejecutarse en distintas plataformas y distintos sistemas operativos como Windows, Mac OS, Linux o Solaris, tal como su eslogan lo dice “Write Now, Run Anywhere” (Programar una vez y ejecutar en cualquier sitio). Los programas Java se compilan a un lenguaje intermedio denominado Bytecode. Este código es interpretado por la máquina virtual de Java del entorno de ejecución (JRE) y así se consigue la portabilidad en distintas plataformas. [11][12]

## 2.5. Protocolo HTTP

Desde un punto de vista más técnico, se necesita un protocolo que permita enviar y traer información en HTML desde un lugar a otro en esta gigantesca red que es la Web.

El protocolo HTTP tiene varias características distintivas que lo han hecho muy perdurable. HTTP es un protocolo de transmisión entre el cliente, que puede ser un browser, un agente, o cualquier herramienta, y el servidor, que es el que almacena o crea recursos como archivos HTML, imágenes, etc. Entre ellos puede haber varios intermediarios, como proxies, gateways y túneles. A través de instrucciones simples, pero poderosas, el cliente indica al servidor qué acciones realizar para recibir o entregar datos. [13]

## 2.6. Servidores Web

Empezando por la funcionalidad básica de un servidor web, su cometido principal es proveer de contenido estático a un cliente que ha realizado una petición a través de un navegador. Para ello, carga un archivo y lo sirve a través de la Red al navegador del solicitante. Este es el funcionamiento habitual cuando lo que se implementa en el servidor es una aplicación web estática (HTML) o dinámica (DHTML), es decir, que el servidor se convierte en un instrumento que proporciona un lugar para guardar y administrar los recursos HTML, que pueden ser accesibles por los usuarios de la Red a través de navegadores. Para que el servidor pueda entender la petición del cliente, éste tiene que realizar una petición “formal”, es decir, la petición tiene que constar de unos elementos concretos y especificados en un orden determinado. Las direcciones de las peticiones suelen ser de tipo URL (Localizador Uniforme de Recurso), que contiene una dirección, la referencia a un cierto protocolo (HTTP, FTP, etc.) y la descripción de un recurso concreto en forma de ruta al objeto que queremos descargar. Opcionalmente podemos incluir un campo adicional (puerto) indicando el número de puerto por el que el servidor está escuchando las peticiones del cliente.[14]

## 2.7. Base de datos

Una base de datos se refiere a un conjunto estructurado de datos relacionados por un determinado aspecto, que se encuentran organizados de forma que sea posible acceder a ellos simultáneamente. Son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron, como por ejemplo, una base de dato de una universidad puede contener archivos de estudiantes o archivos de nóminas, entre otros, almacenando esta información del mismo modo que lo hace la carpeta de un archivador.

Es una herramienta que representa la mejor manera de almacenar, organizar, recuperar, comunicar y administrar información de un modo mucho más eficaz y confiable.[15][16]

## 2.8. Python

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para el desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas.

El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python [17], y puede distribuirse libremente. El mismo sitio contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional.

El intérprete de Python puede extenderse fácilmente con nuevas funcionalidades y tipos de datos implementados en C o C++ (u otros lenguajes accesibles desde C). Python también puede usarse como un lenguaje de extensiones para aplicaciones personalizables [18]

## **2.9. Protocolos de correo electrónico**

El correo electrónico es uno de los servicios más utilizado de Internet, por ello se han desarrollado una serie de protocolos basados en TCP/IP que permiten una fácil administración del enrutamiento del correo electrónico a través de la red como son SMTP, IMAP y POP.

### **2.9.1. Protocolo SMTP**

El protocolo SMTP es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto. Éste es un protocolo que funciona en línea, encapsulado en una trama TCP/IP.

El correo se envía directamente al servidor de correo del destinatario. Funciona con comandos de textos enviados al servidor SMTP (al puerto 25 de manera predeterminada). A cada comando enviado por el cliente (validado por la cadena de caracteres ASCII CR/LF, que equivale a presionar la tecla Enter) le sigue una respuesta del servidor SMTP compuesta por un número y un mensaje descriptivo.

### **2.9.2. Protocolo POP**

El protocolo POP es un modelo Cliente/Servidor que se basa en tomar los correos de un buzón remoto y moverlos a la máquina local.

Existen dos versiones principales de este protocolo, POP2 y POP3, a los que se le asignan los puertos 109 y 110 respectivamente, y que funcionan utilizando comandos de texto diferentes.

Los comandos principales de POP2 son:

- HELLO: Utiliza la dirección IP del remitente.
- FOLDER: Nombre de la bandeja de entrada a consultar.
- READ: Número del mensaje a leer.
- RETRIEVE: Número del mensaje a mover a la máquina del cliente.
- SAVE: Número del mensaje a guardar.
- DELETE: Número del mensaje a eliminar.
- QUIT: Salida del servidor POP2.

Pero en general el protocolo más utilizado es POP3, debido a que las versiones anteriores son antiguas y han quedado obsoletas; al hacer referencia a POP comúnmente se dice POP3 por ser la versión usada actualmente.

Los comandos utilizados en el protocolo POP3 son:

- USER usuario: Nombre de usuario para iniciar sesión
- PASS contraseña: Contraseña del usuario

Una vez ingresados estos dos comandos se pasa al estado de “transacción”. La casilla de correo del usuario queda bloqueada durante toda la conexión, es decir, no puede ser accedida simultáneamente desde otro cliente.

- LIST mensaje: Muestra la lista de mensajes y su estado.
- RETR mensaje: Mueve el mensaje a la máquina del cliente.
- DELE mensaje: Marca el mensaje para ser eliminado.
- QUIT: Una vez ingresado el comando QUIT se pasa al estado de “actualización”, se eliminan los mensajes marcados para eliminar y luego se cierra la conexión TCP.

El servidor responde con sólo dos tipos de mensajes:

- +OK
- -ERR

### 2.9.3. Protocolo IMAP

El protocolo IMAP accede a los correos de un buzón remoto sin moverlos hacia la máquina local a diferencia del POP3.

Este protocolo da la posibilidad de acceder al correo desde cualquier máquina con la posibilidad de seleccionarlos por distintos atributos, por ejemplo, el remitente.

IMAP también responde al modelo Cliente/Servidor iniciando una conexión TCP al puerto 143. Algunas ventajas sobre POP3 son:

- Las transacciones IMAP pueden durar mucho más tiempo.
- El servidor guarda información del estado de los correos (si fueron leídos o no, si fueron guardados en una carpeta, etc.)
- Se pueden definir distintas carpetas para acceder a distintos buzones.
- Se puede conectar más de un cliente al mismo buzón.
- Posee buscadores que se ejecutan en el servidor.

Sin embargo posee ciertas desventajas:

- Es más complejo de implementar que POP3.
- El servidor debe ser más potente para atender a todos los usuarios.
- Consume más recursos de CPU, memoria, etc.

Al estudiar las características de los distintos modelos de las tarjetas Raspberry Pi, se tomaron como referencia los catálogos proporcionados por la empresa manufacturera de estas tarjetas de desarrollo [19], estudio que llevó a comparar las tarjetas principalmente utilizadas:

Características	1ModeloA+	1ModeloB+	2ModeloB
Procesador	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836
CPU	ARM 1176JZF-S	ARM 1176JZF-S	ARM Cortex A7
RAM	256 MB	512 MB	1 GB
Puertos USB	1	4	4

**Tabla 2.1:** Tabla comparativa de modelos de tarjetas Raspberry Pi

Las características de estas tarjetas llevaron a la decisión de trabajar con la tarjeta Raspberry Pi 2 Modelo B, ya que contiene todas las virtudes de las anteriores versiones mejoradas. Entre las características más resaltantes de esta tarjeta se tienen [19]:

- Procesador ARM Cortex A7 que trabaja a 900 MHz y es muy eficiente y con gran potencia, lo que permite que sea compatible con varias distribuciones de GNU/LINUX e, incluso, Windows 10.
- Memoria RAM de 1 GHz.
- Entrada HDMI que permite reproducir contenidos en alta definición.
- Conector Jack 3.5 que permite conectar altavoces y auriculares.
- 4 puertos USB 2.0 para poder conectar gran cantidad de elementos.
- Un puerto Ethernet.
- Interfaz de pantalla.
- Interfaz de cámara.
- Ranura para memoria MicroSD.

Entre las distribuciones disponibles para utilizar con esta tarjeta se encuentran:

1. **Raspbian:** Es un sistema operativo gratis basado en Debian, y optimizado para ser utilizado con las tarjetas Raspberry Pi, haciéndolo una de las opciones más conocidas. Además de poseer el conjunto de programas que permite

la ejecución de esta tarjeta, tiene aproximadamente 35000 paquetes y software pre-compilado y unificado en un formato de fácil instalación [20]. Incluye herramientas de desarrollo tales como IDLE para Python y Scratch para programar videojuegos, entre otras.

2. **Pidora:** Es una distribución de Linux para las computadoras Raspberry Pi. Contiene paquetes de software del sistema Fedora compilados específicamente para la arquitectura del procesador ARMv6 que es utilizado en las Raspberry Pi, que además han sido especialmente escritos o modificados para el uso con estas tarjetas [21].
3. **RISC:** Es uno de los pocos sistemas operativos no basados en Linux que existen para la Raspberry Pi. Fue desarrollado por Acorn Computers (los creadores de ARM) específicamente para los procesadores ARM, y es considerado por muchos como un sistema estable y de fácil entendimiento [22]

Al observar las opciones de las distintas distribuciones disponibles se decidió trabajar con Raspbian debido a que posee herramientas y paquetes que facilitan su utilización, además de ser una de las distribuciones más utilizadas al trabajar con las tarjetas Raspberry Pi.

En el estudio de la tecnología Bluetooth se consultaron las especificaciones técnicas de cada una de las versiones existentes, presentes en la página oficial de dicha tecnología [6] obteniendo como resultado la decisión de trabajar con la versión 4.0, también conocida como Bluetooth LE, esto debido a que presenta las características de las versiones anteriores mejoradas, con el agregado de un ahorro de energía considerable debido a la simplicidad de su arquitectura. Entre las ventajas principales de esta versión con respecto a las anteriores se tiene:

- Interoperabilidad entre distintas marcas y dispositivos.
- Significante ahorro de energía que permite el funcionamiento durante meses con una batería de reloj.
- Utilización de arquitectura de muy bajo costo.

- Alcance de la señal de hasta 100m.
- Tasa de transferencia de datos de hasta 1 Mbps.
- Posee una gran seguridad ya que cuenta con un cifrado AES-128.
- Hace posible el Broadcasting de una trama de bytes sin necesidad de existir apareamiento con los dispositivos remotos.

En cuanto al estudio de los entornos de desarrollo de aplicaciones Android se estableció el lenguaje Java como lenguaje nativo de estas herramientas y se buscó el programa a utilizar para la creación de la aplicación.

Entre los entornos de desarrollo consultados se tienen:

1. **Eclipse:** Entorno creado por IBM en Noviembre de 2001. Es un Software Libre muy utilizado que puede expandirse mediante plugins y productos encontrados en su página [23] para añadir mas funcionalidades a medida que se necesiten.
2. **BlueJ:** Es un ambiente de desarrollo que permite realizar aplicaciones en Java de manera fácil y rápida, con las ventajas de que es simple, interactivo, innovador, continuamente mejorado y puede ejecutarse en distintas plataformas [24].
3. **Android Studio:** Es el entorno de desarrollo oficial para elaborar aplicaciones para Android basado en el software IntelliJ IDEA, con muchas ventajas como por ejemplo que posee un sencillo editor de la apariencia de la aplicación, plantillas para ayudar en la creación y un sencillo sistema de construcción basado en Gradle [25]

Debido a que el entorno Android Studio fue lanzado por Google y declarado como el entorno oficial para realizar aplicaciones, que venía a sustituir a Eclipse en esa posición, se utilizó este programa para la elaboración de la aplicación.

Finalmente se consultaron tutoriales y manuales para desarrolladores de Aplicaciones Android, tomando como principal referencia la página oficial de Android Developers [25], y se pudieron recopilar las librerías e instrucciones necesarias para la elaboración de la Aplicación.

## Capítulo III

# Procedimientos de la Investigación

### 3.1. Revisión Teórica

Esta etapa de la investigación se enfocó en el estudio de diferentes herramientas tecnológicas como son el Bluetooth LE, la tarjeta Raspberry Pi y las aplicaciones para dispositivos Android, así como también en buscar la mejor forma de centralizar todas estas en una sola.

En el estudio de las características de la tarjeta Raspberry Pi se consultaron los catálogos proporcionados por la empresa manufacturera de estas tarjetas [19], donde se tomó la decisión de trabajar con la tarjeta Raspberry Pi 2 modelo B, ya que contiene todas las virtudes de las versiones anteriores mejoradas.

Luego de haber escogido el modelo de la tarjeta de desarrollo se analizaron los distintos sistemas operativos que pudieran ser utilizados en la Raspberry donde prevalecieron distribuciones como Pidora, Risc y Raspbian; obteniéndose como resultado la selección del sistema operativo Raspbian debido a su amplia cantidad de herramientas y paquetes, además de ocupar el primer puesto entre los sistemas operativos más utilizados en tarjetas Raspberry Pi.

Al estudiar la tecnología Bluetooth LE se consultaron las especificaciones técnicas presentes en su página oficial, conociéndose así los protocolos y características

que la definen como una tecnología innovadora y de bajo consumo en comparación con versiones anteriores.

A su vez también se estudiaron las maneras de elaborar una aplicación Android capaz de funcionar como interfaz entre los usuarios y el sistema planteado, para lo cual existen diversas herramientas de desarrollo como Android Studio, que hacen posible su elaboración.

En cuanto a la integración de las tecnologías se consultaron los diferentes métodos y protocolos necesarios para lograr enlazar la tarjeta Raspberry Pi con la tecnología Bluetooth LE y la aplicación para dispositivos Android, estudio donde se destaca el protocolo http bajo protocolos tcp/ip para hacer uso de la base de datos y la instalación de un dongle Bluetooth LE para realizar el broadcasting de la señal de Bluetooth.

Ya habiendo obtenido los conocimientos necesarios para la elaboración del sistema se procedió a realizar el diseño de la plataforma como se muestra a continuación.

### **3.2. Diseño**

Una vez comprendido el funcionamiento de todos los elementos necesarios se procedió a diseñar el funcionamiento del sistema, así como también la estructura que llevaría la base de datos que daría lugar a la información a la que posteriormente accederán los usuarios.

El sistema se plantea de la siguiente forma:

Un elemento central definido por la tarjeta Raspberry Pi, una aplicación para dispositivos móviles y un correo electrónico perteneciente a la institución, y destinado al uso administrativo del sistema, como se muestra en la figura 3.1.

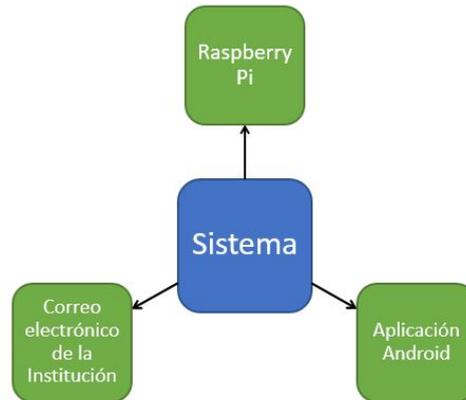


Figura 3.1: Elementos del sistema.

La aplicación android esta definida como la herramienta que logrará mejorar considerablemente la interacción entre los pacientes y la institución. En la figura 3.2 se muestra la estructura de dicha aplicación donde se plantea: un diseño compuesto por cada uno de los elementos de la interfaz gráfica (iconos, listas, botones, menús, etc), y un desarrollo compuesto por todos los elementos que hacen posible el funcionamiento fluido de la aplicación (actividades, clases y librerías).



Figura 3.2: Estructura de la Aplicación.

El siguiente elemento del sistema es la tarjeta Raspberry Pi, que representa el elemento más importante, y se configura bajo la estructura mostrada en la figura 3.3 con el objetivo de dar lugar a la integración de las distintas tecnologías presentes en dicho sistema:

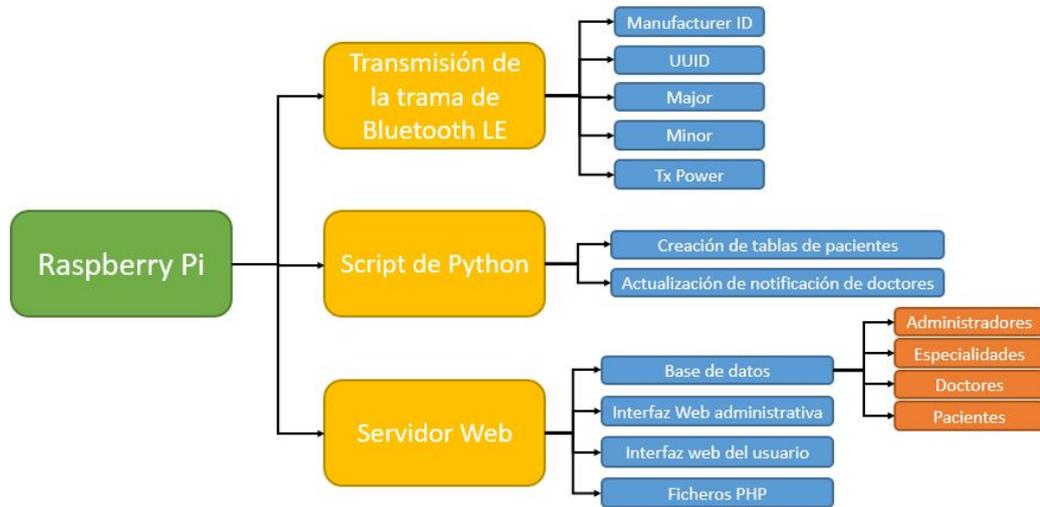


Figura 3.3: Elementos presentes en la Raspberry Pi.

Esta estructura comprende:

- Un transmisor de una señal de Bluetooth LE, como elemento que rige el control de acceso a las listas definitivas de pacientes a atender por cada uno de los doctores para hacer la confirmación de asistencia de los pacientes.
- Un script en lenguaje Python, que se ejecuta en segundo plano dentro de los procesos de la tarjeta Raspberry Pi y que se encarga de la administración y/o actualización de la base de datos.
- Un servidor Web que sirve a través de la red la base de datos, la interfaz web administrativa, la interfaz web del usuario y los ficheros en lenguaje PHP requeridos por la aplicación Android para realizar las consultas a la base de datos.

Por último se tiene el correo electrónico de la institución, este se define como el elemento mediante el cual se recibirá información referente a los doctores y se enviarán los códigos de confirmación de asistencia a los pacientes.

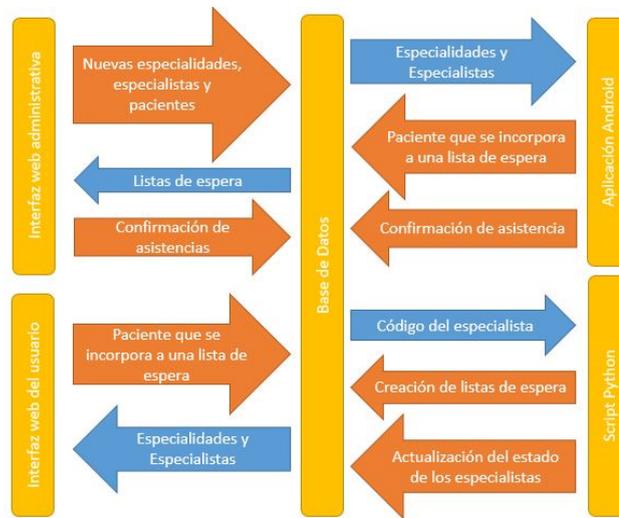
El uso de este correo queda definido por el siguiente esquema:



**Figura 3.4:** Funcionamiento del Correo Electrónico

Ya habiendo definido la estructura del sistema se diseñó el funcionamiento de cada uno de los elementos que lo integran de la siguiente manera:

Uno de los elementos primordiales que se diseñó fue la base de datos, la cual contendrá toda la información requerida por los demás elementos para su funcionamiento, lo que la hace uno de los eslabones más importantes en el sistema. Almacenará la información sobre las especialidades y los especialistas del centro médico para ponerla a disposición de la aplicación Android y la interfaz web del usuario, almacenará los códigos de cada especialista utilizado por el script de Python para crear las listas de espera y leer los correos con las actualizaciones de estado de cada uno de ellos, además recibirá toda la información sobre los pacientes que se incorporan a dichas listas, tal como se muestra en el diagrama de la figura 3.5



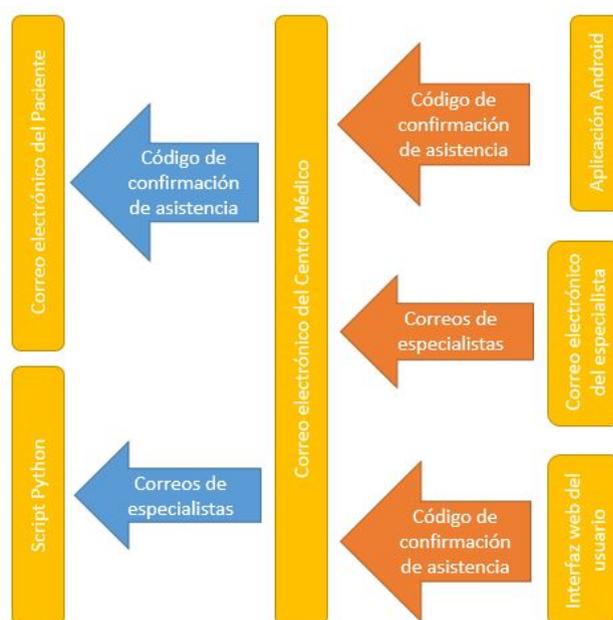
**Figura 3.5:** Diagrama de funcionamiento de la Base de datos.

El siguiente elemento cuyo funcionamiento fue diseñado es la aplicación Android, la cual interactúa con los demás elementos como se muestra en la figura 3.6. Obteniendo la información de las especialidades y especialistas de la base de datos para mostrarla a través de su interfaz a los usuarios y luego llenar las listas de espera con los pacientes que deseen una consulta médica; así mismo la aplicación le envía a través del correo electrónico de la institución el código de confirmación al paciente, el cual dependiendo de la trama publicitaria recibida del transmisor Bluetooth LE, permitirá efectuar la confirmación de la asistencia a dicha consulta.



**Figura 3.6:** Diagrama de funcionamiento de la Aplicación.

El siguiente elemento creado fue el correo electrónico, que se encarga de tomar el código de confirmación de cita generado por la aplicación o por la interfaz web del usuario en el momento del registro de un nuevo paciente y lo envía al correo electrónico de este, almacenado en el momento del registro. También está encargado de recibir los correos electrónicos enviados por los especialistas para actualizar su notificación haciendo uso del Script de Python, como se observa en la figura 3.7.



**Figura 3.7:** Diagrama de funcionamiento del correo electrónico del Centro Médico.

El siguiente elemento definido fue la interfaz web administrativa apreciado en la figura 3.8, que es la encargada de registrar en la base de datos las especialidades y especialistas, así como también los pacientes que solicitan su incorporación a una lista de espera o su confirmación de asistencia por no poder hacerlo él mismo. Para llevar a cabo esta confirmación se debe escribir el código de cita del paciente, posteriormente la interfaz toma este código y lo compara con los códigos de todos los pacientes de todas las listas de espera y al conseguir el que coincida, realiza la confirmación.



**Figura 3.8:** Diagrama de funcionamiento de la interfaz Web administrativa.

Luego se tiene la interfaz web del usuario, que es una interfaz que permite obtener la información de especialidades y especialistas. Esta interfaz recibe esta información de la base de datos y la muestra a los usuarios, y al momento de hacer un nuevo registro de algún paciente, envía esta información a la base de datos al mismo tiempo que envía el código de cita generado al correo de la institución para luego ser enviado al correo del paciente.



**Figura 3.9:** Diagrama de funcionamiento de la interfaz Web del usuario.

Por último se tiene el Script de Python, que recibe los códigos de los especialistas

registrados de la base de datos y crea las listas de espera a partir de estos códigos, así como también accede al correo electrónico de la institución y extrae el cuerpo del mensaje de cada especialista para actualizar la información mostrada a los usuarios.



Figura 3.10: Diagrama de funcionamiento del Script en Python.

Por último, uniendo todos los elementos del sistema mencionados anteriormente se tiene un diagrama del sistema completo, mostrado en la figura 3.11.

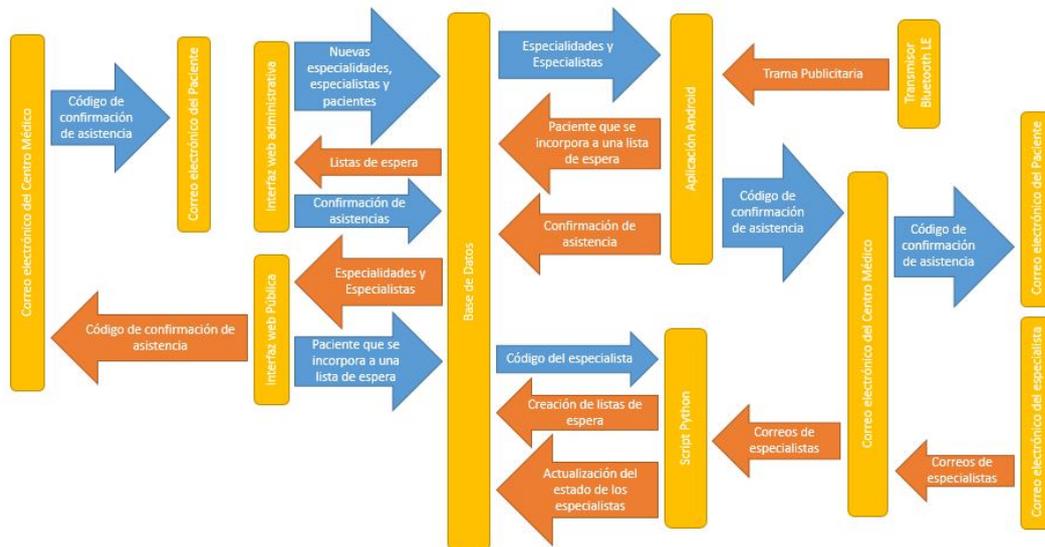


Figura 3.11: Diagrama del sistema completo.

Ya habiendo definido el funcionamiento del sistema, se plantea una arquitectura física para este donde se recomienda ubicar la tarjeta Raspberry Pi en un lugar abierto, como por ejemplo una sala de espera, de modo que se cree una zona de confirmación que excluya a los usuarios que se encuentren fuera de ella, como se muestra en el ejemplo de la figura 3.12:



**Figura 3.12:** Diseño de la arquitectura física del sistema.

Habiendo ya diseñado el funcionamiento y la arquitectura física del sistema como se muestra en la figura 3.12, se procede con la fase de desarrollo donde se presentan cada uno de los elementos desarrollados y configurados para lograr la completa operatividad del sistema.

### 3.3. Desarrollo

Al haber finalizado la etapa de diseño del sistema se procede con la etapa de desarrollo donde se configuró y elaboró los elementos necesarios para hacer funcionar todas las fases del sistema.

El sistema diseñado anteriormente se encuentra centralizado en una tarjeta Raspberry pi, por lo que fue necesaria su integración con cada una de las tecnologías presentes en el mismo, como se muestra en la figura 3.13

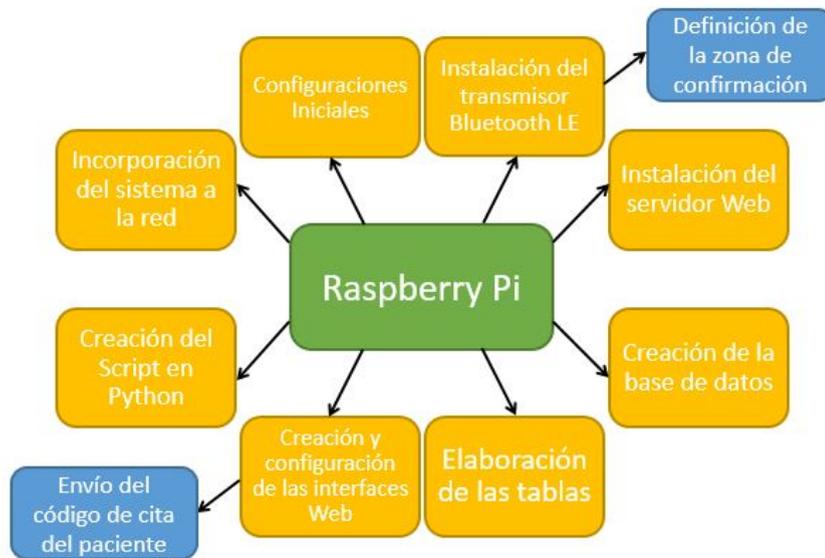


Figura 3.13: Integración de la Raspberry Pi con el sistema.

### 3.3.1. Configuración de la Raspberry:

Antes de empezar la instalación y configuración de las tecnologías es necesario adecuar la tarjeta Raspberry pi para así posteriormente instalar cada una de las tecnologías.

La Configuración de la Raspberry empezó con las configuraciones iniciales encontradas en la página web del fabricante [19], donde destaca la instalación y configuración del sistema operativo. .

Al terminar con el sistema operativo de la tarjeta se empieza con la integración de las tecnologías, donde la primera de las tecnologías a integrar es el Bluetooth LE.

La integración de la Raspberry con la tecnología Bluetooth 4.0 se llevó a cabo mediante la incorporación de un adaptador Bluetooth versión 4.0 a un puerto

USB de la Raspberry, y la configuración de este adaptador manipulando la interfaz HCI0, siendo esta habilitada por la implementación de algunos paquetes y controladores en el sistema operativo de la tarjeta, entre los cuales se destaca el paquete BLUEZ desarrollado para entornos LINUX.

### 3.3.2. Instalación del transmisor Bluetooth LE:

Para la incorporación del dispositivo transmisor de la señal Bluetooth a la tarjeta Raspberry pi, se hace uso de la librería Bluez, la cual hace posible la integración de estas dos tecnologías y permite configurar la transmisión de la señal.

La instalación de las librerías necesarias se realiza desde el terminal de dicha tarjeta de la siguiente manera:

1. Se instalan las primeras librerías
2. Se crea el directorio bluez, se accede a este y se ejecutan otras líneas de comandos.
3. Una vez concluido el paso anterior se introduce el adaptador Bluetooth 4.0 en uno de los puertos USB y se reinicia la Raspberry Pi.
4. Reiniciada la tarjeta se ingresa al terminal y se ejecuta la configuración de la siguiente manera:

```
cd ./bluez/bluez-5.18
sudo tools/hciconfig hci0 up
sudo tools/hciconfig hci0 noscan
sudo tools/hciconfig hci0 leadv 3
sudo tools/hciconfig hci0
```

A este punto ya se encuentra configurado el transmisor para el Broadcasting de la señal Bluetooth LE, y solo hace falta asignar el arreglo de bytes a transmitir, el cual contendrá el Manufacturer Specific Data, el UUID, un MAJOR,

un MINOR y un TxPower que se calibra con el RSSI a un metro de distancia del transmisor. Para asignar estos valores se ejecuta la siguiente línea:

```
sudo tools/hcitool -i hci0 cmd 0x08 0x0008 1E 02 01 1A 1A FF
MSD 02 15 UUID MAJOR MINOR TxPower
```

#### 5. Asignación del Advertising Frequency

```
sudo hcitool -i hci0 cmd 0x08 0x0006 A0 00 A0 00 03 00 00 00 00
00 00 00 00 07 00
sudo hcitool -i hci0 cmd 0x08 0x000a 01
```

Ya culminada la configuración anterior, se posee una Raspberry Pi capaz de transmitir un Broadcast de una señal de Bluetooth LE hacia cualquier dispositivo que se encuentre en su área de cobertura.

Para la implementación del sistema se tomó en cuenta el Manufacturer ID de Google y se establecieron los siguientes parámetros para la configuración de la trama de publicidad (Advertising Frame) del dispositivo:

```
Manufacturer ID: (Google) 224 = 00 E0
UUID: E2 C5 6D B5 DF FB 48 D2 B0 60 D0 F5 A7 10 96 E0
MAJOR: 00 00
MINOR: 00 00
TxPower: -63 dBm = C1
```

Para darle autonomía al sistema, dicha configuración se introduce en el arranque de la tarjeta, para esto se edita el fichero rc.local localizado en el directorio /etc, agregando las siguientes líneas antes de la sentencia exit 0:

```
sudo /home/pi/blues/blues-5.18/tolos/hciconfig hci0 up
sudo /home/pi/blues/blues-5.18/tolos/hciconfig hci0 noscan
sudo /home/pi/blues/blues-5.18/tolos/hciconfig hci0 leadv 3
sudo /home/pi/blues/blues-5.18/tools/hcitool -i hci0 cmd 0x08
```

```
0x0008 1E 02 01 1A 1A FF E0 00 02 15 E2 C5 6D B5 DF FB 48 D2
B0 60 D0 F5 A7 10 96 E0 00 00 00 00 C1
sudo hcitool -i hci0 cmd 0x08 0x0006 A0 00 A0 00 03 00 00 00
00 00 00 00 00 07 00
sudo hcitool -i hci0 cmd 0x08 0x000a 01
```

A la hora de definir la zona de cobertura mediante la cual se asignará el área de confirmación de asistencias se diseñó y desarrolló una aplicación para dispositivos Android, con el objetivo de facilitar las mediciones del RSSI del transmisor Bluetooth anteriormente instalado.

### 3.3.3. ScanBT:

Para esta aplicación se creó una única Actividad llamada MainActivity.java la cual trabaja de la siguiente manera:

- Implementa las librerías de API 21 para la manipulación del adaptador Bluetooth del dispositivo; esto quiere decir que la aplicación sirve solo para dispositivos con Android versión 5.0 en adelante.
- Le proporciona al usuario una interfaz limpia e interactiva para lograr los objetivos.
- Realiza una búsqueda de dispositivos Bluetooth 4.0 (Bluetooth LE)
- Concluida la búsqueda muestra los dispositivos encontrados mediante un ListView y le permite al usuario seleccionar sobre cuál de estos realizar el estudio.

Como servidor web se instaló en la tarjeta Raspberry Pi el servidor Apache en su versión 2 debido a que este viene bajo la licencia de software libre y es compatible con la mayoría de las plataformas. La creación de ficheros PHP y su funcionamiento en conjunto con este servidor bajo los protocolos HTTP, hacen posible la conexión y el direccionamiento de todo el flujo de información a través del sistema, por esto el servidor web representa uno de los elementos primordiales.

### 3.3.4. Instalación del servidor Web:

El primer paso para la instalación del servidor web es el acondicionamiento de la tarjeta, el cual consiste en actualizar el sistema operativo y dar una serie de permisos necesarios para utilizar el servidor; para ello es necesario suministrar los siguientes comandos en el terminal respetando el orden correspondiente:

- Asegurarse de que la tarjeta Raspberry tenga conexión a Internet.  
Si la red en la que se encuentra hace uso de un servidor proxy es seguro que tenga que realizar una autenticación.
- Hacer la instalación y el otorgamiento de permisos al grupo de trabajo.
- Se hace un update de los repositorios y se procede a instalar Apache y PHP.
- Al terminar las instalaciones se reinicia el servicio de Apache.
- Culminada la instalación de Apache y php, se procede a instalar MySQL y phpMyAdmin.
- Al instalarse aparecerá una pantalla de configuración preguntando la contraseña que se quiera usar.
- Luego se vuelve a reiniciar el servicio apache  
En este punto ya se encuentra instalado el servidor web junto con los servicios de php y mysql.
- Para la configuración de la cantidad de conexiones simultáneas permitidas por el servidor puede ser modificado el archivo **apache2.conf** cambiando los valores del parámetro MaxClients del módulo correspondiente; sin embargo no fueron alterados estos valores quedando de forma predeterminada, como se muestra en la figura 3.14.

```
<IfModule mpm_prefork_module>
  StartServers      5
  MinSpareServers   5
  MaxSpareServers   10
  MaxClients        150
  MaxRequestsPerChild 0
</IfModule>
```

**Figura 3.14:** Configuración de la cantidad de conexiones simultáneas permitidas.

En caso de que el Administrador del sistema sea reemplazado y no se cuente con la contraseña del usuario **root** de la base de datos, se debe acceder al fichero **config.inc.php** del directorio llamado **directorio**, donde se especifican las configuraciones necesarias para el inicio de sesión en la interfaz de PhpMyAdmin.

En cuanto a la base de datos utilizada por el sistema, fue diseñada bajo la herramienta PHPMYADMIN la cual tiene una buena compatibilidad con el servidor web implementado; esta brinda una interfaz de fácil manipulación y tiene la capacidad de almacenar toda la información necesaria para el correcto funcionamiento del sistema.

### 3.3.5. Base de datos:

La base de datos establecida se llama **medicos**. En ella se encuentran 3 tablas principales mas una tabla por cada doctor registrado en la institución.

### 3.3.6. Tabla del listado de doctores:

Las columnas de la primera tabla llamada **lista\_doctores** contienen la siguiente información:

1. **ID:** Utilizado para enumerar todos los especialistas registrados en la tabla.
2. **Médico:** En esta sección se coloca el nombre del especialista.
3. **Consultorio:** En esta columna va el número del consultorio del especialista registrado.

4. **Horario:** En esta columna se colocan los horarios y días de consulta del especialista.
5. **Notificación:** Esta es la sección donde se coloca el mensaje que el especialista desea mostrar a los usuarios que accedan a la aplicación.
6. **Especialidad:** La especialidad del doctor, que servirá para separarlos y poder mostrar solo los médicos de las especialidades elegidas por el usuario.
7. **Consultas:** Esta sección puede tener 2 valores: **1** o **2**.
  - a. **1:** Cuando el especialista registrado atiende consultas por previa cita, es decir, que los usuarios no podrán registrarse en la lista de espera debido a que deben dirigirse al consultorio a programar su cita según disponibilidad del especialista.
  - b. **2:** Cuando el especialista registrado atiende consultas por orden de llegada. Esta opción es la que permite que el usuario se registre en la lista de pacientes de ese día, para posteriormente poder ser atendido.
8. **Cod:** Esta última columna se utiliza para definir el código que va a tener cada especialista, con el que podrá posteriormente actualizar el mensaje que podrán ver los usuarios cuando accedan a la aplicación.

Finalmente, uniendo todas las columnas se observa la estructura de la tabla, que se muestra en la imagen a continuación:

id	medico	consultorio	horario	notificacion	especialidad	consultas	cod
----	--------	-------------	---------	--------------	--------------	-----------	-----

Figura 3.15: Estructura Tabla 1.

### 3.3.7. Tabla de registro de especialidades:

La siguiente tabla tiene por nombre **especialidades** y su función es establecer un listado de especialidades tratadas en el centro médico, de la cual el paciente seleccionará su búsqueda.

Las columnas establecidas para esta tabla son:

1. **ID:** Nuevamente esta columna tiene como función enumerar el listado de especialidades registradas.
2. **Opción:** Esta segunda columna funcionará para registrar las especialidades tratadas en la Institución.

Finalmente la apariencia de la tabla será:



id	opcion
----	--------

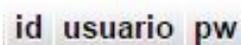
**Figura 3.16:** Estructura Tabla 2.

### 3.3.8. Tabla de administradores:

La siguiente tabla establecida en la base de datos tiene por nombre **administrador**, y en ella se registrará el nombre de usuario y la contraseña de las personas encargadas de actualizar y monitorear las tablas. Este valor será fijo y se establecerá de esa manera para poder controlar el acceso a estas modificaciones.

Cuenta con 3 columnas:

1. **ID:** De igual manera se encuentra la columna de ID que servirá para enumerar los administradores.
2. **Usuario:** Donde se colocará el nombre de usuario de cada persona encargada de la administración de las tablas.
3. **PW:** En esta columna se colocará la contraseña de cada persona encargada de la administración de las tablas.



The diagram shows three adjacent rectangular boxes, each containing a label. From left to right, the labels are 'id', 'usuario', and 'pw'. These labels represent the column headers for Table 3.

Figura 3.17: Estructura Tabla 3.

### 3.3.9. Tablas de registro de pacientes:

Por último se establecieron las tablas que tendrán por nombre el código de cada especialista, donde se registrarán los pacientes para posteriormente ser atendidos, bien sea que decidan registrarse haciendo uso de la aplicación, de la interfaz Web o que sean registrados por uno de los administradores en el Centro Médico.

Consta de 8 columnas:

1. **ID:** Igualmente la columna de ID para enumerar los pacientes registrados.
2. **Nombre:** Nombre completo del paciente a tratar.
3. **Edad:** Edad del paciente.
4. **CI:** Cédula de Identidad del paciente.
5. **Email:** Correo electrónico del paciente, en el que se le enviará un código de confirmación de cita si realiza el registro él mismo.
6. **Cod:** Código de confirmación de cita generado de forma aleatoria y enviado a su correo electrónico.
7. **Conf:** En esta columna se establecerán 2 valores posibles: **3** o **4**
  - a. **3:** Cuando el paciente se ha registrado utilizando la aplicación o la interfaz Web pero no ha confirmado su cita.
  - b. **4:** Cuando el paciente fue registrado en el centro médico por uno de los administradores o se ha registrado él mismo y ya se dirigió al centro médico y confirmó su cita haciendo uso de su código de confirmación de cita y estando en el radio de cobertura del transmisor.

---

id	nombre	edad	ci	email	cod	conf
----	--------	------	----	-------	-----	------

---

Figura 3.18: Estructura Tabla 4.

Para el registro y actualización de la información almacenada en esta base de datos se creó una interfaz web administrativa alojada en la tarjeta Raspberry pi, mediante la cual los administradores designados podrán iniciar sesión para mantener al día toda la información allí almacenada, las cuales fueron creadas con código HTML y PHP.

### 3.3.10. Interfaz administrativa:

La interfaz Web administrativa requiere de nombre de usuario y contraseña para poder iniciar sesión por los administradores, para lo cual previamente se deben haber registrado en la tabla llamada **administrador** los usuarios que tengan autorización para hacer estas modificaciones. Se verifican los datos introducidos con los registros de esta tabla y se permite o no el acceso al sistema.

De haber iniciado sesión correctamente se mostrará un mensaje que diga **Iniciando sesión**. Si el usuario ingresa pero permanece más de 10 minutos sin actividad, la próxima vez que presione un enlace o desee cambiar de página en la interfaz, el sistema le indicará que su sesión ha cerrado por inactividad y lo redireccionará a la página de inicio de sesión.

Se divide en 3 partes principales: Especialidades, Especialistas y Consultas, que determinan y controlan las adiciones y modificaciones de cada una de las tablas registradas en la base de datos.

Para las secciones de Especialidades y Especialistas se utilizaron 5 archivos .php en cada una, de los cuales 3 contienen formularios que permiten registrar, mostrar o modificar la información, y los otros 2 están encargados de controlar la realización de esto.

Ambas secciones poseen una parte de registro donde se deben agregar los Especialistas y Especialidades, según sea el caso. Para hacer esto, el administrador debe llenar todos los campos del formulario y presionar el botón que indica **Registrar**. Al hacer esto, se hace una consulta a la base de datos y se busca la tabla con el nombre indicado, dependiendo del caso, **especialidades** o **lista\_doctores**. Una vez hallada la tabla, se ejecuta el registro en la misma y se muestra un mensaje de alerta que indica **Registro exitoso**.

La siguiente sección muestra una tabla, resultado de una consulta a la base de datos que muestra todos los registros hechos y presenta un cuadro de texto que permite buscar en toda la tabla el registro que coincida con lo introducido.

Se puede apreciar también un enlace que dice **Editar** y otro que indica **Eliminar** en cada fila de la tabla. El enlace editar selecciona la opción presionada, envía la elección por GET, es decir, por el URL, redirecciona a otra página que muestra la información almacenada en ese registro, y permite editar dicha información. Por el contrario, la opción eliminar muestra un mensaje de alerta utilizado para hacer la confirmación del borrado del presente registro. Al seleccionar la opción **Aceptar** se verá un mensaje que dice **Eliminación exitosa** y se verá de nuevo la página con el listado de registros.

Para la sección de Consultas se utilizaron en cambio 9 archivos .php debido a que existen mas elementos a tener en cuenta.

De igual manera posee 5 archivos para agregar, mostrar, editar y/o eliminar los registros que funcionan de la misma manera que la sección anterior, pero a su vez posee una opción para elegir de una lista el médico a consultar, y poder apreciar la lista de pacientes registrados hasta ese momento.

En esta sección se incluye también la opción de confirmar la asistencia de pacientes que no puedan hacerlo haciendo uso de la aplicación, lo cual requerirá escribir el código de cita, y al presionar el botón confirmar se extraerá de la tabla **lista\_doctores** todos los códigos de los especialistas, recorrerá todas las tablas con estos nombres y donde consiga un registro con el código introducido, se procederá a confirmar su asistencia colocando un 4 en la columna de **conf**. Al conseguir el

registro y actualizar la información mostrará un mensaje que indique **Paciente confirmado** y redireccionará a la página principal. De no conseguir ningún paciente no confirmado con el código introducido, mostrará un mensaje indicándolo.

De decidir proceder al registro, el paciente debe seleccionar la opción **Insertar** en el doctor de su elección, lo que lo redireccionará a la página del formulario donde deben escribir sus datos. Si el paciente completa el registro y oprime **Registrar** recibirá su código de cita en su correo electrónico junto con un mensaje de alerta en la interfaz donde se le indicará que su registro fue exitoso.

Como el sistema está orientado a solventar un problema de toda una comunidad, se creó una interfaz web pública alojada de igual forma en la tarjeta Raspberry pi, destinada a todas aquellas personas que aún no tienen acceso a la aplicación Android por no poseer un dispositivo móvil con las características anteriormente mencionadas.

### 3.3.11. Interfaz del usuario:

La interfaz del usuario está encargada de informar a los pacientes que no puedan hacerlo mediante la aplicación, así como también permitirles incluirlos en la lista de espera de los doctores que trabajen con la modalidad de consultas por orden de llegada. De igual manera los pacientes deben dirigirse al centro médico y pedirle a uno de los administradores de las interfaces que le confirmen su asistencia a la consulta.

Esta interfaz funciona similar a la sección de Consultas de los administradores, con la diferencia de que en esta los pacientes verán una lista desplegable proveniente de una consulta con la tabla **especialidades** en la base de datos, en ella deben seleccionar la especialidad que están buscando y presionar el botón **Buscar**, lo que los redireccionará a una página donde se apreciará la información disponible de todos los doctores de esa especialidad, lo que incluye nombre del especialista, consultorio, horarios de consulta, notificación y cantidad de pacientes insertados en su lista de espera, de igual manera que se aprecia mediante la aplicación. Solo podrán registrarse en los médicos que trabajen con consultas por orden de llegada.

A su vez, en el sistema sobresale la fase de confirmación de asistencias a las consultas médicas, para la cual se utilizó la tecnología Beacon, la cual es muy innovadora y está siendo utilizada por muchas compañías para la creación de sistemas en el mundo del Internet de las cosas (IoT); esta tecnología implementa el Bluetooth 4.0 para la creación de zonas de cobertura, con la cual a través del protocolo de información utilizado puede detectar e identificar a los dispositivos. Esto es muy importante ya que mediante éste mecanismo los pacientes ratifican y conservan su puesto en la lista de espera.

Con respecto al método que emplean los médicos para mantener informados a los usuarios de las distintas interfaces, se hizo uso de un correo electrónico destinado a la recepción de esta información, y se implementaron las librerías de protocolos de mensajería en internet como el IMAP en un script en lenguaje Python para la administración de la base de datos. Además se implementó el protocolo de mensajería en internet SMTP, para el envío de los códigos con los cuales los pacientes confirmarán su asistencia al centro médico.

Otro elemento creado es el script en lenguaje Python, fue agregado al arranque de la tarjeta Raspberry pi para proporcionarle al sistema la habilidad de auto administrarse y obtener un sistema que funcione autónomamente con las únicas dependencias de un punto de alimentación y acceso a internet.

### **3.3.12. Script de Python:**

El Script de Python se encarga de realizar las siguientes actividades:

- Leer correos electrónicos y actualizar la base de datos en función de la información recibida, con la finalidad de que los doctores del centro médico puedan enviar un mensaje con la información que desean que los usuarios vean cuando acceden a la aplicación, lo que puede incluir información de sus consultas, disponibilidad de atender pacientes en ese día, o cualquier información que deseen transmitir.

Para esto se utilizaron las librerías **email** e **imaplib**. Se estableció la conexión con la base de datos y se designó el correo al cual los especialistas deben enviar su mensaje para hacer la modificación correspondiente, con la contraseña del mismo para que el programa pueda iniciar sesión.

Funciona de la siguiente manera:

El doctor envía un correo electrónico que debe tener por asunto su código de especialista (asignado aleatoriamente en el momento del registro en la base de datos, que puede ser consultado por uno de los administradores de la interfaz gráfica y no puede ser modificado a menos que se haga un nuevo registro).

En el asunto del correo se debe escribir textualmente el mensaje que desea que aparezca en su registro, para que pueda ser visto por los pacientes al consultarlo.

El sistema trabaja haciendo la consulta de los mensajes no leídos en el correo electrónico designado, y compara el asunto de estos con cada código en el registro de los doctores. Al conseguir uno cuyo código coincida con el asunto del mensaje, actualiza la notificación con la información escrita en el correo.

- Crear las tablas de los doctores registrados en la base de datos, donde se almacenarán los pacientes que hayan realizado el registro por cualquiera de los dos medios disponibles para hacerlo.

Esto ocurre seleccionando en la tabla de los doctores sus códigos y creando las tablas respectivas, a las cuales les colocará de nombre el código de cada especialista para posteriormente almacenar la información de sus pacientes.

Para la implementación de este script se hace uso del fichero `rc.local` de la misma forma en que se utilizó para asignar la configuración del transmisor bluetooth, agregando la línea de comando encargada de ejecutar dicho script, acompañada del símbolo `&` al final de esta para que así se ejecute en segundo plano y no interfiera con las demás tareas del sistema.

Tomando en cuenta a todos aquellos dispositivos que inicialmente vienen con Android inferior a 4.3 y que posteriormente sus dueños logran actualizar sus sistemas operativos de modo que cumplan con este requisito, el código de la aplicación

se ha diseñado para detectar si se cuenta o no con la versión correspondiente del adaptador Bluetooth a la hora de la confirmación de la asistencia, siendo el peor de los casos que este no cuente con Bluetooth 4.0 y por lo cual el usuario deberá dirigirse hasta el puesto del asistente o punto de control para efectuar su confirmación.

Para la incorporación del sistema a la red del centro médico se estudiaron las diferentes configuraciones y estructuras que podrían estar presentes en dicho centro, y se plantearon distintas formas de implementar los protocolos de internet para obtener una óptima conexión, entre las cuales se destacan los servicios DNS, NAT y el redireccionamiento del tráfico a través de los puertos de internet.

### **3.3.13. Incorporar el sistema a la red:**

A la hora de incluir el sistema a la red del centro médico lo primero que hay que hacer es analizar la estructura de la misma, definir que dispositivos están presentes y que tabla de enrutamiento están activas, así como también verificar los protocolos previamente configurados.

La mayoría de las estructuras cuentan con un servidor DNS y Switches con Servicios NAT que re direccionan las peticiones de acuerdo a un dominio en Internet casado a una IP pública; aunque también puede ser implementado el sistema manipulando los puertos HTTP o cualquier otro que se encuentre disponible en la red.

A continuación se detallan los requisitos para implementar ambas configuraciones:

#### 1. Requisitos para implementar la estructura centrada en un Servidor DNS:

- Poseer un dominio en Internet para la IP pública de la red.
- Configurar el dominio dentro del servidor DNS.
- Configurar servicios NAT en los Routers para resolver peticiones hacia el HOST.
- Definir el servicio Virtual Host en el Servidor Apache de la tarjeta Raspberry.

2. Requisitos para implementar el sistema manipulando los puertos de comunicación:

- Obtener un dominio en internet para la IP pública de la red
- Configurar el direccionamiento de las peticiones en los puertos hacia la IP privada de la tarjeta Raspberry
- Escoger un puerto libre dentro de la red para direccionar las peticiones hacia la tarjeta Raspberry
- Definir el servicio Virtual Host en el Servidor Apache de la tarjeta Raspberry.

Las configuraciones presentadas anteriormente no se desarrollan a profundidad, estas mismas son tareas que corresponden a una persona capacitada en el área.

Otro elemento de gran importancia es la aplicación para dispositivos Android la cual fue diseñada y desarrollada a través de la herramienta Android Studio, mediante la cual se implementaron scripts en lenguaje JAVA que procesan toda la información, manipulan el adaptador Bluetooth del dispositivo y se enlazan con ficheros XML que poseen todos los elementos de la interfaz gráfica de esta aplicación.

Cabe destacar que esta aplicación posee restricciones en cuanto a hardware y software las cuales se centran primordialmente en la capacidad del dispositivo para realizar la confirmación de la asistencia a las citas, para lo cual es indispensable contar con Bluetooth 4.0 en dicho dispositivo. Estas limitaciones de hardware hacen que la aplicación esté disponible para dispositivos con sistemas operativos Android 4.3 en adelante debido a que en esta versión es donde se incluye al Bluetooth 4.0 dentro de la plataforma.



Figura 3.19: Creación de la aplicación Android.

### 3.3.14. Aplicación Android:

Esta etapa del proyecto consistió en la elaboración del código que maneja toda la aplicación, concentrando todas las tareas en ficheros JAVA compiladas con extensión de Actividades; estas se encargan de manipular las librerías y los componentes del dispositivo para su funcionamiento.

Esta aplicación se desarrolló concentrando las tareas en 5 Actividades que se presentan a continuación:

#### 3.3.14.1. MainActivity.java:

En esta Actividad se maneja todo lo correspondiente a la pantalla de inicio. Se muestra la pantalla de bienvenida y se le asignan tareas a cada uno de los botones contenidos en la misma.

Uno de los botones se configuró con la capacidad de verificar el adaptador Bluetooth en busca de su compatibilidad con el sistema diseñado, de no cumplir los requisitos se le informa al usuario que debe concluir la operación por medio de otra interfaz, mientras que si cumple con estos se hace inicio de la actividad de confirmación `Confirm_Activity.java`.

Por otro lado el segundo botón da inicio a la consulta de información ejecutando la actividad `SeleccionActivity.java`

#### **3.3.14.2. SelecciónActivity.java:**

Esta Actividad se desarrolló para hacer una consulta HTTP a la base de datos, y mostrarle al usuario a través de un `ListView` todas las especialidades para las cuales hay al menos un médico laborando en el centro médico.

La actividad espera que el usuario seleccione un ítem, e inicia la siguiente actividad (`Listamedicos.java`) con la referencia del ítem previamente seleccionado.

#### **3.3.14.3. Listamedicos.java:**

Esta Actividad se encarga de hacer una consulta HTTP al servidor de modo de establecer una conexión con la base de datos y mostrarle al usuario a través de un `ListView` cada uno de los médicos que laboran en el centro médico y que correspondan con la especialidad previamente seleccionada.

A su vez la actividad espera que el usuario se decida por un médico en específico, al ser pulsado algún ítem de la lista el código se encarga de verificar que el médico al cual corresponde el ítem trabaje con la nueva modalidad, de ser así, mediante una nueva consulta HTTP al servidor se cuenta el número de pacientes en la lista de espera, la muestra por medio de un `AlertDialog` y le plantea al usuario las opciones de incorporarse a la lista o confirmar su asistencia.

De ser el caso que dicho médico no haya decidido trabajar bajo esta modalidad se muestra un `AlertDialog` haciéndole saber al usuario que debe buscar otro medio para conseguir una cita con dicho médico.

#### 3.3.14.4. `InsertarPaciente.java`:

Una vez superadas las actividades anteriores, esta se encarga de tomar los datos de cada `EditText` del formulario presente en el `Layout` y luego de verificar que el usuario haya escrito de manera correcta la información del paciente; ejecuta una consulta `HTTP` al servidor para incorporar sus datos a la lista de espera correspondiente.

Luego de terminado el proceso muestra el `Layout` de despedida con los botones para salir de la aplicación o volver al inicio.

#### 3.3.14.5. `Confirm_Activity.java`:

Esta Actividad es la que le permite al usuario realizar la confirmación de su asistencia al centro médico y luego esperar por ser llamado a la consulta médica.

Al inicio de la actividad el código se encarga de comprobar nuevamente los requerimientos de la aplicación para realizar las tareas y espera que el usuario pulse el botón **Confirmar**.

Al pulsar el botón **Confirmar** el código se cerciora de que el dispositivo posea el `Bluetooth` activado, y de ser así toma los datos del `EditText` del `Layout` correspondiente a un código alfanumérico de 6 dígitos, verifica que este cumpla con las condiciones definidas anteriormente e inicia una búsqueda de dispositivos `Bluetooth 4.0`.

Una vez la aplicación haya detectado la señal de algún dispositivo toma su trama de `Advertising`, extrae su `UUID` y lo compara con el `UUID` definido para el `beacon` designado en la zona de confirmación del centro médico.

De corresponder el UUID con el del beacon y siendo su RSSI mayor a -100 dbm se realiza una consulta HTTP al servidor, para cambiar en la base de datos el estado del paciente, de **no confirmado** a **confirmado** y pueda ser llamado a la hora de la consulta.

Se crearon adicionalmente 2 clases igualmente en JAVA para el manejo de las consultas HTTP y las conversiones de datos:

1. `HttpRequest.java`: Esta clase posee los métodos necesarios para realizar cada una de las consultas HTTP que permiten a la aplicación comunicarse con los ficheros PHP que posee el servidor web.
2. `convert.java`: Esta clase fue creada con el fin de interpretar y manipular datos de tipo hexadecimal y arreglos de bytes necesarios para el manejo de los protocolos de comunicación utilizados por la tecnología Bluetooth.

Para mayor información sobre el código implementado consulte el Apéndice A.

Tomando en cuenta que el código de cita del paciente es lo que le permite a este realizar la confirmación de su asistencia, se desarrolló un método para su envío a través del sistema de correo electrónico como se presenta a continuación:

### 3.3.15. Envío del código de cita del paciente:

En el momento en que el paciente se registra, bien sea haciendo uso de la aplicación o de la interfaz Web, recibe un código de cita con el que podrá posteriormente hacer la confirmación de la misma al llegar al centro médico. Para eso se utilizaron 2 clases llamadas **class.phpmailer** y **class.smtp** y se configuró el sistema para que cada vez que se inserte un nuevo paciente en la lista de espera se genere un código alfanumérico de 6 caracteres y luego se envíe este código hacia la dirección de correo electrónico registrada por el usuario, partiendo desde el correo electrónico establecido para estas tareas, el cual, en el caso de que la Institución tenga su propio servicio de correos electrónicos, debe establecerse con uno de estos debido a que tendrá configurado el puerto smtp con dicho servicio.

Para la creación del código alfanumérico, se estableció un patrón de 6 caracteres con las siguientes características:

1234567890abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

El cual se aprecia que está conformado por un conjunto de números, letras en minúscula y letras en mayúscula, sin incluir caracteres especiales tales como: \*, \$, %, ñ, etc.

## Capítulo IV

# Análisis, interpretación y presentación de los resultados

Considerando las características finales que debe cumplir un sistema de información [26] y tomando como referencia otros sistemas de información [27] se presentan los resultados obtenidos al culminar la elaboración de este proyecto.

Se obtuvo como resultado un sistema que puede ocuparse por sí solo de los distintos escenarios que podrían presentarse, como se aprecia en la figura 4.1 y se indican posteriormente.

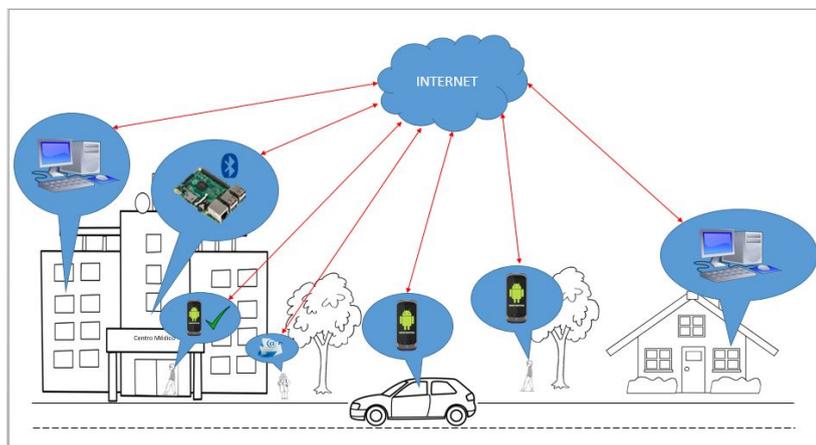


Figura 4.1: Escenarios de funcionamiento del sistema completo.

- El usuario que está consultando la información sobre los especialistas que laboran dentro del centro médico desde su dispositivo Android.
- El usuario de la aplicación Android que está solicitando incorporarse en la lista de espera de algún especialista anteriormente consultado.
- El usuario de la aplicación Android que desea confirmar su asistencia dentro del centro médico.
- El médico que tiene una información que comunicarle a los pacientes que hacen la consulta de información.
- El usuario que no posee la aplicación Android, consulta la información de los especialistas y solicita la incorporación a una lista de espera a través de la interfaz Web.
- El usuario que aún no posee la aplicación Android ni tiene acceso a la interfaz Web, y se ha dirigido hasta el centro médico para solicitar su incorporación a una lista de espera.
- El asistente o encargado de incorporar y editar la información de las especialidades, los médicos que deseen utilizar esta herramienta y los pacientes que se dirigen al centro médico, y de ir llamándolos cuando sea su turno a la consulta.

A continuación se muestran las mediciones realizadas y las pruebas de funcionamiento del sistema.

#### **4.1. Transmisión y recepción de la señal Bluetooth**

Para comprobar la recepción de la señal en los dispositivos Android se utilizó la aplicación **ScanBT** elaborada en el presente proyecto, mediante la cual fue posible medir y establecer las zonas de cobertura colocando el transmisor en el centro de Dimetel sede Bárbula (antigua RedUC) en la Universidad de Carabobo a 1 m de altura y midiendo el RSSI en los distintos puntos mostrados en la figura 4.1.

Los resultados obtenidos se muestran en la figura 4.1, e indican el comportamiento de la señal del transmisor en distintos puntos del recinto; estas mediciones permiten evaluar la capacidad de recepción de los dispositivos móviles, definiéndose así un límite de  $-100\text{dBm}$  en los valores de RSSI aceptados.



**Figura 4.2:** Plano Dimetel sede Bárbula con mediciones del RSSI en diversos puntos.

Este valor de  $-100\text{ dBm}$  se definió como el nivel mínimo de potencia requerido por la aplicación para poder efectuar la confirmación de la asistencia; posteriormente se procedió a realizar las mediciones para definir el límite de la zona de confirmación de asistencia, obteniéndose los resultados mostrados en la figura 4.3, los cuales reflejan un patrón de radiación isotrópico con un radio que varía de acuerdo a las obstrucciones que se encuentren en esa dirección.

Cabe destacar que los límites de la zona de cobertura no pudieron medirse completamente debido a que la zona se encuentra cerrada y no se pudo acceder.

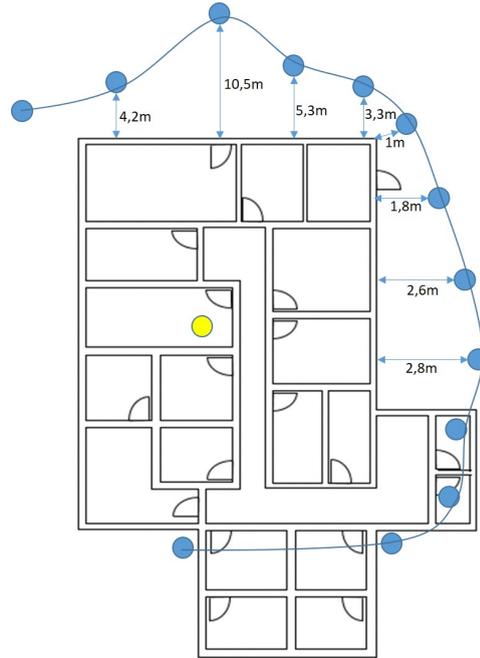


Figura 4.3: Límites de la zona de cobertura del transmisor.

En la siguiente imagen se aprecia el funcionamiento de la aplicación ScanBT, tomando 10 muestras y calculando el promedio obtenido en ellas para los dos ejemplos mostrados:

ScanBT	
REINICIAR	VOLVER
11:24:07 AM	found= 5C:F3:70:6A:0D:BD = -70 dBm
11:24:22 AM	found= 5C:F3:70:6A:0D:BD = -60 dBm
11:24:28 AM	found= 5C:F3:70:6A:0D:BD = -58 dBm
11:24:40 AM	found= 5C:F3:70:6A:0D:BD = -63 dBm
11:24:59 AM	found= 5C:F3:70:6A:0D:BD = -63 dBm
11:25:02 AM	found= 5C:F3:70:6A:0D:BD = -60 dBm
11:25:11 AM	found= 5C:F3:70:6A:0D:BD = -70 dBm
11:25:24 AM	found= 5C:F3:70:6A:0D:BD = -66 dBm
11:25:33 AM	found= 5C:F3:70:6A:0D:BD = -65 dBm
11:25:42 AM	found= 5C:F3:70:6A:0D:BD = -60 dBm
-- RSSI PROMEDIO = -63.0 dBm --	

Figura 4.4: Medición del RSSI a 1 m del transmisor.



**Figura 4.5:** Medición del RSSI a 18 m del transmisor.

Se tuvo como resultado una variación en el tiempo entre las tramas publicitarias recibidas debido a la latencia agregada por las obstrucciones y la distancia entre los dispositivos.

Se nota como a medida que se hacen las mediciones más distantes del transmisor, aumentaba la latencia entre cada una de las tramas, como se aprecia en la comparación de las figuras 4.4 y 4.5 donde se muestra una medición con promedio de -63dBm correspondiente a una distancia de 1 metro del transmisor y otra con promedio de -96dBm correspondiente a una distancia de 18 metros con 2 paredes de drywall y una pared de concreto de por medio.

En el caso en que se desee aumentar la zona de confirmación de asistencias de manera continua se considera el patrón de radiación teórico del transmisor, el cual es isotrópico y permite extender el área de confirmación cumpliendo la siguiente condición:

$$D \leq 2 * R$$

Siendo:

D: distancia entre los transmisores.

R: radio de la zona de confirmación en la dirección que se desea extender dicha zona.

Para el caso de agregar nuevas zonas de confirmación no existen limitaciones en cuanto a separación entre los transmisores.

Debido a la estructura del sistema y la forma que permitió el Bluetooth LE realizar la transmisión de la señal, le proporciona a dicho sistema la posibilidad de brindarle un servicio equitativo a cada uno de los usuarios, siendo como límite de conexiones simultáneas la cantidad de 150.

## 4.2. Base de datos

En la base de datos se crearon la distintas tablas ya mencionadas, y se agregaron registros para posteriormente llevar a cabo las pruebas requeridas. Se observa que la tabla **especialidades** tiene registrado un listado de especialidades, que son las mostradas en la aplicación y la interfaz web al hacer la consulta.

La información de las especialidades y especialistas es referencial, y fue tomada de la Unidad de Atención Médica Integral de la Universidad de Carabobo (UAMI) la cual cuenta con esta información en su página Web [28].

id	opcion
1	Atención Médica Inmediata
2	Cardiología
3	Ecografía
4	Fisiatría
5	Ginecología
6	Mastología
7	Medicina Interna
8	Odontología Especialistas
9	Odontología Generales
10	ORL
11	Pediatría
12	Traumatología
13	Urología

**Figura 4.6:** Pruebas base de datos. Tabla de especialidades.

La tabla **lista\_doctores** también contiene registros de doctores con distintas especialidades, que son mostrados según la elección del usuario pero se encuentran todos almacenados en la misma tabla, como se muestra a continuación:

id	medico	consultorio	horario	notificacion	especialidad	consultas	cod
1	Gloría Camacho	1	Lun a Vie 7 am a 1 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por ...	Atención Médica Inmediata	2	YJ50
2	María Rodríguez	2	Lun a Vie 1 pm a 7 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por ...	Atención Médica Inmediata	2	HY70
3	Alonso Gómez	3	Lun a Vie 7 am a 1 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por ...	Atención Médica Inmediata	2	T985
4	Mariand Rubio	4	Lun a Vie 1 pm a 7 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por ...	Atención Médica Inmediata	2	91D4
5	Nancy Palacios	5	Lun a Vie 7 am a 1 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por ...	Atención Médica Inmediata	2	BVTR
6	Carlos Delgado	6	Mar 7 am a 1 pm y Vie 7 am a 7 pm	30/Apr/2016-09:03:38-Se atienden 9 pacientes por d...	Cardiología	1	R43S
7	Ibrahim Rodríguez	7	Lun a Jue 6 pm a 9 pm	30/Apr/2016-09:03:38-Se atienden 9 pacientes por d...	Cardiología	1	32V9
8	Francisco García	8	Lun 7 am a 10 am. Mar y Jue 5 pm a 7 pm. Vie 7 am ...	30/Apr/2016-09:03:38-Se atienden 9 pacientes por d...	Urología	2	HY34
9	Freddy Sánchez	9	Mar y Mié 2 pm a 4 pm. Vie 4 pm a 6 pm. Sáb 9 am a...	30/Apr/2016-09:03:38-Se atienden 6 pacientes por d...	Urología	2	BTY7
10	Ramón Rivas	10	Mar y Jue 11 am a 2 pm. Lun, Mié y Vie 3 pm a 6 pm...	30/Apr/2016-09:03:38-Se atienden 9 pacientes por d...	ORL	2	FTRE

**Figura 4.7:** Pruebas base de datos. Tabla de especialistas.

Al seleccionar y hacer la solicitud de cita en el listado de un especialista, se registran los datos del paciente y se incluyen en la tabla de cada doctor, de la siguiente manera:

id	nombre	edad	ci	email	cod	conf
1	Juan Jimenez	60	1348213	josem2303@gmail.com	65THlg	3
2	Giorvi Parra	24	20162185	joselinjim@gmail.com	FR5TUB	4

**Figura 4.8:** Pruebas base de datos. Tabla de pacientes.

Y por último se encuentra la tabla llamada **administrador** donde se registran los administradores de la interfaz, encargados de mantener la información actualizada.

id	usuario	pw
1	joselinjim	joselin23
2	josemov	jose23

**Figura 4.9:** Pruebas base de datos. Tabla de administradores.

### 4.3. Aplicación Android

Al ejecutar las pruebas pertinentes de la aplicación en un teléfono Android se aprecia que las consultas a la base de datos son exitosas, lo que trae como resultado en este caso el listado de especialistas laborando en la institución.

Se observa de la siguiente manera:

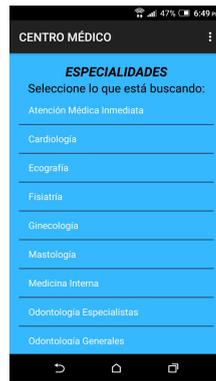


Figura 4.10: Pruebas aplicación en Android Studio. Especialidades.

Al seleccionar una de las especialidades se redirecciona correctamente a la siguiente ventana, lo que proviene de otra consulta en la que se muestran los doctores de la especialidad seleccionada:

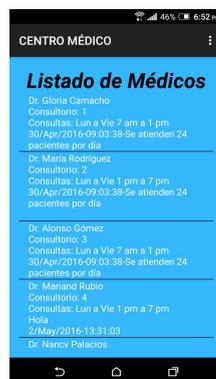


Figura 4.11: Pruebas aplicación en Android Studio. Especialistas.

En esta siguiente sección se apreciarán los especialistas, y al seleccionar uno de ellos pueden ocurrir 2 posibles escenarios:

El primero, que el especialista seleccionado trabaje con la modalidad de consultas por previa cita, lo que mostrará un mensaje indicando:

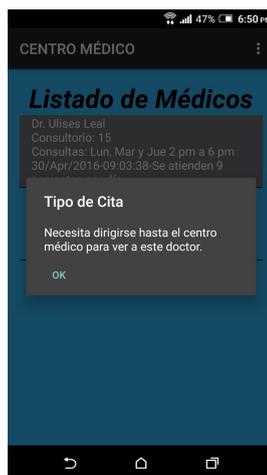


Figura 4.12: Pruebas aplicación en Android Studio. Mensaje previa cita.

Lo que como se observa no le proporciona al usuario acceso para registrarse en la lista de pacientes de ese día.

El segundo escenario ocurre cuando el doctor seleccionado trabaja con consultas por orden de llegada, lo que mostrará:

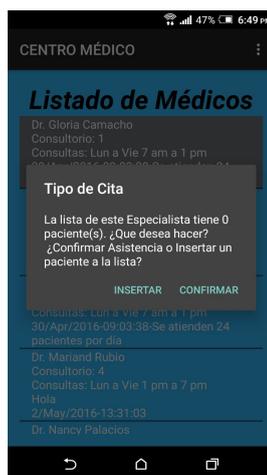
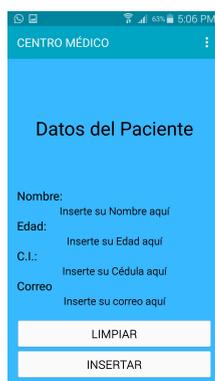


Figura 4.13: Pruebas aplicación en Android Studio. Mensaje orden de llegada.

Aquí se presentan 2 opciones para el paciente. Si selecciona **Insertar** lo trasladará al formulario de registro que le permitirá insertar sus datos e incluirse en la lista de espera.



**Figura 4.14:** Pruebas aplicación en Android Studio. Formulario de registro.

Al completar el registro y presionar la opción **Insertar** se verá un mensaje indicando que el paciente está siendo insertado en la lista de espera.

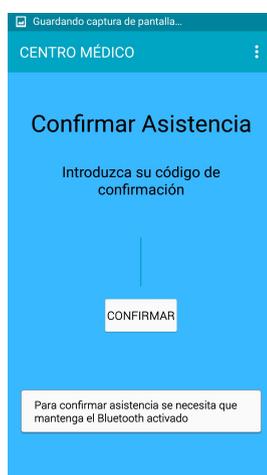
Lo que culminará con un registro exitoso y un mensaje de despedida que dará como culminado este proceso, con un mensaje indicando que sus datos fueron insertados y que debe confirmar su asistencia al llegar al centro médico.



**Figura 4.15:** Pruebas aplicación en Android Studio. Mensaje final.

En este momento el paciente puede salir de la aplicación o volver al inicio a confirmar su cita, si ya se encuentra dentro del centro médico.

La segunda opción mostrada al seleccionar a un especialista indica **Confirmar** y está diseñada para aquellos pacientes que ya se registraron en la lista de espera de algún especialista, se encuentran dentro del centro médico y procederán a confirmar su asistencia. Para esta opción también se puede acceder desde la ventana principal presionando la opción **Confirmar asistencia**. Al oprimir este botón se le requerirá que escriba textualmente su código de cita, al mismo tiempo que le será mostrado un mensaje que le indicará que debe mantener el Bluetooth encendido para proceder con la confirmación.



**Figura 4.16:** Pruebas aplicación en Android Studio. Confirmar asistencia.

Finalmente, al escribir su código correctamente y presionar **Confirmar** se mostrará un mensaje que indica **Confirmando asistencia** lo que posteriormente llevará a una confirmación exitosa mostrando:



**Figura 4.17:** Pruebas aplicación en Android Studio. Paciente confirmado.

Si el paciente escribe un código menor a 6 dígitos, recibirá un mensaje indicando que tiene que insertar un código válido.

Y si no se encuentra dentro del radio de cobertura del transmisor verá un mensaje que le indicará que no se encontró el beacon y que debe asegurarse de estar dentro del centro médico.

#### 4.4. Interfaces Web

Al comprobar el funcionamiento de la interfaz Web administrativa se observa que realiza la consulta a la tabla respectiva, desplegando el listado de especialidades y especialistas al momento de hacer las consultas correspondientes.

En el caso de la adición de nuevos especialistas se requiere que seleccione la especialidad del doctor a registrar, teniendo en cuenta que estas están definidas por la tabla de especialidades que debió ser previamente configurada con las especialidades tratadas en el centro médico.



Figura 4.18: Pruebas interfaz administrativa. Listado de especialidades.

De igual manera en la sección de registro de pacientes se requiere que seleccione el médico tratante, para de esta manera traer su tabla correspondiente y mostrar los pacientes que se han registrado hasta el momento.

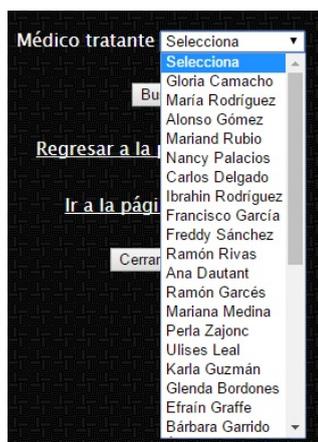


Figura 4.19: Pruebas interfaz administrativa. Listado de especialistas.

En la sección de edición de especialistas se presentan dos listas desplegables que permiten editar la especialidad, lista mostrada en la figura 4.18, y una segunda lista desplegable que permite seleccionar el tipo de consulta por el que atiende el doctor.

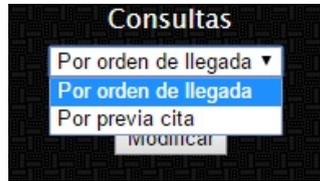


Figura 4.20: Pruebas interfaz administrativa. Tipo de consulta.

Y por último se aprecia el apartado en el que se verán, editarán y eliminarán los registros dependiendo de la sección de la interfaz en la que se encuentre el administrador, que muestra todos los registros establecidos de la tabla requerida, con enlaces que permiten mantener al día la información allí almacenada.

ESPECIALIDAD	MODIFICAR	ELIMINAR
Atención Médica Inmediata	Editar	Eliminar
Cardiología	Editar	Eliminar
Ecografía	Editar	Eliminar
Fisiatría	Editar	Eliminar
Ginecología	Editar	Eliminar
Mastología	Editar	Eliminar
Medicina Interna	Editar	Eliminar
Odontología Especialistas	Editar	Eliminar
Odontología Generales	Editar	Eliminar
ORL	Editar	Eliminar
Pediatría	Editar	Eliminar
Traumatología	Editar	Eliminar
Urología	Editar	Eliminar

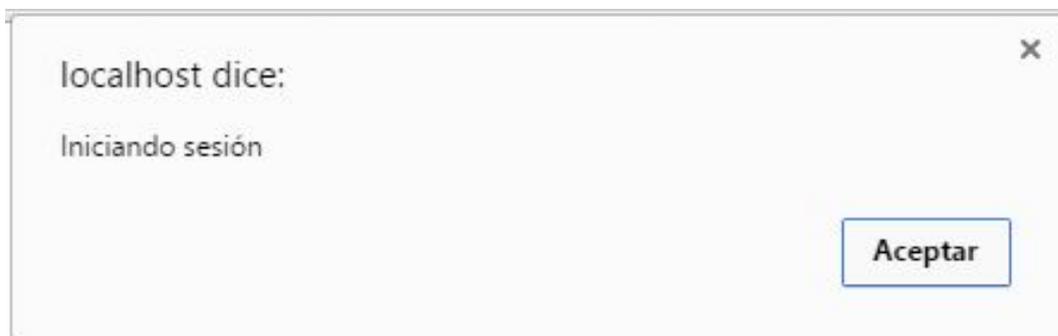
Regresar a la página anterior

[Ir a la página principal](#)

Cerrar sesión

Figura 4.21: Pruebas interfaz administrativa. Tabla de registros.

Durante todo el transcurso de la utilización de la interfaz, el administrador recibirá alertas que le indiquen que se ha ejecutado una acción, así como también que se ha cerrado su sesión cuando ha excedido su tiempo de inactividad. Todas estas notificaciones se aprecian de la siguiente manera, cambiando el mensaje mostrado según sea el caso:



**Figura 4.22:** Pruebas interfaz administrativa. Iniciando sesión.

A diferencia de la notificación que se recibe cuando se desea eliminar un registro, debido a que en esta se debe hacer la comprobación por parte del administrador de que efectivamente desea eliminar ese registro, seleccionando la opción **Aceptar** de así requerirlo.



**Figura 4.23:** Pruebas interfaz administrativa. Eliminando.

Similar a la interfaz administrativa se encuentra la interfaz Web del usuario, que genera la consulta a la base de datos y muestra los registros de las especialidades para permitir la selección

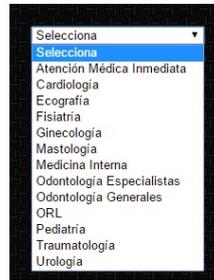


Figura 4.24: Pruebas interfaz usuario. Lista de especialidades.

Y muestra el listado de especialistas registrados para la especialidad seleccionada, con toda la información necesaria para que el usuario tome la decisión de con que doctor consultarse y si registrarse en la lista de espera, lo cual hará seleccionando la opción **Insertar** en el doctor de su elección, y completando los campos del formulario al que será redirigido, recibiendo de igual manera su código de cita en su correo electrónico al completar exitosamente su registro.

**Información sobre: Atención Médica Inmediata**

NOMBRE	CONSULTORIO	HORARIO	NOTIFICACIÓN	CONSULTAS	PACIENTES EN LA LISTA DE ESPERA	INSERTAR PACIENTE EN LA LISTA DE ESPERA
Gloria Camacho	1	Lun a Vie 7 am a 1 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por día	Por orden de llegada	3	Insertar
María Rodríguez	2	Lun a Vie 1 pm a 7 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por día	Por orden de llegada	0	Insertar
Alonso Gómez	3	Lun a Vie 7 am a 1 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por día	Por orden de llegada	0	Insertar
Mariand Rubio	4	Lun a Vie 1 pm a 7 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por día	Por orden de llegada	2	Insertar
Nancy Palacios	5	Lun a Vie 7 am a 1 pm	30/Apr/2016-09:03:38-Se atienden 24 pacientes por día	Por orden de llegada	0	Insertar

Figura 4.25: Pruebas interfaz usuario. Listado de especialistas.

## Capítulo V

# Conclusiones y recomendaciones

### 5.1. Conclusiones

Al concluir la elaboración de este proyecto se obtuvo como resultado un sistema informativo para centros médicos acompañado de una interfaz dinámica que cumple con las expectativas, atacando los problemas de falta de información e interacción entre el médico y sus pacientes.

La creación de este sistema informativo representa un método innovador para proporcionarle a los usuarios una manera eficaz de obtener la información requerida, disminuyendo el tiempo de obtención de esta información en comparación con el método utilizado actualmente.

El sistema se encuentra centralizado en una tarjeta Raspberry con capacidad más que suficiente para emplear este tipo de tareas, por lo que representa una ventaja a la hora de procesar grandes cantidades de información.

La aplicación desarrollada para dispositivos Android cumplió con los objetivos de brindarle al usuario una interfaz dinámica, con la que estos puedan interactuar fácilmente y sin problemas.

La implementación de tecnologías como Bluetooth 4.0 y los protocolos de Internet para la comunicación entre la Raspberry y la Aplicación, hicieron posible crear

un sistema con la capacidad de proporcionarle a los usuarios la oportunidad de solicitar desde el hogar su incorporación en las listas de espera, para optar por una consulta médica mientras este se traslada hasta el centro médico, siendo requisito para ser atendido la confirmación de su asistencia, para así no saturar las listas con pacientes que no asistan al centro médico.

Considerando los requerimientos que los dispositivos móviles de los usuarios deben tener para poder hacer uso de todas las funciones de la aplicación, se creó un sistema paralelo que permite obtener la misma información y solicitar la incorporación en la lista de espera con solo tener un dispositivo con acceso a Internet, para así proporcionar un sistema completo que pueda ser funcional y utilizado por la mayor cantidad de usuarios posibles.

En la búsqueda de crear un sistema automatizado, el script creado en Python y el envío inmediato del código de cita del paciente representan grandes ventajas debido a que permite la creación de las tablas de consultas, la actualización de la notificación de los doctores y la creación y envío del código de cita automáticamente, lo que representa una reducción del trabajo de los encargados del manejo y actualización del sistema.

A pesar de que la aplicación está diseñada para ser utilizada en dispositivos con múltiples tamaños de pantalla, se puede presentar el caso de dispositivos con pantallas muy pequeñas que puedan no mostrar algunos elementos, como por ejemplo el menú de información presente en cada ventana.

## 5.2. Recomendaciones

- Debido a que existen especialistas que trabajan con la modalidad de previa cita, se plantea la configuración de la base de datos para extender las solicitudes de dichas citas para permitirles a los pacientes solicitarla con días de antelación.
- Tomando en cuenta la utilidad de las coordenadas geográficas de los dispositivos, se propone enlazar la aplicación con Google Maps, para reforzar el

mecanismo que permite corroborar la presencia del paciente en el centro médico para confirmar su asistencia a la cita previamente solicitada.

- En busca de un sistema independiente de cableados, autónomo y previendo las fallas que el sistema eléctrico pueda tener en algún momento, se recomienda la incorporación de un adaptador de red inalámbrica que permita la conexión del dispositivo desde cualquier punto del centro médico, así como también la elaboración de un sistema de alimentación y respaldo de energía.
- Se recomienda la elaboración o adquisición de una caja que permita guardar la tarjeta Raspberry Pi, para evitar que se encuentre expuesta y a la intemperie en el lugar en el que se implemente.
- Para minimizar la inversión de adaptadores y descartar problemas de compatibilidad entre estos, se propone la opción de adquirir una tarjeta Raspberry Pi 3 modelo B, ya que viene con un adaptador de Bluetooth 4.1 y un adaptador 802.11n Wireless LAN incorporados.
- En vista de la existencia de centros médicos de gran tamaño que pudieran abarcar un espacio mucho mayor al área de cobertura del transmisor, se recomienda la incorporación de varios transmisores (beacons) al sistema, para que de esta manera se pueda ofrecer una zona de confirmación mayor.
- Complementar el mecanismo de confirmación de asistencia del sistema mediante la implementación de la tecnología NFC.
- Crear la zona de confirmación en base a la señal de un adaptador de red inalámbrica e implementarlo como punto de acceso.
- Implementar el sistema informativo desarrollado anteriormente intercambiando los papeles de transmisor y receptor de la señal Bluetooth

## **Apéndice A**

# **Códigos del sistema**

Para información de los códigos que conforman el sistema, consulte la carpeta correspondiente en el CD adjunto.

# Referencias Bibliográficas

- [1] Juan José Calva González. *Las necesidades de Información*. 2004. URL [http://132.248.242.3/~publica/archivos/libros/necesidades\\_informacion\\_fundamentos.pdf](http://132.248.242.3/~publica/archivos/libros/necesidades_informacion_fundamentos.pdf).
- [2] Esteban García Bustamante. *Nuevas tendencias en la tecnología Bluetooth*. Ciudad Universitaria Rodrigo Facio, Costa Rica, 21 de julio de 2013. URL [http://eie.ucr.ac.cr/uploads/file/proybach/pb2013/pb2013\\_031.pdf](http://eie.ucr.ac.cr/uploads/file/proybach/pb2013/pb2013_031.pdf).
- [3] Daniel Gómez Fernández. *Sistemas de Sensores Inteligentes para detección de automóviles*. Universidad Autónoma de Madrid, Abril 2014. URL <http://arantxa.ii.uam.es/~jms/pfcsteleco/lecturas/20140424DanielGomezFernandez.pdf>.
- [4] Víctor Hernández. *Desarrollo de una aplicación móvil en sistema Android para el control remoto de dispositivos mediante la tecnología Bluetooth 4.0*. 2014. URL [https://riunet.upv.es/bitstream/handle/10251/46610/HERNANDEZ\\_MUNOZ\\_VICTOR.pdf?sequence=1](https://riunet.upv.es/bitstream/handle/10251/46610/HERNANDEZ_MUNOZ_VICTOR.pdf?sequence=1).
- [5] GSMA. *A Guide to Bluetooth Beacons*, Septiembre 2014. URL <http://www.gsma.com/digitalcommerce/wp-content/uploads/2013/10/A-guide-to-BLE-beacons-FINAL-18-Sept-14.pdf>.
- [6] *The Low Energy Technology Behind Bluetooth Smart*, 2015. URL <http://www.bluetooth.com>.
- [7] Kevin Townsend. *Introduction to Bluetooth Low Energy*. Adafruit learning system, Abril 2014. URL <https://learn.adafruit.com/downloads/pdf/introduction-to-bluetooth-low-energy.pdf>.

- [8] Carrie Anne Philbin. *Adventures in Raspberry Pi*. Editorial Wiley, 2014.
- [9] Yeicy Juliana Molina Rivera; Jonathan Sandoval Cardona y Santiago Alberto Toledo Franco. *Sistema Operativo Android: Características y funcionalidad para dispositivos móviles*. Universidad Tecnológica De Pereira, 2012. URL <http://repositorio.utp.edu.co/dspace/bitstream/11059/2687/1/0053M722.pdf>.
- [10] Manuel Báez; Álvaro Borrego; Jorge Cordero; Luis Cruz; Miguel González; Francisco Hernández; David Palomero; José Rodríguez de Llera; Daniel Sanz; Mariam Saucedo; Pilar Torralbo y Álvaro Zapata. *Introducción a Android*. Editorial E.M.E., 2012. URL <http://www.it-docs.net/ddata/18.pdf>.
- [11] Jorge Martínez Ladrón de Guevara. *Fundamentos de programación en Java*. Editorial EME, 2011. URL <https://docs.google.com/file/d/0Byy7aUL9u4fBdnZjVnZ0ampmTjA/edit>.
- [12] Oscar Belmonte Fernández. *Introducción al lenguaje de programación Java. Una guía básica*. Junio 2005. URL <http://www3.uji.es/~belfern/pdidoc/IX26/Documentos/introJava.pdf>.
- [13] *Cómo funciona la Web*. Universidad de Chile, Junio 2008. URL <http://www.ciw.cl/libroWeb-NV.pdf>.
- [14] J.M Vara; Marcos López Sanz; Jesús Javier Jiménez Hernández; Jenifer Verde Marín & Fúque Sánchez. *Desarrollo Web en entorno servidor*. Editorial RA-MA, 2012.
- [15] Abraham Silberschatz; Henry F. Korth & S. Sudarshan. *Fundamentos de Bases de Datos*. Editorial McGRAW-HILL, 2002. URL <https://unefazuliasistemas.files.wordpress.com/2011/04/fundamentos-de-bases-de-datos-silberschatz-korth-sudarshan.pdf>.
- [16] Rafael Camps Paré; Luis Alberto Casillas Santillán; Dolors Costal Costa; Marc Gibert Ginesta; Carme Martín Escofet y Oscar Pérez Mora. *Software Libre*. Fundació per a la Universitat Oberta de Catalunya, Mayo 2005. URL <http://www.uoc.edu/masters/oficiales/img/913.pdf>.

- [17] Python, 2016. URL <http://www.python.org/>.
- [18] Guido van Rossum. *El tutorial de Python*. 2009. URL <http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf>.
- [19] *Raspberry Pi*, 2016. URL <https://www.raspberrypi.org/>.
- [20] Raspbian. URL <https://www.raspbian.org/>.
- [21] Pidora. URL <http://pidora.ca/>.
- [22] Risc. URL <https://www.riscosopen.org/content/>.
- [23] Eclipse. URL <http://www.eclipse.org>.
- [24] BlueJ. URL <http://bluej.org/about.html>.
- [25] Android Developers, 2016. URL <http://developer.android.com/intl/es/index.html>.
- [26] Antonio García Jiménez; Inmaculada Chacón Gutiérrez; Alberto Díaz Esteban; Pablo Gervás Gómez-Navarro. Nuevos sistemas de información: tendencias y evaluación. 2000. URL <http://pendientedemigracion.ucm.es/info/multidoc/multidoc/revista/num9/prensa/jime-chacon.htm>.
- [27] S. Juncosa Font B. Bolívar Ribas. *VALIDACION DEL PROCESO Y RESULTADOS DE UN SISTEMA DE INFORMACION EN ATENCION PRIMARIA*, 1992.
- [28] Unidad de Atención Médica Integral. URL <http://www.uc.edu.ve/uami/index.php>.

