



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

---

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

Control de Permanencia y Asistencia de Personas en un  
Área Laboral mediante la detección de Smartphones

Tesis que para obtener el título de  
Licenciado en Ingeniería en Ciencias de la Computación

PRESENTA:  
Mauro Martín Letras Luna

ASESOR: Nicolás Quiroz Hernández

Septiembre 2013

**BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA**

RESUMEN

**Control de Permanencia y Asistencia de Personas en un Área Laboral  
mediante la detección de Smartphone**

Por **Mauro Martín Letras Luna**

Asesor **Mtro. Nicolás Quiroz Hernández**

En esta tesis se presenta un sistema para la detección de la permanencia y asistencia de un grupo de personas dentro de un área laboral, mediante la localización de *smartphones* (teléfonos inteligentes) de los empleados. Desarrollando una serie de tarjetas controladoras de comunicación Bluetooth con su respectivo firmware.

La elección de este tipo de tecnología es porque se puede utilizar el transceptor Bluetooth que viene integrado en los smartphones, además por el impacto social que los smartphones han ejercido, puesto que la mayoría de las personas cuentan con un dispositivo de estas características.

Para localizar a los empleados se implementara una *red ethernet*; y cada uno de los nodos tendrá un transceptor *Bluetooth*; se realizara un estudio del alcance de la señal Bluetooth y de la estructura física del edificio.

El sistema permitirá al usuario final configurar los horarios y los smartphones a localizar, esto debido a que la mayoría de los lugares de trabajo existe más de un turno laboral.

El uso de la *Ethernet* se justifica porque es necesario recabar toda la información de los nodos para situarla en un punto central (*servidor*), para que a distancia se pueda supervisar a las personas que se encuentran dentro del área de alcance de la red.

La implementación de cada nodo de la *red Ethernet* se hará en un *microcontrolador de 8 bits* para el control de un transceptor Bluetooth y un transceptor Ethernet, el cual servirá en la detección de smartphones y el intercambio de información entre cada uno de los nodos.

## TABLA DE CONTENIDO

Lista de figuras.....	vi
Lista de tablas.....	viii
Capítulo I: Introducción y Estado del Arte.....	15
Definición del Problema.....	15
Propuesta de Solución.....	16
Objetivos.....	17
Alcance.....	18
Limitaciones.....	18
Herramientas.....	19
Métodos Tradicionales de Identificación.....	20
Sistemas basados en Biométricos.....	23
Descripción del Documento.....	26
Capítulo II: Tecnología Bluetooth.....	29
Origen del nombre Bluetooth.....	30
Redes de Área Personal.....	31
Panorama de Bluetooth.....	31
Pila de Protocolos.....	32
Capa Física.....	34
Banda Base.....	37
Protocolos de Alto Nivel.....	41
Protocolo de Detección de Servicios.....	43
Perfiles.....	45
Topologías de Comunicación.....	46
Bluetooth comparado contra otras tecnologías.....	47
Capítulo III: Modelado del Sistema.....	49
Perspectiva del Sistema.....	50
Suposiciones y Dependencia.....	52
Actores.....	52
Casos de Uso.....	53
Diagrama de Secuencia.....	56
Diagrama Entidad Relación.....	60
Capítulo IV: Red para Detección de Smartphones.....	65
Diagrama a Bloques del Sistema.....	66
Transceptor Bluetooth.....	67
Transceptor Ethernet.....	73
Microcontrolador.....	75
Desarrollo de Algoritmo.....	77
Usando Piconet.....	77
Usando Scatternet.....	81
Usando Módulos Bluetooth – Ethernet.....	87
Tarjeta de Circuito Impreso.....	88
Capítulo V: Aplicaciones de Usuario Final.....	91

Aplicación de Escritorio .....	92
Proceso de Búsqueda de Módulos Bluetooth - Ethernet.....	98
Página WEB.....	100
Capítulo VI: Pruebas del Sistema .....	103
Pruebas sobre el Módulo Bluetooth .....	103
Pruebas de Sistema Completo .....	112
Capítulo VII: Conclusiones y Trabajo a Futuro .....	117
Bibliografía.....	119
Apéndice A: Principales comandos AT.....	122
Apéndice B: Firmware Tarjeta Bluetooth - Ethernet .....	126

## LISTA DE FIGURAS

<i>Número</i>	<i>Página</i>
1. Uso de Telefonía Celular en México .....	16
2. Sistema de Manejo de Asistencia, Japón .....	22
3. Logotipo Bluetooth.....	30
4. Modelo a Capas de Bluetooth.....	32
5. Canales de Comunicación WIFI.....	36
6. Canales de Comunicación Bluetooth .....	36
7. División de Tiempo Dúplex .....	38
8. Piconet.....	40
9. Dirección Bluetooth.....	41
10. Protocolo de Detección de Servicios .....	44
11. Perfiles .....	45
12. Piconet con dispositivos móviles .....	46
13. Tipos de Piconet.....	46
14. Scatternet .....	47
15. Modelo de Solución de Problema.....	50
16. Diagramas de Caso de Uso.....	53
17. Diagrama de Secuencia de Administrar Edificio .....	57
18. Diagrama de Secuencia de Administrar Datos Persona .....	58
19. Diagrama de Secuencia para Administrar Red .....	59
20. Diagrama de Secuencia para Monitorear Persona.....	60
21. Diagramas Entidad Relación .....	61
22. Diagrama a Bloques .....	66
23. Diagrama a Bloques Detallado .....	66
24. Transceptor Bluetooth.....	68
25. Pines de Modulo Bluetooth.....	70
26. Transceptor Ethernet.....	73
27. Diagrama Bloques Modulo Ethernet.....	74

28. Encapsulados del PIC18F4550.....	76
29. Solución Implementando Piconet .....	77
30. Diagrama de Flujo Algoritmo Maestro .....	79
31. Diagrama de Flujo Algoritmo Esclavo .....	81
32. Solución propuesta haciendo uso de una Scatternet.....	82
33. Tarjeta de Circuito Impreso.....	89
34. Pantalla Principal de Administración de Información .....	92
35. Pantalla Usuarios .....	93
36. Datos Usuarios.....	94
37. Pantalla Teléfonos .....	94
38. Datos Teléfonos .....	95
39. Pantalla Módulos Bluetooth.....	95
40. Datos Módulos Bluetooth .....	96
41. Pantalla Red de Edificios.....	96
42. Detalle de Edificio.....	97
43. Pantalla Horario .....	97
44. Log in a Página Web.....	100
45. Monitoreo en Tiempo Real .....	100
46. Materiales para prueba Bluetooth .....	106
47. Grafica de prueba a 2.5 m.....	106
48. Grafica de prueba a 5 m .....	107
49. Grafica de prueba a 10 m.....	107
50. Edificio 140 C.....	108
51. Comando AT para Búsqueda.....	110
52. Implementación de tarjeta Bluetooth - Ethernet.....	106

## LISTA DE TABLAS

<i>Número</i>	<i>Página</i>
1. Material para Trabajo de Tesis .....	19
2. Clases de Dispositivos Bluetooth .....	34
3. Caso de Uso Administrar Datos de Edificio.....	54
4. Caso de Uso Administrar Datos de Edificio.....	55
5. Caso de Uso Administrar Red.....	55
6. Caso de Uso Monitorear Persona .....	55
7. Caso de Uso Visualizar Página Web.....	56
8. Características Transceptor Bluetooth .....	69
9. Características UART .....	69
10. Comandos AT para UART .....	71
11. Funciones para el ENC28J60.....	75
12. Características del PIC18F4550.....	76
13. Algoritmo del Módulo Maestro de Solución Piconet.....	78
14. Algoritmo Modulo Esclavo de Solución Piconet.....	80
15. Algoritmo Maestro con Ethernet de Solución Scatternet.....	83
16. Algoritmo Puente de Solución Scatternet.....	85
17. Algoritmo Maestro de Solución Scatternet .....	86
18. Algoritmo Modulo Ethernet - Bluetooth .....	87
19. Elementos de la tarjeta Bluetooth - Ethernet.....	89
20. Algoritmo para Monitoreo Módulos Ethernet.....	98
21. Algoritmo Proceso Comunicación modulo Bluetooth – Ethernet .....	99
22. Características de materiales para pruebas .....	104
23. Resultados pruebas sobre Transceptor Bluetooth .....	105
24. Resultados pruebas sobre Piconet .....	109
25. Búsqueda de Smartphones muestra 1 .....	110
26. Búsqueda de Smartphones muestra 2.....	111
27. Algoritmo búsqueda de múltiples smartphones .....	111
28. Horario usado para la primera prueba de sistema .....	112

## AGRADECIMIENTOS

Antes de agradecer a todo mundo por su valiosa ayuda y cooperación en este proyecto académico y personal, más personal que académico, quisiera hacer mención especial para mi madre y padre ya que sin ellos nada de esto hubiese sido posible –no me refiero sólo al apoyo económico o moral sino que si ellos no se conocen, no estaría escribiendo esto en estos momentos-, gracias Sara Luna Jiménez y José Martín Letras Méndez, solo ahora espero no defraudarles, es una promesa. También mi agradecimiento para cada uno de mis hermanos Dulce “Duche”, Alfredo “Pino” y Eduardo que han sido los primeros amigos y compañeros que he tenido en mi vida.

También quiero extender estos agradecimientos a todas las personas que conocí durante este viaje que hoy termina. A todos los amigos que hice durante estos últimos años, Mario, Misa, Borre, Pipe, Luis, Toño, Charly, “Amibo”, Emma, “Werito”, Fili, los chicos del “pong”, a Raúl y Nalle, los compañeros de la sección 107, y a todas aquellas personas que no estoy mencionando pero que compartieron algún momento conmigo, gracias por sus consejos –me sirvieron o confundieron aún más, pero siempre con la intención de ayudar-que me demostraron que la amistad real si existe y viene en distintas presentaciones, formas y tamaños.

También deseo expresar mi más sincero agradecimiento a los profesores de la facultad con quienes compartí el salón de clases, que me confirmaron que el esfuerzo y la perseverancia son lo principal para poder culminar nuestros objetivos como alumnos. Y que a través de señalar mis errores me enseñaron que las cosas siempre pueden ser mejores.

Al M.C. Nicolás Quiroz Hernández le agradezco por la confianza y paciencia depositada en mí para la realización de este proyecto, por sus consejos y correcciones en cada una de las etapas de esta tesis, y puedo decirle que he ganado un amigo en usted.

A los integrantes del jurado, los profesores Mario Bustillo Díaz y Apolonio Ata Pérez por el tiempo y apoyo brindado para la revisión de este proyecto. Su opinión fue muy importante y sus correcciones me servirán para mejorar académicamente.

Y para no hacer más largo esto, quiero concluir con unas palabras que alguien me dijo recientemente, y se me quedaron grabadas, “solo se el cambio que tú quieres ver en tu mundo” y quiero ser comprensión, perseverancia, fortaleza, esfuerzo, amor, computación, música y valentía. Gracias a todos, gracias a la vida.

*I am not fully known to myself, because part of  
what I am is the enigmatic trace of others*

## GLOSARIO

**Algoritmo.** Conjunto ordenado y finito de operaciones que permite hallar la solución de un problema.

**Biometría.** El concepto biometría proviene de las palabras *bio* (vida) y *metría* (medida), por lo tanto con ello se infiere que todo equipo biométrico mide e identifica alguna característica propia de la persona

**Bluetooth.** Es una especificación industrial para Redes Inalámbricas de Área Personal que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda de los 2,4 GHz.

**Dirección MAC.** (Siglas en inglés de media access control; en español "control de acceso al medio") es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una tarjeta o dispositivo de red.

**Dúplex.** Término utilizado para definir a un sistema que es capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea.

**Ethernet.** Es una popular tecnología LAN (Red de Área Local) que utiliza el Acceso múltiple con portadora y detección de colisiones entre estaciones con diversos tipos de cables.

**Firmware.** Es un bloque de instrucciones de máquina para propósitos específicos, grabado en una memoria, normalmente de lectura y/o escritura (ROM, EEPROM, flash, etc.), que establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo de cualquier tipo.

**Half Duplex.** Significa que el método o protocolo de envío de información es bidireccional pero no simultáneo.

**Hardware.** Conjunto de los componentes que integran la parte física de una computadora.

**Microcontrolador.** Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

**PCB.** Tarjeta de circuito impreso o PCB (del inglés printed circuit board), es una superficie constituida por caminos o pistas de material conductor laminadas sobre una base no conductora.

**PIC.** Son una familia de microcontroladores tipo RISC fabricados por Microchip Technology.

**Piconet.** Una piconet es una red cuyos nodos se conectan utilizando Bluetooth. Una piconet puede constar de dos a siete dispositivos. En una piconet, habrá siempre un maestro y los demás serán esclavos.

**Protocolo.** Un protocolo de comunicaciones es un conjunto de reglas y normas que permiten que dos o más entidades de un sistema de comunicación se comuniquen entre ellos para transmitir información.

**Scatternet.** Es un tipo de red de computadoras ad-hoc que está formada por dos o más piconets.

**Smartphone.** Es un teléfono móvil con una mayor capacidad de almacenar datos y realizar actividades semejantes a una computadora.

**Software.** Conjunto de programas, instrucciones y reglas informáticas para ejecutar ciertas tareas en una computadora.

**Transceptor.** Es un dispositivo de comunicación que cuenta con un transmisor y un receptor que comparten parte de la circuitería o se encuentran dentro del mismo dispositivo.

**UART.** Son las siglas de "Universal Asynchronous Receiver-Transmitter" (en español, Transmisor-Receptor Asíncrono Universal). Éste controla los puertos y dispositivos serie. Se encuentra integrado en la placa base o en la tarjeta adaptadora del dispositivo.





---

# INTRODUCCIÓN

En este capítulo se abordan las motivaciones personales y profesionales en el desarrollo de este sistema para control de asistencia. Se ofrece una justificación del uso de Bluetooth como la tecnología para conseguir dicho propósito. Además se define un objetivo general el cual culminara en base al cumplimiento de una serie de objetivos particulares. Después se muestran las ventajas de la solución propuesta sin olvidar las desventajas, para mejorar el sistema en un trabajo futuro. También se hará un estudio de la manera en que se lleva el control de asistencia, y los factores que se ven involucrados al realizar esta tarea. Se mostraran las distintas maneras en que se realiza el pase de lista y las distintas tecnologías involucradas, desde el uso de una simple lista impresa en papel hasta sistemas que identifican el acceso de las personas en base a características físicas únicas (sistemas basados en biométricos). Ya para finalizar se hace una descripción de cómo está estructurado este documento.

## **Definición del problema**

Un trabajador en un ambiente laboral tiene la obligación de asistir, llegar puntual y permanecer en el trabajo, estos aspectos repercuten directamente en el rendimiento de la empresa en la que trabaja, esto puede representar ganancias o pérdidas (si el empleado es impuntual y falta mucho). En base al cumplimiento de los puntos mencionados anteriormente a un trabajador se le puede premiar (con algún tipo de bono de puntualidad y/o asistencia) o castigar (con algún descuento en el salario). Es aquí donde surge la necesidad

de llevar un registro de la asistencia y puntualidad de un grupo de trabajadores, y proporcionar un mecanismo que permita garantizar la permanencia de los mismos en su lugar de trabajo durante un horario asignado.

### Propuesta de Solución

La solución propuesta para esta tesis consiste en el uso de tecnología *Bluetooth*. El objetivo a perseguir radica en la localización de *smartphones* (que cuentan en su mayoría con un transceptor *Bluetooth*), mediante el SDP (*Service Discovery Protocol* por sus siglas en inglés), que es un protocolo que manejan los dispositivos *Bluetooth* para descubrir dispositivos cercanos, se pretende estar supervisando constantemente un grupo de *smartphones* que estarán ligados con un grupo de trabajadores. Para garantizar la asistencia se asignará un *smartphone* únicamente con un trabajador, aprovechando que los dispositivos *Bluetooth* tienen asignada una dirección única llamada *dirección Bluetooth* (muy similar a la dirección *MAC* que también es única para cada tarjeta de red *Ethernet*).



Figura 1. En esta figura se hace una comparativa del uso de telefonía celular y el servicio de internet, se puede observar que hay un muy notable crecimiento de usuarios de tecnología móvil, datos obtenidos de la Comisión Federal de Telecomunicaciones [1]

Uno de los motivos para usar *smartphones* es el gran impacto que han tenido en la sociedad moderna. Actualmente hay seis billones de usuarios de tecnología celular a nivel mundial, lo cual representa tres cuartas partes de la población

mundial [2]. Además hay un incremento acelerado del uso de telefonía celular en países en vías de desarrollo en comparación con países desarrollados. En lo que respecta a México, la población registrada en 2010 por INEGI (*Instituto Nacional de Estadística y Geografía*) fue de 112 millones 336 mil 538, de esta población según cifras de la COFETEL (*Comisión Federal de Telecomunicaciones*) hay 97.6 millones usuarios de telefonía celular. Este indicador nos quiere decir que de cada 100 mexicanos 86.9 tienen acceso a un equipo celular (Figura 1).

Además realizando una consulta de los equipos que ofrece la más grande empresa de telefonía celular en el país, hay 216 equipos disponibles en el mercado de los cuales 180 cuentan con un *transceptor Bluetooth* [3].

Estas razones fueron el motivo por el cual usar Bluetooth, ya que no es necesario comprar algún tipo de transceptor adicional, usando la tecnología disponible y debido a la gran difusión de los *smartphones* se puede dar solución a este problema.

## Objetivos Generales y Específicos del Proyecto

### Generales:

Desarrollar un sistema empotrado (*microcontrolador*) para detección del tiempo de permanencia y asistencia de un grupo de personas en un área laboral, mediante una *red Ethernet*, donde cada nodo tendrá un *transceptor Bluetooth* que estará detectando los smartphones de los empleados.

### Específicos:

- Analizar e implementar una *red Ethernet*, haciendo uso de microcontroladores, para detección de smartphones.
- Determinar la ubicación adecuada de los *nodos de la red* dentro de un edificio para obtener el alcance de la señal Bluetooth, permitiendo conocer la cantidad de módulos requeridos.
- Desarrollar la tarjeta controladora que servirá de nodo de la red que estará formada por un *transceptor Ethernet y uno Bluetooth*. Diseñar *hardware* y *firmware* para la *tarjeta Ethernet*.
- Analizar y estudiar protocolos de comunicación para hacer una elección de la manera óptima de intercambiar información entre los nodos de la *red*.
- Diseñar un protocolo para la recolección de información y comunicación entre los nodos en la *red*

### **Alcance**

El alcance del proyecto está basado en los objetivos que se pretenden culminar. Hacer un sistema basado en tecnología *Ethernet*, la cual recolectará los datos necesarios para brindar información estadística de la asistencia, puntualidad y permanencia de un grupo de trabajadores en un área laboral.

Se pretende que a través de una base de datos toda la información sea guardada para consultas posteriores.

El sistema solamente podrá reconocer a *smartphones* que anteriormente hayan sido registrados en el sistema, cualquier otro dispositivo será ignorado.

La búsqueda de dispositivos se realiza mediante el *transceptor Bluetooth*, los resultados serán enviados por el *transceptor Ethernet* para posteriormente ser almacenados en la base de datos.

Dependiendo de la infraestructura del edificio se configuraran las conexiones de cada nodo de la red y se ubicaran dependiendo de la zona que abarquen.

### **Limitaciones**

Las principales limitaciones que se presentan son físicas, de los *transceptores Bluetooth* ya que el radio que abarca la señal es de 10 m.

Otra limitación importante es que la red se crea de manera estática, los *módulos con comunicación Ethernet* están comunicados directamente con el servidor. Debido a esto si se pierde la conexión con un módulo este quedara aislado del sistema hasta solucionar el problema de manera manual.

La estructura física de los edificios donde se implantará el sistema juega un papel importante porque dependiendo de los materiales usados en la construcción es muy probable que la señal disminuya, pasando a ser menor de 10 m de radio de la señal.

## Herramientas Usadas

En la tabla 1 se hace un listado del material que se necesitó para llevar a cabo este trabajo de investigación.

Equipo	Comentario
Computadora	En este equipo es donde se realizará la programación del hardware y software involucrado en el sistema. Para el uso de redacción de documentos
Smartphone	Un <i>smartphone</i> con Bluetooth, para poder realizar pruebas
Compilador C	Herramienta de software para la programación de <i>microcontroladores</i>
Visual Studio	Para realizar la implementación de la aplicación que administrara los horarios y una página web donde visualizar resultados.
Transceptores Bluetooth	Los dispositivos Bluetooth con los cuales se montara la red.
Microcontrolador de 8 bits	Micro controlador de la empresa Microchip
Fuente de Voltaje	Para trabajar con elementos que necesiten alimentación eléctrica
Software de PCB	Herramienta de Software para el diseño de PCB
Transceptor Ethernet	El dispositivo <i>Ethernet</i> para enviar datos al servidor

Tabla 1. Material utilizado en el trabajo de tesis

## Control de Asistencia

El control de asistencia consiste en verificar que una persona va a estar en un lugar por determinado tiempo realizando una tarea en específico (trabajo, escuela, etc.). No solo es necesario indicar que la persona asistió a un sitio a una hora determinada sino saber cuánto tiempo permaneció en su área de trabajo, cuánto tiempo estuvo ausente de las misma y la hora en la cual se retiró del sitio. Por eso un sistema que se encargue de controlar la asistencia debe de garantizar por lo mínimo los siguientes puntos.

- Registrar hora de entrada, para comprobar puntualidad
- Registrar permanencia en el área de trabajo
- Registrar salida

La manera más sencilla para realizar este cometido la podemos encontrar en la mayoría de las aulas de escuelas del país. Consiste en que el profesor llega a clase con una lista de los alumnos (impresa o en formato digital) y empieza de manera oral verificando la asistencia de los alumnos. Si el alumno se encuentra presente responde a su nombre alzando la mano o indicando su presencia, entonces el profesor registra en su lista la asistencia o falta del alumno dependiendo el caso. Una de las principales ventajas de este enfoque, es que

garantiza el primer punto que debe de cumplir un sistema de estas características, que es comprobar la puntualidad, debido a que la lista solo se pasa una vez al inicio de la clase. Pero tiene desventajas en garantizar la permanencia de los alumnos en el salón de clases, porque una vez checada la lista los alumnos pueden salir del salón sin que su asistencia se vea afectada. Además es imposible verificar la salida de los alumnos solo que el profesor pase de nuevo lista al finalizar la clase pero esta circunstancia rara vez se ve en un salón.

Otra manera de pasar lista es cuando el profesor llega a clase, hace pasar una hoja con fecha e indica a los alumnos que escriban su nombre, matrícula y firma. Este enfoque tiene muchas desventajas comparado con el mencionado anteriormente. Debido a que no garantiza con certeza ninguno de los puntos que debe de cumplir un sistema de estas características. Empezando por la puntualidad y permanencia, porque alguien más puede pasar lista por un compañero y falsificar la firma, además el profesor no cuenta con un registro de las firmas de cada alumno para compararlas y descartar un posible fraude.

En ambos ejemplos mencionados anteriormente, no se puede realizar consultas posteriores que nos muestren de manera estadística la respuesta del grupo en asistencia y puntualidad.

En comparación con ambientes laborales, la manera de comprobar la asistencia se vuelve más estricta, debido a que este aspecto puede generar ganancias o pérdidas a una empresa. La impuntualidad y la inasistencia pueden dar una imagen a la empresa de irresponsabilidad, de baja productividad y de incumplimiento. Además tiene efectos negativos en la calidad del trabajo, la confianza y el respeto.

En las siguientes líneas se mostraran sistemas que se implementan en empresas e innovaciones alrededor del mundo. Estos sistemas hacen uso de la tecnología disponible en el mercado para garantizar la asistencia de un grupo de personas.

## **Métodos Tradicionales de Identificación**

### **Tarjetas con Banda Magnética**

En términos de control de acceso, hay tarjetas que emplean código de barras, una banda magnética, banda magnética con marca de agua y tecnología *Wiegand*. Las diferentes tecnologías ofrecen varios grados de capacidad de almacenamiento de datos y seguridad contra duplicación. Código de barras y las bandas magnéticas ofrecen poca resistencia contra duplicación fraudulenta aunque *Wiegand* y la banda magnética con marca de agua requieren equipo especializado para decodificar y duplicar.

Las tarjetas son consideradas inestables para su uso en ambientes difíciles porque los datos ubicados en la superficie de la tarjeta pueden ser fácilmente dañados.

## **Tarjetas de proximidad**

Las tarjetas de proximidad o de otra manera conocidas como etiquetas de radio frecuencia (ID RF), y los lectores de radiofrecuencia proporcionan una manera de identificación sin ningún contacto. El lector emite radiación de radiofrecuencia dentro de una zona y cuando una tarjeta de proximidad pasa cerca de esta zona, la circuitería dentro de la tarjeta transmite datos al lector. Con esta información se puede construir un sistema basado en estas características que cumpla con el reconocimiento de un individuo.

A continuación se muestran algunos sistemas desarrollados de manera tradicional para el control de asistencia y permanencia.

## **Sistema de Registro en Línea**

Este sistema está basado en el uso de *aplicaciones web* y *una base de datos*, se encarga de checar la asistencia a un grupo de alumnos. En la base de datos se carga la información correspondiente a los alumnos, los horarios y salones de clase.

El funcionamiento de este sistema es de la siguiente manera, el profesor con su computadora portátil conectada a internet, realiza el pase de lista de manera tradicional, pero los datos de la asistencia y la hora de entrada quedan registrados en la base de datos. Con los datos guardados se puede realizar un análisis estadístico sobre la respuesta del grupo. Además los alumnos pueden hacer consulta de su historial de faltas y asistencia vía una *página web*. Una ventaja de este sistema es que puede indicar al instructor cuando un alumno ha alcanzado las faltas o impuntualidades toleradas, para poder aplicar una sanción.

La desventaja principal sigue siendo el tiempo en el que se registra la asistencia de los alumnos ya que se necesita la colaboración de alumnos e instructor para tener los datos de la asistencia en el sistema. [3]

## **Sistema para registro de asistencia usando teléfonos móviles y Bluetooth**

El enfoque mostrado en este sistema es similar al que se desarrolla en esta tesis. Se hace uso de la tecnología *Bluetooth* que viene integrada en la mayoría de los teléfonos celulares modernos. El contexto de este sistema es un salón de clases donde el profesor necesita registrar asistencia. Justificándose en el impacto que han tenido los celulares, el sistema registra asistencia a través de los celulares indicando que cada celular corresponde solamente con un alumno. El instructor llega al salón de clases y desde su computadora portátil corre una aplicación desarrollada en *Python*, que se encarga de buscar los celulares dentro del aula. Si la aplicación encuentra el celular que está ligado con un alumno indica la asistencia en el sistema, en caso contrario pone falta al alumno.

La principal ventaja de este sistema radica en que se elimina el tiempo que el profesor gastaba en estar mencionando el nombre de cada alumno, ya que este trabajo lo realizan los *dispositivos Bluetooth*.

La desventaja de este sistema es que necesita tener una computadora portátil en cada una de las aulas o como mínimo por cada profesor para poder realizar el pase de lista, además si un alumno se encuentra dentro del rango del transceptor Bluetooth que toma lista, pero esta fuera del salón de clases se le tomara como asistencia, de esta manera registrando un asistencia falsa. [4]

### **Sistema para manejo de asistencia usando teléfonos móviles**

El contexto de este sistema también es para checar asistencia dentro de una escuela. Esta aplicación desarrollada en la Universidad de Kansai en Japón, verifica que la persona se encuentre en el lugar y el tiempo de permanencia. La solución propuesta consiste en que mediante el uso de teléfonos móviles con cámara fotográfica y conexión a internet se envíe evidencia de la asistencia a la clase. El primer paso es que el alumno tome una fotografía del pizarrón del salón a donde asiste, después el alumno envía la fotografía vía correo electrónico indicando en el asunto el código de la clase y en el contenido del correo el nombre y la matrícula del alumno.



Figura 2. Foto tomada desde el telefono movil de un alumno, la cual confirma la presencia y asistencia a la clase

Después que la clase finaliza, en el servidor de la escuela se obtiene toda la información enviada vía correo electrónico por parte de los alumnos. Con este mecanismo se garantiza que el alumno esté presente en su clase por un tiempo determinado. Para evitar posibles fraudes se checará la información correspondiente a la hora en que se tomó la fotografía, el tamaño de la misma, el equipo celular que la obtuvo, fecha y hora. Si estos datos coinciden exactamente entre dos fotografías es muy probable que alguien esté haciendo trampa y quiera registrar una falsa asistencia, en este caso el sistema alerta de esta situación para corroborar la asistencia con más detalle.

La desventaja principal de este sistema es que se necesita tener forzosamente un teléfono móvil con conexión a internet y aplicaciones de correo electrónico. [5]

## **Sistemas basados en Biométricos**

Los sistemas basados en biométricos son sistemas automatizados que verifican o reconocen a una persona por algún aspecto conductual (habla, firma, etc.) o característica fisiológica (huella digital, geometría de la mano, etc.).

Aplicados a sistemas de acceso y control de asistencia, los sistemas biométricos están orientados para identificar a la persona que requiere el acceso.

Los sistemas biométricos operan en dos modos. En *modo de reconocimiento*, el sistema compara una muestra obtenida en tiempo real con todas las plantillas en la base de datos y determina la mejor coincidencia. En *modo verificación*, el sistema requiere que la persona se identifique primero, con una clave que le fue asignada con anterioridad, después el sistema busca las plantillas correspondientes de la base de datos y acepta o rechaza la asistencia en base a las coincidencias.

Incluso para bases de datos pequeñas, el *reconocimiento* es muy demandante en comparación con la *verificación*. Las circunstancias en las cuales usar uno u otro enfoque dependen en gran medida a los requerimientos del proyecto. Pero es notable que al usar un enfoque de *verificación* el tiempo de respuesta del sistema sea muy rápido pero la desventaja es que requiere intervención directa de la persona a la cual se toma asistencia.

Un resumen de los más difundidos sistemas de reconocimiento de biométricos se listan a continuación. [6]

## **Reconocimiento de huella digital**

Los sistemas de reconocimiento de huella digital son abundantes. Estos sistemas realizan una comparación de la huella del individuo con una imagen grabada con anterioridad, si las características concuerdan se confirma el reconocimiento.

Desafortunadamente, daños en la piel o cicatrices, suciedad en los surcos del dedo o incluso el uso de crema para las manos puede resultar en falsos rechazos.

Además, en gente con piel seca no se puede obtener una imagen clara de la huella porque es necesario que haya un poco de humedad en el dedo para tener un buen contacto con el sensor.

### **Geometría de la Mano**

Los sistemas basados en la geometría de la mano miden las dimensiones de la mano como la longitud de los dedos y el área que abarcan.

Uno de los más populares dispositivos para capturar la geometría de la mano, consiste en que el usuario desliza su mano sobre una superficie. La mano es ubicada usando unas barras que sirven de guía. Con la ayuda de espejos, una cámara digital situada arriba de la mano captura tomas de todos los lados para tener una imagen completa de la mano.

### **Reconocimiento de Iris**

El iris en un ojo humano es una textura extremadamente fina y es única como la huella digital. A diferencia de los dedos, el iris rara vez sufre de daños causado por los sensores y es menos propenso a dar falsos resultados debido al daño físico.

Sin embargo, como en la mayoría de los sistemas basados en biométricos, hay condiciones médicas que pueden impedir que ciertos individuos usen el sistema, incluidos aquellos que padezcan enfermedades que obstruyan el iris, como las cataratas.

El escáner de iris ilumina y entonces toma una fotografía del ojo humano. Aunque los lentes de contacto no afectan el rendimiento de este sistema, las gafas pueden causar resultados falsos porque los lentes crean una reflexión que es capturada sobre la imagen del iris.

### **Escáner de Retina**

El escáner de retina almacena el patrón de venas de la retina. Esto requiere de una fuente de luz que cruce a través de la pupila hasta el fondo del ojo. El

principal problema con el escáner de retina es que los usuarios lo consideran intrusivo y se muestran renuentes a mirar por el escáner, mientras su retina es escaneada. Hasta hace poco, el escáner de retina era solamente asociado con estaciones militares y espaciales. Ahora se está dirigiendo al mercado comercial.

## **Reconocimiento Facial**

Los sistemas basados en reconocimiento facial analizan la geometría del rostro y sus características. Variaciones en la iluminación, el ángulo de la cara y la expresión facial hacen del reconocimiento facial una tarea compleja.

En la actualidad, hay disponibles en el mercado sistemas basados en reconocimiento facial que necesitan que el individuo permanezca frente a la cámara, pero hay sistemas más sofisticados en fase de prototipo que son capaces de reconocer a una persona mientras está en movimiento.

## **Reconocimiento de Voz**

Una de las pruebas biométricas menos invasivas es el reconocimiento de voz. La manera en que pronunciamos las palabras depende de muchos factores incluyendo influencias geográficas, las cuerdas vocales y la forma de la boca. Haciendo que un individuo repita un conjunto de palabras o frases, es posible obtener una muestra de voz que identifique a una persona.

Una desventaja del sistema es que puede rechazar a usuarios autorizados debido a ruido excesivo, resfriados o dolores de garganta, o un defecto del habla. Es posible que una grabación pueda engañar al sistema pero para evitar esto, el usuario estará obligado a repetir distintas palabras seleccionadas de manera aleatoria.

## **Firma Dinámica**

La firma dinámica es una modalidad de biométricos que es usada, para propósitos de reconocimiento, de la anatomía y de las características del comportamiento que un individuo exhibe cuando está firmando su nombre (o alguna otra frase). Los dispositivos de firma dinámica no deben de ser confundidos con sistemas de captura de firma electrónica.

Datos como la dirección, el grosor, la presión y la forma de la firma pueden servir para identificar a una persona.

A continuación se listan algunos ejemplos de proyectos basados en la detección de biométricos para el reconocimiento de individuos y toma de asistencia.

## Medición de asistencia en Sistemas de Gestión de Aprendizaje basado en biométricos

Este proyecto realizado en la Universidad de Vigo en España, está diseñado para adaptarse a Sistemas de Gestión de Aprendizaje (SGA como *Moodle*). Para corroborar la asistencia de un alumno se hacen uso de diferentes pruebas como el *reconocimiento facial*, *reconocimiento de voz* y *reconociendo de huella dactilar*. El principal objetivo es modelar el comportamiento del estudiante, permitiendo saber exactamente cuánto tiempo pasa el alumno examinando los contenidos del SGA.

La *identificación y verificación* del rostro son tareas que se llevan a cabo en *modo no colaborativo*. La *verificación de huella dactilar y de voz* se realiza bajo demanda con el objetivo de evitar influencia negativa de factores ambientales o de comportamiento humano como falsos ingresos al sistema.

Estas circunstancias pueden frustrar la *verificación facial* en modo no colaborativo. Este proyecto aborda el problema de la autenticación incluso en ambientes adversos para la *verificación facial*.

La solución consiste en colocar en cada estación una *cámara*, un *lector de huella digital* y un *micrófono*. El primer paso consiste en que el usuario inicie sesión en el SGA, el cual pedirá que coloque su dedo sobre el *lector de huella*. Si la sesión se inició con éxito, el sistema en cierto periodo de tiempo tomara fotografías para confirmar que la persona se encuentra en la estación asignada, y por ultimo de manera colaborativa pedirá que el usuario repita una serie de palabras para verificar con una plantilla de voz guardada con anterioridad en la base de datos.

En comparación con algunas tecnologías de acceso de control y asistencia, la tecnología de biométricos está en su infancia.

Se puede decir que los sistemas biométricos más maduros, son aquellos basados en la *geometría de la mano* y en *huella dactilar*, son los más probados, teniendo aplicaciones a largo del mundo en bancos, control de inmigración, etc. [7]

## DESCRIPCION DEL DOCUMENTO

Se ha dividido la totalidad de este trabajo de investigación en siete capítulos en los cuales se ha detallado de manera completa todo lo relacionado con el diseño y la implementación del mismo.

En el presente capítulo se definieron los objetivos principales que se persiguen dentro de este trabajo de investigación así como el uso de las herramientas necesarias. Se hizo un recorrido por las diferentes maneras en la que se aborda el problema del pase de asistencia y las tecnologías usadas como solución para el mismo.

En el capítulo dos titulado *Tecnología Bluetooth*, se describe de manera detallada todo lo relacionado con este protocolo de comunicación y es sobre el cual se desarrollará nuestra solución al problema. Se explicará de qué manera la señal Bluetooth, es enviada por ondas de radio y la manera que se propaga. También se ve el uso de los distintos protocolos para distintos tipos de datos, las distintas clases de dispositivos (dependiendo de la potencia del dispositivo se pueden alcanzar mayores distancias) y la manera de asociar distintos dispositivos en redes *piconet o scatternet*.

El capítulo tres *Modelado del Sistema*, se definen todo los requisitos del sistema. Para esto se hace uso de UML que es el lenguaje unificado para modelado. Se definen los requerimientos haciendo uso de herramientas de UML como lo son diagramas de casos de uso y diagramas de secuencia. Para finalizar esta sección se hace uso de diagramas entidad relación para definir las relaciones y la base de datos que usará el sistema.

En *Red para Detección de Smartphones*, se abordara todo lo relacionado con el desarrollo de los módulos encargados de tomar la asistencia (módulos Bluetooth - Ethernet) y la manera en que estos se organizan para formar una red. Se detallará en los transceptores y el microcontrolador usados para implementar cada una de las tarjetas Bluetooth – Ethernet. Se verá la implementación del algoritmo que será programada como firmware de cada uno de los módulos.

En el capítulo cinco *Aplicaciones de Usuario Final*, se explica cómo se realizó la implementación del modelado del sistema definido en el capítulo tres. El proceso se dividió en tres apartados, el primero Aplicaciones para Administración de Datos, Proceso para recuperar la información de los módulos y una página web para observar los resultados en tiempo real. Se muestra el resultado final a través de la descripción de las pantallas que se obtuvieron después del proceso de implementación.

En *Pruebas del Sistema*, se explica cómo se realizó la integración del sistema y como se procedió a tomar mediciones y hacer pruebas sobre el rendimiento del sistema. Las principales pruebas que se ven, son sobre el transceptor Bluetooth de manera individual y sobre los módulos Bluetooth – Ethernet.

En el último capítulo titulado Conclusiones y Trabajo a Futuro, se expresan las experiencias obtenidas y a manera de conclusión se plasma el impacto que

puede tener este proyecto de investigación, y se cierra la redacción de este documento con una serie de ideas que podrían mejorar este trabajo.

### **CONCLUSIONES**

En el presente capítulo se hizo una definición del objetivo principal de esta tesis, que es la implementación de un sistema para el control de asistencia. Se ha justificado el motivo por el cual se usa *Bluetooth* como la tecnología base de este proyecto. Ya que el uso está justificado debido al impacto que los teléfonos celulares (que incluyen un transceptor Bluetooth) han ejercido sobre la sociedad moderna.

Después del análisis realizado a diferentes maneras en que se aborda este problema, podemos indicar que los sistemas basados en biométricos todavía son poco confiables en tareas no intrusivas y que para reforzar su funcionamiento se tienen que agregar pruebas intrusivas (donde haya colaboración directa de la persona a monitorear), por esto, se ha optado por realizar el monitoreo mediante *Bluetooth*, de una manera no intrusiva, el usuario solamente tendría que activar su *Bluetooth* antes de entrar a su área laboral, y el modulo *Bluetooth* se encargaría de encontrarlo.



---

# TECNOLOGÍA BLUETOOTH

En este capítulo se expondrá la tecnología base utilizada en esta tesis que es *Bluetooth*, se explicará la manera en la que funciona. Se mostrará como *Bluetooth* trabaja a nivel físico y lógico. Se expondrán estos temas de manera en la cual no profundizar demasiado ya que no es el objetivo de este trabajo, se pretenderá ilustrar la capa lógica a través de una pila de protocolos y servicios; que se comunica directamente con la capa física que es la encargada de enviar la señal de radiofrecuencia. Para finalizar este tema se hará una comparativa de *Bluetooth* con otras tecnologías inalámbricas disponibles en el mercado.

Debido al amplio uso de las computadoras, cada oficina y hogar es un laberinto de cables conectados a la computadora y periféricos. Esta situación puede crear muchos problemas de mantenimiento. Realizar la interconexión de estos dispositivos inalámbricamente a través de ondas de radio ofrece una gran ventaja en términos de menor mantenimiento, más fiabilidad y mejor apariencia. Por estas necesidades se creó *Bluetooth*, que es un sistema de comunicación de corto alcance para el intercambio de datos, entre dispositivos cercanos.

Desarrollado originalmente por *Ericsson*, *Bluetooth* ha experimentado una serie de cambios progresivos en su especificación original, llevados a cabo por el

Grupo de Interés Especial SIG (por sus siglas en inglés) de *Bluetooth* y por la IEEE bajo la referencia del estándar IEEE 802.15.1

En 1994, *Ericsson* publicó un estudio sobre la factibilidad de crear una interfaz de radiofrecuencia de baja potencia para ser usada entre teléfonos móviles y sus accesorios. Para 1998, *Intel*, *IBM*, *Nokia* y *Toshiba* se unieron a *Ericsson* y en mayo del mismo año crearon el SIG. A partir de entonces a la fecha varias compañías líderes no solamente del ámbito computacional sino incluso compañías automotrices se unieron al SIG para desarrollar este estándar. Al día de hoy el SIG cuenta con más de 2500 miembros.

### Origen del Nombre “Bluetooth”

Uno de los aspectos que llaman la atención es el nombre de *Bluetooth*, la mayoría de las iniciativas industriales tienen un nombre que describe a la tecnología a la cual están asociadas o un acrónimo describiendo el nombre completo, un ejemplo es *WI-FI* que significa *wireless fidelity*. En lo correspondiente a *Bluetooth* hay varias historias que describen el origen del nombre, la más aceptada es la siguiente.

Harold Blatand fue rey de Dinamarca aproximadamente del año 940 al 985. Durante su reinado, Blatand unificó a Dinamarca y a Noruega y expandió el cristianismo en Escandinavia. La traducción más cercana de Blatand sería “Blue Tooth”. Los orígenes de este nombre son inciertos pero aunque era común en esa época que algunos reyes usaran nombres distintivos (algunos relatos dicen que el nombre se atribuye a la tez oscura de Harold otros más dicen que el rey Harold era conocido por sus dientes color azul debido a su afición a los arándanos, probablemente todo esto sea folklore). Para una tecnología que tiene sus orígenes en Escandinavia, parece apropiado que los fundadores del SIG hayan optado por este nombre ya que *Bluetooth* tiene la intención de unificar compañías multinacionales en un estándar de comunicación comparado con la manera en la que el rey Harold unificó los reinos. Así es como se otorgó este nombre, que en un inicio era un nombre no oficial, pero al día de hoy ya es una marca registrada, en la siguiente figura se muestra el logotipo oficial, el cual está basado en las iniciales HB, que corresponden al nombre del rey Harold. [8]



Figura 3. El logotipo de *Bluetooth* que fue desarrollado por el *Special Interest Group* SIG, se puede observar que el diseño está basado en las iniciales *Harold Blatand*

## Tecnología Bluetooth

### Redes de Área Personal PAN<sup>i</sup>

En la mayoría de los hogares se cuenta con numerosos aparatos electrónicos, computadora, laptop, impresora, modem, periféricos, etc. Estos dispositivos están interconectados a través de cables con el fin de usar un servicio (imprimir) o compartir información (transferir un archivo de una computadora a otra), todos estos dispositivos forman una *PAN*. Una PAN es una red de uso específico, debido a que la topología y el número de nodos no son los mismos en cada momento, cambian dinámicamente. La mayoría de los problemas asociados con la administración de estas redes pueden solucionarse si se utiliza radiofrecuencia para la comunicación y así obtener información de las características de los dispositivos y su función. Podemos ver el uso y necesidad de PAN en oficinas, hogares e incluso en los automóviles.

### Panorama de Bluetooth

Bluetooth es una tecnología de bajo costo, baja potencia, corto alcance y que usa radiofrecuencia para la formación de *redes de área personal PAN*.

Bluetooth está diseñado para sustituir las conexiones con cables además la mayoría de dispositivos con *Bluetooth* usan baterías en consecuencia el uso de radiofrecuencia consume poca energía debido a la baja potencia del dispositivo.

Los dispositivos *Bluetooth* pueden ser hechos por distintos fabricantes debido a que es un estándar abierto, por ende dispositivos de distintos fabricantes pueden realizar intercambio de datos sin ningún inconveniente. Por esto Bluetooth se ha convertido en una de las más populares tecnologías para la formación de *redes inalámbricas de área personal*.

### Aplicaciones

La telefonía móvil es uno de los principales mercados para esta tecnología. Los *transceptores Bluetooth* son fácilmente añadidos en teléfonos celulares. Como se mencionó anteriormente se usa, para sustituir el uso cables en el intercambio de datos entre periféricos (ratón, teclado, impresoras, audífonos, etc.), cámaras digitales, pagos en banco y sistemas de automóviles (para la lectura de datos obtenidos por algunos sensores). A cada una de estas aplicaciones le llamaremos *perfiles*. El *perfil* da detalles acerca de los parámetros para el intercambio de datos y la configuración del dispositivo para que se establezca la comunicación de una manera uniforme.

---

<sup>i</sup> El término *PAN* hace referencia a las siglas en inglés de *Personal Area Network*, se utilizara esta abreviación de aquí en adelante para hacer referencia a las *Redes de Área Personal*

## Comunicaciones Inalámbricas basadas en Radiofrecuencia

La comunicación basada en radiofrecuencia emplea transmisores y receptores sintonizados para producir y usar, respectivamente ondas de radio en una frecuencia dada.

La potencia del transmisor y la sensibilidad del receptor ayudan a determinar la distancia sobre la cual se pueden comunicar. Transmisores de alta potencia son usados para comunicaciones de largo alcance como la señal de televisión mientras que las comunicaciones de corto alcance requieren menos potencia, y por ende están diseñadas para comunicarse a unos metros de distancia, los cuales pueden ser alimentados por la batería de cualquier dispositivo móvil.

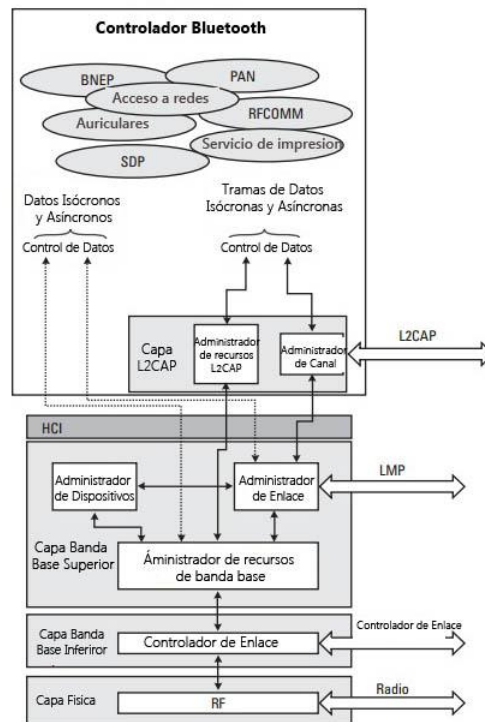


Figura 4. Diagrama que muestra la división en capas del protocolo Bluetooth, obsérvese que la Interfaz de Controlador de Host sirve como intermediario entre las capas físicas y las capas lógicas

Otra característica que es relevante, es la habilidad de las ondas de radio para penetrar distintos objetos. Algunos obstáculos reflejan las ondas de luz usadas en tecnologías como *infrarrojo*, pero otras ondas como las de radiofrecuencia en general pueden (con cierta limitación) penetrar varios obstáculos (aunque en algunos casos las ondas de radio pueden difractar o ser desviadas por objetos). Por lo tanto, las ondas de radiofrecuencia pueden traspasar muchos

## Tecnología Bluetooth

obstáculos como ropa, cuerpos, paredes, puertas, etc. Esto significa que no es necesario alinear el receptor y el transmisor como en el caso del *infrarrojo*.

La señal de radio en FM, por ejemplo, opera en un espectro de frecuencias que van desde los 88 megahertz (MHz) hasta los 108 MHz; algunos teléfonos inalámbricos operan en el espectro de frecuencias de los 900 MHz, el *Bluetooth* y otras tecnologías inalámbricas operan en los 2.4 gigahertz (GHz). Debido a que el espacio de frecuencias es finito, la mayoría de los gobiernos regulan su uso, otorgando concesiones para transmitir a esas frecuencias a determinados niveles de potencia. Algunas frecuencias están reservadas para el uso sin licencia bajo ciertas condiciones, por ejemplo los 900 MHz y los 2.4 GHz no requieren de licencias para su uso en casi cualquier parte del mundo.

### Pila de Protocolos

Bluetooth maneja una implementación de capas que se muestra en la figura 4. Hasta el fondo podemos observar la *capa física*, aquí es donde las señales de radio son procesadas. El alcance de la señal, la tolerancia al ruido y los filtros están implementados en esta capa.

Arriba de la *capa física* se encuentra la *banda base*, la cual está dividida en parte baja y parte alta. Es en esta donde a los paquetes de datos se les da un formato: cabeceras, campo de detección de errores, retransmisión de paquetes y opcionalmente, encriptación y descifrado. El *controlador de enlace* es la entidad que implemente los protocolos y procedimientos de la *banda base*. Las conexiones *Bluetooth* son administradas por el *administrador de enlace*. Los dispositivos configuran, negocian y administran las conexiones usando el *protocolo de administración de enlace LMP*<sup>ii</sup>. Cantidades grandes de datos son divididas en pequeñas unidades de información para ser transmitidas sobre enlaces Bluetooth, esta es la responsabilidad del *protocolo de adaptación y comunicación de enlace lógico (L2CAP)*<sup>iii</sup>.

En la mayoría de los casos, la funcionalidad de Bluetooth está integrada en un anfitrión que tiene capacidad de computo pero carece de un transmisor de radio. Por este motivo, los *transceptores Bluetooth* solo manejan las capas inferiores.

Por otro lado, una laptop por ejemplo es capaz de manejar las capas altas de la pila de protocolos, se le puede adicionar un *transceptor Bluetooth* para así tener acceso a enviar ondas de radio, usar la *banda base* y L2CAP.

---

<sup>ii</sup> LMP corresponden a las siglas en inglés *Link Manager Protocol*

<sup>iii</sup> L2CAP O LLCAP corresponde a las siglas en inglés de *Logical Link Communication and Adaptive Protocol*

Para realizar la comunicación entre las capas altas y bajas, se ha diseñado la interfaz controladora de host (HCI). [9]

## Capa Física

### Alcance de la Comunicación

Los dispositivos Bluetooth están divididos en tres clases, la principal diferencia radica en los niveles de potencia usados para la transmisión de la señal de radio. La siguiente tabla hace un resumen de estas diferencias. Casi todos los dispositivos con Bluetooth como teléfonos móviles, audífonos, laptop son dispositivos de clase dos. Dispositivos de clase uno también están disponibles para los consumidores, los dispositivos de clase tres rara vez son usados.

Clase	Potencia (mW)	Alcance de la señal sugerido
1	100	100 m
2	2.5	10 m
3	1	1 m

Tabla 2. Se muestra el alcance de la señal sugerido para cada una de las clases

Los alcances de la señal dados en la tabla 2 son solamente aproximaciones usadas para fines informativos, el fabricante garantiza este alcance bajo ciertas condiciones. En la práctica, el alcance de la señal puede variar significativamente en base al ambiente y las condiciones de uso. Paredes, edificios, arboles, muebles y otras estructuras pueden afectar la intensidad de la señal. Los hornos de microondas en particular, pueden hacer inservibles algunos *dispositivos Bluetooth*. Además las personas como tal, interfieren en la *señal Bluetooth*, ya que el agua (que constituye el 60% del cuerpo humano) es muy buena absorbiendo ondas de radio a las frecuencias usadas por Bluetooth.

### Velocidad de la Comunicación

Es algo difícil dar un estimado fiable del ancho de banda de un canal de comunicación *Bluetooth*, pero las cifras aproximadas ayudan. Teóricamente dos dispositivos Bluetooth, tienen una velocidad máxima de datos asimétrica de 723.2 kilobits por segundo (kbps) y una velocidad máxima de datos simétrica de 433.9 kbps. En este contexto, asimétrico significa que solamente un dispositivo está transmitiendo y simétrico que ambos están intercambiando datos uno con otro. En la práctica, la tasa de intercambio son menores, debido a que siempre va haber ruido en los canales de comunicación inalámbrica. Como en todos los métodos de comunicación inalámbrica, la intensidad de la señal de *Bluetooth* se deteriora cuadráticamente en razón de la distancia de la fuente. Las señales débiles son mucho más fáciles de ser

## Tecnología Bluetooth

corrompidas por el ruido, la velocidad máxima de comunicación entre dos dispositivos está muy limitada por la distancia de separación. Para este proyecto aunque lo anteriormente mencionado suene como a desventaja, es algo que se puede sacar provecho cuando la aplicación requiere limitar el área de intercambio de datos como es nuestro caso.

### Radiofrecuencia y saltos de canal

Todos los *dispositivos Bluetooth*, como se ha mencionado antes, operan en el espectro de frecuencias de los 2.4 GHz o mejor conocida como la *–banda industrial, científica y médica (ISM<sup>iv</sup>)*. Esto significa que usan el mismo espectro que *WI-FI* y algunos teléfonos inalámbricos<sup>v</sup>. La diferencia de *Bluetooth* con otras tecnologías es que divide el espectro de frecuencias de la banda ISM en 79 canales y emplea una técnica llamada salto de canales<sup>vi</sup> por lo que los *dispositivos Bluetooth* siempre están cambiando la frecuencia en la que reciben y transmiten datos.

La interferencia en una canal no debería importar mucho, ya que en el salto al siguiente canal no debería de existir esta interferencia.

En comparación, la manera en que *WI-FI<sup>vii</sup>* opera, el espectro de frecuencias de la banda ISM está dividida en 14 canales, cada uno de 5MHz. Cuando una red inalámbrica es configurada, el administrador de red escoge uno de estos canales y los dispositivos en la red siempre transmitirán en la frecuencia de radio para ese canal; si hay muchas redes inalámbricas en la misma área por ejemplo en un edificio de apartamentos donde cada apartamento en el peor de los casos tiene un modem inalámbrico, existen la probabilidad de que algunas de estas redes colisionen unas con otras y el rendimiento puede verse afectado.

Bluetooth, como *WI-FI*, también divide el espectro de frecuencias de la banda ISM en canales, pero es aquí donde la similitud termina. Para empezar, *Bluetooth* usa 79 canales en vez de 14, y estos canales son más pequeños (1 MHz en vez de 5 MHz). La principal diferencia radica en que los *dispositivos Bluetooth* nunca permanecen en el mismo canal. En una comunicación promedio entre *dispositivos Bluetooth*, se hace un cambio de canal cada 625  $\mu$ s (1600 veces por segundo).

---

<sup>iv</sup> Estas siglas corresponden a las palabras en inglés *Industrial, Scientific and Medical*

<sup>v</sup> El termino teléfono inalámbrico en este contexto no hace referencia a los teléfonos móviles o celulares, sino a los teléfonos utilizados para sustituir a los equipos de telefonía fija, que permiten cierta movilidad dentro de una casa u oficina

<sup>vi</sup> La traducción más cercana del termino en inglés *channel hopping*

<sup>vii</sup> Por ejemplo los estándares 802.11b y 802.11g de la IEEE

Los *saltos de canal* se realizan de manera aleatoria por lo que ningún canal es usado más que otro. Cuando dos *dispositivos Bluetooth* se están comunicando uno con otro, ambos deben de sincronizarse para saltar al mismo canal para que siempre estén transmitiendo y recibiendo en la mismas frecuencias. Podemos ver este comportamiento en la siguiente figura.

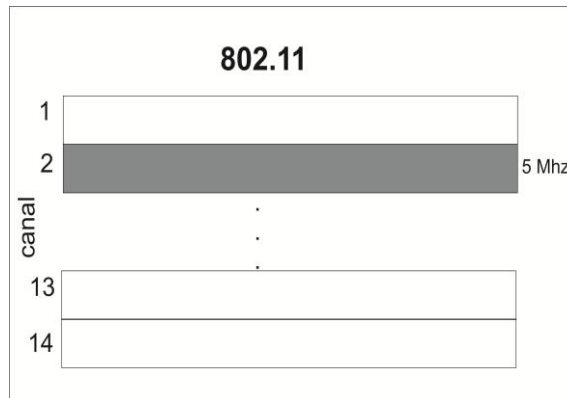


Figura 5. Se muestra en esta figura la manera en que *WiFi* divide el espectro de los 2.4 GHz en 14 canales y la comunicación se concentra en un solo canal

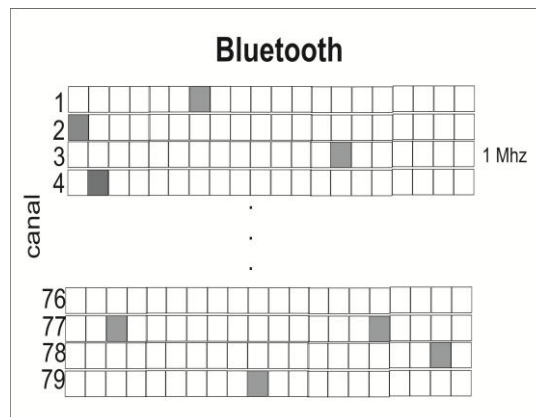


Figura 6. Se muestra la manera en que *Bluetooth*, divide el espectro de los 2.4 GHz en 79 canales y como se van realizando los saltos de un canal a otro

En teoría, todos estos saltos en frecuencia hacen a *Bluetooth* mucho más robusto contra interferencia de fuentes cercanas de ruido, y permite a muchas *redes Bluetooth* coexistir en el mismo lugar. Las últimas versiones de *Bluetooth*, van más allá y usan un esquema *de saltos de frecuencia adaptativo*<sup>viii</sup>, donde los dispositivos tienen la capacidad de evitar canales con ruido o con altos niveles de interferencia (por ejemplo canales que coinciden con dispositivos *WiFi*).

<sup>viii</sup> La traducción más cercana para el termino *adaptive frequency hopping*

## Tecnología Bluetooth

### Banda Base

#### Controlador de Enlace

El controlador de enlace lleva a cabo protocolos de la banda base y otras rutinas de enlace de bajo nivel. La información es intercambiada entre los dispositivos Bluetooth en forma de paquetes, y cada paquete es transmitido en una frecuencia diferente. Por lo regular cada paquete es transmitido en un periodo de tiempo de  $625\mu\text{s}$ , aunque algunos pueden extenderse por 5 periodos de tiempo. Para realizar la comunicación *full dúplex*,<sup>ix</sup> un esquema de *División de Tiempo Dúplex (TDD<sup>x</sup>)* es usado. Normalmente, en los sistemas de comunicación por radio, dos frecuencias diferentes son usadas, una para transmitir y la otra para recibir. Esto es conocido como *División de Frecuencia Dúplex (FDD<sup>xi</sup>)*, en TDD solamente un frecuencia es usada para la recepción y transmisión de datos. Durante un periodo de tiempo un dispositivo puede enviar datos, los cuales son recibidos por el otro dispositivo, en el siguiente dispositivo los papeles se intercambian.

#### Periodo de Tiempo

Antes de entrar a explicar el periodo de tiempo usado en la transmisión de datos, téngase en cuenta al *maestro* como el dispositivo Bluetooth que controla la comunicación y al *esclavo* como el dispositivo que está a la espera de conexiones para realizar un intercambio de información. Una vez que se haya enablado una conexión ambos pueden recibir y transmitir datos mientras el maestro no corte la conexión. Se profundizara en este tema en líneas siguientes.

Cada periodo de tiempo tiene una duración de  $625\ \mu\text{s}$ . El *maestro* inicia la comunicación en periodos de tiempo pares, enviando un paquete con su dirección al *esclavo*. El *esclavo* puede enviar paquetes una vez enablada la conexión, en periodos de tiempo impares. Un paquete por lo general ocupa un periodo de tiempo pero puede extenderse hasta cinco. Si un paquete se extiende más de un periodo de tiempo, la frecuencia de salto siempre será la misma para todo el paquete.

La transmisión de un paquete es mostrada en la siguiente figura. En esta figura el *maestro* transmite en el periodo de tiempo 0 y lo hace a través de la

---

<sup>ix</sup> Se entiende a la manera en que el transmisor y receptor intercambian datos, en una comunicación full dúplex en teoría se puede recibir y transmitir datos en el mismo instante de tiempo

<sup>x</sup> Siglas que corresponden en inglés a *Time Division Duplex*

<sup>xi</sup> Siglas que corresponden en inglés a *Frequency Division Duplex*

frecuencia  $f_1$ . Después el *esclavo* transmite en el periodo de tiempo 1 y lo hace a través de la frecuencia  $f_2$ , y así sucesivamente,

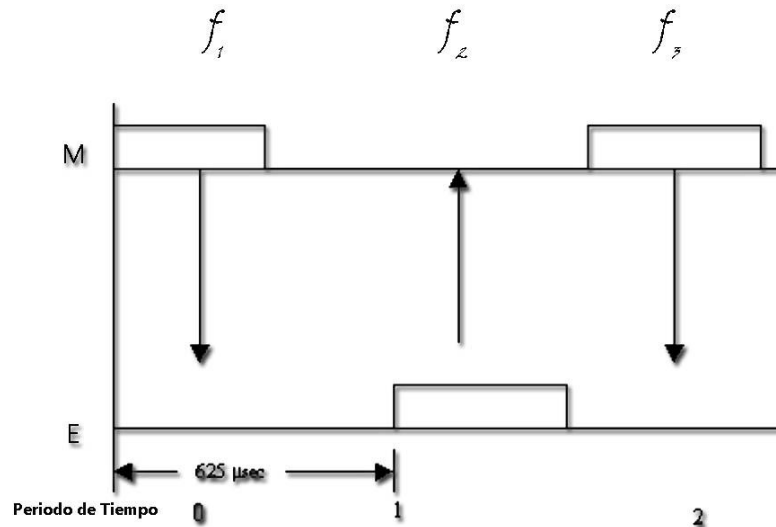


Figura 7. Se ilustra cómo se realiza el intercambio de información entre un maestro y un esclavo a través de División de Tiempo Dúplex

## Estados de un dispositivo Bluetooth

Un dispositivo Bluetooth puede estar en uno de dos principales estados: conectado y en espera. Cuando un dispositivo se encuentra conectado puede comunicarse con otros dispositivos intercambiando paquetes. Cuando un dispositivo está en espera como su nombre lo indica está a la espera de conexiones con otros dispositivos y permanece en un modo de ahorro de energía. Este es el estado por default. Además de estos dos estados hay siete sub-estados: *página*, *búsqueda de página*, *petición*, *búsqueda de peticiones*, *respuesta de maestro*, *respuesta de esclavo* y *respuesta de petición*<sup>xii</sup>.

Para iniciar la comunicación, un dispositivo Bluetooth puede entrar al estado de *petición*, para buscar otros dispositivos próximos a él. Para responder a una *petición*, el dispositivo deberá entrar de manera periódica en el estado *búsqueda de peticiones*. Cuando la petición es exitosa, el dispositivo entra en el *estado de respuesta de peticiones*. Cuando un dispositivo desea conectarse a otro, debe de estar en el *estado de página* para tratar de entablar comunicación con otro dispositivo. En este estado el dispositivo pasa a ser un *maestro* y puede conectarse a otros dispositivos. Cuando el *maestro* haya entablado comunicación con el otro dispositivo, el *maestro* entra en el estado de *respuesta de maestro*. Para el caso de un dispositivo *esclavo*, este tiene que estar

<sup>xii</sup> Estos términos corresponden a la traducción más cercana para *page*, *page scan*, *inquiry*, *inquiry scan*, *master response*, *slave response*, *inquiry response*

## Tecnología Bluetooth

periódicamente entrando al estado *búsqueda de página*, inmediatamente después de que el *esclavo* recibe un *paquete de búsqueda de página*, pasa al estado *respuesta de esclavo*.

Después de que la paginación de dispositivos se ha realizado, el *maestro* y el *esclavo* entran en un *estado activo*, durante el cual es realizado el intercambio de información. La conexión también puede ser puesta en uno de los siguientes tres modos: *retenido*, *específico* y *detenido*<sup>xiii</sup>. En el modo retenido, el dispositivo permanecerá sin recibir tráfico de datos por un lapso de tiempo específico, para que otros dispositivos puedan hacer uso del canal de comunicaciones, después que expire el lapso de tiempo el dispositivo volverá a recibir datos de nuevo. En modo *específico*, se indica al dispositivo que paquetes debe leer y cuales ignorar, todo en base a una condición previamente especificada. La conexión puede estar en *modo detenido* y el dispositivo solamente lee una señal proveniente del maestro, la cual sirve para mantenerse sincronizados pero no hay intercambio de datos.

El procedimiento para realizar la comunicación entre ambos dispositivos Bluetooth es el siguiente.

1. El dispositivo envía una petición
2. Otro dispositivo que está en búsqueda de peticiones, responde a la petición anterior enviando un paquete. Este paquete contiene la información necesaria para realizar la conexión.
3. El dispositivo que envió la petición solicita la conexión con el dispositivo que respondió a dicha petición.
4. La paginación o emparejamiento es usada para iniciar la conexión con el dispositivo seleccionado
5. El dispositivo seleccionado, el cual ha entrado en el estado *búsqueda de página*, responde al emparejamiento
6. Una vez realizada la conexión, el *maestro* se sincroniza con el *esclavo* para realizar los saltos en frecuencia al mismo tiempo. [10]

### Roles de Maestro y Esclavo

A nivel de la *banda base*, cuando dos dispositivos establecen un enlace *Bluetooth*, uno actúa en el rol de *maestro* y el otro en el rol de *esclavo*. La especificación permite que cualquier *transceptor Bluetooth* funcione en ambos roles, como *maestro* para un enlace de comunicación y como *esclavo* para otro.

La función principal de un *maestro* es generar un patrón para realizar los saltos en frecuencia. Todos los esclavos comunicados con un maestro, tienen que realizar dichos saltos al unísono con el maestro.

---

<sup>xiii</sup> Se usaron estos términos en que corresponden a las palabras en inglés *hold*, *sniff* y *park*, que su traducción literal al castellano sería *sujetar*, *olfatear* y *estacionar*

Un maestro puede comunicarse con múltiples esclavos, siete en *estado activo* y con 255 en *modo detenido*. Todos los esclavos comunicándose con un solo maestro forman lo que la especificación llama *piconet*. Solo puede haber un *maestro* en una sola *piconet*. La relación *maestro – esclavo* es necesaria a bajo nivel en la comunicación Bluetooth. Cuando dos dispositivos establecen una conexión, el rol que cada uno asume es a menudo irrelevante para los protocolos de alto nivel o el usuario final del dispositivo. En algunos escenarios como es este proyecto puede ser una ventaja o incluso necesario indicarle a un dispositivo un rol particular a asumir. La siguiente figura muestra a un *esclavo* y *maestro* dentro de una *piconet*. [11]

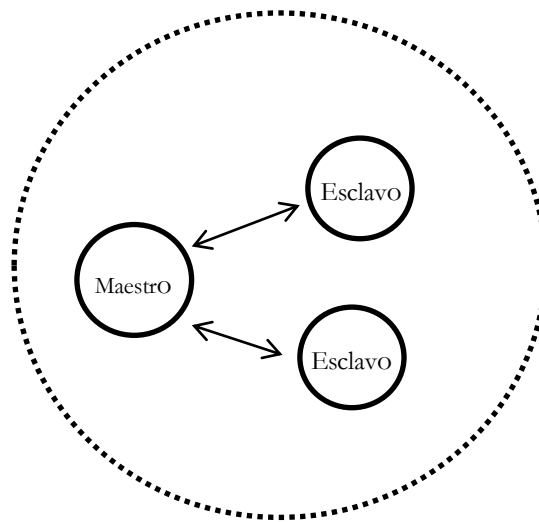


Figura 8. *Piconet* formado por tres dispositivos Bluetooth, uno actúa como *maestro* mientras que los otros dos sirven como *esclavos*

## Direccionamiento Bluetooth

Todos los *transceptores Bluetooth* tienen una dirección única que consta de 48 bits. Dividida en *Parte Baja de la Dirección LAP*<sup>xiv</sup> con 24 bits, *Parte Alta de la Dirección UAP*<sup>xv</sup> con 8 bits, *Parte No Significativa de la Dirección NAP*<sup>xvi</sup> formada de 16 bits.

El formato de la dirección se muestra en la figura 9. Esta dirección es asignada al *transceptor Bluetooth* por el fabricante y consiste del identificador de la compañía y un número asignado a la compañía. En la mayoría de los textos relacionados con Bluetooth, la dirección es referida como BT\_ADDR.

<sup>xiv</sup> Siglas en inglés para el termino *Lower Address Part*

<sup>xv</sup> Siglas en inglés para el termino *Upper Address Part*

<sup>xvi</sup> Siglas en inglés para el termino *No Significant Address Part*

## Tecnología Bluetooth

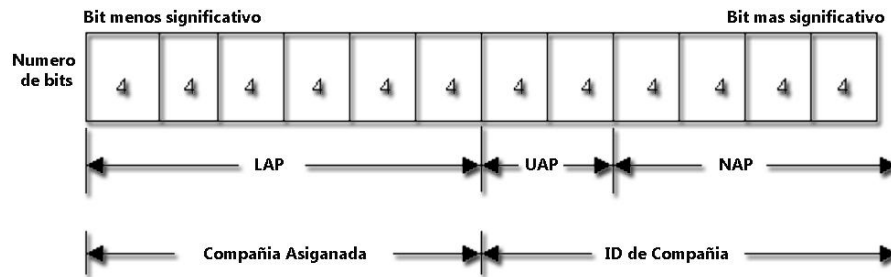


Figura 9. Ejemplo de la distribución de la dirección Bluetooth en cada uno de los campos que la conforman, en esta figura cada bloque de la dirección es igual a cuatro bits para formar los 48 bits de la dirección

Además de BT\_ADDR, tenemos otras dos direcciones que son significativas para el funcionamiento de Bluetooth.

LT\_ADDR: la dirección que corresponde a los miembros activos de una piconet. Está conformada por tres bits y es la razón por la cual solamente pueden existir ocho<sup>xvii</sup> nodos en una piconet (contando al maestro).

PM\_ADDR: es una dirección reservada para los miembros no activos de la piconet. Esta dirección está formada de ocho bits y corresponde a los esclavos que están en *modo detenido*, a lo más pueden 255 direcciones por piconet.

## Protocolos de Alto Nivel

Después de haber visto la manera en que Bluetooth organiza la comunicación a bajo nivel, se va a explicar los protocolos de alto nivel o de transporte más usados en aplicaciones con Bluetooth.

Diferentes aplicaciones tienen diferentes necesidades, por lo tanto diferentes protocolos de transporte.

## RFCOMM

El protocolo de *Comunicación de Radio Frecuencia* (RFCOMM<sup>xviii</sup>), es un protocolo seguro basado en comunicación de flujo de datos. La especificación de Bluetooth indica que fue diseñado con la intención de emular las capacidades del protocolo de comunicación serial RS-232 (para hacer más fácil a los fabricantes agregar funciones de Bluetooth a sus existentes puertos seriales).

<sup>xvii</sup> Son ocho debido a que  $2^3 = 8$ , direcciones que se pueden formar con tres bits

<sup>xviii</sup> RFCOMM corresponde a *Radio Frequency Communications*

En otras palabras RFCOMM es un protocolo de transporte de propósito general que trabaja bien emulando puertos seriales. En el manejo de puertos, RFCOMM solamente usa treinta.

### L2CAP

El *Protocolo de Adaptación y Control de Enlace Lógico (L2CAP)*, es un protocolo basado en paquetes que puede ser configurado en varios niveles de seguridad. El tamaño default de cada paquete es de 672 bytes, pero el tamaño puede ser negociado a lo máximo a 65,535 bytes después que la conexión es establecida.

La seguridad es implementada con el esquema de transmisión – reconocimiento, en el cual paquetes que no llegaron al receptor pueden ser o no retransmitidos. Hay solo tres posibles políticas para la retransmisión de paquetes:

- Nunca retransmitir (pero se puede obtener el mejor rendimiento)
- Retransmitir siempre mientras la conexión sea exitoso o se rompa la comunicación (enfoque más seguro, configurado por default)
- Descartar al paquete y moverlo a la cola si el paquete no ha sido reconocido después de un periodo de tiempo límite especificado (0-1279 ms). Este enfoque es útil cuando los datos deben de ser transmitidos en tiempo y forma.

Estas políticas con frecuencia parecen útiles a primera vista, pero hay algunas consideraciones a tomar en cuenta. En primer lugar, si se modifica solamente una conexión L2CAP a un dispositivo remoto, por ende todas las conexiones L2CAP hacia dicho dispositivo se modificarían, conexiones a otros dispositivos no son afectadas. Segundo, L2CAP sirve como protocolo de transporte de RFCOMM, por lo que todas las conexiones RFCOMM se encuentran encapsuladas en una conexión L2CAP. Este hecho, combinado con lo anteriormente mencionado, significa que al cambiar deliberadamente las condiciones para los enlaces L2CAP, estaríamos haciendo lo mismo para los enlaces RFCOMM.

### ACL<sup>xix</sup>

El protocolo de transporte Conexión Lógica Orientada (ACL) no está disponible a nivel de usuario por lo que rara vez se usa de manera directa, pero es importante mencionarlo ya que todas las conexiones L2CAP se encuentran encapsuladas en dentro de una conexión ACL. Por lo que, las conexiones RFCOMM también están encapsuladas dentro de conexiones

---

<sup>xix</sup> Siglas que corresponden a *Asynchronous Connection-oriented Logical*

## Tecnología Bluetooth

ACL. Dos dispositivos Bluetooth a lo más pueden tener una conexión ACL la cual es usada para transportar paquetes de L2CAP y RFCOMM.

### SCO<sup>xx</sup>

El último protocolo de transporte que se mencionara es Conexión Orientada Síncrona. Este protocolo es el que mejor rendimiento tiene y es exclusivamente usado para la transmisión audio con calidad para voz, no cualquier tipo de audio, sólo a una velocidad de 64 kb/s.

No es apto para transmitir audio con alta calidad porque su tasa de transmisión es muy pequeña, pero es lo suficiente para transmitir llamadas telefónicas. La mayoría de manos libres inalámbricos hacen uso de una conexión SCO. Si se usan audífonos con Bluetooth para escuchar música muy probablemente se esté usando una conexión L2CAP en vez de SCO.

Los paquetes SCO no son seguros y nunca son retransmitidos pero la calidad del servicio está garantizada. Las conexiones SCO siempre están garantizadas para tener una tasa de transmisión de 64 kb/s. Si alguna otra aplicación y conexión están conteniendo para usar la capa física, por ejemplo transmitir un archivo, se le dará prioridad a las conexiones SCO para usar el canal de comunicación. Para garantizar que todas las conexiones SCO tienen esta garantía, un dispositivo Bluetooth a lo más puede tener tres conexiones SCO activas a la vez.

SCO deberá ser siempre usado para transmitir audio con calidad de voz y nunca para ningún tipo de dato.

## Protocolo de Detección de Servicios

El Protocolo de Detección de Servicios (SDP), provee a Bluetooth la capacidad de desarrollar redes de uso específico. Este protocolo es usado para la localización de servicios disponibles o provistos por algún dispositivo Bluetooth. El SDP facilita lo siguiente:

- La habilidad de buscar servicios que se encuentran en la piconet.
- Los servicios a ser descubiertos están basados en clases de servicios (como el servicio de impresión)
- El mecanismo de ver cuando un servicio no está disponible por que el dispositivo está fuera del alcance de las ondas de radio.
- Los detalles del servicio como la clase de servicio y los atributos
- La capacidad de descubrir servicios de otro dispositivo sin la necesidad de consultar un tercero

---

<sup>xx</sup> Siglas que corresponden a *Synchronous Connection-Oriented*

Antes de explicar el funcionamiento de SDP se hace referencia al *cliente* como la aplicación que necesita de cierto servicio y como a *servidor* al prestador de dicho servicio. El funcionamiento del SDP se describe en la siguiente figura. Cuando un dispositivo quiere buscar un servicio, la aplicación de software corriendo en el lado de *cliente* envía una solicitud SDP al *servidor* (el dispositivo que puede proveer dicho servicio). El *cliente* y el *servidor* SDP entonces intercambian mensajes.

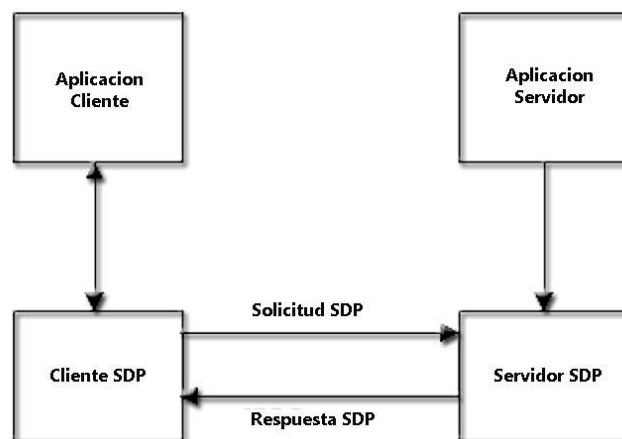


Figura 10. Intercambio de mensajes usando SDP entre dos dispositivos haciendo una analogía entre sistema cliente servidor

El servidor es el que se encarga de mantener una lista de los servicios disponibles. Un número de 32 bits único por cada elemento de la lista identifica a un solo servicio. Cada elemento de esta lista contiene un número de atributos. Los atributos son el identificador de clase de servicio (el tipo de servicio), descripción del protocolo (protocolo usado para este servicio), un nombre del servicio, y la descripción del servicio. Cada atributo consta de dos elementos, el identificador de atributo y el valor del mismo.

Supongamos que una laptop, requiere de un servicio de impresión. La laptop es un cliente en busca de un servicio de impresión dentro de una red Bluetooth. El proceso se ejecutaría de la siguiente forma:

1. El cliente envía una solicitud de búsqueda de servicio especificando el identificador de la clase de servicio de impresión al servidor.
2. El servidor envía una respuesta a la solicitud de búsqueda de servicio indicando que dos servicios están disponibles.
3. El cliente solicita los detalles de alguno de los dos servicios.
4. El servidor envía la respuesta indicando que el servicio de impresión está disponible.

## Tecnología Bluetooth

El SDP es el núcleo de Bluetooth. Ya que permite descubrir la habilidad y detalles de diferentes servicios junto con información necesaria, por ejemplo los protocolos necesarios para realizar el intercambio de datos.

### Perfiles

El estándar Bluetooth no está limitado para usarse en casos o aplicaciones específicas. Sin embargo, para proveer soporte para todo tipo de aplicaciones, el SIG ha desarrollado un conjunto de *perfiles*. Un *perfil* define la descripción de una interfaz de comunicación para un servicio en particular. Los perfiles que definen los procedimientos fundamentales para conexión Bluetooth y los perfiles para distinto tipo de servicios han sido definidos. Un nuevo perfil puede ser creado a partir de los existentes permitiendo reusar perfiles existentes. Esto da lugar a una estructura jerárquica de perfiles que se muestra en la siguiente figura.

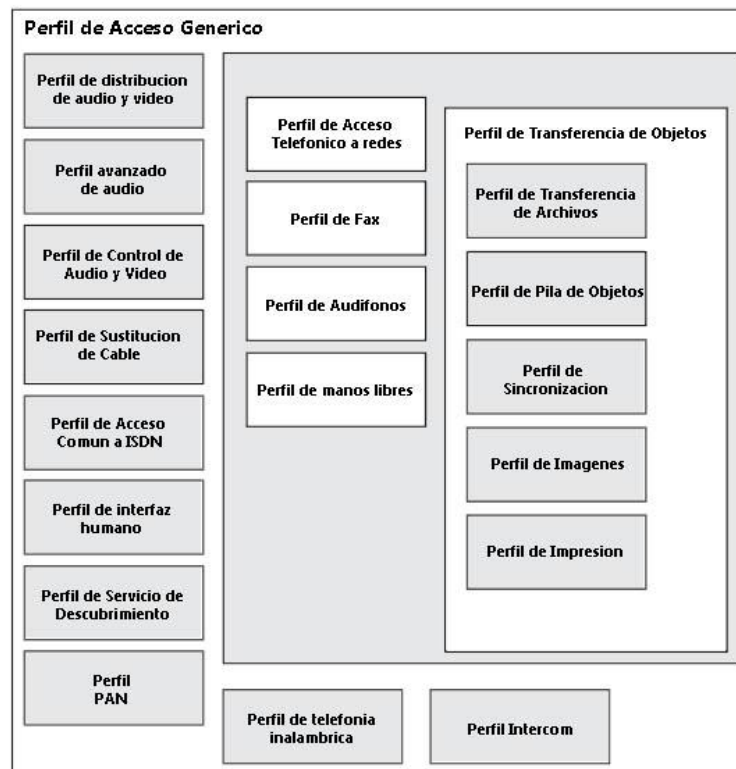


Figura 11. En esta figura se observa la manera jerárquica en la que se organizan los perfiles teniendo como base el perfil de acceso genérico

El objetivo principal de Bluetooth era sustituir los cables. La sustitución de cables es ofrecida por el perfil de puerto serial a través de emular el protocolo RS232. Distintos perfiles, como el de redes de área personal y el perfil de posicionamiento local hacen uso del perfil de puerto serial.

## Topologías de Comunicación

Cuando dos o más dispositivos Bluetooth establecen una conexión, una *red de área personal* se forma. Una *red de área personal* puede ser una *piconet* o una *scatternet*. La siguiente figura muestra los dispositivos dentro de una *piconet*.



Figura 12. Ejemplo de piconet que se puede formar con dispositivos que encontramos en los hogares o en oficinas

En una piconet hay un dispositivo llamada *maestro*, y otros dispositivos llamados *esclavos*.

Una piconet puede soportar dos tipos de comunicaciones, como se muestra en la siguiente figura. En la parte a), la piconet solamente tiene un maestro y un esclavo. En este caso solamente estos dos dispositivos están envueltos en la comunicación, por lo que es una comunicación punto a punto. En la parte b) la piconet tiene un maestro y más de un esclavo. En este caso, el maestro puede comunicarse con todos, por lo que es una comunicación de un punto a varios puntos, por lo que todos los dispositivos comparten un mismo canal. Definiéndolo de manera formal, una piconet es una red en la cual dos o más dispositivos comparten el mismo canal.

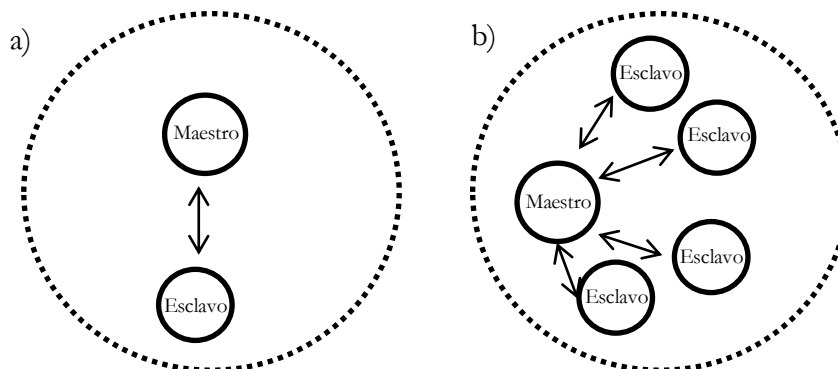


Figura 13. Se muestra en la parte a) una piconet formada por dos dispositivos en un enfoque punto a punto, en la parte b) una con enfoque punto a varios puntos

Para comunicarse el maestro y el esclavo deben de estar sincronizados en la misma frecuencia y tiempo. El esclavo deberá saber a qué frecuencia transmite el maestro y deberá sintonizar dicha frecuencia. En una piconet se permiten a lo más siete

## Tecnología Bluetooth

dispositivos que puede permanecer como *activos*, y varios más pueden estar en modo *detenido*. Los esclavos en modo *detenido* no se encuentran activos en el canal de comunicación, mas sin embargo se mantienen sincronizados con el maestro. El maestro es quien controla el acceso al canal. Un esclavo y maestro pueden intercambiar roles.

Una *scatternet* se forma por la superposición de la cobertura de varias *piconets*, como se muestra en la siguiente figura. Cada *piconet* tiene un maestro, pero el maestro en una *piconet* puede ser el esclavo en otra. Cada *piconet* tiene su propia secuencia de saltos de frecuencia, por lo que no hay interferencia entre dos o más *piconets*. [12]



Figura 14. Una scatternet se puede formar por la superposición de dos o más piconets

## Bluetooth comparado contra otras tecnologías

### Bluetooth e Infrarrojo

Un predecesor al Bluetooth es el infrarrojo. Con esta tecnología varios dispositivos como PDA y teléfonos celulares realizaban intercambio de datos. Una desventaja del infrarrojo es que los dispositivos involucrados debían estar a lo más treinta centímetros de distancia y lo más importante, cada transceptor de infrarrojo debían verse uno a otro. Si ninguna de las condiciones anteriores no se respeta, la transmisión se perdía. Bluetooth soluciona el primer limitante pues puede alcanzar los 10 metros. También soluciona el segundo impedimento, porque trabaja con ondas de radio y la transmisión es omnidireccional.

### Bluetooth y 802.11

Bluetooth y 802.11 fueron creados para atender distintas necesidades pero ambos operan en la banda ISM.

El principal objetivo de 802.11 son dispositivos separados por una distancia larga y que cuentan con la potencia necesaria para transmitir a velocidades altas. Comúnmente se usa para conectar, por ejemplo dos laptops a una distancia de cien metros a una velocidad de 11Mbps. Esta tecnología es muy conveniente para administradores de redes de área local donde es muy caro o inconveniente instalar cables.

Por otro lado, la intención de Bluetooth es conectar dispositivos pequeños como teléfonos celulares con un alcance de diez metros a una velocidad de 1 Mbps. El uso de una baja tasa de transferencia y un alcance corto, lo convierten en una tecnología inalámbrica de baja potencia. Comparado con dispositivos 802.11, algunos dispositivos Bluetooth pueden consumir 500 veces menos energía, lo que hace una gran diferencia en el tiempo de uso de dispositivos que usan baterías.

Bluetooth nunca sustituirá a 802.11, porque no fue diseñado para hacer las siguientes tareas:

- Transferencia de archivos grandes
- Comunicación a largas distancias (solamente dispositivos de clase uno pueden alcanzar los cien metros)

802.11 nunca sustituirá a Bluetooth, porque no fue diseñado para:

- Comunicarse con periféricos
- Requiere mucha potencia para la comunicación
- En un principio no fue diseñado para la transmisión de voz[13]

## CONCLUSIONES

Para finalizar este capítulo, podemos decir que Bluetooth es una tecnología de radiofrecuencia de corto alcance y baja potencia, permitiendo alcanzar cobertura de señal de 10 a 100 metros. Esta característica es importante para la solución del problema expuesto en esta tesis, ya que se quiere limitar el área sobre la cual buscaremos smartphones. Además al realizar la búsqueda mediante señales de radio, se realizaría de manera no intrusiva, solo sería necesario que los usuarios prendieran el Bluetooth que viene integrado con el smartphone.

Bluetooth ofrece lo que son perfiles, en este proyecto se hará uso del perfil de Puerto Serial, que se usará para comunicar el *transceptor Bluetooth* con un *microcontrolador*.



---

# MODELADO DEL SISTEMA

Después de haber visto las ventajas de usar Bluetooth y justificar de sobremanera su uso, en el siguiente capítulo se mostrara como se diseñó la solución para este proyecto de investigación. Para este propósito se hará uso de UML, que es el Lenguaje Unificado de Modelado. UML cuenta con herramientas como diagramas de caso de uso, que nos mostraran las funciones de cada uno de los actores implicados en la toma de asistencia, diagramas de secuencia para ver los posibles escenarios involucrados, diagrama entidad relación para definir los requerimientos de la base de datos.

Antes de indicar la manera en que se implementó el sistema es necesario entrar en la parte de diseño, esto mostrara por que se eligió implementar de este modo el sistema. Como se mencionó anteriormente se persigue la realización de un sistema que se encargue de pasar asistencia a un grupo de personas dentro de un área laboral.

En base a este objetivo principal y lo definido anteriormente en el capítulo uno, empezaremos a ver los factores involucrados para hacer un mejor análisis del problema.

UML es un lenguaje para la especificación, visualización, construcción y documentación, de los factores involucrados en el desarrollo de un sistema. Como lenguaje UML tiene la intención de comunicar el comportamiento de un sistema discutiendo su entorno y su contexto. En cuanto a modelos, UML

se enfoca en el sistema, vía la formulación de un modelo. Un modelo captura un conjunto de ideas o abstracciones sobre el sistema, incluida toda la información necesaria para su comprensión y excluye cualquier información irrelevante o incidental que pueda obstaculizar nuestra comprensión.

Usando modelos, podemos gestionar mejor la complejidad de trabajar con sistemas grandes. En cuanto a las especificaciones, podemos usar UML para “lo” que se requiere de un sistema y el “como” un sistema satisfará sus requisitos. UML también hace uso del lenguaje visual valiéndose de que “una imagen vale más que mil palabras”, esto a través de diagramas y esquemas que ilustran las necesidades de diseño del sistema. [14]

### Perspectiva del Sistema

El proyecto consiste en desarrollar una serie de módulos equipados con comunicación *Ethernet* y *Bluetooth*. Cada uno de estos módulos estará distribuido por todo un edificio. Una vez colocados todos los módulos, se debe proseguir a tomar lectura de los *smartphones* dentro del área de cobertura de cada módulo. Hasta este punto podemos dividir el proyecto en la parte *lógica* (que incluye las aplicaciones de software necesarias para que funcione el sistema) y la parte *física* (formada de tarjetas controladores para comunicación *Bluetooth - Ethernet*). En la figura 15 se presenta un acercamiento del sistema final, compuesto de una red de módulos Bluetooth, servidor, base de datos y pagina web.

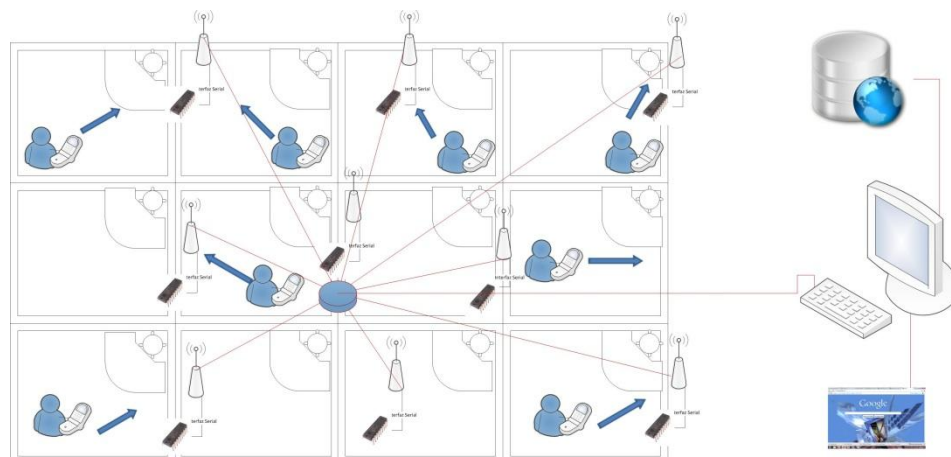


Figura 15. Solución propuesta para la solución de pase de asistencia, consta de una red de módulos equipados con Bluetooth, una base de datos y una página web

*Módulos Bluetooth Ethernet:* estos módulos son el corazón de este proyecto, integran un transceptor Bluetooth para la detección de smartphones y un transceptor Ethernet para enviar información al servidor. Al montar esta red

## Modelado del Sistema

de elementos vemos que todos los *módulos Bluetooth – Ethernet*, forman un sistema distribuido.

*Servidor*: es el encargado de sincronizarse y solicitar la información a los *módulos Bluetooth - Ethernet*. También lleva control de la base de datos y la página web. También es necesario implementar una aplicación para la configuración de la red que se formara con los módulos *Bluetooth - Ethernet*, para asignar horarios y smartphones a verificar.

*Base de Datos*: es necesario implementar una base de datos de los smartphones para guardar la información de monitoreo, indicando la fecha en que se realizó y la hora. Todo con el fin de tener datos para propósitos estadísticos, por ejemplo ver qué persona es la que llega más tarde o falta mucho.

*Página Web*: esta aplicación es necesaria para mostrar en tiempo real<sup>xxi</sup>, el monitoreo realizado a los smartphones. Esta aplicación solo es de consulta, debido a que el usuario final solo podrá ver información mas no realizar cambios sobre esta.

## Capacidades Principales del Sistema

Los módulos *Bluetooth - Ethernet* contarán con las siguientes características:

- Transceptor Bluetooth para la búsqueda de dispositivos Bluetooth cercanos
- Transceptor Ethernet para comunicación al servidor
- Microcontrolador que controlara ambos transceptores

El servidor contará con las siguientes características:

- Dar de alta personas y smartphones a monitorear
- Administrar edificio y salas donde se instalaran los módulos *Bluetooth - Ethernet*
- Asignar direcciones IP para la comunicación con los módulos
- Administrar los usuarios que tendrán derecho a consultar la información en la página web

La base de datos contará con las siguientes características:

- Almacenar la información de las personas y smartphones a monitorear

---

<sup>xxi</sup> El termino tiempo real muchas veces es muy ambiguo dependiendo de la finalidad del problema, para tareas que son críticas el margen en el cual se toma la información y da respuesta el sistema debe de ser lo más corto posible, en otros más si se llega a perder información no es un gran problema mientras el sistema siga funcionando

- Almacenar información del monitoreo
- Crear y almacenar vistas para facilitar el acceso a la información

La página web contará con:

- Se podrá visualizar en tiempo real el monitoreo obtenido por los módulos

### Suposiciones y Dependencias

Algo muy importante que podría entorpecer los resultados finales del sistema, es que las personas a monitorear no activen el Bluetooth de sus smartphones. Esto podría dar resultados falsos, debido a que la persona si acude a su área laboral pero el sistema no se está contabilizando su permanencia. Otra suposición es que el poseedor del smartphone siempre permanezca con él, ya que puede dejar el smartphone en el área de trabajo y la persona puede abandonar su área de trabajo sin que se vea afectado el tiempo de permanencia en el mismo.

### Actores

Los actores son entidades que intervienen directamente con el sistema, y realizan funciones como intercambio de información y cambio de variables. En base a este sistema podemos listar los siguientes actores:

- Usuario de Sistema
- Persona a monitorear
- Módulos *Bluetooth – Ethernet*

Se identifica como usuario del sistema, a la persona que puede administrar el sistema, puede modificar, insertar o borrar información del mismo. Puede realizar también consultas desde la página web.

Las personas a monitorear, son los trabajadores que acudirán a un área laboral y serán los portadores de los smartphones.

Los módulos *Bluetooth – Ethernet* son los actuadores que obtendrán la información del sistema

Una vez que se han definido los roles de cada uno de los actores se prosigue a describir cada uno de los casos de uso.

## Modelado del Sistema

### Casos de Uso

Un caso de uso es una clase que representa una especificación de una funcionalidad del sistema en general, que se puede observar y medir, proporcionando una secuencia completa de acciones que un actor lleva a cabo en una interacción con el sistema.

Una instancia de un caso de uso es llamado escenario y representa el uso específico de las funciones del sistema a partir de una secuencia de acciones. Un caso de uso puede tener atributos, operaciones, asociaciones binarias y puede ofrecer interfaces que definen un subconjunto de interacciones.

A continuación en la figura 15 hacemos uso de esta herramienta para ilustrar como los actores interactúan directamente con el sistema.

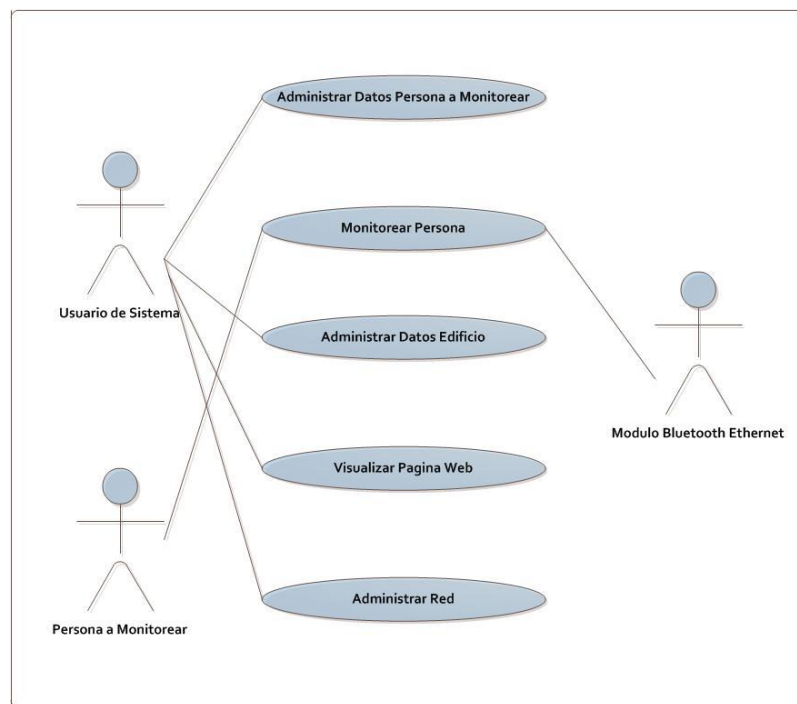


Figura 16. Diagrama de Casos de Uso que muestra los posibles escenarios donde los actores del sistema intervienen

Como se observa, puede haber casos de uso en los que se necesite de la interacción de uno o de más actores y también puede haber una secuencia, donde un caso de uso sea necesario antes que otro. A continuación se hace la descripción detallada de estos casos.

<b>Caso de Uso: Administrar Datos Edificio</b>	
Requisitos:	
Actores: Usuario de Sistema	
Flujo de Datos 1	
1.1	Se solicita a usuario de sistema realizar el alta de los datos de un edificio
1.2	Usuario de Sistema ingresa los datos como nombre de edificio, número, número de salas
1.3	Se procede a ingresar los datos de los salas, como capacidad y nombre
1.4	Se validan y almacenan datos en el sistema
1.5	Fin de Flujo de Datos 1
Flujo de Datos 2	
2.1	Se solicita a usuario de sistema realizar cambios sobre los datos de los edificios donde se instalará el sistema
2.2	Usuario de sistema realiza la búsqueda de los datos requeridos
2.3	Una vez encontrado, se realizan los cambios, posiblemente un cambio de nombre, numero o modificar la nomenclatura asignada a cada una de las salas
2.4	Se validan los datos y se almacenan en el sistema

Tabla 3. Descripción de caso de uso Administrar Datos de Edificio

<b>Caso de Uso: Administrar Datos Persona a Monitorear</b>	
Requisitos: Usuario de Sistema	
Actores: Usuario de Sistema, Persona a Monitorear	
Flujo de Datos 1	
1.1	Persona a monitorear acude con Usuario de Sistema para dar sus datos personales
1.2	Usuario de Sistema ingresa los datos como nombre completo, horario de trabajo, asigna un lugar de trabajo y guarda información del smartphone
1.3	Se validan los datos y se almacenan en el sistema
1.4	Fin de Flujo de Datos 1
Flujo de Datos 2	
2.1	Persona a monitorear, acude para realizar cambios en su información
2.2	Usuario de sistema realiza la búsqueda de los datos requeridos
2.3	Una vez encontrado, se realizan los cambios, posiblemente un cambio de horario o cambio de smartphone
2.4	Se validan los datos y se almacenan en el sistema
Flujo de Datos 3	
3.1	Se solicita a Usuario de sistema dar de baja a una persona a monitorear, posiblemente porque ya no acude a trabajar
3.2	Usuario de Sistema realiza la búsqueda de los datos solicitados

## Modelado del Sistema

Continuación de tabla 4.

<b>Caso de Uso: Administrar Datos Persona a Monitorear</b>
Requisitos: Usuario de Sistema
Actores: Usuario de Sistema, Persona a Monitorear
3.3 Una vez encontrados procede a borrar todos los datos relacionados a la persona a monitorear
3.4 Sistema pide que usuario confirme movimiento
3.5 Cambios se registran en el sistema

Tabla 4. Descripción de caso de uso Administrar Datos de Edificio

<b>Caso de Uso: Administrar Red</b>
Requisitos: Administrar Datos Edificio, Administrar Datos de Persona a monitorear
Actores: Usuario de Sistema , Modulo Bluetooth – Ethernet
Flujo de Datos 1
1.1 Se le asigna a usuario de sistema la tarea de ubicar cada uno de los módulos Bluetooth dentro de las salas de cada edificio
1.2 Una vez instalado cada módulo, se procede a registrar en el sistema a que edificio corresponde y a que sala
1.3 Se asigna una dirección IP irrepitible a cada módulo para la comunicación vía TCP/IP
1.4 Se asignara un horario al módulo que estará monitoreando a smartphones
1.5 Fin de Flujo de Datos 1

Tabla 5. Descripción de caso de uso Administrar Red

<b>Caso de Uso: Monitorear Persona</b>
Requisitos: Administrar Datos Edificio, Administrar Datos de Persona a monitorear, Administrar Red
Actores: Persona a monitorear , Modulo Bluetooth – Ethernet
Flujo de Datos 1
1.1 Persona a monitorear ingresa a la sala donde labora que tiene instalado un módulo Bluetooth Ethernet
1.2 Inmediatamente que ingrese a la sala, la persona a monitorear debe activar el Bluetooth de su smartphone
1.3 El modulo Bluetooth Ethernet recibe una indicación de sistema que le indica que debe de empezar a buscar un smartphone
1.4 Si encuentra a dicho smartphone internamente el modulo guarda un valor 1 en caso contrario guarda un 0.
1.5 El paso 1.4 se repite hasta que el horario de trabajo de la persona a monitorear termine
1.6 Fin de Flujo de Datos 1

Tabla 6. Descripción de caso de uso Monitorear Persona

<b>Caso de Uso: Visualizar Pagina Web</b>	
Requisitos: Administrar Datos Edificio, Administrar Datos de Persona a monitorear, Administrar Red, Monitorear Persona	
Actores: Usuario de Sistema	
Flujo de Datos 1	
1.1	El usuario de sistema quiere visualizar el monitoreo en tiempo real
1.2	Usuario de sistema debe de abrir un navegador web e ingresar una dirección que lo enviara, a la página web de sistema
1.3	Tiene que iniciar sesión y seleccionar el edificio a monitorear
1.4	De manera gráfica se muestran los edificios y el estado de cada módulo Bluetooth
1.5	Usuario puede consultar los demás edificios hasta que decida cerrar el navegador
1.6	Fin de Flujo de Datos 1

Tabla 7. Descripción de caso de uso Visualizar Página Web

Con la descripción de los caso de uso, se delimitan acciones que cada uno de los actores tiene sobre el sistema y se definen bien los roles.

## Diagrama de Secuencia

Los diagramas de secuencia se usan para enfatizar la interacción con el sistema a través de un enfoque temporal, incluyendo roles y su interacción. Los diagramas de secuencia son útiles para la visualización de patrones de interacción y del rendimiento global del comportamiento del sistema.

A continuación, se ilustran los diagramas de secuencia de cada uno de los casos de uso.

1. Diagrama de Secuencia para el caso de uso “Administrar Edificio”

## Modelado del Sistema

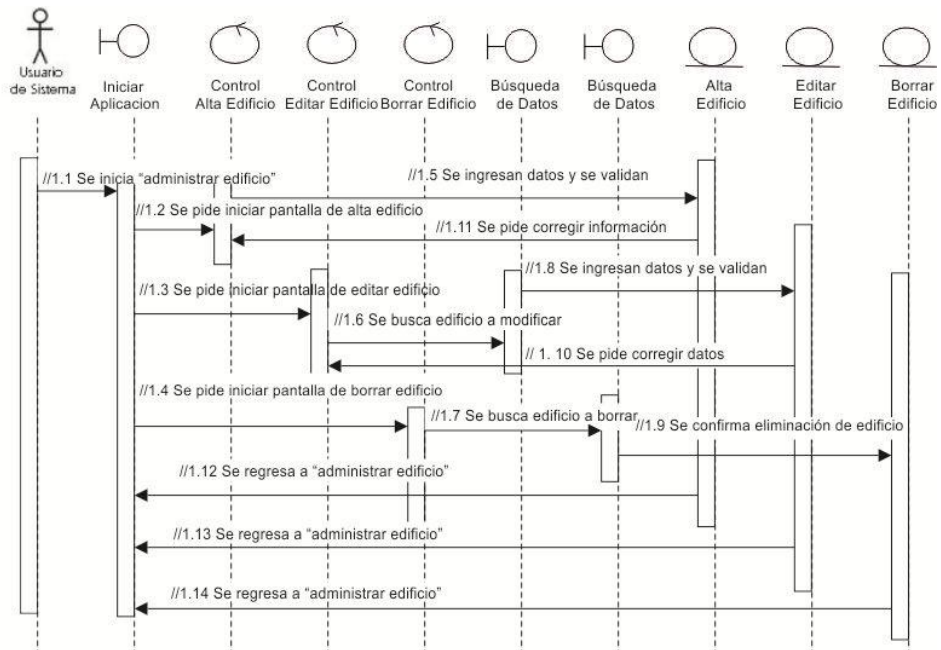


Figura 17. Diagrama de Secuencia de Administrar Edificio, a través de este diagrama podemos entender mejor las alternativas que el usuario del sistema puede elegir

En el diagrama de secuencia anterior muestra la interacción entre cada uno de los módulos involucrados en la alta de un nuevo edificio en el sistema, observando la parte donde se validan los datos; si los datos son correctos se regresa al menú principal, en caso contrario se regresa a alta de edificio para solicitar de nuevo los datos.

## 2. Diagrama de Secuencia para el caso de uso “Administrar Datos de Persona”

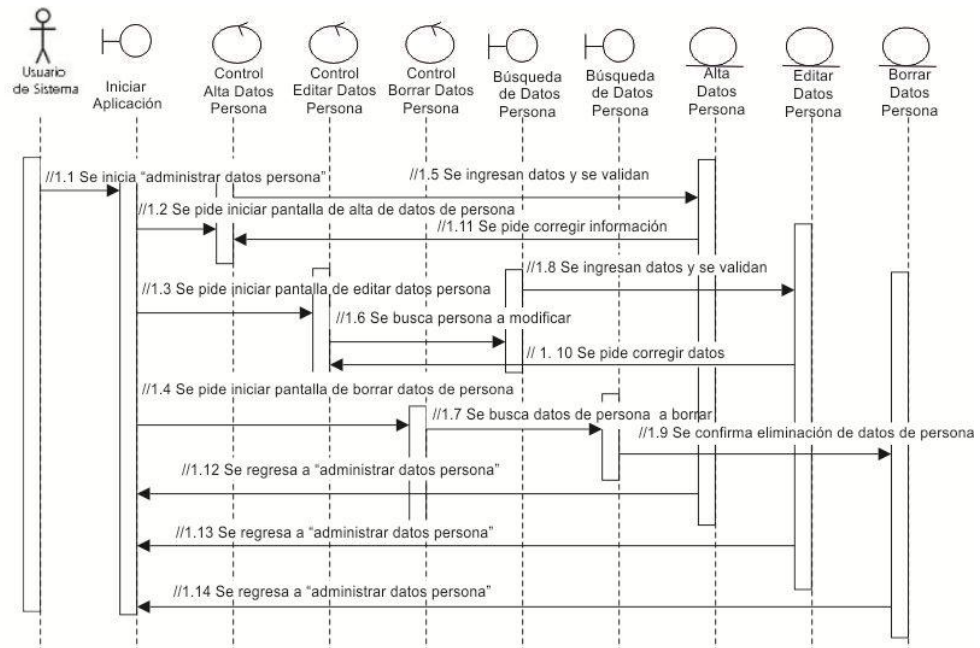


Figura 18. Diagrama de Secuencia de Administrar Datos Personas, en el diagrama se puede observar los pasos necesarios para realizar cada una de las operaciones

En este diagrama de la figura 18 se observa que el sistema debe tener reglas que definan cuando los datos de una persona son válidos o no, por ejemplo dos personas no pueden tener asignado el mismo smartphone. Igual que en el diagrama anterior se ofrecen las alternativas para corregir la información ingresada al sistema.

### 3. Diagrama de Secuencia para el caso de uso “Administrar Red”

## Modelado del Sistema

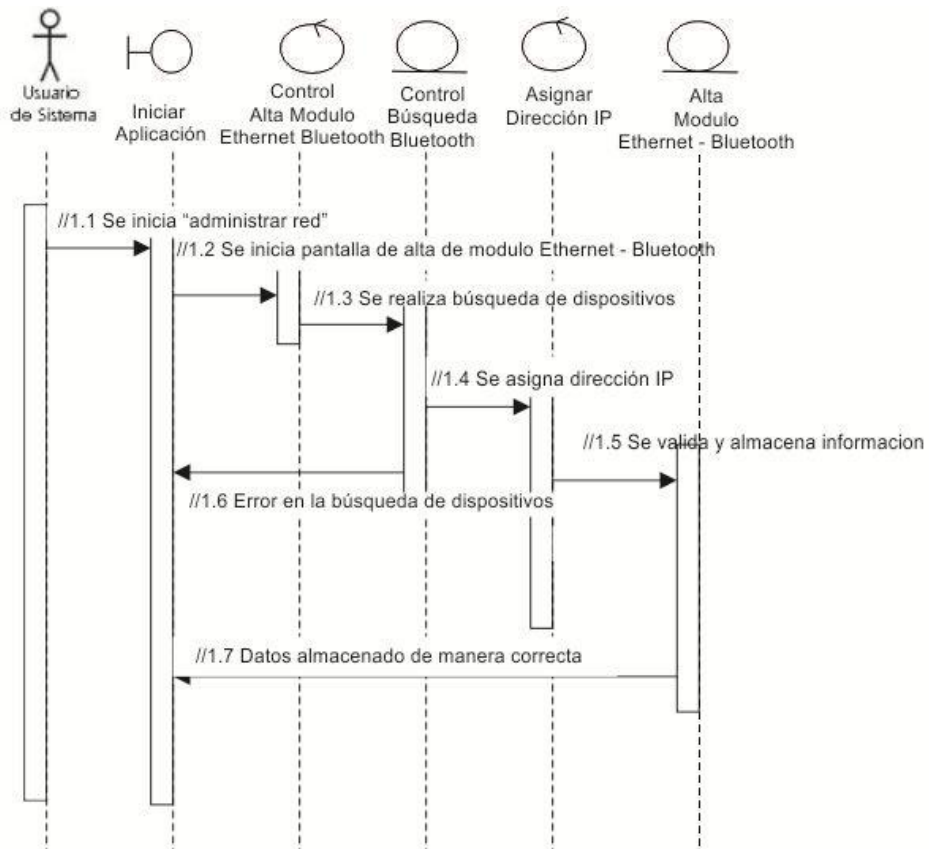


Figura 19. Diagrama de Secuencia para Administrar Red, en este diagrama la tarea más importante es la asignación de la dirección IP, que servirá para comunicarse con el modulo

En el diagrama de secuencia para Administrar Red, se da por hecho que el modulo ya fue instalado en el edificio, solo se toma en cuenta el proceso para darlo de alta en el sistema y un paso importante que es la asignación de la dirección IP con la cual se comunicará, se tiene que ser cuidadoso ya que esta dirección tiene que ser única para cada módulo.

#### 4. Diagrama de Secuencia para el caso de uso "Monitorear Persona"

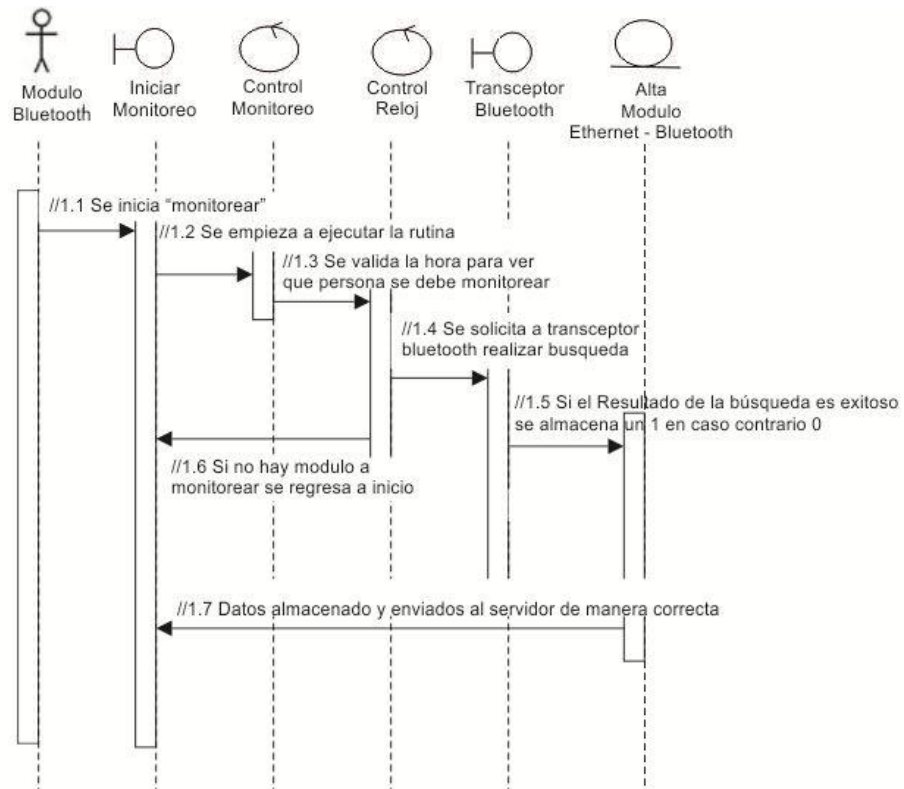


Figura 20. Diagrama de Secuencia para Monitorear Persona, se observa que si se encuentra al smartphone se guarda un valor de uno y cero en caso contrario

En este diagrama de secuencia, se observa el proceso de toma de asistencia, esta rutina consiste en primero validar día y hora para verificar si alguien tiene que estar en estos momentos en el área laboral, en caso de ser así se procede a buscar al smartphone, en caso de estar presente se guarda un valor de uno en caso contrario cero. Después se procede a enviar el resultado haciendo uso del transceptor Ethernet.

### Diagrama Entidad Relación

Después de haber realizado los casos de uso e identificar las acciones que cada uno de los actores ejerce sobre el sistema. Se va a empezar a definir la base de datos de este sistema. Se hará uso del Diagrama de Entidad Relación para precisar la información a tratar.

## Modelado del Sistema

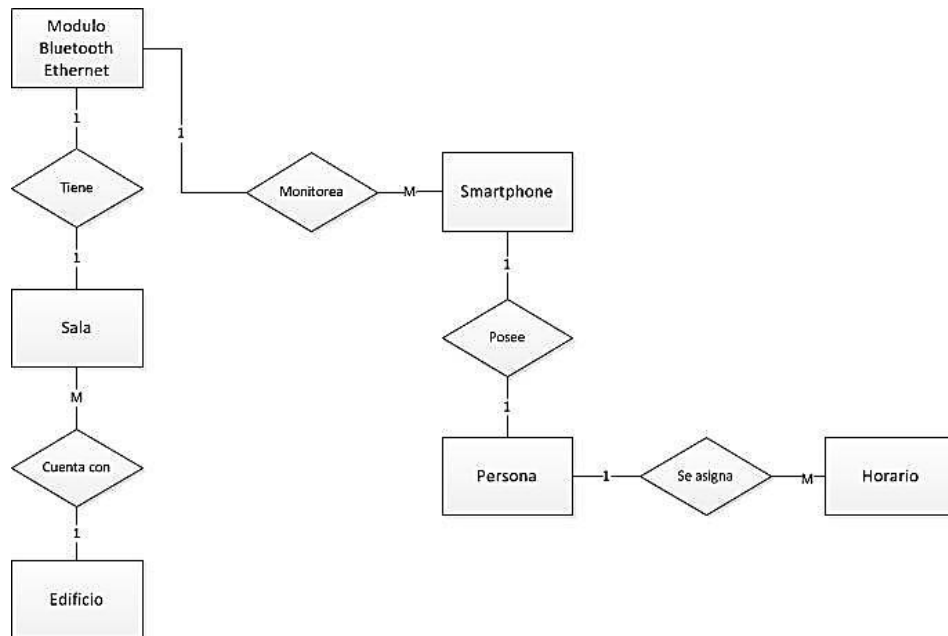


Figura 21. Diagrama Entidad Relación que define la relación existente entre cada una de las entidades del sistema

Las entidades que se pueden apreciar en la figura 21 y sus atributos son los siguientes:

- Edificio
- Sala
- Módulo Bluetooth Ethernet
- Smartphone
- Persona
- Horario

### Edificio

ID_Edificio	Nombre	Numero de salones
-------------	--------	-------------------

En esta entidad se guardarán los datos de los edificios, cuenta con una llave primaria *ID\_Edificio*, que servirá para identificar cada uno de los edificios de manera única

### Sala

ID_Sala	Sala	Edificio	Piso
---------	------	----------	------

En esta entidad se almacenarán los datos de cada una de las salas de cada edificio, tiene una llave primaria *ID\_Sala*, y una llave foránea *Edificio* que sirve para hacer la relación con la entidad Edificio.

Modulo Bluetooth Ethernet

ID_Modulo	Nombre Modulo	BT_ADDR	Dirección IP	Sala
-----------	---------------	---------	--------------	------

Esta entidad sirve para ilustrar cada uno de los módulos Bluetooth – Ethernet, obsérvese que se guarda la información de la dirección BT\_ADDR que corresponde a la dirección del transceptor Bluetooth y la dirección IP que se asignará al transceptor Ethernet. Cuenta con una llave primaria para distinguir cada uno de los módulos y una llave foránea que hace referencia a la tabla Sala, debido a que el modulo tiene que estar instalado en una sala.

Persona

ID_Persona	Nombre (s)	Apellido Paterno	Apellido Materno
------------	------------	------------------	------------------

En la entidad persona se almacena toda la información personal de cada una de las personas a monitorear.

Smartphone

ID_Smartphone	Persona	Nombre Smartphone	BT_ADDR
---------------	---------	-------------------	---------

En Smartphone se define la información que corresponde a cada uno de los smartphones, se almacena la dirección BT\_ADDR del smartphone para poder buscarlo posteriormente con algún modulo Bluetooth – Ethernet. Se hace una llave foránea para la tabla Persona para asociar a un smartphone con una persona, que es el propietario del mismo.

Horario

ID_Horario	Modulo	Hora Inicio	Hora Fin	Día	Smartphone
------------	--------	-------------	----------	-----	------------

En la entidad horario, se guardan los horarios en que una persona tiene que estar laborando en alguna de las salas. Tiene llaves foráneas a módulo y smartphone.

Después de haber realizado el análisis UML, se ha llegado a definir la base de datos que corresponde al sistema. Definiendo cada una de las tablas (entidades) y los atributos que las conforman.

## Modelado del Sistema

### **CONCLUSIONES**

En este capítulo se ha realizado el modelado del sistema en base a un estándar que es UML.

Haciendo uso de diagramas como de caso de uso y secuencia se delimitaron los actores involucrados y la serie de pasos que se debe seguir para poder interactuar con el sistema.

En base a todo este análisis, se pudo definir la base de datos de este proyecto de tesis, que servirá como base del desarrollo del trabajo que se verá en las siguientes secciones.



# IV

---

## RED PARA DETECCIÓN DE SMARTPHONES

Como se mencionó anteriormente se abordará la solución del proyecto de tesis dividiéndolo en parte física y lógica. En este capítulo se describe cómo se desarrolló la parte física. Se mencionara el tipo de transceptores empleados, y la manera en que estos operan. Se hará un paréntesis para indicar el microcontrolador. También se mostrara el algoritmo que se estará ejecutando en el firmware del módulo Bluetooth - Ethernet. Por último se mostrara el diseño en PCB que corresponde a las tarjetas controladoras para la comunicación Bluetooth - Ethernet.

La capa física siendo una de las más importantes se le ha dado un capítulo entero de este trabajo de investigación. Se entiende por capa física como a todo lo que rodea la implementación de la red para detección de smartphones. Durante este capítulo se hablará del diseño de la tarjeta controladora de comunicación *Bluetooth - Ethernet*.

## Diagrama a Bloques del Sistema

Un diagrama a bloques es una representación gráfica del funcionamiento interno de un sistema mediante el uso de bloques y las relaciones entre estos, y que se define en base de sus entradas y salidas.

El primer acercamiento del diagrama de bloques para el módulo *Bluetooth - Ethernet*, de manera muy abstracta sería el siguiente:

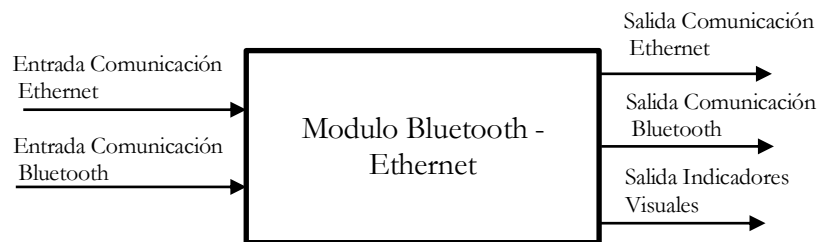


Figura 22. Un primer acercamiento al diagrama de bloques, mostrando las entradas de la comunicación Bluetooth, Ethernet y sus salidas correspondientes

Como se observa en la figura 22, se requiere tener comunicación al exterior mediante el uso de dos tecnologías. La comunicación vía Ethernet para poder comunicarse con el servidor que recibirá todos los datos de la búsqueda de smartphones y comunicación Bluetooth para detectar los smartphones.

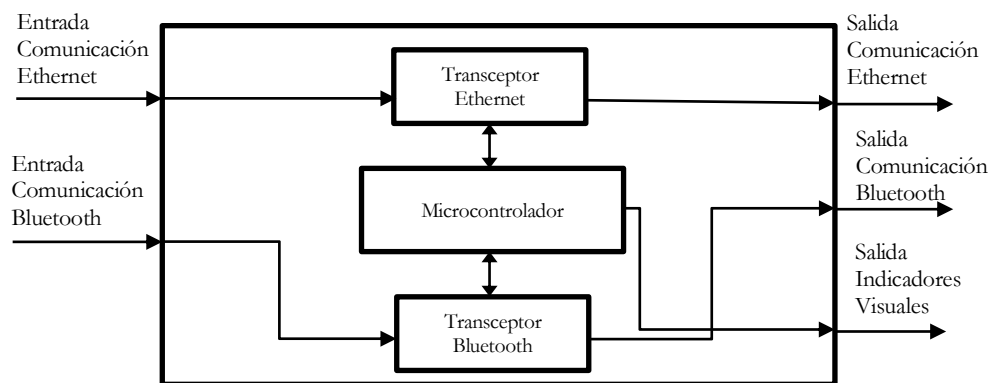


Figura 23. Diagrama a bloques que incluye los transceptores y el microcontrolador y la manera en que interactúan

Para esto se hace uso de una serie de transceptores que se encuentran disponibles en el mercado. Ahora se puede observar otra necesidad es poder controlar estos transceptores para procesar la información que nos ofrecen y ejecutar en este, el firmware del módulo Bluetooth - Ethernet. A través de un microcontrolador se maneja tanto el transceptor Bluetooth como el Ethernet para obtener y enviar información.

## Red para Detección de Smartphones

Incluyendo estos elementos en el diagrama de bloques, la manera en la que se comunican e interactúan cada uno de estos elementos es la siguiente.

En el diagrama de la figura 23 se muestra la conexión que hay entre cada uno de los módulos. El transceptor Bluetooth está conectado a la entrada y salida de comunicación Bluetooth. Pero también al microcontrolador con el cual puede intercambiar información. La información recibida por el transceptor Bluetooth es solicitada por el microcontrolador y este la procesa; después el microcontrolador puede responder y solicita al transceptor Bluetooth el envío de la información. Un caso similar ocurre con el transceptor Ethernet, en este caso el transceptor Ethernet al recibir una conexión le indica al microcontrolador de esta situación, entonces el microcontrolador toma el mando para especificar sobre que protocolo comunicarse y se procede al intercambio de información.

Por último la salida de indicadores visuales servirá como apoyo para saber cuándo y que transceptor está intercambiando información con el microcontrolador. Esta señal solamente es de manera informativa para el desarrollador.

Obsérvese que a la hora de implementar el modulo se debe procurar que la comunicación con ambos transceptores no se vea interrumpida, además que cuando se esté intercambiando información con un transceptor no se entorpezca el tiempo de respuesta con otro transceptor.

En líneas siguientes se hablara de cada uno de estos módulos que conforman el diagrama a bloques para después realizar su integración y mostrar el modulo funcionando de manera correcta.

### **Transceptor Bluetooth**

El termino transceptor hace referencia a un dispositivo de comunicación que tiene la capacidad de transmitir y/o recibir información, en otras palabras que cuenta con la circuitería necesaria para realizar estas operaciones.

En la siguiente figura se ilustra el transceptor Bluetooth utilizado en este proyecto.



Figura 24. Transceptor Bluetooth que se utiliza en este proyecto de investigación

Las especificaciones principales de este transceptor se explican en la siguiente tabla<sup>xxii</sup> :

Característica	Descripción
Versión de Bluetooth v 2.0	Este chip cuenta con características de la versión 2 de la especificación, la característica más importante que cuenta es <i>saltos de frecuencia adaptativo</i> .
Alimentación de 3.3 v	La alimentación del transceptor es de 3.3 v, puede soportar 5 v pero esto puede reducir la vida del dispositivo por lo que es recomendable usar un regulador de voltaje para no dañarlo
Interfaz I2C, UART y USB	Para comunicarse desde otro dispositivo (host) con el transceptor se puede usar tres tipos de comunicación serial, en este proyecto de investigación se usa la interfaz UART
Tamaño 29.9mm por 13 mm por 2.2 mm	El tamaño reducido representa una ventaja para proyectos donde no haya mucho espacio como en nuestro caso que deseamos implementar un sistema empotrado
Dispositivo de Clase 2	Es un dispositivo de baja potencia que puede ser alimentado incluso con baterías, su alcance es de un radio de 10 metros que delimita la distancia donde se monitorean smartphones

<sup>xxii</sup> Se agrega al final de este trabajo como anexo, la hoja de datos del fabricante del transceptor Bluetooth para una consulta más exhaustiva

## Red para Detección de Smartphones

Comandos AT <sup>xxiii</sup>	Viene incluido con un conjunto de comandos denominados AT que sirven para configurar y realizar las tareas del módulo. Con estos comandos podemos realizar configuraciones como cambiar nombre al dispositivo o cambiar el rol (Esclavo o maestro) y tareas como conectarse a otro dispositivo.
Roles	Los roles soportados por este transceptor son: esclavo, maestro y esclavo cíclico (este rol solo reenvía la información que recibe sin realizar ningún tipo de procesamiento sobre la misma)

Tabla 8. Características principales con las que cuenta el transceptor Bluetooth

Como se aprecia en el diagrama a bloques (figura 23) es necesario comunicar al transceptor Bluetooth con un microcontrolador. La tabla 8 muestra que este transceptor tiene integrado tres tipos de interfaces de comunicación UART, USB e I2C. Para el desarrollo de este proyecto se utiliza la interfaz UART<sup>xxiv</sup>.

Las características del UART del transceptor Bluetooth son las siguientes:

Característica	Descripción
Velocidad de Transmisión	El modulo maneja las siguientes velocidades de transmisión de información: 4800, 9600, 19200, 38400, 57600, 115200, 230400, 460800, 912600, 1382400 bps.
Bits de Parada	Usa la opción de uno o dos bits de parada.
Bit de Paridad	Se puede configurar como ninguno, impar y par

Tabla 9. Características del UART del transceptor Bluetooth

<sup>xxiii</sup> Los comandos AT son instrucciones codificadas que conforman un lenguaje de comunicación entre el hombre y un Terminal MODEM. AT corresponde a la abreviación de la palabra en ingles *attention*, en caso del transceptor estos comando sirven para configurarlo

<sup>xxiv</sup> UART es un acrónimo de *Universal Asynchronous Receiver Transmitter* que es quien se encarga de puertos y dispositivos seriales en una computadora

En este proyecto los transceptores Bluetooth estarán configurados de la siguiente manera, con una velocidad de 38400 bps, un bit de parada, sin bit de paridad y ocho bits de datos.

Para poder realizar cualquier tipo de configuración o acción con el transceptor Bluetooth se hace uso de los comandos AT.

Los comandos AT son instrucciones codificadas que comúnmente conforman un lenguaje de comunicación entre el hombre y un terminal modem. En un principio, el juego de comandos AT fue desarrollado como un interfaz de comunicación con un modem para así poder configurarlo y proporcionarle instrucciones, tales como marcar un número de teléfono. Los comandos AT se denominan así por la abreviatura de *attention*.

Aunque la finalidad principal de los comandos AT es la comunicación con módems, la telefonía móvil también ha adoptado como estándar este lenguaje para poder comunicarse con sus terminales. De esta forma, todos los teléfonos móviles poseen un juego de comandos AT específico que sirve de interfaz para configurar y proporcionar instrucciones a las terminales.

Si se alimenta la entrada *key* del transceptor Bluetooth con 3.3 v, se activa el modo de comandos AT.

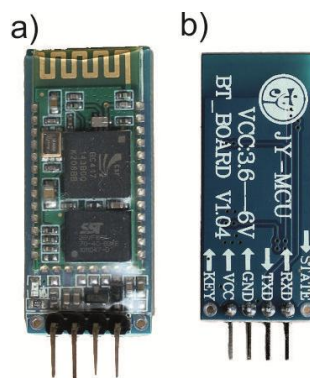


Figura 25. Pines del módulo Bluetooth, en a) se ve el anverso del módulo, en b) de izquierda a derecha, *key* corresponde al pin de AT, VCC al voltaje de alimentación, GND a la tierra del dispositivo, RXD y TXD, transmisor y receptor del UART.

El comando AT encargado de la configuración del UART es el siguiente:

## Red para Detección de Smartphones

Comando	Resultado	Argumentos
AT+UART=<Parametro1>,<Parametro2>,<Parametro3> AT+UART?	Ok  +UART: <Parametro1>,<Parametro2>,<Parametro3> Ok	Parametro1 es la velocidad de transmisión, parámetro 2 el bit de parada y por último el bit de paridad

Tabla 10. Comando AT para la configuración del UART

Una vez que se ha configurado la velocidad de cada uno de los transceptores se procede a realizar la conexión con otros dispositivos.

Los pasos que se deben seguir para realizar la interconexión entre dos dispositivos son los siguientes, téngase en cuenta que esta secuencia se ejecuta del lado del transceptor Bluetooth que este configurado como maestro. También recuerde que la comunicación se realiza de maestro a esclavo, y no se puede realizar de maestro a maestro.

1. Inicializar transceptor
2. Emparejar dispositivos
3. Enlazar dispositivos
4. Intercambiar Información
5. Desconectar Dispositivo

Los comandos AT para cada una de estas tareas se listan a continuación:

Comando	Resultado	Argumento
AT+INIT	1. OK 2. FAIL	Ninguno

El comando INIT indica al transceptor que tiene que prepararse para realizar la conexión con otro dispositivo remoto.

Comando	Resultado	Argumento
AT+PAIR=<Parametro1>,<Parametro2>	1. Ok 2. FAIL	Parametro1: dirección del dispositivo remoto, Parametro2 tiempo de espera para el emparejamiento

En comando PAIR se ejecuta el emparejamiento de ambos dispositivos, en este proceso es cuando se entra en el estado de página y se negocia cada cuanto tiempo se realizaran los saltos en frecuencia y la manera de determinar estos saltos.

Comando	Resultado	Argumento
AT+LINK=<Parametro1>	1. Ok 2. FAIL	Parametro1: dirección del dispositivo remoto a conectarse

En el comando LINK se realiza la conexión completa entre ambos dispositivos una vez que este proceso se ha ejecutado de manera exitosa, ambos dispositivos podrán realizar intercambio de información hasta que alguno de los dos rompa el enlace.

Comando	Resultado	Argumento
AT+DISC	1. +DISC:SUCCESS 2. +DISC:LINK_LOSS 3. +DISC:NO_SLC 4. +DISC:TIMEOUT 5. +DISC:ERROR	Ninguno

El comando DISC, se ejecuta desde el modulo maestro que está llevando el control de la comunicación, el enlace puede terminar de cinco maneras distintas, exitosamente, se perdió la conexión, no\_slc, se agotó el tiempo de espera y por algún error.

Ahora sustituyendo los comandos en la secuencia de conexión, los pasos serían los siguientes:

1. AT+INIT
2. AT+PAIR=1234,56,abcdef,20
3. AT+LINK= 1234,56,abcdef
4. Intercambiar Información
5. AT+DISC

Nótese que el parámetro de dirección se tiene que pasar en el formato LAP (16 bits), UAP (8 bits) y NAP (48 bits), cada campo se representa en hexadecimal.

El último comando que se observa es aquel que indica cuando un dispositivo Bluetooth se encuentra dentro del área de alcance de nuestro transceptor.

Comando	Resultado	Argumento
AT+RNAME? <Parametro1>	1. +RNAME: <Parametro2> OK 2. FAIL	Parametro1: dirección del dispositivo remoto a conectarse Parametro2: nombre del dispositivo remoto

## Red para Detección de Smartphones

El comando RNAME, recibe como parámetro la dirección del dispositivo remoto a monitorear, si el dispositivo está dentro del radio del transceptor entonces devuelve el nombre del dispositivo, en caso de no estar presente responde con un FAIL, para ejecutar este comando no es necesario realizar una nueva conexión.

### Transceptor Ethernet

El transceptor Ethernet utilizado en este proyecto es el ENC28J60 de la compañía Microchip. El ENC28J60 es un controlador Ethernet autónomo que hace uso del estándar SPI<sup>xxv</sup>. Está diseñado para servir como una interfaz de red Ethernet para cualquier controlador equipado con SPI.

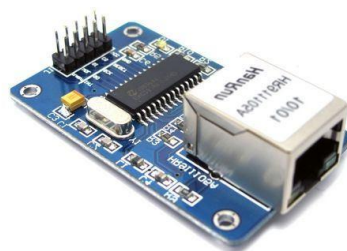


Figura 26. Transceptor Ethernet ENC28J60 de la marca Microchip, diseñado para funcionar con microcontroladores

El ENC28J60 cumple con las especificaciones descritas en el estándar 802.3<sup>xxvi</sup> de la IEEE.

El módulo cuenta con seis bloques importantes:

1. Una interfaz SPI, que sirve como canal de comunicación entre el controlador principal (microcontrolador) y el ENC28J60
2. Registros de control que se utilizan para controlar y supervisar el ENC28J60
3. Un búfer de RAM para recibir y transmitir paquetes de datos
4. La interfaz de bus que interpreta los datos y comandos recibidos a través de la interfaz SPI
5. Módulo MAC (Control de Acceso al Medio) que implementa la lógica IEEE 802.3 MAC
6. El módulo PHY (capa física) que codifica y decodifica los datos analógicos que están presentes en el cable de par trenzado

---

<sup>xxv</sup> SPI corresponde a las siglas *Serial Peripheral Interface*, es un estándar de comunicación serial, usado principalmente para la transferencia de información entre circuitos integrados en equipos electrónicos.

<sup>xxvi</sup> Corresponde al primer estándar realizado por la IEEE para las comunicaciones Ethernet

A continuación se muestra como se realiza el intercambio de información entre el transceptor y el microcontrolador.

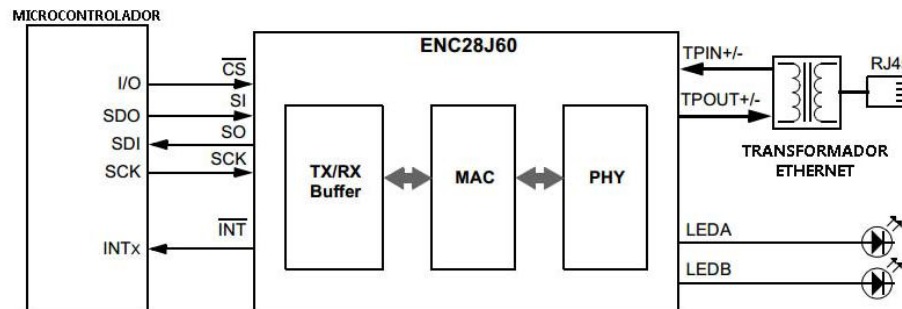


Figura 27. Se puede apreciar los módulos que conforman este transceptor Ethernet y las señales que se utilizan en la comunicación SPI

En la figura 27 se observa cómo funciona el módulo, la capa física y la interfaz MAC que son los encargados de obtener los datos que se almacenan en búfer, en la parte izquierda de la figura se muestra la interconexión con un microcontrolador y las señales necesarias para realizar el intercambio de datos con SPI.

- SCK es la señal de reloj y envía un pulso para marcar la sincronización.
- SO o MISO (*Master Input Slave Output*) salida de datos del maestro y entrada de datos del esclavo (SPI también maneja un esquema de esclavo maestro, en nuestro caso el maestro es el microcontrolador y el esclavo el módulo Ethernet).
- SI o MOSI (*Master Output Slave Input*) salida de datos del maestro y entrada de datos del esclavo
- CS (*chip select*) señal que sirve para seleccionar a un esclavo o decirle a un esclavo que se active

Para poder hacer uso de este módulo es necesario usar una librería en la programación del microcontrolador que ofrezca la posibilidad de elegir un protocolo entre todos los que conforman TCP/IP para realizar el intercambio de información. Microchip nos ofrece esta librería donde se implementa toda la pila de protocolos TCP/IP y nos ofrece una serie de funciones para poder comunicarnos con el módulo. A continuación se hace un listado de las funciones más importantes.

## Red para Detección de Smartphones

<b>Función</b>	<b>Descripción</b>
<b>MACAddrInit ()</b>	Esta función sirve para definir la dirección MAC de nuestro transceptor, al asignar la dirección debemos ser cuidadosos y verificar que no coincida con la de algún otro dispositivo de la red
<b>IPAddrInit()</b>	Esta función sirve para definir la dirección IP y el puerto sobre el cual nos estaremos comunicando y está formada de cuatro bytes.
<b>TCPListen(Puerto)</b>	Esta función se ejecutará continuamente, ya que esta función permanecerá a la espera de conexiones entrantes.
<b>TCPIsConnected(MYTCPSocket)</b>	Función que sirve para verificar si un socket se encuentra conectado remotamente
<b>TCPIsGetReady(MYTCPSocket)</b>	Esta función devuelve verdadero cuando el transceptor Bluetooth está listo para recibir información
<b>TCPGet(MYTCPSocket, &amp;c)</b>	Lee un byte del socket indicado y lo almacena en el segundo parámetro llamado c
<b>TCPIsPutReady(MYTCPSocket)</b>	Esta función devuelve verdadero cuando el transceptor está listo para enviar información
<b>TCPPut(MYTCPSocket,c)</b>	Se escribe el byte c al socket indicado, la función TCPIsPutReady se debe ejecutar antes de esta función
<b>TCPDisconnect(MYTCPSocket)</b>	Función que realiza la desconexión del socket indicado

Tabla 11. Se hace un listado de las funciones más importantes para el manejo del transceptor Bluetooth

## Microcontrolador

En la solución del sistema se hizo uso de un microcontrolador PIC de la empresa Microchip, del PIC18F4550. Este dispositivo ofrece un alto rendimiento de cómputo a un precio económico.

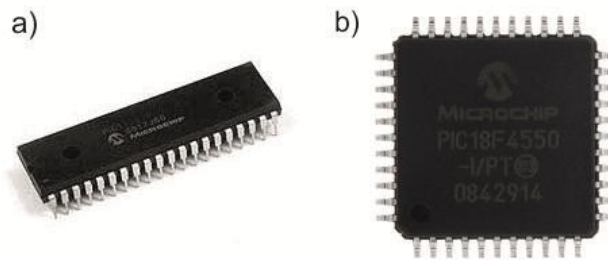


Figura 28. Se muestra al microcontrolador PIC18f4550, a) podemos observarlo en su presentación de 40 pines PDIP, en b) el encapsulado de superficie

Las características más importantes del PIC18F4550 son las siguientes:

Característica	Descripción
USB	Contiene un puerto USB esclavo
Tres interrupciones externas	Tiene asignado tres pines para que cuando haya un cambio de voltaje se genere una interrupción
Cuatro Temporizadores (Timer0 al Timer3)	Su función es generar una interrupción cada determinado lapso de tiempo
UART	Soporta comunicación serial vía protocolo RS – 232
I2C	Cuenta con un puerto para la comunicación I2C, que es un protocolo de comunicación basado en el esquema esclavo- maestro
SPI	Cuenta con un puerto SPI, que permite transmitir y recibir simultáneamente 8 bits de información de manera síncrona.
Convertidor Analógico a Digital	Tiene un convertidor de analógico a digital, con un umbral ajustable en un rango de 0 a 5v
Memoria de Programa	Cuenta con una memoria Flash de 32 Kbyte
Memoria RAM	Con una memoria de acceso aleatorio de 2 Kbyte
Memoria EEPROM	Adicionalmente tiene una memoria EEPROM de 256 bytes

Tabla 12. Características de PIC18F4550, de vemos que en dentro de la tabla tenemos las interfaces de comunicación para intercambiar datos con el tranceptor Bluetooth y Ethernet

## Red para Detección de Smartphones

De las características mencionadas anteriormente, se puede utilizar el puerto serial (UART) para comunicarse con el transceptor Bluetooth. Hay un puerto SPI que funciona para intercambiar datos con el transceptor Ethernet, además se cuenta con una serie de temporizadores que se usan para realizar la búsqueda de smartphones por un determinado lapso de tiempo.

## Desarrollo de Algoritmo

A continuación se detalla el algoritmo que se implementó dentro de cada una de las tarjetas controladoras. Originalmente este proyecto estaba pensado para ejecutarse haciendo uso de redes Bluetooth que son *piconet* y *scatternet*. Pero al final del trabajo se decidió hacer la integración de un transceptor Ethernet a cada uno de los módulos. En las siguientes líneas se muestra el algoritmo de cada uno de estos intentos para justificar el cambio realizado.

## Usando Piconet

Como se mencionó en capítulos anteriores una piconet puede tener ocho dispositivos en comunicación de manera activa. Uno de estos dispositivos corresponde a un maestro y siete esclavos.

El maestro será el único que contará con un transceptor Bluetooth y con este recibirá las instrucciones de iniciar el monitoreo e indicar que personas monitorear. Los esclavos serán los encargados de realizar el monitoreo y cada determinado tiempo realizaran la búsqueda.

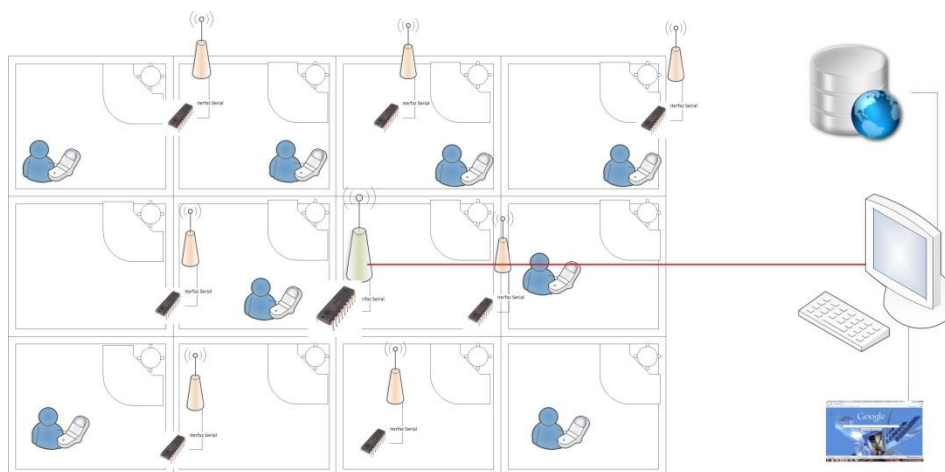


Figura 29. Se muestra la solución propuesta implementando una piconet, observe que solo un módulo tiene comunicación con el servidor

En la figura 29 se aprecia un maestro (color verde) y siete esclavos (color café), cada uno de estos esclavos tiene comunicación directa con el maestro e intercambian información a través de una serie de comandos.

El algoritmo para realizar la búsqueda lo podemos dividir en dos, en la parte del servidor y en la parte del esclavo.

A continuación se muestra el algoritmo que se ejecutará de parte del maestro

Algoritmo 1. Maestro de Piconet
<pre> Arreglo esclavos [7]; Cadena comando ← Leer_Puerto_Ethernet();  Si (comando es igual a Activar Sondeo)      Hacer i ← 0 hasta 7         activarSondeo(esclavos[i]);  Fin Si  Si (comando es igual a Enviar Dirección a Monitorear)     Hacer i ← 0 hasta 7         dirección_Bluetooth ← Leer_Direccion_Ethernet();         enviarDireccionSondeo(dirección_Bluetooth);     Fin Hacer Fin Si  Si (comando es igual a Solicitar Sondeo)     resultado ← 0;     Hacer i ← 0 hasta 7         resultado ← solicitarSondeo(esclavo[i]);     Fin Hacer     enviarSondeoEthernet(resultado); Fin Si  Si (comando es igual a Desactivar Sondeo)     Hacer i ← 0 hasta 7         desactivarSondeo(esclavo[i]);     Fin Hacer Fin Si                     </pre>

Tabla 13. Se aprecia el pseudocódigo del programa implementado en el módulo maestro

## Red para Detección de Smartphones

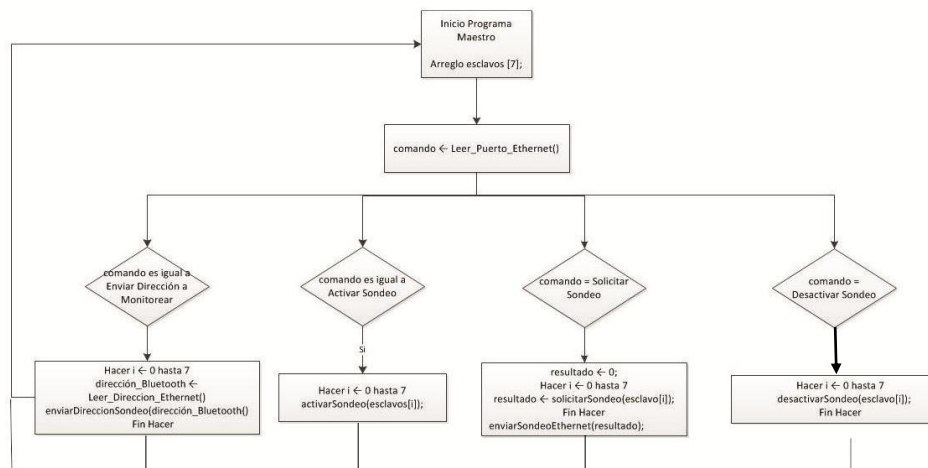


Figura 30. Se ilustra el mismo pseudocódigo pero con una representación gráfica como un diagrama de flujo, obsérvese que no hay una etiqueta fin de programa debido a que programa tiene que estar ejecutándose constantemente

En el algoritmo anterior vemos que cada tarea se realiza sobre un conjunto de esclavos. En la primera opción que es activar sondeo, indica a todos los módulos de la piconet deben de empezar a buscar smartphones. Es recomendable ejecutar antes enviar dirección a monitorear, ya que esta dirección corresponde al smartphone a verificar. El comando solicitar sondeo le indica al maestro que ejecuta este algoritmo, que solicite la información de los resultados del sondeo que están ejecutando cada uno de los esclavos. Por ultimo desactivar sondeo como su nombre lo indica, envía una señal a todos los esclavos para pedirles que detengan el sondeo.

El siguiente algoritmo muestre el funcionamiento del esclavo.

### Algoritmo 2. Esclavo de Piconet

```

Cadena dirección_Smartphone;
Booleano sondeoActivo;
Cadena comando ← Leer_Bluetooth();
Entero resultado;

Si (comando es igual a Activar Sondeo)
    sondeoActivo ← verdadero
Fin Si

Si (Temporizador Agotó Tiempo y sondeoActivo es verdadero)
    Resultado ← sondearDireccion(direccionSmarphone)
Fin Si

Si (comando es igual a Enviar Resultado)

```

**Continuación Algoritmo 2. Esclavo de Piconet**

```

Escribir_Bluetooth(Resultado);
Fin Si

Si (comando es igual a Recibir Dirección a Monitorear)
    dirección_Smartphone ← Leer_Bluetooth();
Fin Si

Si (comando es igual a Detener Sondeo)
    sondeoActivo ← falso
Fin si

```

Tabla 14. Se aprecia el pseudocódigo del programa implementado en el módulo esclavo

En este algoritmo se aprecia que se hace uso de un temporizador el cual cada determinado tiempo indica que se debe realizar la búsqueda de smartphones. Se observa también que el intercambio de información es realizada por el maestro. Cuando el maestro envía un comando, el modulo esclavo verifica de que comando se trata. Los comandos activar y desactivar sondeo indican en que horas tenemos que estar obteniendo información de los smartphones. El comando recibir dirección a monitorear, sirve para que el modulo esclavo reciba la instrucción del maestro para indicarle a que smartphone buscar. Y por último el comando enviar información de sondeo, como su nombre lo indica envía la información de la búsqueda al maestro.

## Red para Detección de Smartphones

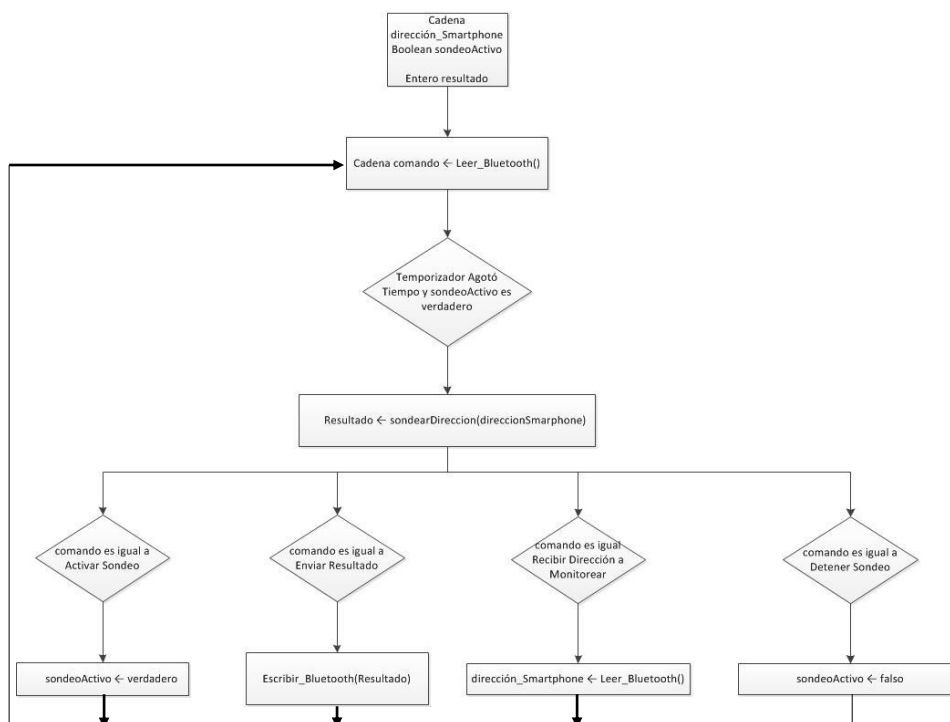


Figura 31. Diagrama de flujo que corresponde al algoritmo del módulo esclavo

## Usando Scatternet

En capítulos anteriores se comentó que una *scatternet* es un conjunto de piconets interconectadas entre sí. Tomando este concepto en el sistema, vemos que necesitamos tres tipos de módulos: esclavo, maestro y puente.

El esclavo y maestro se comportan de la misma manera que en una piconet, el tercer elemento que introducimos es un esclavo que servirá como puente entre dos maestros. Anteriormente se vio que dos maestros no pueden realizar intercambio de información de manera directa, por eso se introduce un tercer módulo llamada puente. A continuación se muestra de manera gráfica como se implementó este enfoque.

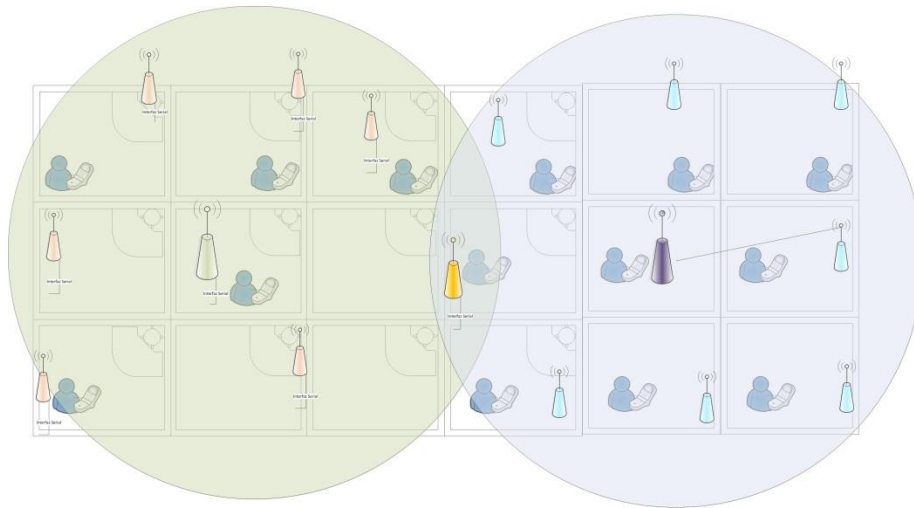


Figura 32. Solución propuesta haciendo uso de una scatternet

En la figura 32 se tiene dos piconets, el maestro A de color verde y el maestro B de color azul. Dentro del radio de alcancen de las dos piconets hay un espacio donde ambas se interceptan, es aquí donde se ha colocado al módulo puente (de color amarillo). Solo un maestro es el que tendrá comunicación con el servidor para indicar las acciones a realizar.

Los módulos en consideración para esta solución son los siguientes:

- Módulo Maestro – Ethernet
- Módulo Maestro
- Módulo Puente
- Módulo Esclavo

A continuación se definen los algoritmos para cada uno de estos módulos:

El algoritmo para el modulo maestro - Ethernet es el siguiente:

## Red para Detección de Smartphones

### Algoritmo 3. Maestro - Ethernet de Scatternet

```
Arreglo esclavos [7];
Cadena comando ← Leer_Puerto_Ethernet();
Cadena Puente;
Cadena MiDireccion;
Cadena Direccion_Destino;
Cadena Datos;

Si (comando es igual a Activar Sondeo)
  Hacer i ← 0 hasta 7
    escribirBluetooth(esclavos[i], Activar Sondeo);
  Fin Hacer
  escribirBluetooth(Puente, MiDireccion, Activar Sondeo);
Fin Si

Si (comando es igual a Enviar Dirección a Monitorear)
  Direccion_Destino ← Leer_Puerto_Ethernet();
  Datos ← Leer_Puerto_Ethernet();
  Si (Direccion_Destino es igual a MiDireccion)
    Hacer i ← 0 hasta 7
      escribirBluetooth(esclavos[i], Enviar Dirección a Monitorear, Datos);
    Sino
      escribirBluetooth(Puente, Enviar Dirección a Monitorear, MiDireccion,
Direccion_Destino, Datos);
    Fin Si
  Fin Si

Si (comando es igual a Solicitar Sondeo)
  resultado ← 0;
  resultado ← leerBluetooth(puente, Solicitar Sondeo);
  Hacer i ← 0 hasta 7
    resultado +← escribirBluetooth(esclavo[i], Solicitar Sondeo);
  Fin Hacer
  escribirEthernet(Solicita Sondeo, resultado);
Fin Si

Si (comando es igual a Desactivar Sondeo)
  Hacer i ← 0 hasta 7
    escribirBluetooth(esclavo[i], Desactivar Sondeo);
  Fin Hacer
  escribirBluetooth(Puente, MiDireccion, Desactivar Sondeo);
Fin Si
```

Tabla 15. Se muestra el algoritmo correspondiente al Maestro con Ethernet de la Scatternet

Como comentario del pseudocódigo podemos ver que al agregar más de una piconet al control del monitoreo, el problema se vuelve más complejo. Los comandos son prácticamente los mismos que los que se usaron en los algoritmos con piconet. La única diferencia es que si la información no pertenece a esta piconet se tiene que enviar a un puente para que este después la canalice con el siguiente maestro y así sucesivamente hasta arribar a su destino. Ejemplo de esto es el comando solicitar sondeo, a resultado se le

asigna el resultado de sondeo de la piconet actual más sin embargo se pide los resultados de sondeo al puente. Una vez recibida la información se hace envío de la misma vía Ethernet.

```

Algoritmo 4. Puente de Scatternet
Cadena MiDireccion;
Cadena Maestro1;
Cadena Maestro2;
Cadena Origen;
Cadena BufferM1;
Cadena BufferM2;

Cadena Datos;

Cadena comando ← LeerBluetooth();

Si (comando es igual a Activar Sondeo)
Origen ← LeerBluetooth();

    Si (Origen es igual a Maestro1 )
        escribirBluetooth(Maestro2, Activar Sondeo);
    Fin Si

    Si (Origen es igual a Maestro2)
        escribirBluetooth(Maestro1, Activar Sondeo);
    Fin Si

Fin Si

Si (comando es igual a Enviar Dirección a Monitorear)
    Origen ← LeerBluetooth();
    Direccion_Destino ← Leer_Puerto_Ethernet();
    Datos ← Leer_Puerto_Ethernet();

    Si (Origen es igual a Maestro1 )
        escribirBluetooth(Maestro2, Enviar Dirección a Monitorear, Datos);
    Fin Si

    Si (Origen es igual a Maestro2)
        escribirBluetooth(Maestro1, Enviar Dirección a Monitorear, Datos);
    Fin Si

Fin Si

Si (comando es igual a Solicitar Sondeo)
resultado ← 0;
    Origen ← LeerBluetooth();

    Si (Origen es igual a Maestro 1)
        escribirBluetooth(BufferM1);
    Fin Si

    Si (Origen es igual a Maestro2)
        escribirBluetooth(BufferM2);

```

## Red para Detección de Smartphones

Continuación Algoritmo 4. Puente de Scatternet
Fin Si
Fin Si
Si (comando es igual a Enviar Sondeo)
resultado $\leftarrow$ 0;
Origen $\leftarrow$ LeerBluetooth();
Si (Origen es igual Maestro 1)
BufferM2 $\leftarrow$ leerBluetooth();
Fin Si
Si (Origen es igual Maestro1)
BufferM1 $\leftarrow$ leerBluetooth();
Fin Si
Fin Si
Si (comando es igual a Desactivar Sondeo)
Origen $\leftarrow$ LeerBluetooth();
Si (Origen es igual a Maestro1 )
escribirBluetooth(Maestro2, Activar Sondeo);
Fin Si
Si (Origen es igual a Maestro12)
escribirBluetooth(Maestro1, Activar Sondeo);
Fin Si
Fin Si

Tabla 16. Algoritmo que Corresponde al puente de la Scatternet

El algoritmo anterior se realiza con una limitante solamente permite dos nodos maestros conectados a él puente, se observan las variables *bufferM1* y *bufferM2*, que son los búferes que servirán para el almacenamiento de datos temporales para cada uno de estos dos maestros.

Se observa que durante cada comando se realiza una verificación para saber quién es el que manda la información. Esto para saber en qué búfer guardar la información para cuando algún maestro la solicite simplemente se tenga que vaciar el búfer correspondiente.

Por ejemplo en el comando desactivar sondeo, si origen es el maestro 1 se procede a indicar a maestro 2 que desactive el sondeo, en caso contrario maestro 2 indica a maestro 1 que se detenga en el monitoreo de smartphones.

**Algoritmo 5. Maestro de Scatternet**

```

Arreglo esclavos [7];
Cadena PuenteEntrada;
Cadena PuenteSalida;
Cadena MiDireccion;
Cadena Direccion_Destino;
Cadena Datos;

Cadena comando ← LeerBluetooth(PuenteEntrada);

Si (comando es igual a Activar Sondeo)
  Hacer i ← 0 hasta 7
    escribirBluetooth(esclavos[i], Activar Sondeo);
  Fin Hacer
  escribirBluetooth(PuenteSalida, MiDireccion, Activar Sondeo);
Fin Si

Si (comando es igual a Enviar Dirección a Monitorear)
  Direccion_Destino ← LeerBluetooth(PuenteEntrada);
  Datos ← LeerBluetooth(PuenteEntrada);
  Si (Direccion_Destino es igual a MiDireccion)
    Hacer i ← 0 hasta 7
      escribirBluetooth(esclavos[i], Enviar Dirección a Monitorear, Datos);
    Sino
      escribirBluetooth(PuenteSalida, Enviar Dirección a Monitorear, MiDireccion,
Direccion_Destino, Datos);
    Fin Si
  Fin Si

Si (comando es igual a Solicitar Sondeo)
  resultado ← 0;
  resultado ← leerBluetooth(puenteSalida, Solicitar Sondeo);
  Hacer i ← 0 hasta 7
    resultado +← escribirBluetooth(esclavo[i], Solicitar Sondeo);
  Fin Hacer
  escribirBluetooth(PuenteEntrada, Solicita Sondeo, resultado);
Fin Si

Si (comando es igual Desactivar Sondeo)
  Hacer i ← 0 hasta 7
    escribirBluetooth(esclavo[i], Desactivar Sondeo);
  Fin Hacer
  escribirBluetooth(PuenteSalida, MiDireccion, Desactivar Sondeo);
Fin Si

```

Tabla 17. Algoritmo Maestro Scatternet

Este maestro función de la misma manera que el maestro de la piconet visto en los algoritmos anteriores, a diferencia que todo el intercambio de información se realiza a través de comunicación Bluetooth. Este maestro cuenta con 7 esclavos sobre los cuales llevara la información sobre el

## Red para Detección de Smartphones

monitoreo en tiempo real smartphones. El Algoritmo de esclavo no fue incluido porque es el mismo que se ilustra anteriormente.

### Usando Módulos Bluetooth – Ethernet

Este enfoque es similar al que se muestra en la figura 14. En esa figura se observa una serie de módulos Bluetooth – Ethernet conectados hacia un servidor vía un router. Este es el diseño que se utilizó para la implementación de las tarjetas controladoras para la comunicación Bluetooth. Debido a que agregando un transceptor Ethernet se tiene comunicación directa desde el servidor a cada uno de los módulos. Además al usar una red inalámbrica como piconet y scatternet, la fiabilidad no era garantía debido a que en el enfoque de piconet, si se pierde comunicación con el maestro no habría sistema. Ahora en el enfoque de scatternet, el problema son los enlaces y la cobertura de la señal. Para poder aplicar este enfoque se necesitaría asegurar que la cobertura del radio Bluetooth es la necesaria para garantizar que los enlaces no se van a romper. Otra razón de su elección es que agregando estos módulos la complejidad del sistema se reduce considerablemente.

A continuación el algoritmo del módulo Bluetooth – Ethernet.

#### Algoritmo 6. Módulo Bluetooth – Ethernet

```
Cadena dirección_Smartphone;  
Boolean sondeoActivo;  
Cadena comando ← Leer_Bluetooth();  
Entero resultado;  
  
Si (comando es igual a Activar Sondeo)  
    sondeoActivo ← verdadero  
Fin Si  
  
Si (Temporizador Agotó Tiempo y sondeoActivo es verdadero)  
    Resultado ← sondearDireccion(direccionSmarphone)  
Fin Si  
  
Si (comando es igual a Enviar Resultado)  
    Escribir_Ethernet(Resultado);  
Fin Si  
  
Si (comando es igual a Recibir Dirección a Monitorear)  
    dirección_Smartphone ← Leer_Ethernet();  
Fin Si  
  
Si (comando es igual a Detener Sondeo)  
    sondeoActivo ← falso  
Fin Si
```

Tabla 18. Algoritmo que se ejecuta dentro de los módulos Bluetooth – Ethernet usados en la solución final de este proyecto

Como se observa la complejidad del algoritmo se reduce bastante y la respuesta del sistema es inmediata, ya que el sistema toma características de un sistema distribuido. La ventaja de usar un enfoque distribuido es, que sin importa que alguno de los módulos deje de funcionar, el sistema completo puede seguir funcionando y se puede ganar tiempo para solucionar el problema presente.

En la implementación de este algoritmo se hizo en el lenguaje de programación PIC C, que es un compilador de Lenguaje C para microcontroladores. Básicamente PIC C maneja el mismo estilo de programación que C solamente que contiene un conjunto especial de instrucciones para poder trabajar con microcontroladores. Si se desea consultar el código fuente se sugiere ver el apéndice B donde aparece el código completo.

### **Tarjeta de Circuito Impreso**

La tarjeta de circuito impreso es uno de los principales medios de montaje de interconexión para la mayoría de los circuitos electrónicos.

Está formada por un substrato de materia aislante sobre el cual se montan los componentes y se trazan los caminos conductores que proveen las conexiones necesarias para el circuito.

Una vez definido cada uno de los elementos de la tarjeta para comunicación Bluetooth – Ethernet, se realiza el ensamblaje para así diseñar la tarjeta de circuito impreso que se ve en la siguiente figura.

## Red para Detección de Smartphones

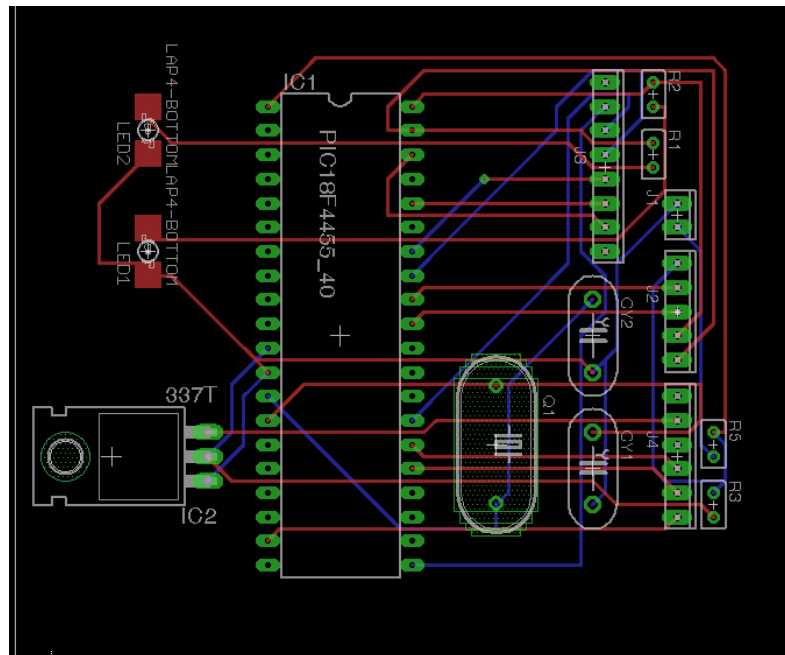


Figura 33. Se muestra el diseño de la tarjeta de circuito impreso, mostrando todos los componentes necesarios y el espacio para las interconexiones con cada uno de los transceptores

En la siguiente tabla se hace una descripción de los elementos de la figura 33.

Elemento	Descripción
<b>LED1, LED2</b>	Indicadores LED que sirven como indicadores visuales, para indicar que el sistema se encuentra realizando algunas tareas.
<b>PIC18F4550</b>	El microcontrolador sobre el cual se ejecutará el firmware del módulo Bluetooth – Ethernet.
<b>Q1</b>	Corresponde al cristal de 20 MHz. Utilizado por el microcontrolador.
<b>IC2</b>	Corresponde a un regulador de voltaje de 3.3 voltios. Necesario porque los transceptores Bluetooth y Ethernet necesitan 3.3 voltios en la entrada de alimentación y no 5 que son los que alimentan al microcontrolador.
<b>CY1, CY2</b>	Son capacitores necesarios para el funcionamiento del microcontrolador.

Tabla 19. Elementos de la tarjeta Bluetooth - Ethernet

## CONCLUSIONES

Durante este capítulo se ha explicado que transceptores se han usado para dar solución al problema. Hemos observado que el transceptor Bluetooth funciona con el uso de una serie de comandos AT y que el transceptor Ethernet hace uso de SPI. Se ha definido el uso del PC18F4550 por que soporta comunicación RS 232 y SPI para que podamos hacer uso de los transceptores además de que tiene una memoria relativamente grande.

Se ha mostrado los diferentes enfoques (*piconet, scatternet*) que se le dieron a este problema hasta antes de decidir la solución final del problema que es basada en la idea de un sistema distribuido.



# APLICACIONES DE USUARIO

En este capítulo se mostrara como se hizo la implementación de las aplicaciones que usa el usuario final ya sea para realizar la configuración del sistema o la visualización de información. Se muestra como se realizó y esta ordenada la aplicación para la alta, modificación y baja de información. También se describe la aplicación que realiza la petición de la información de cada uno de los módulos Bluetooth – Ethernet. Por último se mostrara la página web que sirve para realizar consultas de la información de la búsqueda en tiempo real.

El software de usuario final es una de las partes más sensibles de este proyecto después de la implementación de los módulos Bluetooth - Ethernet, debido a que es la cara de nuestro sistema que se presenta a los usuarios finales que son los encargados de usar el sistema. Además de cubrir con todas las necesidades funcionales y de diseño mostradas en el capítulo tres, también se tiene que cubrir la parte visual, ofreciendo una interfaz fácil de usar y amigable a la vista.

Durante el desarrollo de esta parte del proyecto se hizo uso de una herramienta conocida como Microsoft Visual Studio 2010®, que nos ofrece la opción de desarrollar tanto aplicaciones de escritorio como aplicaciones web.

Para cubrir el punto de “amigable a la vista”, usamos WPF<sup>xxvii</sup> (*Windows Presentation Foundation*) para las aplicaciones de escritorio y Silverlight<sup>xxviii</sup> para las aplicaciones web. Tanto WPF como Silverlight tienen características para realizar interfaces de usuario elaboradas y que tengan un impacto en la aplicación como conjunto, realizando la interactividad con el usuario.

Debido al uso de tecnología .NET en este proyecto se decidió implementar la base de datos en Microsoft SQL Server, para proporcionar integración con Visual Studio. Y aprovechando SQL Server para la implementación de procedimientos almacenados para administrar la base de datos.

### Aplicación de Escritorio

En este apartado se hace el mapeo del diseño realizado en el capítulo tres a un lenguaje de programación. Cada uno de los casos de uso tiene su contraparte con alguna de las pantallas o una serie de pantallas dentro de la implementación final del sistema.

La primera pantalla es la principal que aparece en forma de menú indicando cuatro opciones que son: usuarios, teléfonos, módulos y red.

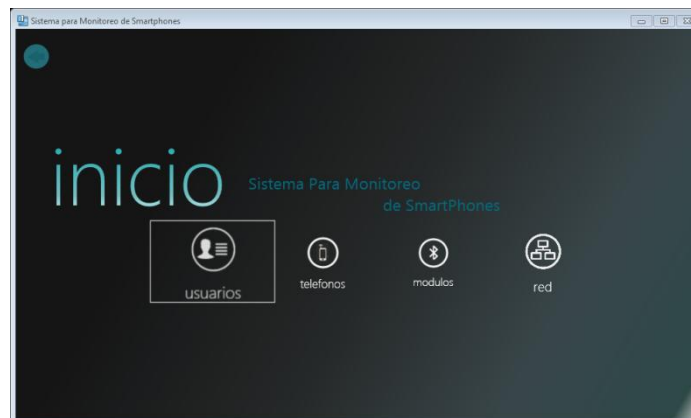


Figura 34. Se muestra la pantalla principal donde se realiza la administración del sistema

Las acciones de cada uno de estas opciones del menú ofrecen las siguientes alternativas:

---

<sup>xxvii</sup> Windows Presentation Foundation (WPF) proporciona a los programadores un modelo de programación unificado, en las que se incorpora la interfaz de usuario, multimedia y documentos, para más información visitar sitio de MSDN en internet

<sup>xxviii</sup> Silverlight es una herramienta de desarrollo de gran alcance para crear atractivas experiencias de usuario interactivas para aplicaciones web y móviles, para más información visitar sitio de MSDN en internet

## Aplicaciones de Usuario

- En usuarios, se podrá administrar los datos que el sistema maneja de los usuarios que tendrán derecho de ingresar al sitio web.
- En teléfonos, se podrá manejar la información de los smartphones a monitorear, y se podrá asignar a cada usuario solamente un smartphone.
- En módulos, se tendrá la información de cada uno de los módulos Bluetooth - Ethernet, aquí se le asigna a cada módulo la dirección IP y se almacenará la BT\_ADDR (dirección Bluetooth) de cada uno de los transceptores.
- En red, se lleva el control de la red que se formara con los módulos Ethernet – Bluetooth, aquí se asigna a un edificio cierta cantidad de módulos indicando en que sala se encuentra cada uno de estos.

En la figura 35 podemos apreciar la captura de la pantalla usuarios. Esta pantalla servirá para administrar los datos de los usuarios que tendrán acceso a la página web. Las acciones que se podrán realizar son agregar (para dar de alta un nuevo usuario), editar (para modificar los datos) y borrar (para dar de baja del usuario).

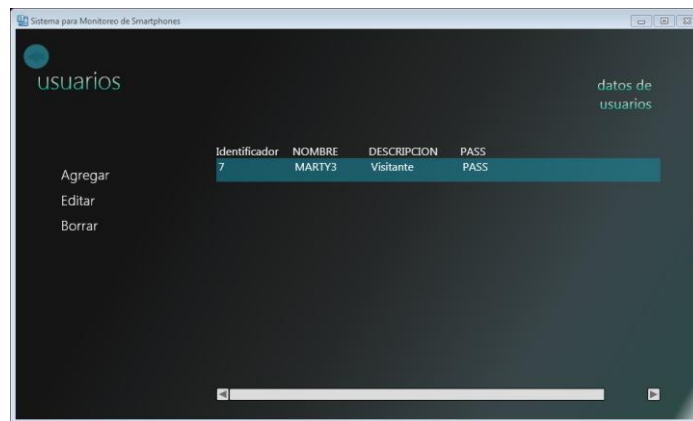


Figura 35. Se aprecia la pantalla de usuarios, en la parte derecha se puede apreciar la lista de usuarios que están registrados en el sistema y las acciones a ejercer sobre cada uno de estos

En la figura 36 se muestra que los datos que corresponden a un usuario, se tiene nombre que es una cadena de caracteres, tipo de usuario que puede ser administrador o visitante y la contraseña de ingreso al sistema. Toda esta información es enviada a la base de datos para que a través de un procedimiento almacenado se guarde la información.

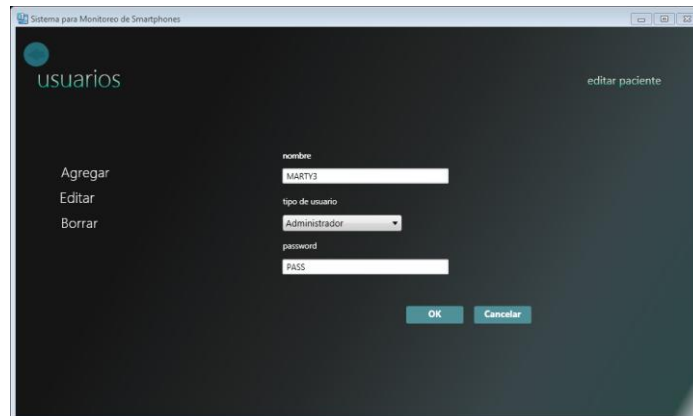


Figura 36. Se muestra los datos que maneja la pantalla de usuarios

La siguiente pantalla es la que corresponde a los teléfonos o smartphones a monitorear. Si el smartphone no está dado de alta aquí, no estará disponible para monitorearlo posteriormente. Las opciones a realizar sobre este conjunto de datos son agregar (dar de alta una nueva persona con un smartphone), modificar (hacer alguna modificación sobre los datos de la persona o el smartphone) y borrar.

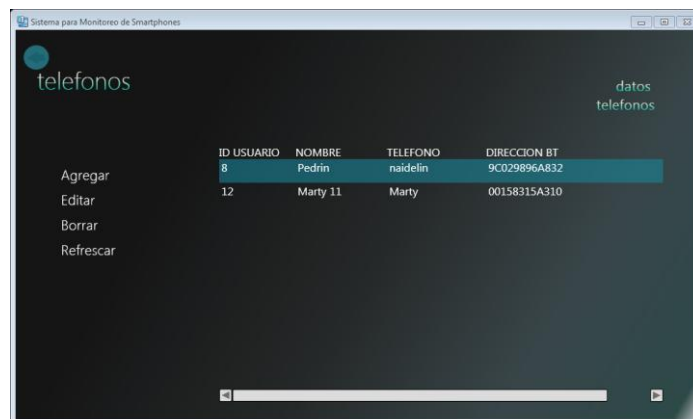
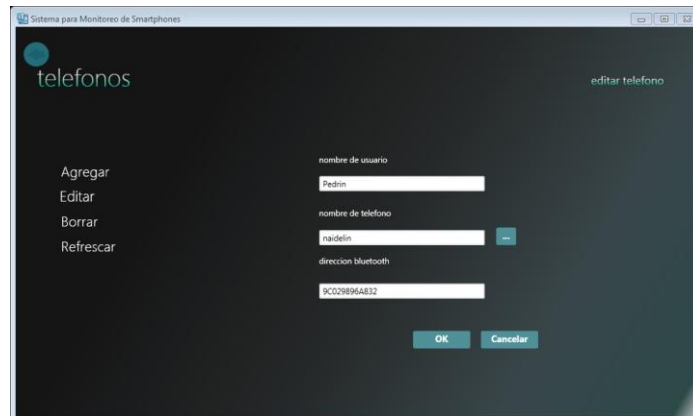


Figura 37. Se muestra la pantalla de teléfonos, en la parte derecha se puede observar la lista de todas las personas a monitorear y el smartphone asignado, en la parte izquierda se observa las opciones a realizar sobre este conjunto de datos

En la figura 38 se observa los campos que componen cada uno de los registros para un smartphone, tenemos nombre usuario compuesto de una cadena de caracteres y que servirá para almacenar el nombre de la persona a monitorear. Después nombre de teléfono que es el nombre con el cual identificamos al smartphone que es una cadena de caracteres y por último la dirección Bluetooth que corresponde a BT\_ADDR y es una cadena de caracteres de longitud doce.

## Aplicaciones de Usuario



Sistema para Monitoreo de Smartphones

telefonos editar telefono

Agregar  
Editar  
Borrar  
Refrescar

nombre de usuario  
Pedrin

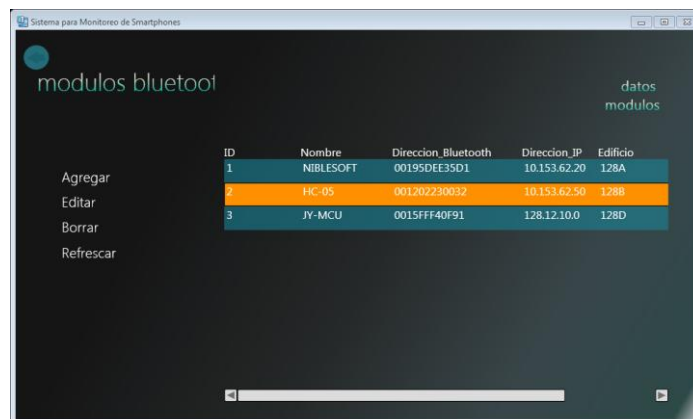
nombre de telefono  
haidelin

direccion bluetooth  
9C029894E832

OK Cancelar

Figura 38. Campos que se usan para cada una de las personas a monitorear

La siguiente pantalla se ha llamado módulos Bluetooth que se ve en la figura 39, en esta se hace la administración de los datos de todos los módulos Bluetooth - Ethernet. Las operaciones sobre este conjunto de datos son las correspondientes a agregar que servirá para agregar un nuevo módulo indicando en que sala y edificio se ubicara, editar para modificar los datos de cada módulo y borrar.



Sistema para Monitoreo de Smartphones

modulos bluetooth datos modulos

Agregar  
Editar  
Borrar  
Refrescar

ID	Nombre	Direccion Bluetooth	Direccion IP	Edificio
1	NIBLESOFT	00195DEE35D1	10.153.62.20	128A
2	HC-05	001202230032	10.153.62.50	128B
3	JY-MCU	0015FFF40F91	128.12.10.0	128D

Figura 39. Pantalla correspondiente a módulos Bluetooth, en la parte derecha se muestra la lista de todos los módulos Ethernet – Bluetooth registrados, de la parte izquierda las operaciones que se pueden realizar sobre este conjunto de datos

Los campos que conforman cada uno de los registros para un módulo Bluetooth son los siguientes: nombre de modulo que es nombre del transceptor Bluetooth, dirección Bluetooth que corresponde a la BT\_ADDR del transceptor Bluetooth, la dirección IP que estará asignada al a transceptor Ethernet para poder realizar el intercambio de información, y por último la sala y edificio donde estará ubicado el modulo.

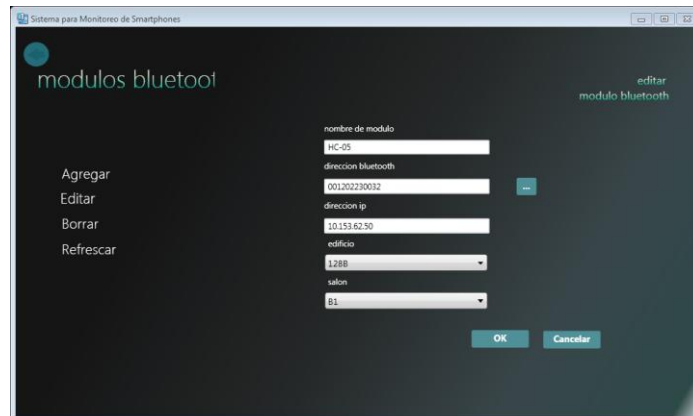


Figura 40. Se muestra cada uno de los atributos que se guardan cada vez que da de alta un nuevo módulo Bluetooth

La última pantalla es la red de módulos Bluetooth, en primera instancia se muestra todos los edificios involucrados en el monitoreo de smartphones, esto se puede apreciar en la figura 41. A partir de aquí el usuario puede seleccionar cualquier edificio para asignar los módulos Bluetooth que estarán presentes en dicho edificio.

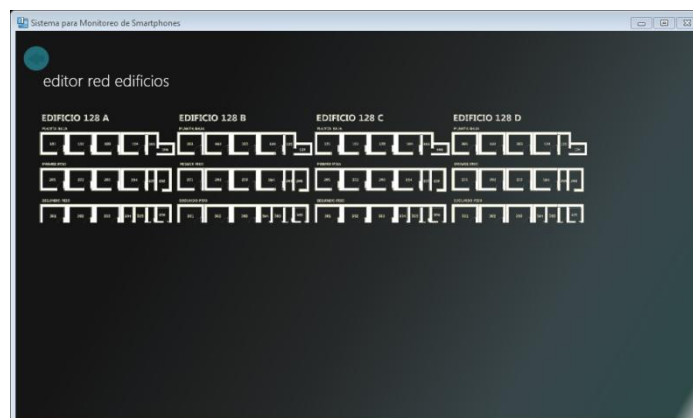


Figura 41. Pantalla que corresponde a la red, se puede apreciar el conjunto de edificios donde se ha instalado la red

En la figura 42 se puede observar, que hay cuatro módulos Ethernet – Bluetooth asignados al edificio 128B. Se hace esta representación gráfica de los módulos y edificios para facilitar la información al usuario final. En la parte inferior izquierda se puede observar los comandos que podemos ejecutar sobre un edificio que son: agregar (agrega un nuevo módulo al escenario del edificio), editar (aquí podemos hacer la asignación del elemento del escenario con un módulo que hayamos registrado con anterioridad) y borrar.

## Aplicaciones de Usuario



Figura 42. Se muestra a detalle un edificio, podemos observar los módulos Ethernet – Bluetooth, en color amarillo

En la figura 43 se observa la pantalla de editar modulo, aquí se puede realizar la asignación del elemento del escenario con un módulo registrado anteriormente. Esta pantalla permite visualizar información como a que sala pertenece, la dirección BT\_ADDR y la IP asignada al edificio.

Inmediatamente después se tiene el horario de asistencia, aquí se puede asignar un smartphone distinto por cada hora del día. Permitiendo flexibilidad por si la persona a monitorear tiene un conjunto de tareas a realizar en distintas salas del edificio o entre distintos edificios.



Figura 43. Detalle de la pantalla horario, se muestra la asignación de un smartphone diferente por cada día y hora

Toda la información que se almacena a través de todas las pantallas es importante y es utilizada por otras aplicaciones como el proceso que se encarga de juntar la información de la búsqueda a los smartphones o la página web que muestra los resultados del mismo monitoreo.

## Proceso de Búsqueda de Módulos Bluetooth - Ethernet

El proceso de búsqueda de módulos Bluetooth – Ethernet se encarga de recabar la información del monitoreo de todos los módulos que conforman la red. La intención de este proceso es que se ejecute de manera permanente en el servidor para que de manera automática y dependiendo de los horarios establecidos se envíe a cada módulo la instrucción de realizar la búsqueda.

A continuación se presente el algoritmo desarrollado para realizar esta tarea:

### Algoritmo 7. Monitoreo a Módulos Bluetooth – Ethernet

```

Entero retardo_Segundo = 60;
Booleano activo ← falso;
Carácter comando;
Proceso procesoMonitoreo;

Comando ← Leer_Caracter();

Si Comando es igual a Activar Monitoreo entonces
    Activo ← verdadero;
    Hacer i ← 0 hasta total de modulosBluetooth
        procesoMonitoreo ← nuevo Proceso(modulosBluetooth[i]);
        procesoMonitoreo.Iniciar();
    Fin Hacer
Fin Si

Si Comando es igual a Desactivar Monitoreo entonces
    Activo ← falso;
Fin Si

Si Comando es igual a cambiar retardo entonces
    retardo_Segundo ← Leer_Entero();
Fin Si
    
```

Tabla 20. Se presenta el algoritmo para realizar el monitoreo a los módulos

En la tabla 20 podemos observar que este algoritmo solo está compuesto de tres comandos. El comando cambiar retardo, que nos sirve para cambiar una variable que usaran los procesos que realizan el monitoreo. Después tenemos activar sondeo, en este comando se activara una bandera que le indicara a los procesos encargados del monitoreo cuando iniciar sus acciones, también se aprecia que dentro de este comando se realiza la creación de un proceso que corresponde únicamente a un módulo con el cual entablara comunicación. Por último tenemos al comando desactivar sondeo que solamente ponen en falso el valor de la bandera activo. No hay ninguna función que detenga los procesos anteriormente creados debido a que cada proceso tendrá acceso a la

## Aplicaciones de Usuario

información de nuestro proceso principal y se podrá validar cuando la variable activo es falsa o verdadera.

Para hacer más fácil la explicación del algoritmo en su totalidad se presenta en la tabla 21 el algoritmo que corresponde a las acciones que cada proceso realizara.

<b>Algoritmo 8. Proceso Comunicación modulo Bluetooth – Ethernet</b>
<pre>BasedeDatos bd; ModuloEthernetBluetooth minodo; Horario horario_actual; Hacer     Horario_actual ← bd.horario(hora actual, minodo, día de la semana);     Si Horario_actual es diferente de nulo entonces         escribir_Socket(Activar Monitoreo, minodo.IP);      Si(enviar_BTADDR_Smarphone(         horario_actual.smartphone.BT_ADDR, minodo)) entonces         hacer             monitorear_Smartphone(minodo.IP, horario_actual);             mientras (horario_actual.HORA_FIN &gt; hora actual y activo                 sea verdadero)      Si no         Proceso.dormir(3 minutos);     Fin Si     Fin Si Mientras (activo sea verdadero);</pre>

Tabla 21. Algoritmo correspondiente al proceso de monitoreo

En el algoritmo empezamos definiendo un elemento llamado base de datos que es una instancia de nuestra base de datos y un objeto módulo Bluetooth - Ethernet.

Se observa que el hilo de ejecución estará activo mientras la bandera activo sea verdadera, esta bandera se definió en el anterior algoritmo.

La primera instrucción que se ejecuta dentro de este ciclo es obtener de la base de datos el horario actual, esto corresponde a la hora de inicio de un turno, la hora final y el smartphone a monitorear durante este tiempo.

Inmediatamente después se le indica al módulo Bluetooth – Ethernet correspondiente que ponga en activo su bandera para el monitoreo después se hace el envío de la dirección del smartphone.

Y se entra en un nuevo ciclo donde se estará indicando al módulo que realice la búsqueda de smartphones. La función monitorear smartphone será la

encargada de escribir a la base de datos los resultados obtenidos. Este ciclo terminara cuando el tiempo del turno haya acabado.

Este proceso se repetirá indefinidamente hasta que en el proceso principal se asigne falso a la bandera activo.

## Página WEB

La página web desarrollada para este proyecto solo servirá para mostrar los resultados del monitoreo a smartphones en tiempo real. Toda la información será obtenida de la base de datos que previamente fue llenada por el proceso de monitoreo.



Figura 44. Página web del sistema, la primera pantalla corresponde al inicio de sesión de un usuario

En la figura 44 se puede apreciar la pantalla de inicio de sesión donde un usuario previamente registrado puede tener acceso a la información de la búsqueda de smartphones de nuestro sistema. Los únicos datos solicitados son el nombre de usuario y una contraseña que solamente el usuario a ingresar sabe.



Figura 45. Pantalla que muestra el monitoreo en tiempo real de smartphones desde la página web

## Aplicaciones de Usuario

En la figura 45 vemos la interfaz que el usuario tendrá disponible, en la parte superior se encuentra en forma de menú cada uno de los edificios que están dentro de la red, al dar clic sobre alguno, en la parte inferior se mostrara el escenario que corresponde a ese edificio junto con los módulos que tiene instalados. A partir de aquí la página se estará actualizando cada minuto para mostrarnos los cambios que se genere en la información, posiblemente el arribo de una persona, su partida o ausencia. El icono de cada módulo cambiara de color para indicarnos cada una de estas acciones de la siguiente manera:

- El amarillo corresponde para indicar que no se ha podido entablar comunicación con el modulo Bluetooth – Ethernet, esta señal nos servirá para corroborar si pasa algo físicamente con dicho modulo
- El rojo indica la ausencia del smartphone y la persona de la sala indicada
- El verde indica que el smartphone y la persona se encuentran en la sala indicada
- El color morado indica que en este momento nadie tiene que estar en la sala

## CONCLUSIONES

En el presente capítulo se mostró la manera en que se implementó el sistema a nivel de aplicaciones. Después de haber realizado un buen diseño de hardware la parte de software no se podía quedar atrás. Se hizo uso de herramientas que se contaban disponibles en el momento de la elaboración de este proyecto y que el desarrollador tenia previo conocimiento. Se observa que a través del uso para desarrollo de interfaces de usuario como WPF o Silverlight se ha obtenido un buen resultado final que cumple con las características planteadas en el párrafo introductorio de este capítulo. En el siguiente capítulo se realizara la integración de la parte de hardware y las aplicaciones de usuario para realizar las pruebas y obtener resultados.



---

# PRUEBAS DEL SISTEMA

En el presente capítulo se aborda las pruebas finales realizadas sobre el sistema. Primero se realiza un estudio de las distancias alcanzadas por los módulos Bluetooth, aunque la especificación nos indica que la distancia alcanzada por el transceptor Bluetooth es de 10 metros en la práctica este dato difiere mucho de la realidad debido a la interferencia provocada por paredes y objetos. Después se realizan las pruebas del sistema completo, para observar los resultados obtenidos por el mismo y si garantizan la asistencia de las personas.

## **Pruebas sobre el Módulo Bluetooth**

Como se mencionó en capítulos anteriores, el transceptor Bluetooth usado durante todo ese proyecto es de clase dos. Esto quiere decir que la señal tiene un alcance de 10 metros. Este dato nominal es el que el fabricante nos ofrece indicando su funcionamiento en un ambiente controlado, en otras palabras es la cobertura máxima garantizada que nos ofrece.

Pero viendo el funcionamiento del transceptor en el mundo real, los datos del alcance pueden diferir mucho dependiendo del ambiente donde se instale. Para una señal de radio no es lo mismo atravesar una pared de bloques, que una de tabla roca o peor aún un techo de concreto. Dependiendo de los

distintos materiales se obtienen alcances distintos y por ende resultados distintos.

La primera prueba que se realizó, consta de medir el alcance de la señal del transceptor Bluetooth. Para dicha prueba se construyeron pequeñas cajas de distintos materiales (Blocks, uniceL y madera). En lo que respecta a madera no hay una medida estándar por unidad. En uniceL, el material para las cajas se obtuvo de placas de 1 metro de largo por un metro de ancho y 1 cm. o 2 cm. de grosor. La medida de los blocks usados son de 20 cm. de altura, 40 cm. de longitud y 14 cm. de grosor. A continuación se presentan las dimensiones de cada una de estas cajas.

<b>Material</b>	<b>Dimensiones de la Caja</b>	<b>Grosor del Material</b>
<b>Madera</b>	Longitud: 25 cm. Anchura: 18 cm. Altura: 10 cm.	2 cm.
<b>UniceL</b>	Longitud: 19.5 cm. Anchura: 12 cm. Altura: 10 cm.	1 cm.
<b>UniceL</b>	Longitud: 30 cm. Anchura: 22 cm. Altura: 16 cm.	3 cm.
<b>Block</b>	Longitud: 40 cm. Anchura: 40 cm. Altura: 34 cm.	14 cm.

Tabla 22. Se muestran las dimensiones de cada una de las cajas creadas para medir el alcance de la señal Bluetooth.

Después se procedió a colocar el módulo Bluetooth dentro de cada una de estas cajas. La razón, es para asegurarnos que la señal de radio pase por al menos uno de los lados de la caja y garantizar que la señal atraviesa cada uno de los materiales en cuestión.

El procedimiento de esta prueba fue el siguiente:

- Se configura el transceptor Bluetooth en modo esclavo y se coloca dentro de una caja.
- Desde la computadora con otro transceptor Bluetooth conectado por puerto serie, se procede a la búsqueda.
- El paso anterior se realiza nueve veces más para tener un total de diez muestras.

## Pruebas del Sistema

En la siguiente tabla se muestran las pruebas realizadas a cada uno de los materiales, a distintas distancias. El módulo se colocó a 2.5, 5 y 10 metros de distancia de un segundo módulo que se encargó de obtener los resultados:

Material	Distancia	Numero de Intentos	Intentos Exitosos	Intentos Fallidos
Sin Obstáculos	2.5m	10	10	0
Sin Obstáculos	5m	10	10	0
Sin Obstáculos	10m	10	10	0
Madera de dos centímetros de grosor	2.5m	10	10	0
Madera de dos centímetros de grosor	5m	10	10	0
Madera de dos centímetros de grosor	10m	10	7	3
Unicel de un centímetro de grosor	2.5m	10	10	0
Unicel de un centímetro de grosor	5m.	10	10	0
Unicel de un centímetro de grosor	10m	10	5	5
Unicel de tres centímetros de grosor	2.5m	10	10	0
Unicel de tres centímetros de grosor	5m	10	9	1
Unicel de tres centímetros de grosor	10m	10	5	5
Bloques de 11.5 cm de grosor	2.5m	10	10	0
Bloques de 11.5 cm de grosor	5m	10	6	4
Bloques de 11.5 cm de grosor	10m	10	2	8

Tabla 23. Se muestran los resultados después de haber realizados muestras sobre el alcance del transeptor Bluetooth y diferentes materiales



Figura 46. Se muestra la caja de madera y las dos cajas de unicel utilizadas en la medición del alcance de la señal Bluetooth

De la tabla 23 se puede concluir que la señal Bluetooth garantiza la detección de smartphones en un radio de aproximadamente diez metros, con esto garantizamos el área de cobertura del sistema. A continuación, en base a los resultados anteriores se muestran una serie de gráficas que ayudaran a describir los resultados obtenidos.

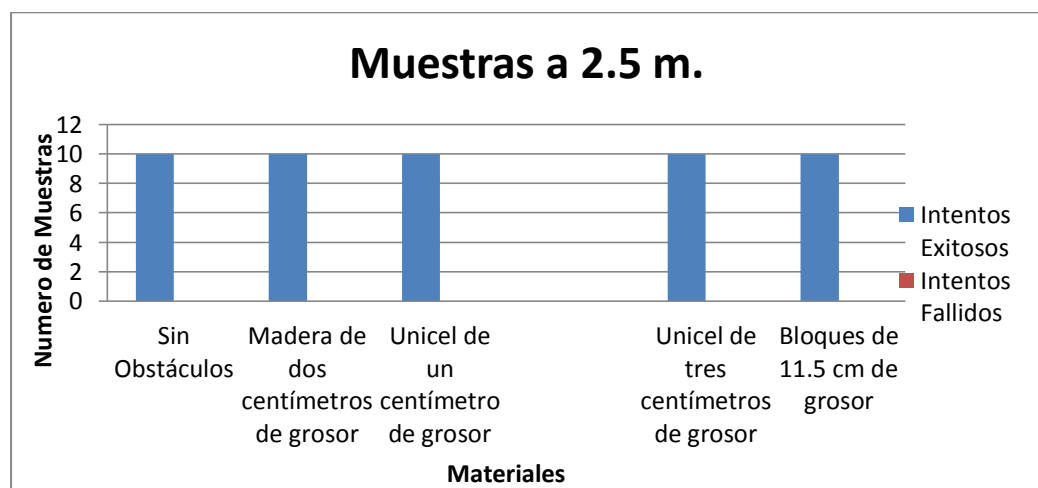


Figura 47. Se muestran los resultados obtenidos realizando pruebas con cinco materiales a una distancia de 2.5m entre transceptores Bluetooth

Es visible que a una distancia corta siempre se puede comunicar con un módulo Bluetooth como se observa en la gráfica de la figura 46, de los diez intentos realizados en los mismo diez se obtuvieron resultados positivos. Esto garantiza que a esta distancia siempre vamos a encontrar los smartphones de los trabajadores y el sistema cumplirá su cometido.

## Pruebas del Sistema

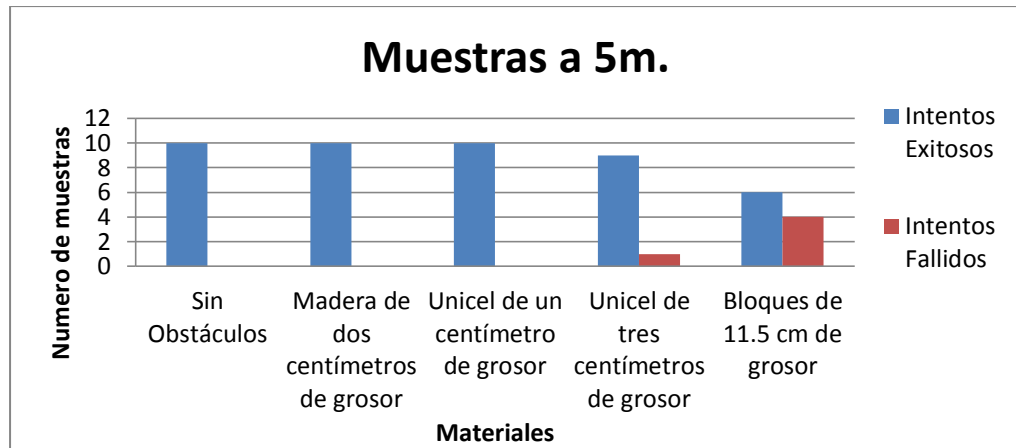


Figura 48. Se muestran los resultados obtenidos realizando pruebas con cinco materiales a una distancia de 5m entre transceptores Bluetooth

Ahora a una distancia de cinco metros en la gráfica de la figura 48 podemos observar que la madera y el unicel de un centímetro de grosor no impiden que la señal pase, pero los demás materiales impiden el paso de la señal en algunas muestras. El caso más significativo es cuando se usó blocks. De las 10 muestras, 6 dieron un resultado positivo mientras que las otras 4, a pesar de que en algunos intentos sí se encontró el dispositivo, se considera como falla debido al tiempo de respuesta. Si el módulo tardaba más de 30 segundos se considera que no se encontró el dispositivo.

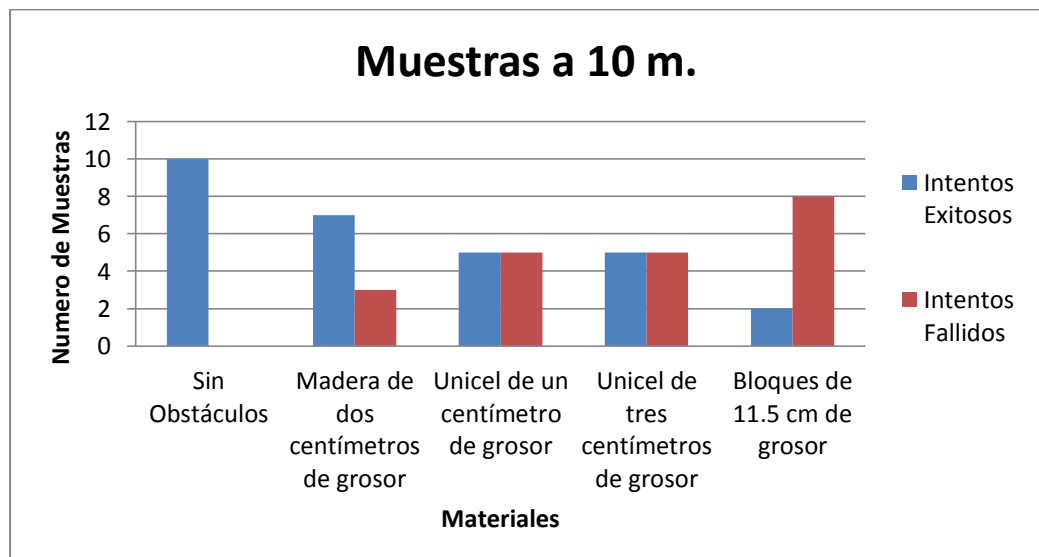


Figura 49. Se muestran los resultados obtenidos realizando pruebas con cinco materiales a una distancia de 10m entre transceptores Bluetooth

A una distancia de diez metros se observa que la señal no se encuentra garantizada en todos los intentos de búsqueda como se observa en la figura

49, un dato curioso es que el unicelel aparentemente aísla más la señal Bluetooth que la madera, para distancias de diez metros, mientras que con los bloques como era de esperarse solo dos intentos fueron exitosos. Como en el caso anterior se considera como búsqueda fallida, si la búsqueda excede más de los 30 segundos. A continuación se muestran resultados de alcance sobre la señal Bluetooth que se realizaron sobre una piconet. La instalación de la piconet se hizo en el edificio “140C” de la Facultad de Ciencias de la Computación

### EDIFICIO 140C

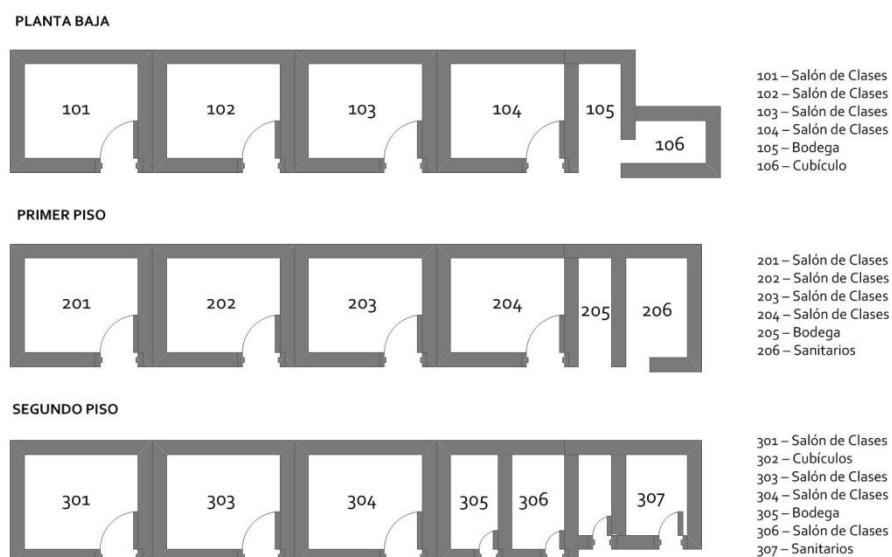


Figura 50. Plano del Edificio 140C donde se realizó las pruebas de la piconet

Grosor de Muros	20 cm. Aproximadamente
Número de Pisos	3
Numero de Salones	18
Materiales Usados en la Elaboración de los Muros	Tabiques
Materiales Usados en la Elaboración de los Techos	Concreto y malla

## Pruebas del Sistema

Posición Maestro	Posición Esclavo	Distancia Aproximada en metros	Tiempo de Búsqueda del Dispositivo(solo)	Tiempo de Búsqueda con Red Completa	Número Máximo de Muros a Cruzar	Numero Minimo de Muros a Cruzar	La señal pasa por algún muro falso	Numero de Muros Falsos que cruza la señal
103	102	5	4 seg.	8 seg.	1	1	No	0
103	104	5	2.5 seg.	8 seg.	1	1	No	0
103	101	10	3.1 seg.	7 seg.	2	2	No	0
103	203	5	3.7 seg.	7.3 seg.	1	1	No	0
103	202	8	-	-	1	1	No	0
103	201	15	-	-	2	1	No	0
103	203	5	3.9 seg.	8.3 seg.	1	1	No	0
103	204	8	-	-	1	1	No	0
203	102	7	2.8 seg.	8.5 seg.	2	1	No	0
203	103	5	3.1 seg.	8.1 seg.	1	1	No	0
203	104	8	4 seg.	8.5 seg.	2	1	No	0
203	201	15	-	-	2	2	No	0
203	202	5	2.5 seg.	7.1 seg.	1	1	No	0
203	204	5	3.2 seg.	7.5 seg.	1	1	No	0
203	205	9	-	-	2	2	Si	1
203	206	12	-	-	3	3	Si	1
203	302	8	4 seg.	7.2 seg.	2	1	No	0
203	303	5	3 seg.	8.5 seg.	1	1	No	0
203	304	8	3.2 seg.	7.7 seg.	2	1	No	0
303	202	8	4 seg.	7.6 seg.	2	1	No	0
303	203	5	3 seg.	7 seg.	1	1	No	0
8303	204	8	3.5 seg.	8 seg.	2	1	No	0
303	301	15	-	-				
303	302	5	3 seg.	8 seg.	1	1	No	0
303	304	5	2.9 seg.	7.3 seg.	1	1	No	0
303	305	7	5 seg.	8.1 seg.	2	2	Si	1

Tabla 24. Resultados relacionados con la prueba de una piconet en un edificio

Se observa que en salones contiguos la señal llega de manera correcta porque solamente tiene que atravesar sólo un muro. Cuando es necesario que la señal atraviese más de un muro, es posible todavía entablar comunicación dependiendo de la distancia de ambos transceptores. Si la distancia es mayor a los ocho metros y se tiene que atravesar más de un muro es posible que la comunicación no se garantice. Los muros falsos, hechos por lo regular de tabla roca, no son gran impedimento para que la señal viaje a través de ellos.

### Pruebas de Módulo Bluetooth y Comandos AT

Las siguiente serie de pruebas se realizó sobre los comandos AT y la manera en que el modulo Bluetooth respondía a estos. Se hizo una prueba sobre la búsqueda de Smartphones, de manera aislada, sin el uso de algún microcontrolador se conectó directamente el transceptor Bluetooth hacia una computadora mediante puerto RS-232, haciendo uso de una híper terminal se observaron los resultados.

```
at+rname?3cc2,43,60f108
+RNAME:Nokia 201
OK
```

Figura 51. Se muestra el comando AT para la búsqueda de dispositivos

En el primer escenario de prueba se indica al módulo buscar cada uno de los smartphones. El módulo debe de ser capaz de detectar todos los smartphones debido a que el Bluetooth de estos ha sido activado.

Smartphone a Buscar	Dirección	Resultado Búsqueda	Resultado Esperado
LG A200	70:05:14:9a:da:00	Encontrado	Encontrado
Nokia 201	3c:c2:43:60:f1:08	Encontrado	Encontrado
Samsung Galaxy SIII Mini	9c:02:98:96:a8:32	Encontrado	Encontrado

Tabla 25. Resultados obtenidos durante una búsqueda con los tres smartphones presentes

La tabla anterior muestra los resultados obtenidos indicando que el resultado corresponde con lo que se espera. En nuestro segundo caso de prueba se desactiva el Bluetooth de uno de los smartphones.

## Pruebas del Sistema

Smartphone a Buscar	Dirección	Resultado Búsqueda	Resultado Esperado
LG A200	70:05:14:9a:da:00	Encontrado	Encontrado
Nokia 201	3c:c2:43:60:f1:08	No encontrado	No encontrado
Samsung Galaxy SIII Mini	9c:02:98:96:a8:32	Encontrado	Encontrado

Tabla 26. Muestra los resultados de una búsqueda donde uno de los tres Smartphones no estaba presente

Ahora en esta segunda prueba de búsqueda observamos que los resultados corresponden a lo esperado. De esto se puede concluir que la búsqueda de smartphones se puede hacer de manera selectiva indicándole al transceptor que smartphones buscar, todo dependiendo de una lista anteriormente almacenada, en nuestro caso en la memoria del microcontrolador. A continuación se presenta un algoritmo para la búsqueda de smartphones sin importar cuantos sean estos, siempre y cuando previamente se encuentren almacenados en memoria.

Algoritmo. Búsqueda múltiple de Smartphones
<pre> Cadena smartphones[n]; Booleano Resultado[n]; Cadena aux; Entero i; Para cada i =0 hasta n Hacer     aux = Buscar(Smartphones[n]);     Si ( aux != "FAIL")         Resultado[n] = verdadero;     else         Resultado[n] = falso; Fin_Hacer                     </pre>

Tabla 27. Se muestra el algoritmo desarrollado para la implementación de la búsqueda de múltiples smartphones resultado de observar en comportamiento de búsqueda del transceptor Bluetooth

El algoritmo describe como sobre un conjunto de smartphones que están almacenados en un arreglo se realiza la búsqueda de manera individual y el resultado se almacena en un segundo arreglo, para posteriormente ser utilizado.

## Pruebas de Sistema Completo

Una vez que se realizó la integración del sistema, se prosiguió a realizar las pruebas del sistema en su totalidad para esto se realizaron mediciones en base a un horario hipotético.

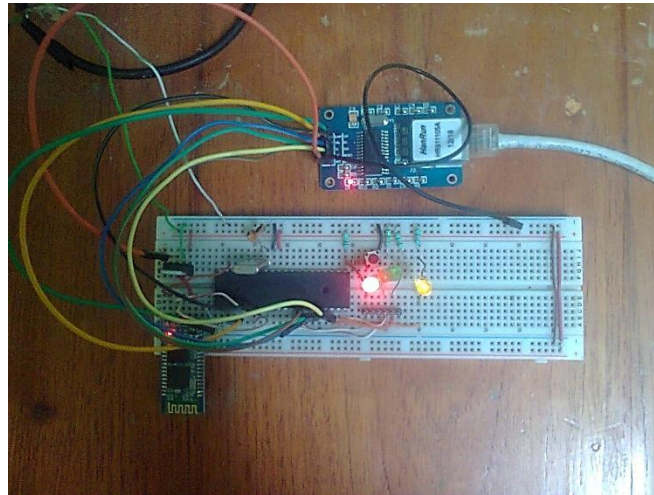


Figura 52. Implementación de la tarjeta controladora de la comunicación Bluetooth – Ethernet usada para las pruebas

Esta prueba se realizó para dos salas donde se colocó cada uno de los módulos Bluetooth – Ethernet. A continuación se presentan los horarios y los resultados por separado.

A la primera sala la nombraremos A1, que tiene el siguiente horario registrado y se obtuvieron los siguientes resultados.

Hora	Smartphone	Dirección
7:00 - 8:00	Smartphone1	3c:c2:43:60:f1:08
8:00 - 9:00	Smartphone2	9c:02:98:96:a8:32
9:00 - 10:00	Smartphone1	3c:c2:43:60:f1:08
10:00 - 11:00	Smartphone1	3c:c2:43:60:f1:08
11:00 - 12:00	Smartphone2	9c:02:98:96:a8:32

Tabla 28. Se describe el horario para el primer módulo

## Pruebas del Sistema

Smartphone	Hora Registrada Por El Sistema	Resultado del Sistema	Distancia Aproximada del Modulo	Resultado Esperado
Smartphone1	7:39:31	1	3 m.	1
Smartphone1	7:43:32	1	3 m.	1
Smartphone1	7:53:01	1	3 m.	1
Smartphone1	7:59:47	-1	3 m.	1
Smartphone2	8:18:06	0	-	0
Smartphone2	8:23:23	0	-	0
Smartphone2	8:28:24	0	-	0
Smartphone2	8:33:36	0	-	0
Smartphone2	8:59:15	0	-	0
Smartphone1	9:05:16	1	5 m.	1
Smartphone1	9:11:17	1	5 m.	1
Smartphone1	9:18:17	1	5 m.	1
Smartphone1	9:23:18	1	5 m.	1
Smartphone1	9:31:13	1	5 m.	1
Smartphone1	9:40:41	0	12 m.	0
Smartphone1	9:48:48	1	3 m.	1
Smartphone1	9:54:13	1	3 m.	1
Smartphone1	10:14:06	1	3 m.	1
Smartphone1	10:40:33	1	3 m.	1
Smartphone1	10:47:35	1	3 m.	1
Smartphone1	10:56:37	1	3 m.	1
Smartphone2	11:02:38	0	-	0
Smartphone2	11:07:38	0	-	0
Smartphone2	11:13:39	0	-	0
Smartphone2	11:20:39	0	-	0
Smartphone2	11:27:40	0	-	0
Smartphone2	11:33:41	0	-	0
Smartphone2	11:38:41	0	-	0
Smartphone2	11:42:42	0	-	0
Smartphone2	11:48:42	0	-	0
Smartphone2	11:54:58	0	-	0
Smartphone2	12:00:00	0	-	0

Tabla 29. Resultados de la búsqueda de dispositivos

En esta primera prueba, hicimos que el smartphone1 si acudiera a su lugar de trabajo mientras que el smarphone2 no lo hace. En la tabla 29 identificamos como asistencia un 1 y como falta un 0. Podemos ver la correspondencia

entre el resultado obtenido y los resultados esperados. Estos datos presentados solo son un subconjunto de los datos obtenidos, podemos observar peculiaridades como la que ocurrió a las 7:59, donde el sistema no pudo conectarse con el modulo pero a la siguiente prueba se pudo enlazar la comunicación.

Hora	Smartphone	Dirección
7:00 - 8:00	Smartphone1	3c:c2:43:60:f1:08
8:00 - 9:00	Smartphone2	9c:02:98:96:a8:32
9:00 - 10:00	Smartphone2	9c:02:98:96:a8:32
10:00 - 11:00	-	-
11:00 - 12:00	Smartphone1	3c:c2:43:60:f1:08

Tabla 30. Se presenta el horario de prueba para el segundo modulo

Smartphone	Hora Registrada Por El Sistema	Resultado del Sistema	Distancia Aproximada del Modulo	Resultado Esperado
Smartphone1	7:39:31	1	4 m.	1
Smartphone1	7:40:32	1	4 m.	1
Smartphone1	7:43:32	1	4 m.	1
Smartphone1	7:51:33	1	4 m.	1
Smartphone1	7:57:33	1	4 m.	1
Smartphone2	8:02:34	0	-	0
Smartphone2	8:07:35	0	-	0
Smartphone2	8:17:08	0	-	0
Smartphone2	8:23:09	0	-	0
Smartphone2	8:32:10	0	-	0
Smartphone2	8:59:15	0	-	0
Smartphone2	9:05:19	0	-	0

## Pruebas del Sistema

Continuación de tabla 31.

Smartphone	Hora Registrada Por El Sistema	Resultado del Sistema	Distancia Aproximada del Modulo	Resultado Esperado
Smartphone2	9:13:08	0	-	0
Smartphone2	9:22:36	0	-	0
Smartphone2	9:32:16	0	-	0
Smartphone2	9:41:45	0	-	0
Smartphone2	9:55:17	0	-	0
Smarphone1	11:07:08	1	7 m.	1
Smarphone1	11:17:12	1	7 m.	1
Smarphone1	11:23:13	1	7 m.	1
Smarphone1	11:35:15	1	7 m.	1
Smarphone1	11:43:16	1	7 m.	1
Smarphone1	11:48:16	1	7 m.	1
Smarphone1	11:54:17	1	7 m.	1
Smarphone1	11:59:17	1	7 m.	1

Tabla 31. Resultados obtenidos en la búsqueda en la sala B1

En las tablas 30 y 31 se muestran los datos relacionados con la búsqueda de smartphones dentro de la sala B1. También en este experimento solo hicimos que el smartphone1 acudiera a trabajar mientras que smartphone2 no, por lo cual tenemos registrada la asistencia e inasistencia de da uno. Durante esta prueba también se generaron desconexiones con el modulo Bluetooth – Ethernet pero el sistema pudo recuperarse y seguir realizando la búsqueda. En base a esto, podemos decir que el sistema se comporta de la manera esperada, indicando la inasistencia de las personas y la ausencia temporal de estas. Ademes hay casos donde se pierde la conexión con los módulos Bluetooth pero el sistema es capaz de recuperarse y continuar trabajando.

## CONCLUSIONES

En este capítulo se ha presentado datos que sustentan la funcionalidad del sistema desarrollado. Se han desarrollado pruebas, tanto de manera individual como colectiva de cada una de las partes del sistema. Con las pruebas realizadas en la última parte de este capítulo observamos que el sistema está cumpliendo con el propósito planteado al inicio de este trabajo de investigación. Por su puesto es necesario mejorar el sistema pero como primera etapa el sistema cumple totalmente con las necesidades requeridas.



---

# CONCLUSIONES Y TRABAJO A FUTURO

Después de realizar este proyecto y haber cubierto cada una de las etapas consideradas en la definición de problema, podemos concluir que podemos realizar un sistema basado en la detección de smartphones para controlar la asistencia de un área laboral.

El uso de microcontroladores nos facilitó demasiado la programación de los módulos Bluetooth – Ethernet, debido al uso de un lenguaje de programación de alto nivel como lo es C. A través de C y las librerías diseñadas por Microchip para el control del transceptor Ethernet, se hizo uso de cada una de las funciones del transceptor. Además nos permitió implementar una librería dedicada para la comunicación con el transceptor Bluetooth por comunicación serial. Concluimos que fue una buena elección un microcontrolador para realizar la parte del sistema empotrado.

Para finalizar podemos concluir que el sistema respondió a lo que se esperaba. Y esto lo podemos observar en el capítulo seis donde se realizaron una serie de pruebas sobre el transceptor Bluetooth de manera individual y sobre la implementación final del sistema. A través de los datos obtenidos podemos decir que el sistema cumple con los requisitos de garantizar la permanencia y asistencia de un grupo de personas en su área trabajo. Además se tiene la página web desde la cual podemos observar en tiempo real quien se encuentra

laborando. Se desarrolló una serie de aplicaciones con interfaces amigables al usuario para no solamente cumplir con la parte funcional del sistema, sino también abarcar la parte estética del mismo.

Una de las principales ventajas de este sistema comparado con otros es que nos puede ofrecer información estadística real, sobre la asistencia de las personas y la permanencia de éstas en su lugar de trabajo. A comparación de métodos tradicionales (mencionados en el capítulo 1), estos solo tienen información de la posible asistencia de las personas pero personas deshonestas podrían firmarse en el sistema y después abandonar el área de trabajo, cosa que nuestro sistema no permite ya que constantemente se encuentra monitorizando a las personas (lleva un control de la permanencia).

Una posible desventaja sería que una persona deshonesto dejara su smartphone dentro de su área de trabajo y así poder burlar el sistema. Pero definitivamente esta variable no puede ser controlada por el sistema ya que entra en juego la honestidad y la integridad de las personas a las cuales está dirigida el sistema.

### **Trabajo a Futuro**

Independientemente de haber concluido en su totalidad el sistema, mirando de manera autocrítica se puede decir que todavía hay aspectos por mejorar. Uno sería referente a la alimentación del módulo Bluetooth – Ethernet, en el proyecto se hace de a través de un eliminador que nos ofrece un voltaje de 5 volts. Esto implica que tenemos que conectar el eliminador a la corriente eléctrica, para hacer más portátil el sistema se puede alimentar a través de baterías de litio o haciendo uso de alimentación sobre Ethernet, esto nos ofrece una gran ventaja ya que se podría usar el mismo cableado que tenemos para la comunicación Ethernet solo tenemos que garantizar que los switches o routers usados soporten esta función.

Otros aspecto que se podría mejorar es la comunicación Ethernet, al implementar una red LAN con los transceptores Ethernet conlleva al uso de cable UTP, para evitar usar mucho cable podría sustituir el transceptor Ethernet actual por un transceptor que permita implementar WIFI para tener un sistema totalmente inalámbrico.

## BIBLIOGRAFIA

- <sup>1</sup> Comision Federal de Telecomunicaciones. (Julio de 2012). *Cofetel*. Recuperado el 17 de 01 de 2013, de Cofetel: <http://www.cft.gob.mx:8080/portal/>
- <sup>2</sup>World Bank. (2012). *worldbank.org*. Recuperado el 20 de 01 de 2013, de worldbank.org: <http://www.worldbank.org/ict/IC4D2012>
- <sup>3</sup>Radiomovil Dipsa SA de CV. *Telcel*. Recuperado el 18 de 01 de 2013, de Tece:<http://www.telcel.com/portal/equipos/equiposResult.do?hits=&offset=0&tipoTerminal=&historico=&tipoPlan=&marca=&tecono=&caract=Bluetooth&queryEq=>
- <sup>4</sup> Shafie, E., Ai-Ajlan, A., Aldrawiesh, K., Bajahzar , A., & Ai-Saawy, A. (2011). *Online Registration System*. Reino Unido: De Montfort University.
- <sup>5</sup>Zonggang, He, Junsheng, Zheng (2010). *Design and Implementation of Student Attendance Management System Based on MVC*. China.
- <sup>6</sup>Moseley, Kay (2010). *Access Control and Monitoring in a Prison Environment*. Reino Unido.
- <sup>7</sup>Anil J. Patel (2005). *The Impact of Bluetooth on Networking*. Estaos Unidos: St. Edward's University.
- <sup>8</sup> Jamil, T. (2011). *Automatic Attendance Recording System Using Mobile Telephone*. Serbia: IEEE.
- <sup>9</sup> Shibata, H. (2005). *An Attendance Management System Using Mobile Phones*. Japón: Universidad de Kansai.
- <sup>10</sup> Li, T.-C., Wu, H.-W., & Wu, T.-S. (2012). *The Study of Biometric Technology Applied in Attendance Management Systems*. Taiwan: St. John's University.
- <sup>11</sup> González, E., Argones, E., & Alba, J. (2010). *Multimodal Biometrics-based Student Attendance Measurement in Learning Management System*. España: Universidad de Vigo.
- <sup>12</sup> Millert, B., & Bisdikian, C. (2001). *Bluetooth Revealed*. Prentice Hall.
- <sup>13</sup>Mettala , R. (1999). *Bluetooth Protocol Architecture*. SIG.
- <sup>14</sup>Hopkins, B., & Anthony, R. (2003). *Bluetooth for Java*. Apress.

- <sup>15</sup>Team, D. S. (2002). *WAP, Bluetooth and 3G Programming*. Hungry Minds.
- <sup>16</sup>Rudolph, L. (2007). *Bluetooth Scentials for Programmers*. Cambridge University Press.
- <sup>17</sup> Ching-Fang, Hsu. Shu-Ming, Hsu (2006). *A Novel Hold-Mode-based Adaptive Inter-Piconet Scheduling Algorithm in Bluetooth Scatternets*. Taiwan: National Cheng Kung University.
- <sup>18</sup>Yang Zeng, Ning Han, (2007). *Hybrid UWB Multi-Piconet Network Interference Modeling and Performance Enhancement by 6-Bands MB-OFDM*. Corea: Inha University.
- <sup>19</sup>Suman Saha, Mitsuji Matsumoto (2006). *An Inter-Piconet Scheduling Algorithm for Bluetooth Scatternets*. Japón: Waseda University.
- <sup>20</sup>Stefano Basagni, Maurizio A. Nanni, Chiara Petrioli (2006). *Flow-fair Intra-Piconet Scheduling for Communications in Personal Area Networks*. Italia: Universit`a di Roma.
- <sup>21</sup>Gerhmann, C., & Persson, J. (2004). *Bluetooth Security*. Artech House.
- <sup>22</sup>Si Alhir, S. ( 2002). *Guide to Applying the UML*. Springer.
- <sup>23</sup>Gaudiozzo, Victor (2010). *Foundation Expression Blend 4 with Silverlight*. Springer.
- <sup>24</sup>Microchip (2007). *PIC18F2455/2550/4455/4550 Data Sheet*. Microchip Technology Inc.
- <sup>25</sup>LoCasio, Ted (2008). *Microsoft Expression Design on Demand*. Pearson Education.
- <sup>26</sup>Holden, Greg (2008). *Introducing Microsoft Expression Studio: Using Desing, Web, Blend, and Media to Create Professional Digital Content*. Course Technology.

---

# Apéndice A: Principales Comandos AT

A continuación se realiza un listado de los comandos AT más utilizados. Los comandos AT como se mencionó, sirven para la configuración de características especiales del transceptor Bluetooth y para entablar comunicación.

Comando	Resultado	Argumento
AT	OK	Ninguno

Este primer comando sirve para verificar que exista comunicación serial con el modulo. Si se obtiene la respuesta por parte del módulo, se garantiza que la comunicación es correcta.

Comando	Resultado	Argumento
AT+VERSION?	+VERSION<Parametro1> OK	Parametro1, es la versión de software del modulo Bluetooth.

## Principales Comandos AT

El comando anterior solo nos ofrece la versión de software de transceptor Bluetooth desarrollado por el fabricante.

Comando	Resultado	Argumento
AT+ORGL?	OK	Ninguno
Comando	Resultado	Argumento
AT+INIT	1. OK 2. FAIL	Ninguno

Este comando se encarga de restaurar los valores por default del transceptor Bluetooth. Configurándolo de la siguiente manera:

- Clase de Dispositivo: 0
- Código de Búsqueda: 0x009e8b33
- Modo de Dispositivo: Esclavo.
- Puerto Serie: 38400 bps, un bit de parada, sin paridad.
- Código de Emparejamiento: "1234"

Comando	Resultado	Argumento
AT+ADDR?	+ADDR:<Parametro1> OK	Parametro1: BT_ADDR (dirección Bluetooth)

Este comando nos devuelve el valor de la dirección Bluetooth del transceptor. Se obtiene el dato en el formato NAP, UAP, LAP. En valores hexadecimales.

Comando	Resultado	Argumento
AT+RNAME? <Parametro1>	1. +RNAME: <Parametro2> OK 2. FAIL	Parametro1: dirección del dispositivo remoto a conectarse Parametro2: nombre del dispositivo remoto

El comando RNAME, recibe como parámetro la dirección del dispositivo remoto a monitorear, si el dispositivo está dentro del radio del transceptor entonces devuelve el nombre del dispositivo, en caso de no estar presente responde con un FAIL, para ejecutar este comando no es necesario realizar una conexión.

Comando	Resultado	Argumento
AT+ROLE= <Parametro1>	OK	Parametro1: 0 – Esclavo 1 – Maestro 2 – Puente Cíclico
AT+ROLE?	+ROLE:<Parametro2> OK	

El comando ROLE, nos ofrece dos variantes. En la primera podemos asignar a un transceptor la tarea de esclavo, maestro o puente cíclico. En el segundo caso podemos consultar en qué modo se encuentra el transceptor.

Comando	Resultado	Argumentos
AT+UART=<Parametro1>, <Parametro2>, <Parametro3> AT+UART?	Ok  +UART: <Parametro1>, <Parametro2>, <Parametro3> Ok	Parametro1 es la velocidad de transmisión, parámetro 2 el bit de parada y por último el bit de paridad

El comando UART también tiene dos variantes. En la primera se configura el módulo UART del transceptor. Se asignan los valores de velocidad de transmisión, bit de parada y bit de paridad. En la segunda opción se obtienen los resultados para hacer la consulta.

Comando	Resultado	Argumento
AT+INIT	3. OK 4. FAIL	Ninguno

El comando INIT indica al transceptor que tiene que prepararse para realizar la conexión con otro dispositivo remoto.

Comando	Resultado	Argumento
AT+PAIR=<Parametro1>, <Parametro2>	3. Ok 4. FAIL	Parametro1: dirección del dispositivo remoto, Parametro2 tiempo de espera para el emparejamiento

En comando PAIR se ejecuta el emparejamiento de ambos dispositivos, en este proceso es cuando se entra en el estado de página y se negocia cada cuanto tiempo se realizaran los saltos en frecuencia y la manera de determinar estos saltos.

Comando	Resultado	Argumento
AT+LINK=<Parametro1>	3. Ok 4. FAIL	Parametro1: dirección del dispositivo remoto a conectarse

En el comando LINK se realiza la conexión completa entre ambos dispositivos una vez que este proceso se ha ejecutado de manera exitosa,

## Principales Comandos AT

ambos dispositivos podrán realizar intercambio de información hasta que alguno de los dos rompa el enlace.

Comando	Resultado	Argumento
AT+DISC	6. +DISC:SUCCESS 7. +DISC:LINK_LOSS 8. +DISC:NO_SLC 9. +DISC:TIMEOUT 10. +DISC:ERROR	Ninguno

El comando DISC, se ejecuta desde el modulo maestro que está llevando el control de la comunicación, el enlace puede terminar de cinco maneras distintas, exitosamente, se perdió la conexión, no\_slc, se agotó el tiempo de espera y por algún error.

---

# Apéndice B: Firmware Tarjeta Bluetooth – Ethernet

En este apéndice se muestra el código fuente del firmware de la tarjeta Bluetooth – Ethernet. La implementación del programa se hizo en un lenguaje de alto nivel. Se hizo uso del compilador PIC C. Todo el código fuente fue escrito en C para microcontroladores. Valiéndonos de herramientas y librerías disponibles por el compilador se hizo este firmware.

```
/*  
PROGRAMA : MASTER RAIZ V3.0  
ESTE MASTER CONTIENE UN TRANSCPTOR BLUETOOTH PARA REALIZAR  
LA BUSQUEDA DE SMARTHONES Y UN TRANSCPTOR ETHERNET EL CUAL  
LE LLEGARA TODA LA INFORMACION DE LA RED Y LA ENVIARA A UN  
CLIENTE  
MEDIANTE UN PUERTO ETHERNET  
EN ESTE PROGRAMA SE CORRE UN SERVIDOR TCP EN LA  
DIRECCION  
IP : 169.254.42.111  
GATEWAY : 169.254.42.1  
MASCARA DE SUBRED : 255.255.0.0  
Microchip VendorID, MAC : 00-04-A3-XX-XX-XX  
MAC ADDRESS : 00-04-A3-00-00-01
```

```
*/
```

## Firmware Tarjeta Bluetooth - Ethernet

```
#INCLUDE <18F4550.h>
#FUSES
HSPLL,MCLR,NOWDT,NOPROTECT,NOLVP,USBDIV,PLL5,NODEBUG,CPUDIV
1,VREGEN // Ocupa cristal 4Mhz = pll1, cambiar a PLL5
para ocuparlo con 20Mhz.
#USE delay(clock=48000000)
#USE RS232(xmit=PIN_C6,rcv=PIN_C7,BAUD=38400,BITS=8,PARITY
=N,STOP =1) // puerto serie para la comunicacion con el
modulo BT

//DEFINICION DE VARIABLES GLOBALES

char c,cont,I;
int TAM;

//*****
char DIRRECCION_SONDEO[13] = "-----";
boolean ACTIVO = FALSE,COM_PUENTE= false,B2= false,result =
false;

//*****
char MIDIRECCION[13] = "00195dee35d1"; // ES EL NIBLESOFT

char BUFFERD[40];
char miBUFFER[60];
unsigned int32 CONTEOTMR2=0;
//"001202230032"; // ES EL HC
static char content[250];
#define ON output_high
#define OFF output_low

// Conexión entre el PIC18F4550 y el Modulo ENC28J60 (Se
conecta directamente sin adaptador de voltajes).
// Se adjunta el datasheet del Modulo ENC28J60.

#define PIN_ENC_MAC_SO PIN_D4 // Conectar con PIN
MISO del ENC28J60. *original era c7 pero interfiere con
serial se cambia por d4
#define PIN_ENC_MAC_SI PIN_B0 // Conectar con PIN
MOSI del ENC28J60. *
#define PIN_ENC_MAC_CLK PIN_B1 // Conectar con PIN SCK
del ENC28J60. *
#define PIN_ENC_MAC_CS PIN_B3 // Conectar con PIN CS
del ENC28J60. *
#define PIN_ENC_MAC_RST PIN_B5 // Conectar con PIN RST
del ENC28J60.*
#define PIN_ENC_MAC_INT PIN_D2 // Conectar con PIN INT
del ENC28J60.*
#define PIN_ENC_MAC_WOL PIN_D3 // Conectar con PIN WOL
del ENC28J60.
```

```

//=====PUERTOS=====
#define AT PIN_D0 // pin que controla el
modo AT *****CAMBIAR POR EL QUE
MEJOR CONVenga**
#define CONFIGURACION PIN_B7 // SALIDA LED QUE INDICA
CUANDO ESTAMOS EN MODO CONFIGURACION
#define INDICADOR PIN_B6 // SALIDA LED QUE INDICA
CUANDO ESTAMOS EN MODO SONDEO
//=====

//Protocolos a utilizar.

#define STACK_USE_MCPENC TRUE
#define STACK_USE_ARP TRUE
#define STACK_USE_ICMP TRUE
#define STACK_USE_TCP TRUE

#include "tcpip/stacktsk.c"
#define MY_PORT 20000 // aqui se asigna el puerto en
nuestro caso usaremos el puerto libre 20000

int8 MYTCPSocket=INVALID_SOCKET;
#include <math.h>
#include "LIBRERIAS/utils.c"
#include "LIBRERIAS/direccionesEEPROM.c"
#include "LIBRERIAS/comunicaciones.c"

// Microchip VendorID, MAC: 00-04-A3-XX-XX-XX

void MACAddrInit(void)
{
    MY_MAC_BYTE1=0x00;
    MY_MAC_BYTE2=0x04;
    MY_MAC_BYTE3=0xA3;
    MY_MAC_BYTE4=0x00;
    MY_MAC_BYTE5=0x00;
    MY_MAC_BYTE6=0x02;
}

void IPAddrInit(void)
{
    /* normal cuando conecto directo a pc
    MY_IP_BYTE1=169;
    MY_IP_BYTE2=254;
    MY_IP_BYTE3=42;
    MY_IP_BYTE4=111;

    //Puerta de Enlace
    //169.254.42.1
    MY_GATE_BYTE1=169;
    MY_GATE_BYTE2=254;
    MY_GATE_BYTE3=42;

```

## Firmware Tarjeta Bluetooth - Ethernet

```
MY_GATE_BYTE4=1;

//Mascara de Subred
MY_MASK_BYTE1=255;
MY_MASK_BYTE2=255;
MY_MASK_BYTE3=0;
MY_MASK_BYTE4=0;
*/
//mod 1 10.153.62.20 ethernet grnade
//mod 2 10.153.62.50
MY_IP_BYTE1=192;
MY_IP_BYTE2=168;
MY_IP_BYTE3=1;
MY_IP_BYTE4=20;

//Puerta de Enlace
//169.254.42.1
MY_GATE_BYTE1=192;
MY_GATE_BYTE2=168;
MY_GATE_BYTE3=1;
MY_GATE_BYTE4=254;

//Mascara de Subred
MY_MASK_BYTE1=255;
MY_MASK_BYTE2=255;
MY_MASK_BYTE3=255;
MY_MASK_BYTE4=0;
}

void MYTCPput(char c)
{
    TCPput(MYTCPsocket, c);
}

void MYTCPtask(void)
{
    static enum {ST_CONNECT=0, ST_WAIT_CONNECT=1, ST_GET=2,
ST_PUT=3, ST_DISCONNECT=4} state=0;
    static TICKTYPE timeout_counter;
    //static char lc, lc2;

    if (MYTCPsocket==INVALID_SOCKET)
        state=ST_CONNECT;
    else if (!TCPIsConnected(MYTCPsocket))
        state=ST_WAIT_CONNECT;
    else if (TickGetDiff(TickGet(), timeout_counter) >
TICKS_PER_SECOND*60)
        state=ST_DISCONNECT;

    switch(state)
    {
        case ST_CONNECT:
            MYTCPsocket=TCPListen(MY_PORT);
            if (MYTCPsocket!=INVALID_SOCKET)
            {
```

```

        //printf(usb_cdc_putc, "\r\nHTTP: Listening");
        state=ST_WAIT_CONNECT;
        timeout_counter=TickGet();
    }
    break;

    case ST_WAIT_CONNECT:
        timeout_counter=TickGet();
        if (TCPIsConnected(MYTCPSocket))
        {
            state=ST_GET;
            //printf(usb_cdc_putc, "\r\nHTTP: Connected");
        }
        break;

    case ST_GET:
        if (TCPIsGetReady(MYTCPSocket))
        {
            while (TCPGet(MYTCPSocket, &c))
            {
                switch(c)
                {
                    case 'a': // prendemos un led
                        ACTIVO = TRUE;
                        ON(INDICADOR);
                        state = ST_DISCONNECT;
                        break;

                    case 'b': // apagamos led

                        LEER_CADENA_SOCKET(); // LEEEMONS UNA
CADENA ENTERA DESDE EL SOCKET
                        for(I=0;I<12;I++)
                            DIRRECIION_SONDEO[I] = content[I+1];
// pasamos la direccion al bufferd para ejecutar la funcion
de busqueda

                        state = ST_PUT;
                        break;

                    case 'c':
                        state=ST_PUT;
                        //aca solo se enviaria el dato pero
aqui sondearemos
                        break;

                    case 'd':
                        ACTIVO = FALSE;
                        OFF(INDICADOR);
                        state = ST_DISCONNECT;
                        break;

                    case 's':
                        state = ST_DISCONNECT;
                        break;
                }
            }

```

## Firmware Tarjeta Bluetooth - Ethernet

```
    }
}
break;

case ST_PUT:
    if (TCPIsPutReady(MYTCPSocket))
    {
        //printf(usb_cdc_putc, "\r\nHTTP: Putting Page
Response");
        // solo mandamos una cadena
        switch(c)
        {
            case 'b':
                //TCPPut(MYTCPSocket,12);
                //for(c = 0; c<12; c++)
                //    TCPPut(MYTCPSocket,
DIRRECCION_SONDEO[c]);
                TCPPut(MYTCPSocket,cont); //comentar si
noquieres recibir un caracter extraño por que c# me manda el
tamaño de la cadena
                for(c = 0; c<cont; c++)
                    TCPPut(MYTCPSocket, content[c]);

                TCPFlush(MYTCPSocket);
                state = ST_DISCONNECT;

            break;

            case 'c':
                if(result)
                {
                    //escribir 1
                    TCPPut(MYTCPSocket, '1');
                }
                else
                {
                    //escribir 2
                    TCPPut(MYTCPSocket, '0');
                }

                TCPFlush(MYTCPSocket);
                state = ST_DISCONNECT;
            break;
        }
    }
break;

case ST_DISCONNECT:
    if (TCPIsPutReady(MYTCPSocket))
    {
        //printf(usb_cdc_putc, "\r\nHTTP: Force
Disconnect");
        TCPDisconnect(MYTCPSocket);
    }
}
```

```

        state=ST_WAIT_CONNECT;
    }
    break;
}
}

#int_TIMER2
void interrupcion_timer2 ()
{
    set_timer2(0);
    CONTEOTMR2++;

    if(CONTEOTMR2==7353)//40 seg
    {
        CONTEOTMR2 = 0;
        COM_PUENTE = TRUE;
        //ACTIVAR LA BANDERA PARA QUE SE HAGA LA INSTRUCCION
    }
}

void main(void)
{
    ON(AT); //ACTIVAMOS EL MODO AT
    guardarDireccion(0,MIPUENTE); // guardamos en la
    posicion 0 de la memoria eeprom
    ON(INDICADOR);
    OFF(CONFIGURACION);

    MACAddrInit();
    IPAddrInit();
    StackInit();

    ON(CONFIGURACION);
    OFF(INDICADOR);
    HABILITAR_TIMER2(TRUE);
    while(TRUE)
    {
        StackTask();
        MYTCPTask();

        if(COM_PUENTE && ACTIVO)
        {

            result = CHECAR_DISPOSITIVO();

            if(B2)
            {
                B2 = FALSE;
                ON(CONFIGURACION);
            }
            else

```

## Firmware Tarjeta Bluetooth - Ethernet

```
        {
            B2 = TRUE;
            OFF(CONFIGURACION);
        }

        COM_PUENTE = FALSE;
    }
}
}
```

A continuación el código que corresponde a la librería de comunicaciones. Es en esta librería donde se implementan las funciones para la comunicación Bluetooth.

```
// ARCHIVO DE LOS COMANDOS DE COMUNICACION DEL
// NODO MASTER

//primer funcion es que cheque la direccion del master
int CHECAR_DIRECCION_MIA()
{
    if(CADENAEQUALS(MIDIRECCION,BUFFERD,12))
        return TRUE;

    return FALSE;
}

//FUNCION QUE LEE UNA CADENA DE UN SOCKET
void LEER_CADENA_SOCKET()
{
    cont =0; //EL TAMNAÑO DE LA CADENA
    TAM = 0; //EL DATO QUE SE VA A LEER

    while (TCPGet(MYTCPSocket, &TAM))
        content[cont++] = TAM;
}

void escribir_ArregloSerial(char cadena[],int tam)
{
    int I;
    for(I = 0; I< tam; I++)
        putc(cadena[I]);
}

void leer_CadenaSerial(char cadena[],int *tam)
{
    *tam = 0;
    do
    {
        {
            cadena[*tam++] = getch();
        }while(cadena[*tam-1]!=10 && *tam < 40);
    }
}

void leer_CadenaSerial()
```

```

{
TAM = 0;
do
{
miBUFFER[TAM++] = getch();
}while(miBUFFER[TAM-1]!=10 && TAM < 40);
}

void CHECAR_ESTADO_PICONET ()
{
int X,i,j;
for(X=0;X<8;X++)
{

leerDireccion(X);

printf("at+rtype?%c%c%c%c,%c%c,%c%c%c%c%c%c\r\n",bufferDire
ccion[0],bufferDireccion[1],

bufferDireccion[2],bufferDireccion[3],bufferDireccion[4],bu
fferDireccion[5],bufferDireccion[6],

bufferDireccion[7],bufferDireccion[8],bufferDireccion[9],bu
fferDireccion[10],bufferDireccion[11]);
j = 0;
i = 0;

leer_CadenaSerial();
// aqui empieza la comparacion de las
cadenas para saber el estado del
// resultado del modo AT
if(substring(miBUFFER,TAM,rname,5))
//encontramos el dispositivo BLUETOOTH pero tenemos que
leer otra linea XD
{
leer_CadenaSerial();
OKFAILRname = TRUE; // marcamos la
BANDERA
ERRORRname = FALSE;
}
else if(substring(miBUFFER,TAM,error,5))
// vemos si hay algun error por parte del MODULO
{

OKFAILRname = FALSE; //si hay
errores marcamos la
ERRORRname = TRUE; //bandera
correspondiente
//obtenerCodigoError(cadena,i); //
mandamos la cadena para obtener el codigo del error
}
else if(substring(miBUFFER,TAM,fail,4))
// si no se encuentra el dispositivo BLUETOOTH
{
//output_low(BLANCO_ENCONTRADO);

```

## Firmware Tarjeta Bluetooth - Ethernet

```

                                OKFAILRname    = FALSE; // marcamos
BANDERAS
                                ERRORRname     = FALSE;
                                }
                                }
                                }

int CHECAR_NODO_PICONET(int NODO)
{
    if(NODO>=0 && NODO<NDISP)
    {
        leerDireccion(NODO);

printf("at+rtype?%c%c%c%c,%c%c,%c%c%c%c%c%c\r\n",bufferDire
ccion[0],bufferDireccion[1],

bufferDireccion[2],bufferDireccion[3],bufferDireccion[4],bu
fferDireccion[5],bufferDireccion[6],

bufferDireccion[7],bufferDireccion[8],bufferDireccion[9],bu
fferDireccion[10],bufferDireccion[11]);
        //j = 0;
        //i = 0;

        leer_CadenaSerial();
        // aqui empieza la comparacion de las
cadenas para saber el estado del
        // resultado del modo AT
        if(substring(miBUFFER,TAM,rname,5))
//encontramos el dispositivo BLUETOOTH pero tenemos que
leer otra linea
        {
            leer_CadenaSerial();
            OKFAILRname    = TRUE; // marcamos
la BANDERA
            ERRORRname     = FALSE;
        }
        else
if(substring(miBUFFER,TAM,error,5)) // vemos si hay algun
error por parte del MODULO
        {
            OKFAILRname    = FALSE; //si hay
errores marcamos la
            ERRORRname     = TRUE;
//bandera correspondiente
        }
        else
if(substring(miBUFFER,TAM,fail,4)) // si no se encuentra el
dispositivo BLUETOOTH
        {
            //output_low(BLANCO_ENCONTRADO);
            OKFAILRname    = FALSE; // marcamos
BANDERAS

```

```

        ERRORRname      = FALSE;
    }

    if (OKFAILRname)
        return 1;
    else
        return 0;
}
return -1;
}

void INICIALIZAR_COMUNICACION ()
{
    output_high(AT);
    delay_ms(100);
    printf("at+init\r\n");
    leer_CadenaSerial();
    printf("at+inqc\r\n");
    leer_CadenaSerial();
}

int EMPAREJAR_NODO(int NODO)
{
    leerDireccion(NODO);

    printf("at+pair=%c%c%c%c,%c,%c,%c,%c,%c,%c,10\r\n",bufferDi
reccion[0],bufferDireccion[1],

bufferDireccion[2],bufferDireccion[3],bufferDireccion[4],bu
fferDireccion[5],bufferDireccion[6],

bufferDireccion[7],bufferDireccion[8],bufferDireccion[9],bu
fferDireccion[10],bufferDireccion[11]);
    leer_CadenaSerial();

    if (substring(miBUFFER,TAM,ok,2)) //encontramos el
dispositivo BLUETOOTH pero tenemos que leer otra linea
    {
        output_high(CONFIGURACION);
        return TRUE;
    }
    return FALSE;
}

int EMPAREJAR_NODO ()
{
    //PREVIAMENTE SE DEBERA DE OBTENER LA DIRECCION PARA
ENVIAR

    printf("at+pair=%c%c%c%c,%c,%c,%c,%c,%c,%c,10\r\n",bufferDi
reccion[0],bufferDireccion[1],

```

## Firmware Tarjeta Bluetooth - Ethernet

```
bufferDireccion[2],bufferDireccion[3],bufferDireccion[4],bu
fferDireccion[5],bufferDireccion[6],

bufferDireccion[7],bufferDireccion[8],bufferDireccion[9],bu
fferDireccion[10],bufferDireccion[11]);
    leer_CadenaSerial();
    parpadeo_INDICADORES();
    if(substring(miBUFFER,TAM,ok,2)) //encontramos el
dispositivo BLUETOOTH pero tenemos que leer otra linea
    {
        output_high(CONFIGURACION);
        return TRUE;
    }
    return FALSE;
}

int LIGAR_NODO(int NODO)
{
    leerDireccion(NODO);

printf("at+link=%c%c%c%c,%c%c,%c%c%c%c%c%c\r\n",bufferDirec
cion[0],bufferDireccion[1],

bufferDireccion[2],bufferDireccion[3],bufferDireccion[4],bu
fferDireccion[5],bufferDireccion[6],

bufferDireccion[7],bufferDireccion[8],bufferDireccion[9],bu
fferDireccion[10],bufferDireccion[11]);
    leer_CadenaSerial();

    if(substring(miBUFFER,TAM,ok,2))
    {
        output_high(INDICADOR);
        return TRUE;
    }
    return FALSE;
}

int LIGAR_NODO()
{
    // previamente se debera de obtener la direccion

printf("at+link=%c%c%c%c,%c%c,%c%c%c%c%c%c\r\n",bufferDirec
cion[0],bufferDireccion[1],

bufferDireccion[2],bufferDireccion[3],bufferDireccion[4],bu
fferDireccion[5],bufferDireccion[6],

bufferDireccion[7],bufferDireccion[8],bufferDireccion[9],bu
fferDireccion[10],bufferDireccion[11]);
    leer_CadenaSerial();

    if(substring(miBUFFER,TAM,ok,2))
    {
```

```

        output_high(INDICADOR);
        return TRUE;
    }
    return FALSE;
}

int desconectar_Dispatch_Piconet ()
{
    output_low(AT);
    delay_ms(500);
    output_high(AT);
    delay_ms(500);
    printf("at+disc\r\n"); //enviamos el comando AT para
desconectar
    leer_CadenaSerial();
    if(substring(miBUFFER,TAM,disc,4))
    {
        leer_CadenaSerial();// hay que leer un ok
        output_low(CONFIGURACION);
        output_low(INDICADOR);
        return TRUE;
    }
    return FALSE;
}

int ENVIAR_CARACTER_NODO(int NODO,char C)
{
    if(EMPAREJAR_NODO(NODO))
    {
        if(LIGAR_NODO(NODO))
        {
            //ENVIAMOS EL CARACTER T
            delay_ms(1000);
            putc(C);
            //espero una confirmacion para poder continuar
            getch();
            if(desconectar_Dispatch_Piconet())
                return TRUE;
            return FALSE;
        }
    }

    return FALSE;
}

int CHECA_DIRECCION_N_BUFFERD()
// A LA VIEJA USANZA DE C SI NO ENCUENTRA LA DIRECCION
REGRESA -1
// SINO REGRESA EL INDICE DE LA DIRECCION
{
int H;

```

## Firmware Tarjeta Bluetooth - Ethernet

```
//COMPARA BUFFERD CON TODAS LAS DIRECCIONES GUARDADAS
for (H=0;H<NDISP;H++)
{
    leerDireccion(H);
    if(CADENAEQUALS (BUFFERD,bufferDireccion,12))
        return H;
}
return -1;
}

int ENVIAR_COMANDO_NODO (int NODO,char COMMAND)
{
    if(NODO!=-1)
    {
        leerDireccion(NODO);
        INICIALIZAR_COMUNICACION();
        if(EMPAREJAR_NODO())
        {
            if(LIGAR_NODO())
            {
                //ENVIAMOS EL COMANDO
                delay_ms(1000);
                putc(COMMAND);
                //espero una confirmacion para poder continuar
                getc();
                if(desconectar_Disb_Piconet())
                    return TRUE;
                return FALSE;
            }
            return FALSE;
        }
    }
    return FALSE;
}

// FUNCION QUE SE ENCARGA DE PEDIR DATOS DE SONDEO E
// INDICAR EL SONDEO PARA CADA NODO
// DE LA PICONET
// RETORNA EN UN ENTERO DE 8 BITS EL RESULTADO DEL SONDEO
// UN BIT POR CADA ELEMENTO SONDEADO
int SONDEAR()
{
    int RESULT=0,MUL=0,SON=0,H;
    INICIALIZAR_COMUNICACION();
    for (H = 0; H<NDISP; H++)
    {
        leerDireccion(H);
        SON = MUL = 0;
        if(EMPAREJAR_NODO())
            if(LIGAR_NODO())
            {
                MUL = 1;
                delay_ms(100);
            }
    }
}
```

```

        putc('C'); // COMANDO QUE INDICA QUE ME DE LOS
RESULTADOS DE SU BANDERA SONDEO
        SON = getc();
        putc('D'); //COMANDO PARA INDICARLE AL NODO QUE
EJECUTE LA BUSQUEDA
        getc(); // RECIBIMOS NOTIFICACION
        desconectar_Disb_Piconet();
    }
    RESULT |= ((int)pow(2,H))*MUL*SON;
}

return RESULT;
}

BOOLEAN CHECAR_DISPOSITIVO()
{
ON(AT);
delay_ms(10);

printf("at+rname?%c%c%c%c,%c%c,%c%c%c%c%c%c\r\n",DIRRECIION_
SONDEO[0],DIRRECIION_SONDEO[1],

DIRRECIION_SONDEO[2],DIRRECIION_SONDEO[3],DIRRECIION_SONDEO[4]
,DIRRECIION_SONDEO[5],DIRRECIION_SONDEO[6],

DIRRECIION_SONDEO[7],DIRRECIION_SONDEO[8],DIRRECIION_SONDEO[9]
,DIRRECIION_SONDEO[10],DIRRECIION_SONDEO[11]);
    //j = 0;
    //i = 0;

    leer_CadenaSerial();
    // aqui empieza la comparacion de las
cadenas para saber el estado del
    // resultado del modo AT
    if(substring(miBUFFER,TAM,rname,5))
//encontramos el dispositivo BLUETOOTH pero tenemos que
leer otra linea
    {
        leer_CadenaSerial();
        OKFAILRname = TRUE; // marcamos
la BANDERA
        ERRORRname = FALSE;
    }
    else
if(substring(miBUFFER,TAM,error,5)) // vemos si hay algun
error por parte del MODULO
    {
        OKFAILRname = FALSE; //si hay
errores marcamos la
        ERRORRname = TRUE;
//bandera correspondiente
    }
}

```

## Firmware Tarjeta Bluetooth - Ethernet

```

                                else
if(substring(miBUFFER,TAM,fail,4)) // si no se encuentra el
dispositivo BLUETOOTH
    {
        //output_low(BLANCO_ENCONTRADO);
        OKFAILRname    = FALSE; // marcamos
BANDERAS
        ERRORRname    = FALSE;
    }

    OFF(AT);
    delay_ms(10);
    if(OKFAILRname)
        return TRUE;
return FALSE;
}
```