Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Laboratorio ICARO

BlueCiens: Una Herramienta para el Escaneo y la Distribución de Anuncios Usando Bluetooth

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por el Bachiller:

Eduardo Diego Díaz Castillo
C.I.: 18.269.760
E-mail: guachos2003@gmail.com

para optar al título de Licenciado en Computación

Tutor: Profa. María E. Villapol

Caracas, Mayo 2012

Universidad Central de Venezuela

Facultad de Ciencias

Escuela de Computación

Laboratorio ICARO

ACTA DEL VEREDICTO

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por el bachiller Eduardo Diego Díaz Castillo C.I.: 18.269.760, con el título **"BlueCiens: Una Herramienta para el Escaneo y la Distribución de Anuncios Usando Bluetooth"**, a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 29 de Mayo de 2012, a las 11:00 am, para que su autor lo defendiera en forma pública, en el aula PB 3 de la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, lo cual este realizó mediante una exposición oral de su contenido, y luego respondió satisfactoriamente a las preguntas que le fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el 29 de Mayo de 2012, dejándose también constancia de que actuó como Coordinador del Jurado la Profesora Tutor María E. Villapol.

Profa. María E. Villapol
(Tutor)

Prof. Sergio Rivas
(Jurado Principal)

Profa. Karima Velasquez
(Jurado Principal)

Resumen

Título:
BlueCiens: Una Herramienta para el Escaneo y la Distribución de Anuncios Usando Bluetooth.

Autor:
Eduardo Diego Díaz Castillo

Tutor:
Prof. María E. Villapol

La presente investigación se fundamentó en el uso de la tecnología Bluetooth. Esta es una tecnología de comunicaciones inalámbrica de corto alcance, la cual puede ser empleada para diversos fines. Entre sus ventajas resalta la posibilidad que brinda para que un dispositivo pueda recolectar datos de otros que estén dentro de su radio de alcance, que en conjunto con otros datos, tales como el momento y el lugar donde fueron recolectados, definen una técnica llamada escaneo Bluetooth. A su vez, Bluetooth permite que un dispositivo pueda enviar archivos a otros que estén dentro de su radio de alcance. Dicha funcionalidad puede ser empleada como una alternativa para enviar anuncios y comunicarse con las personas.

Las ventajas antes mencionadas pueden ser aprovechadas utilizándolas dentro del ámbito que nos rodea. En tal sentido, el propósito de este trabajo fue desarrollar una herramienta para el escaneo y la distribución de anuncios en la Facultad de Ciencias de la Universidad Central de Venezuela, la cual se denominó BlueCiens.

Con este propósito se concibió una herramienta que funciona a través de tres componentes: unos componentes periféricos que pueden ser ubicados en varios lugares al mismo tiempo y se encargan de realizar las labores de escaneo y distribución de anuncios; un componente central que coordina el trabajo de los componentes periféricos, y un componente que permite al usuario administrador hacer uso de la herramienta.

El trabajo se apoyó en la siguiente metodología tradicional de desarrollo: en primer lugar se definieron los requerimientos; seguidamente se hizo el diseño de la herramienta; luego se realizó la implementación de la misma, y finalmente se llevaron a cabo las pruebas para evaluar el cumplimiento de los requerimientos planteados en un principio.

Los resultados de las pruebas revelaron una gran cantidad de datos provenientes del escaneo de los dispositivos Bluetooth que se encontraban dentro del radio de alcance de la herramienta, con un pico máximo de 24 dispositivos escaneados en un día. De igual modo, se alcanzó la distribución de anuncios utilizando Bluetooth, al conseguir enviarlos a los dispositivos que se encontraban dentro del radio de alcance de la herramienta.

De este modo podemos concluir que se alcanzaron los objetivos propuestos, obteniendo como producto final una herramienta que permite el escaneo y la distribución de anuncios utilizando Bluetooth dentro de la Facultad de Ciencias de la Universidad Central de Venezuela.

Palabras Claves: Tecnología Bluetooth, Escaneo Bluetooth, Anuncios Usando Bluetooth

Tabla de Contenido

Índice de Figuras

Índice de Tablas

1. Introducción

Bluetooth es una tecnología de comunicaciones inalámbrica de corto alcance. Nace con el firme propósito de sustituir los cables que son utilizados para interconectar gran variedad de dispositivos. El proyecto tiene sus inicios en el año 1998 cuando se conforma un grupo denominado SIG (*Special Interest Group*, Grupo de Interés Especial). Este grupo se compuso inicialmente por cinco grandes compañías del ámbito de la computación y de la telefonía celular cuyo principal enfoque es el desarrollo de la naciente tecnología.

Bluetooth se caracteriza por ser una tecnología de bajo costo y corto alcance. Opera en la banda abierta de los 2.4 GHz, y tiene como principal ventaja proporcionar la libertad al usuario de prescindir de los cables para interconectar gran variedad de dispositivos. A su vez, al emplear ondas de radio, puede atravesar obstáculos como paredes, techos, vegetación, entre otros.

Para conseguir la interoperabilidad de las aplicaciones, los dispositivos remotos deben ejecutarse sobre la misma pila de protocolos, que se define en la especificación de Bluetooth. Entre los protocolos que conforman la especificación Bluetooth resalta el de la Banda Base, que introduce varios conceptos claves. Uno de ellos es la definición de las *piconets*, que consisten en las estructuras que forman varios dispositivos que se encuentran interconectados entre sí. Del mismo modo, también se definen los procesos *inquiry* (pregunta) y *page* (búsqueda). El primero de ellos es fundamental para el proceso de escaneo Bluetooth. El principal propósito de un dispositivo al utilizar el proceso *inquiry* es recolectar información sobre otros dispositivos Bluetooth cercanos. Así mismo, el objetivo del proceso *page* para un dispositivo es invitar a otro a unirse a una *piconet*.

Dentro de las características de Bluetooth encontramos la posibilidad que tiene cada dispositivo mediante el proceso *inquiry* de recolectar información de otros que se encuentren dentro de su radio de alcance. Esta información puede resultar bastante útil para diversos fines. Hoy en día existen aplicaciones que hacen uso de esta técnica para obtener información que puede ser utilizada en: las redes sociales, el rastreo y la localización bajo techo, la medición del flujo de personas y los sistemas de computación ubicua, entre otros.

Este proceso conocido como escaneo Bluetooth consiste en la recolección de datos claves de los dispositivos que se encuentren al alcance de aquel que actúe como escáner. Entre los datos recolectados se pueden mencionar: la BD_ADDR que identifica unívocamente al dispositivo; el nombre del dispositivo; la clase del dispositivo que puede indicar si el dispositivo es: un teléfono celular, una computadora portátil, una computadora de escritorio, o cualquier otro; los distintos servicios que ofrece el dispositivo; y opcionalmente también se puede conocer la potencia de la señal del dispositivo que se esté escaneando.

Los datos recolectados podrán ser almacenados y asociados con la hora y el lugar donde fueron registrados. De este modo se consiguen generar análisis que nos pueden indicar las características y comportamiento de los dispositivos escaneados. Aunque se puede utilizar esta información para inferir también las características y comportamiento de las personas,

nunca se podrá utilizar para asociarla con una persona en específico, ya que los dispositivos pueden pasar de una persona a otra.

En la actualidad, y aprovechando la masificación que ha experimentado la tecnología Bluetooth, han surgido gran cantidad de aplicaciones que hacen uso de la mencionada tecnología para interactuar con los usuarios. Muestra de ello son las aplicaciones que permiten la distribución de anuncios digitales por este medio. Estos sistemas de anuncios no intrusivos buscan llevar de forma transparente información de interés general o particular a los usuarios. Sistemas de este tipo se han desplegado en recintos como centros comerciales, hospitales, y museos, entre otros.

1.1. Planteamiento del Problema

Bluetooth, como ya se dijo, es una tecnología usada para el reemplazo de cables usados para la interconexión de diversos dispositivos. Sin embargo, hoy en día esta tecnología ha encontrado otro uso, como es el caso del escaneo Bluetooth, el cual se describe en detalle en este trabajo.

Por otra parte, la tecnología Bluetooth está siendo usada para la distribución de anuncios en diversos sitios tales como: centros comerciales, museos, hospitales, e instituciones educativas. Esta aplicación explota el hecho de que muchos de los dispositivos móviles, tales como los teléfonos celulares, vienen con la tecnología Bluetooth integrada. Hace algunos años hubo un intento por realizar una aplicación que permitía el despliegue de anuncios en la Facultad de Ciencias de la UCV. Sin embargo, este trabajo fue limitado por la falta de lenguajes que soportaran el desarrollo de programas para Bluetooth **[15]**.

Por lo anteriormente expuesto, en este trabajo se pretende dar respuesta a dos problemas. Por un lado, la falta de un mecanismo automatizado adaptable que permita el estudio de la localización de usuarios de dispositivos Bluetooth con diversos fines en la Facultad de Ciencias de la UCV. Por otra parte, la falta de un mecanismo alternativo para la distribución de anuncios académicos, administrativos, de investigación y extensión en la Facultad de Ciencias de la UCV que aporte una alternativa a los tradicionales mecanismos como lo son la forma de distribución de anuncios de forma manual y a través del uso del correo electrónico.

1.2. Objetivo General

El objetivo de este trabajo es desarrollar una herramienta que soporte el escaneo Bluetooth y la distribución de anuncios en la Facultad de Ciencias de la Universidad Central de Venezuela.

1.3. Objetivos Específicos

Los objetivos específicos de este trabajo son:

- Determinar los requerimientos para el desarrollo de la herramienta de software.

- Diseñar la herramienta de software en base a los requerimientos planteados.
- Implementar la herramienta de software que este acorde al diseño planteado.
- Realizar las pruebas de funcionamiento de la herramienta de software para verificar que cumpla con los requerimientos planteados.
- Analizar los resultados.

1.4. Justificación

Bluetooth se puede calificar como la tecnología para las comunicaciones inalámbricas de corto alcance con mayor presencia entre los dispositivos móviles de la actualidad. Las estadísticas de diversos estudios indican que entre un 7% y un 10% de los peatones poseen un dispositivo con capacidad Bluetooth activada **[14]**. Por tales motivos surgen diversas aplicaciones que se apoyan en la mencionada tecnología para lograr mayor contacto con y entre las personas. Esta tecnología puede otorgar información del comportamiento y características de los dispositivos que estén a su alcance; lo que podría suponer en información relacionada a los dueños de tales dispositivos.

Pensar en un sistema que facilite el proceso de escaneo Bluetooth con diversos fines propicia el desarrollo de un mayor número de aplicaciones que se apoyen en esta tecnología. Hasta ahora el universo de posibilidades no tiene límites. La tendencia apunta a una mayor masificación de la tecnología Bluetooth.

Aprovechando la penetración de esta tecnología dentro de la población se puede imaginar en utilizarla como un medio para facilitar la comunicación con las personas. Por lo tanto se puede concebir un sistema que permita la distribución de anuncios dentro del recinto de la Facultad de Ciencias de la Universidad Central de Venezuela mediante el uso de la tecnología Bluetooth. Esto puede permitir un mayor contacto con la comunidad que hace vida dentro de estas instalaciones.

1.5. Estructura del Documento

Este trabajo se organiza de la siguiente manera:

Capítulo 1 - Introducción: Se coloca en contexto al lector, se define el planteamiento del problema, se describe el objetivo general y los objetivos específicos, se expone la justificación del trabajo y por último se explica cómo se encuentra estructurado este documento.

Capítulo 2 - Tecnología Bluetooth: Se explica el funcionamiento y los protocolos empleados para lograr la comunicación entre los dispositivos Bluetooth.

Capítulo 3 - Escaneo Bluetooth (*Bluetooth Scanning*): Se explica en detalle en qué consiste este método y cuáles son los trabajos relacionados que se han realizado.

Capítulo 4 – Metodología y Herramientas de Hardware y Software: Se explica en detalle la metodología utilizada y las diversas herramientas de software y hardware que fueron empleadas con el objetivo de desarrollar la herramienta BlueCiens.

Capítulo 5 - Análisis y Diseño de la Aplicación: Se describe el proceso de análisis que se llevó a cabo para el desarrollo de la herramienta para el escaneo y la distribución de anuncios usando Bluetooth, denominada BlueCiens.

Capítulo 6 - Implementación de la Aplicación: Se explica cómo se realizó la implementación de la herramienta BlueCiens.

Capítulo 7 - Realización de pruebas y Análisis de Resultados: Se aborda el proceso de pruebas, desde la definición del entorno de pruebas, su despliegue, la ejecución de los casos de prueba y el posterior análisis de los resultados obtenidos.

Capítulo 8 - Conclusiones: Se hacen algunas consideraciones finales, incluyendo las limitaciones encontradas y las sugerencias para trabajos futuros.

2. Tecnología Bluetooth

Bluetooth es una tecnología de comunicaciones inalámbrica, que fue concebida para remplazar los cables utilizados para interconectar gran variedad de dispositivos, como teléfonos celulares, dispositivos de manos libres, monitores, equipos médicos, entre otros. El proyecto nació en 1998 **[3]** al crearse un grupo compuesto por varias compañías, enfocadas en el desarrollo de dicha tecnología.

Bluetooth se caracteriza por ser una tecnología de bajo costo y corto alcance, que opera en la banda abierta de los 2.4 GHz. La misma tiene como principal ventaja proporcionar la libertad de prescindir de los cables y al emplear ondas de radio puede atravesar obstáculos, como paredes, techos, vegetación, entre otros.

2.1. Arquitectura de Hardware

El hardware que compone el dispositivo Bluetooth está formado por dos partes: un dispositivo de radio, encargado de modular y transmitir la señal y un controlador digital que está compuesto por un CPU, por un procesador de señales digitales (DSP - *Digital Signal Processor*) llamado LC (*Link Controller*, Controlador de Enlace) y de las interfaces con el dispositivo anfitrión, como se muestra en la Figura 2.1 **[8]**.

El LC está encargado de hacer el procesamiento de la Banda Base y del manejo de los protocolos ARQ (*Automatic Repeat reQuest*) y FEC (*Forward Error Correction*) de la capa física. Además, se encarga de las funciones de transferencia (tanto asíncrona como síncrona), codificación de audio y cifrado de datos.

Por otro lado, el CPU del dispositivo se encarga de atender las instrucciones relacionadas con Bluetooth del dispositivo anfitrión, para así simplificar su operación. Para ello, sobre el CPU corre un software denominado *Link Manager* (Gestor del Enlace) que tiene la función de comunicarse con otros dispositivos por medio del protocolo LMP (*Link Manager Protocol*, Protocolo de Gestión del Enlace).

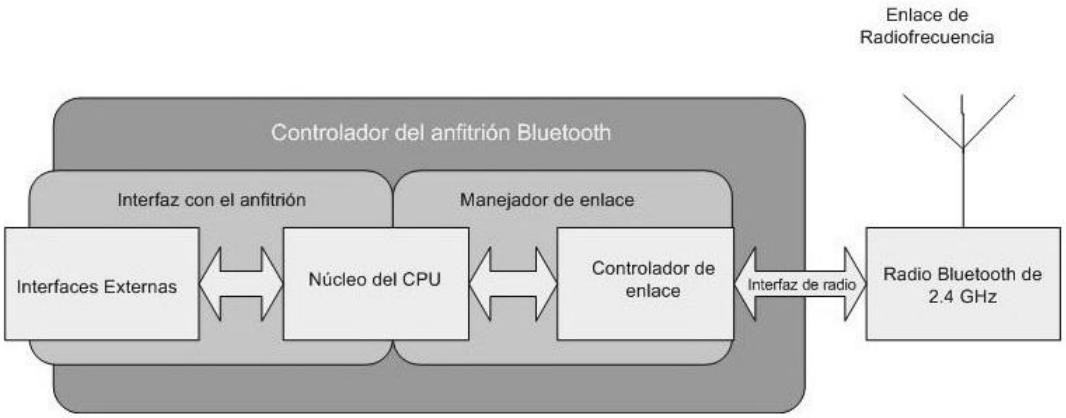

Figura 2.1 Arquitectura de hardware Bluetooth.

2.2. Pila de Protocolos

El objetivo principal de la especificación de Bluetooth es que todas las aplicaciones y dispositivos que implementan esta tecnología sean capaces de operar entre sí. Para conseguir esta interoperabilidad, las aplicaciones en dispositivos remotos deben ejecutarse sobre la misma pila de protocolos que se muestra en la Figura 2.2. La pila de protocolos de Bluetooth se puede dividir en cuatro capas lógicas, las cuales se resumen en la Tabla 2.1.

Capa lógica del protocolo	Protocolos involucrados
Núcleo de Bluetooth	Banda Base, LMP, L2CAP, SDP
Sustitución del cable	RFCOMM
Control de telefonía	Comandos AT, TCS
Protocolos adoptados	PPP, UDP/TCP/IP, WAP, WAE, OBEX, vCal, vCard

Tabla 2.1 Capas lógicas de Bluetooth

Diferentes pilas de protocolo son usadas para diferentes aplicaciones. Sin embargo los protocolos del “Núcleo de Bluetooth” deben ser implementados por la mayoría de los dispositivos Bluetooth.

La especificación de Bluetooth también define una interfaz entre el dispositivo anfitrión (laptop, teléfono celular, etc.) y el dispositivo Bluetooth como tal (chip Bluetooth) denominada HCI (*Host Controller Interface*). Esta se puede ubicar como aparece en la Figura 2.2 debajo de la capa L2CAP o también puede posicionarse encima de esta **[18]**.

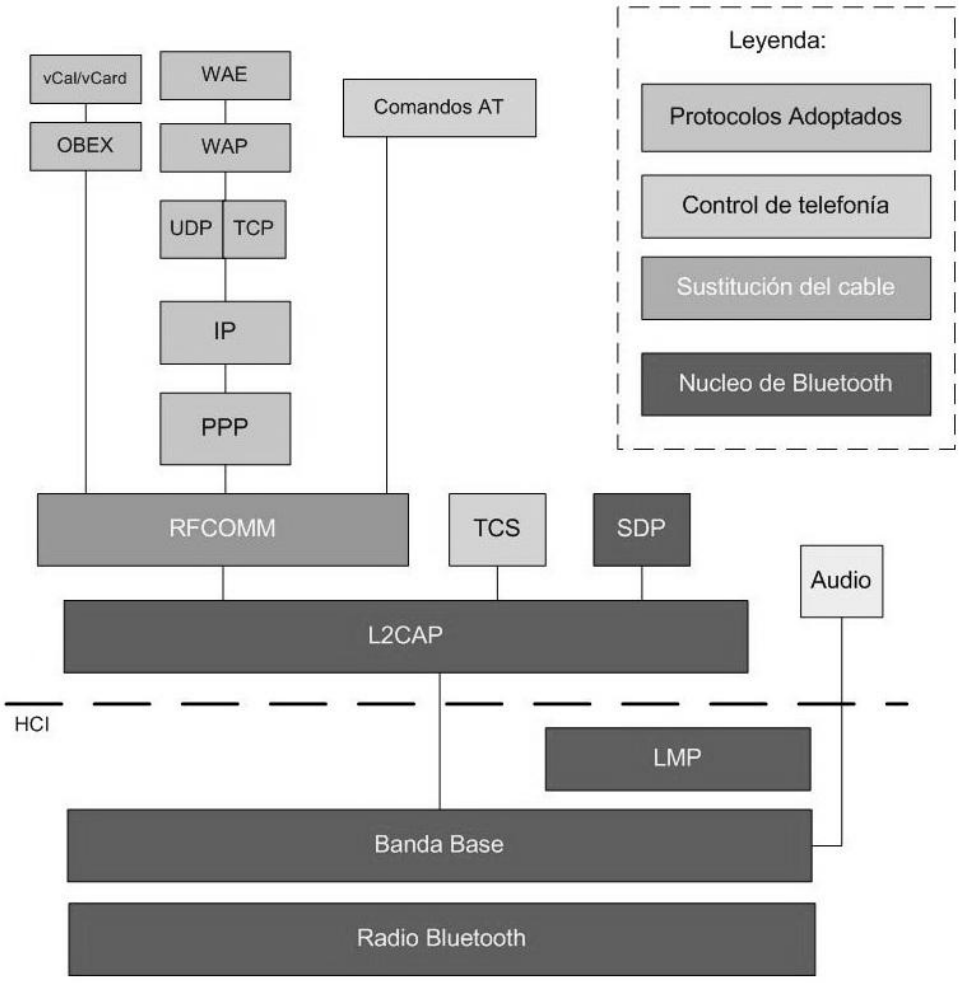

Figura 2.2 Pila de protocolos Bluetooth.

A continuación se explica en detalle cada uno de los protocolos que forman la pila de protocolos de Bluetooth.

2.2.1. Radio de Bluetooth

Bluetooth trabaja en la banda ISM (*Industrial, Scientific and Medical*) de 2.4 GHz. Las bandas ISM están destinadas para su uso en el área industrial, científica o médica. Las autoridades reguladoras alrededor del mundo han abierto estas bandas para su uso en sistemas de bajo poder de emisión, que pueden trabajar sin la necesidad de una licencia, sin embargo, bajo una estricta regulación.

Bluetooth trabaja con el método FHSS (*Frequency Hopping Spread Spectrum*, Transmisión de Saltos de Frecuencia de Espectro Ensanchado) que opera sobre 79 canales, de 1 MHz de ancho cada uno, comenzando en la mayoría de los países en la frecuencia de

2.402 GHz y acabando en la frecuencia de 2.4835 GHz. Regulaciones en ciertos países pueden contraer la banda ISM de 2.4 GHz. Por tal motivo el radio de Bluetooth y el protocolo de Banda Base (que se describe más adelante) pueden adaptarse a dos alternativas, una donde opera con 79 canales y otra con 23 canales, cada uno de 1 MHz de ancho. Y para tener en cuenta las emisiones fuera de banda se establecen las siguientes bandas de guarda al principio y al final de la banda de frecuencias de trabajo (ver Tabla 2.2).

Banda de guarda inferior	Banda de guarda superior
2 MHz	3.5 MHz

Tabla 2.2 Bandas de guarda Radio Bluetooth.

Para FHSS en la banda ISM de 2.4 GHz las regulaciones restringen un pico máximo de poder de salida del emisor de no más de 1 watt (30 dBm) **[5]**. Por otra parte, 75 de los 79 canales de frecuencia deben ser usados de forma pseudo-aleatoria con un tiempo total sobre cada canal que no exceda los 0,4 segundos dentro de un período de 30 segundos. Bluetooth utiliza el máximo número de canales disponibles, con una alta tasa de saltos, de 1.600 saltos pseudo-aleatorios por segundo a través de todas estas frecuencias. De este modo se puede lograr una gran resistencia al ruido.

Bluetooth emplea GFSK (*Gaussian Frequency Shift Keying*, Modulación por Desplazamiento de Frecuencia Gaussiana), con las siguientes características:

- El índice de modulación debe estar entre 0.28 y 0.35.
- Un uno binario se representa por una desviación positiva de frecuencia y un cero binario como una desviación negativa.
- La desviación mínima no ha de ser menor de 115 KHz.
- El error de los pasos por cero (diferencia de tiempo entre el período de símbolo ideal y el tiempo de cruce medido) debe ser menor de ±1/8 del período de símbolo.

En la Tabla 2.3 se resumen las características del Radio Bluetooth:

Modulación	Gaussian Frequency Shift Keying (GFSK).	Producto BT [1] 0.5 índice de modulación: 0.28 – 0.35.
Tasa de salto de frecuencia	1.600 saltos por segundo.	Tiempo sobre cada canal: 625 µseg.
Potencia de transmisión	Clase 3: 0 dBm (1 mW).	Un radio de Bluetooth típico; Control de potencia por debajo de los -30 dBm opcional.
	Clase 2: 4 dBm (2.5 mW).	Control de potencia opcional como el anterior.
	Clase 1: 20 dBm (100 mW).	Control de potencia por encima de los 4 dBm. Por debajo es opcional.
Sensibilidad del receptor	Un receptor Bluetooth debe alcanzar una tasa de error puro de 0,1% con un nivel de señal de entrada de -70 dBm o inferior.	El nivel de sensibilidad de -70 dBm se debe lograr para cualquier señal de entrada generada por cualquier transmisor Bluetooth compatible.

Tabla 2.3 Características de radio Bluetooth.

2.2.2. Protocolo Banda Base

En la sección anterior se describió el protocolo de Radio de Bluetooth que se encarga de definir la forma como se envían y reciben los datos. Sin embargo, no se han mencionado consideraciones como qué datos se transmiten y cuándo, qué datos se esperan recibir y en qué momento, y cuál es la frecuencia portadora y qué potencia se va a utilizar para transmitir estos datos. Esta responsabilidad recae sobre el LC, el cual implementa el protocolo de Banda Base y los procesos relacionados con este.

Este protocolo introduce el concepto de *piconets* y como estas son creadas, las cuales se explican en detalle más adelante. A su vez, debido a que Bluetooth hace uso de FHSS para transmitir los paquetes en espacios de tiempo definidos sobre un conjunto de frecuencias determinadas, este protocolo define los procesos *inquiry* (pregunta) y *page* (búsqueda) para sincronizar la secuencia de saltos de frecuencia y el reloj entre dos dispositivos Bluetooth. También se definen los paquetes de bajo nivel usados en Bluetooth.

Paquetes

Los paquetes usados por el protocolo de Banda Base tienen un formato general que se muestra en la Figura 2.3, durante la transmisión de estos paquetes no ocurre ningún salto de frecuencia y la transmisión de estos comienza con el bit menos significativo continuando hasta el bit más significativo.

[1] El término "producto BT" no es la abreviatura de "producto Bluetooth". Es un parámetro que describe la calidad de la onda de transmisión expresada como el producto del ancho de banda de la modulación del filtro y el tiempo de bit.

Figura 2.3 Formato paquete Banda Base.

Los campos que conforman el paquete son los siguientes:

- **Código de acceso:** Este campo se utiliza para la sincronización, la compensación y la identificación. Este campo típicamente tiene una longitud de 72 bits incluyendo la cola de 4 bits. Si el campo no va seguido de una cabecera, y por lo tanto, tampoco de una carga útil, el campo de código de acceso prescindirá de dicha cola de 4 bits y en consecuencia su tamaño será de solo 68 bits que son utilizados para transmitir paquetes de identificación (los cuales se explicarán en detalle más adelante). Se compone de un preámbulo de 4 bits una palabra de sincronización 64 bits y una cola de 4 bits en el caso que sea necesario.

- **Cabecera:** Este campo contiene información de control de enlace, utilizando 6 campos que se describen en la Tabla 2.4.

Campo	Tamaño	Descripción
AM_ADDR	3 bits	Dirección temporal de 3 bits de los dispositivos activos dentro de una *piconet* asignada por el maestro, siendo 000 la de *broadcast*.
TIPO	4 bits	Define uno de los 16 tipos de paquetes, dependiendo de la carga útil que se transporte.
FLUJO	1 bit	Se usa para control de flujo, si es 0 detiene la transmisión.
ARQN	1 bit	Se utiliza para reconocimiento de paquetes y determinar si estos llegaron a su destino. 1 para ACK y 0 para NACK.
SEQN	1 bit	Número de secuencia que se utiliza para detectar paquetes repetidos.
HEC	8 bits	Comprobación de error de la cabecera (HEC, *Header Error Check*) creada por un generador polinomial $G_{HEC}(x) = x^8 + x^7 + x^5 + x^2 + x + 1$

Tabla 2.4 Campos de la cabecera del protocolo Banda Base.

- **Carga útil:** Este campo contiene los datos que serán transportados.

Entre los paquetes que se muestran en la Figura 2.4 y que son utilizados por el protocolo de Banda Base destacan los siguientes:

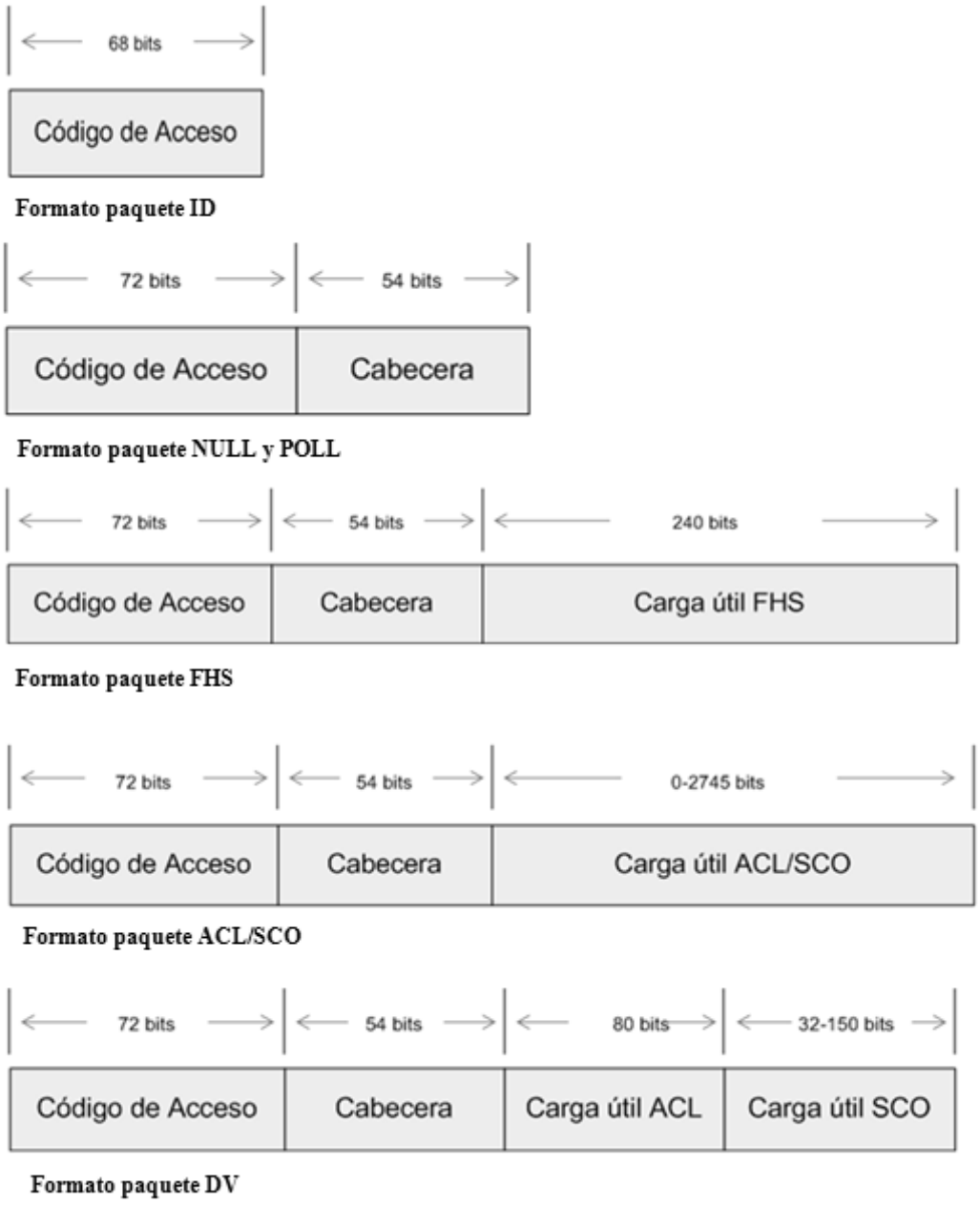

Figura 2.4 Formato de paquetes de la Banda Base.

- **Paquete de identificación:** También llamado paquete ID es usado durante el proceso de *inquiry* y *page* para la sincronización de los dispositivos Bluetooth. Está compuesto únicamente por el código de acceso, con una longitud de 68 bits.
- **Paquete NULL**: Este paquete es utilizado para confirmar una transmisión y no es necesario trasmitir carga útil. Tiene una longitud de 126 bits compuestos por el código de acceso y la cabecera.

- **Paquete POLL:** Es muy parecido al paquete NULL y es utilizado por el maestro de una *piconet* para hacer un sondeo de los esclavos, al contrario del paquete NULL este debe recibir una confirmación de vuelta.
- **Paquete FHS:** El paquete de secuencia de saltos de frecuencia (*Frequency-Hopping Sequence*) es utilizado durante la creación de una *piconet* para pasar la dirección AM_ADDR (la cual se explica en detalle más adelante) e información de reloj entre los futuros maestros y esclavos de una *piconet*. La carga útil de este paquete consta de 240 bits de longitud.
- **Paquetes ACL:** El paquete asíncrono sin conexión, se utilizan para el transporte regular de datos de usuarios, se designan como paquetes D(M|H), donde “DM” se utiliza para enviar datos a media velocidad y contienen CRC, y “DH” es utilizado para enviar datos a alta velocidad y por lo tanto no contienen CRC.
- **Paquetes SCO:** El paquete síncrono orientados a conexión, se designan como paquetes HV y están destinados para el transporte de voz. Contiene 10 bytes de información, 240 bits de carga útil.
- **Paquetes DV:** Paquete de datos y voz, que contiene una carga útil del tipo ACL y SCO. Es transmitido cuando se necesita enviar datos ACL al receptor durante una trasmisión SCO. Más adelante se explica en detalle en qué consiste una transmisión ACL y una SCO.

Canales Físicos

A nivel de la Banda Base, cuando dos dispositivos establecen una conexión, generalmente el dispositivo que inicia la comunicación asume el papel de maestro y el otro dispositivo asume el rol de esclavo. Un dispositivo puede asumir cualquiera de estos dos roles. Por lo tanto se puede ser el maestro para un enlace y ser el esclavo para otro. Ser maestro de una conexión no implica poseer privilegios especiales o autoridad alguna sobre los demás dispositivos. Este dispositivo solo va a regir la sincronización de las comunicaciones las cuales utilizan, como ya se ha mencionado el sistema FHSS. El dispositivo maestro va a determinar el patrón de saltos de frecuencia haciendo uso de su BD_ADDR y también va a determinar la fase en la secuencia de saltos en base a su reloj.

Un dispositivo maestro puede comunicarse con varios esclavos, de los cuales puede haber hasta siete esclavos activos y hasta 255 en estado *park*. Estos esclavos en estado *park* no se encuentran activos en el canal, pero continúan sincronizados con el maestro y pueden pasar a un estado activo sin necesidad de establecer una nueva conexión. Tanto para los esclavos en estado activo o *park*, el acceso al canal es controlado por el maestro. Todos los esclavos que se comunican con un maestro forman lo que la especificación denomina *piconet* (como se puede observar en la Figura 2.5). La cual se compone de dos o más dispositivos que comparten el mismo canal físico. También es importante mencionar que los dispositivos que no están asociados a ninguna *piconet* se encuentran en un estado prevenido.

Un dispositivo Bluetooth puede participar en dos o más *piconets* simultáneamente utilizando multiplexación por división de tiempo. Un dispositivo Bluetooth nunca podrá ser maestro de más de una *piconet* al mismo tiempo. Sin embargo, puede ser esclavo en múltiples *piconets*. Un dispositivo Bluetooth que es miembro de dos o más *piconets* se dice que se encuentra en una *scatternet* como se observa en la Figura 2.5. Involucrarse en una estructura

de este tipo no implica alguna capacidad o función de enrutamiento por parte del dispositivo, ya que los protocolos del núcleo Bluetooth no se encargan de esto, ni pretenden ofrecer dicha funcionalidad. Esta responsabilidad puede recaer en los protocolos de las capas superiores y está fuera del alcance de la especificación Bluetooth.

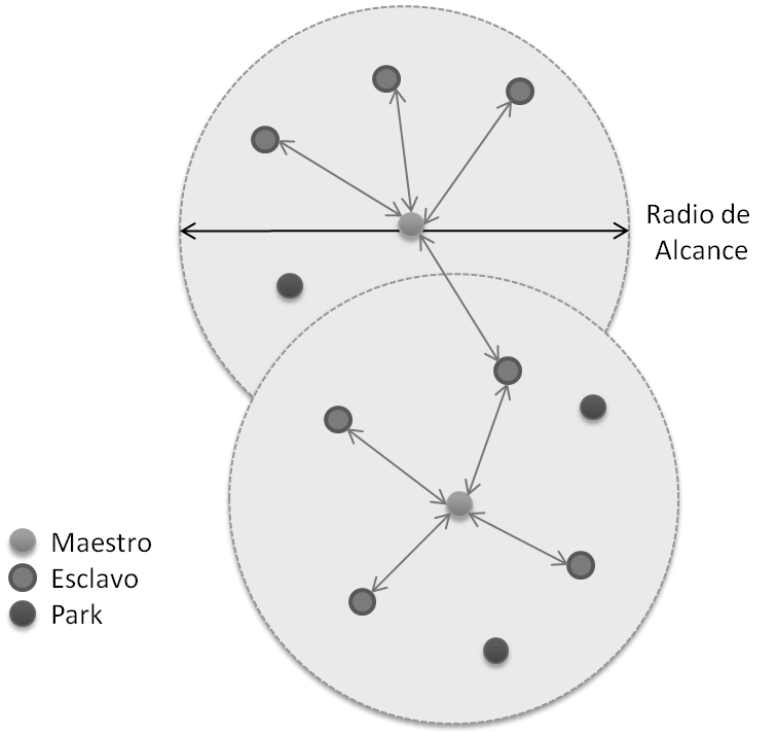

Figura 2.5 *Piconets* y *scatternets*.

Se utiliza un esquema TDD (*Time Division Duplex*) que es una implementación de la multiplexación por división de tiempo que permite separar las señales de ida de las señales de vuelta. Con este esquema se puede emular un enlace full-duplex sobre un enlace half-duplex. El canal es dividido en ranuras de tiempo, cada una de 625 µs. Las ranuras de tiempo son numeradas según el reloj del maestro de la *piconet* dentro de un rango de 0 a 2^{27}-1 y son cíclicas. El maestro y el esclavo transmiten de forma alterna. El maestro debe iniciar la transmisión en las ranuras numeradas con números pares, y los esclavos deben hacerlo en las ranuras numeradas con números impares. El inicio de la transmisión del paquete debe alinearse al comienzo de una ranura de tiempo y puede ocupar 1, 3 ó hasta 5 ranuras, tal como se muestra en la Figura 2.6.

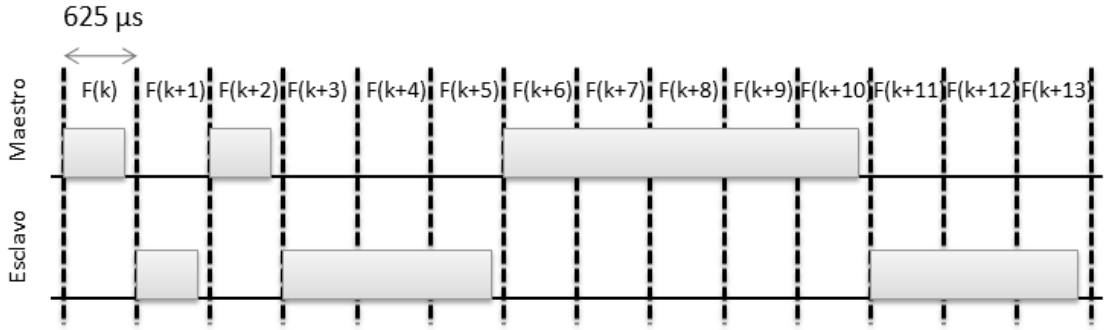

Figura 2.6 Envío de paquetes Banda Base.

Enlaces Físicos

El protocolo de Banda Base maneja dos tipos distintos de enlaces físicos: un enlace SCO (*Synchronous Connection-Oriented*, Orientado a Conexión Síncrono) y un enlace ACL (*Asynchronous Connection-less*, No Orientado a Conexión Asíncrono).

El enlace SCO establece un enlace punto a punto entre el maestro y un esclavo específico, dicho maestro mantiene el enlace mediante el uso de ranuras de tiempo que son reservadas en intervalos de tiempo regulares. Un enlace SCO normalmente lleva la información de voz, aunque puede llevar también una combinación de voz y datos.

El enlace SCO se divide en dos tipos. El primero es simétrico, donde el maestro puede soportar hasta tres enlaces SCO simultáneos, mientras que los esclavos pueden soportar hasta tres enlaces SCO de un mismo maestro o dos si los enlaces provienen de diferentes maestros. Los paquetes SCO usan transmisión de 64 Kbps y nunca son retransmitidos, pero si pueden utilizar FEC. Sólo se puede tener un enlace ACL sencillo. El modo ACL se usa únicamente para transmitir datos y puede soportar retransmisión cuando esta sea necesaria.

Corrección de Errores

Existen tres esquemas de corrección de errores definidos para Bluetooth:

- FEC con una tasa de 1/3, donde cada bit se repite 3 veces.
- FEC con una tasa de 2/3, se codifica con un código Hamming acortado con una tasa 2/3.
- Esquema ARQ para los datos, donde se transmite un paquete hasta que se obtenga un acuse de recibo o se exceda de un límite de tiempo.

El propósito del esquema FEC (*Forward Error Correction*, Corrección de Errores hacia Adelante) en la carga útil es para reducir el número de retransmisiones. Sin embargo, en un ambiente razonablemente libre de errores, FEC produce una sobrecarga innecesaria que reduce el rendimiento, debido a que este esquema añade bits de redundancia. Por lo tanto, las definiciones de paquetes se mantienen flexibles para usar FEC en la carga útil o no. Esto da

como resultado varios tipos de paquetes, DM y DH para el transporte lógico ACL, paquetes HV para el transporte lógico SCO, y paquetes EV para el transporte lógico eSCO (SCO extendido). La cabecera del paquete siempre está protegida por FEC con una tasa de 1/3 ya que contiene información valiosa de los enlaces y está diseñada para soportar mayor cantidad de errores de bits.

Bluetooth emplea acuses de recibo, rápidos y no numerados, donde hay acuses de recibo positivos y negativos estableciendo valores ARQN (Go-Back-N ARQ) adecuados. Si se excede el contador, Bluetooth marca el paquete y continúa con el siguiente.

Canales Lógicos

Los canales lógicos se refieren a los diferentes tipos de canales que pueden correr sobre un enlace físico ACL o SCO. Por tal motivo se definen cinco tipos de canales para transmitir datos de control e información de usuario.

Los canales lógicos LC y ACL-LC son utilizados en las capas de LC (*Link Control*) y LM (*Link Manager*), respectivamente. Hay que mencionar que los canales lógicos ACL-LC y ACL-U utilizan un identificador de enlace lógico llamado LLID, campo que se ubica en la cabecera de cada paquete. Hay que destacar que el canal lógico LC es transportado en la cabecera del paquete y todos los demás canales lógicos son transportados en la carga útil del paquete. A continuación se describe en detalle en qué consiste cada uno de estos canales:

- **Canal LC (*Link Control*, Control de Enlace):** Este canal transporta información de control como: ARQ, control de flujo, y características de la carga útil. Este canal como ya se mencionó es transportado en la cabecera de cada paquete, menos en los paquetes ID.
- **Canal ACL-LC (*Asynchronous ConnectionLess-Link Control*, Control de Enlace-Sin Conexión Asíncrono):** Este canal lleva información de control que se intercambia entre los gestores de enlace del maestro y del esclavo. Este canal lógico utiliza los paquetes DM1 o DV. También hay que mencionar que este canal se identifica con el código LLID 11.
- **Canal ACL-U (*Asynchronous/Isochronous ConnectionLess-User*, Usuario-Sin Conexión Asíncrono/Isócrono):** El canal lógico ACL-U llevará los datos L2CAP asíncronos e isócronos del usuario. Estos mensajes se pueden transmitir en uno o más paquetes de la Banda Base. Si los mensajes están fragmentados, el paquete de inicio utiliza el código LLID 10. Y para los paquetes restantes se utiliza un código LLID 01. Si no hay fragmentación todos los paquetes utilizan el código LLID 10.
- **Canal SCO-S (*User Synchronous Data Logical Link*, Enlace Lógico de Datos Síncronos de Usuario):** El canal lógico SCO-S lleva los datos síncronos del usuario. Este es transportado sobre un enlace SCO.
- **Canal eSCO-S (*User Extended Synchronous Data Logical Link*, Enlace Lógico de Datos Síncronos Extendido de Usuario):** El canal lógico eSCO-S también lleva los datos síncronos del usuario, pero este canal lógico es transportado sobre un enlace eSCO.

Medios de Conexión

Como ya se discutió, una *piconet* puede estar formada por un máximo de siete esclavos activos y muchos más en modo *park*. Además existen otros dos tipos de modos llamados *sniff* (escucha) y *hold* (espera). Esta cantidad de modos facilita el ahorro de energía de los dispositivos. Estos modos de bajo consumo son en realidad tres métodos diferentes para entrar y salir de un estado de bajo consumo. También hay que resaltar que dichos modos se aplican solo a una determinada conexión Bluetooth y no al dispositivo en su conjunto. Debido a que no es necesario tener un enlace de comunicación activo con todos los dispositivos al mismo tiempo, estos modos de la Banda Base permiten colocar un mayor número de dispositivos dentro de un mismo radio de proximidad. Todos estos cuatro modos (activo, *sniff*, *hold,* y *park*) solo aplican cuando la Banda Base está en un estado conectado. Cuando no lo está, la Banda Base se encuentra en un estado prevenido, que no debe ser confundido con ninguno de los modos en los que puede encontrarse el estado conectado.

En el modo **activo** un esclavo esencialmente siempre está escuchando el medio en espera de transmisiones provenientes del maestro. Los esclavos activos reciben paquetes que los mantienen sincronizados con el maestro y les informan cuando pueden transmitir paquetes dirigidos al maestro. Un esclavo activo debe leer todos los paquetes que provengan del maestro, aunque se permite una optimización donde los esclavos activos no necesiten leer el paquete entero (solo la cabecera del paquete) cuando este conozca que existen otros esclavos activos comunicándose con el maestro. El estado activo típicamente provee el tiempo de respuesta más rápido pero también consume la mayor cantidad de energía, ya que siempre está recibiendo paquetes y siempre está preparado para transmitir paquetes.

El modo ***sniff*** se emplea para reducir el consumo de energía. En este modo un esclavo esencialmente se convierte en esclavo activo en intervalos de tiempo periódicos. El maestro acuerda transmitir paquetes destinados a un esclavo en particular solo en intervalos regulares de tiempo (aunque puede que no transmita paquetes en todos estos intervalos). Luego, el esclavo solo necesita escuchar a la espera de paquetes del maestro al inicio de cada uno de estos intervalos (siempre sujeto a cierta tolerancia de tiempo). Si el esclavo recibe paquetes al inicio de dicho intervalo este continuará escuchando y recibiendo paquetes; por el contrario puede “dormirse” hasta el inicio del siguiente intervalo. El modo *sniff* reduce el consumo de energía ya que reduce el promedio de ciclo de trabajo del radio Bluetooth, pero es seguramente menos rápido que el modo activo. El consumo de energía y la capacidad para responder rápidamente en el modo *sniff* van a depender del tamaño del intervalo de escucha.

En modo ***hold*** el dispositivo esclavo puede detener enteramente la escucha de paquetes por un intervalo de tiempo. El maestro y el esclavo acuerdan un período de tiempo determinado para detener el enlace de comunicación. Durante este tiempo que el esclavo se encuentra detenido no necesita escuchar los paquetes que provengan del maestro y puede dedicarse a hacer otras cosas como establecer enlaces con otros dispositivos, o puede solo “dormirse” durante este tiempo. Al concluir el período durante el cual el esclavo se encuentra detenido, este reanuda la escucha de paquetes provenientes del maestro. En este modo se disminuye la

capacidad para responder rápidamente incluso más que en el modo *sniff*, por lo cual puede permitir un mayor ahorro de energía, aunque esto también va a depender del tiempo que el esclavo se encuentre en este modo y que esté haciendo durante este tiempo.

En el modo ***park*** un esclavo mantiene la sincronización con el maestro pero no es considerado mas como activo (ya que los esclavos en modo activo, *sniff,* y *hold* son considerados activos). Ya que solo puede haber hasta siete esclavos activos dentro de una *piconet* en un mismo momento, el uso del modo *park* permite al maestro orquestar comunicaciones con más de siete dispositivos al intercambiar esclavos activos y en modo *park* y así mantener hasta siete conexiones activas mientras que el resto de los esclavos de la *piconet* permanecen en modo *park*. Un esclavo en modo *park* todavía necesita mantener la sincronización con el maestro y lo logra al escuchar al maestro periódicamente. El modo *park* es el modo con la menor capacidad para responder rápidamente de todos los modos de conexión, ya que el esclavo debe hacer la transición para convertirse en un miembro activo de la *piconet* antes de reanudar las comunicaciones generales, pero el modo *park* permite el mayor ahorro de energía.

La Figura 2.7 muestra la relación entre los diferentes modos en términos de su capacidad de responder rápidamente versus el consumo de energía. Sin embargo, tanto el consumo de energía y la capacidad de responder rápidamente en estos modos es altamente dependiente de factores como la cantidad de tráfico y los períodos de espera, que pueden afectar los ciclos de trabajo de los radios Bluetooth. Como regla general, los esclavos activos van a consumir la mayor cantidad de energía pero tienen mayor capacidad para responder rápidamente, mientras que los esclavos en modo *park* tendrán la menor capacidad de respuesta de los cuatro modos. La Figura 2.7 ilustra la tendencia general, aunque esta relación puede variar en casos específicos.

Figura 2.7 Comparación de modos de conexión.

Direccionamiento

Existen básicamente cuatro tipos de direcciones a nivel de la Banda Base. La primera, la BD_ADDR (dirección del dispositivo Bluetooth), es una dirección de 48 bits, que se considera la entidad más estática para estos dispositivos, ya que es gravada electrónicamente en cada dispositivo. Una BD_ADDR es única y para garantizar esta unicidad, una autoridad numeradora asigna dichas direcciones. Esta dirección es similar a la dirección MAC (*Medium Access Control*, Control de Acceso al Medio) de IEEE 802.xx usada en los dispositivos de red de área local.

El campo de direccionamiento de 48 bits, que se muestra en la Figura 2.8 desde el bit menos significativo hasta el bit más significativo, es particionado en tres partes: la parte más baja de la dirección LAP (*Lower Address Part*), la parte más alta de la dirección UAP (*Upper Address Part*), y la parte no significativa de la dirección NAP (*Non-significant Address Part*). Los 24 bits de la parte UAP Y NAP constituyen la parte del identificador único de la organización OUI (*Organization Unique Identifier*) que es asignado por la autoridad numeradora a diferentes organizaciones. La parte LAP es asignada internamente por cada organización. Las distintas partes de la BD_ADDR están involucradas en casi todas las operaciones de la Banda Base para la identificación de una *piconet*, para el control de errores de la cabecera, para la autenticación y para la generación de llaves de cifrado.

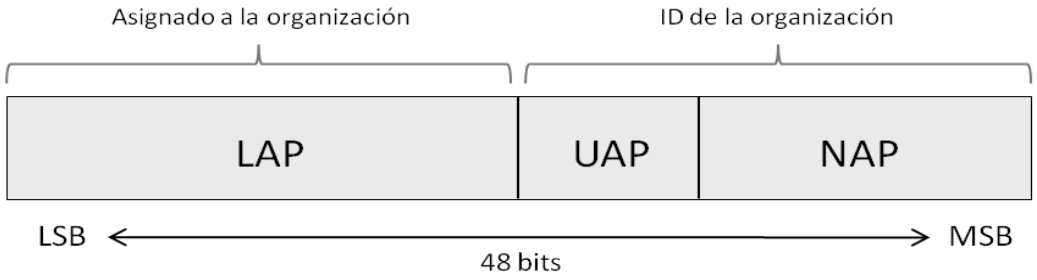

Figura 2.8 Formato de dirección Bluetooth.

Más allá de la BD_ADDR existen otros tres tipos de direcciones que se le pueden asignar a los dispositivos:

- AM_ADDR (*Active Member Address*, Dirección de Miembro Activo): Dirección de 3 bits de un miembro activo de una *piconet*. Todos los bits de este campo en cero están destinados para mensajes de Broadcast.
- PM_ADDR (*Parked Member Address*, Dirección de Miembro en modo *park*): Dirección de 8 bits destinada para esclavos en modo *park* de una *piconet*.
- AR_ADDR (*Access Request Address*, Dirección de Acceso Requerida): La dirección de solicitud de acceso es usada por los esclavos en modo *park* para determinar la segunda mitad de la ranura de tiempo de ventana de acceso para mandar mensajes de petición.

Procedimientos de Acceso

Un dispositivo Bluetooth puede asumir distintos estados operacionales. Un dispositivo se encuentra en un estado conectado cuando es miembro de una *piconet*. Por otro lado, cuando un dispositivo no está asociado a ninguna *piconet* o no se encuentra en el proceso que resulte en la formación de una, se dice que se encuentra en un estado prevenido, que es el estado operacional por defecto para todo dispositivo Bluetooth, en el cual, típicamente un dispositivo permanece ocioso, con solo su reloj nativo operando en un modo de bajo consumo de energía.

Para que un dispositivo cambie a un estado conectado, este va a atravesar por un estado *inquiry* (pregunta) y un estado *page* (búsqueda), que son instanciados de forma distinta pero complementaria. En el estado *inquiry*, un dispositivo indaga sobre la identidad de otros dispositivos vecinos; estos otros dispositivos deben encontrarse en un estado *inquiry scan* para escuchar y en consecuencia responder los mensajes *inquiry* que reciba. En el estado *page*, un dispositivo explícitamente invita a otro a participar en una *piconet*, donde el maestro será aquel dispositivo que hace la invitación; el otro dispositivo debe estar en un estado de *page scan* para escuchar los mensajes de búsqueda y en consecuencia responder a los mismos. Como se muestra en la Figura 2.9, un dispositivo puede saltar el estado *inquiry* si la identidad del dispositivo que va a ser "buscado" ya se conoce. La figura también sugiere que mientras un dispositivo es miembro de una *piconet*, este puede continuar realizando "*inquiries*" y "*pages*" a dispositivos adicionales para que se unan a esta o a otra *piconet*, y eventualmente llegar a formar una *scatternet*.

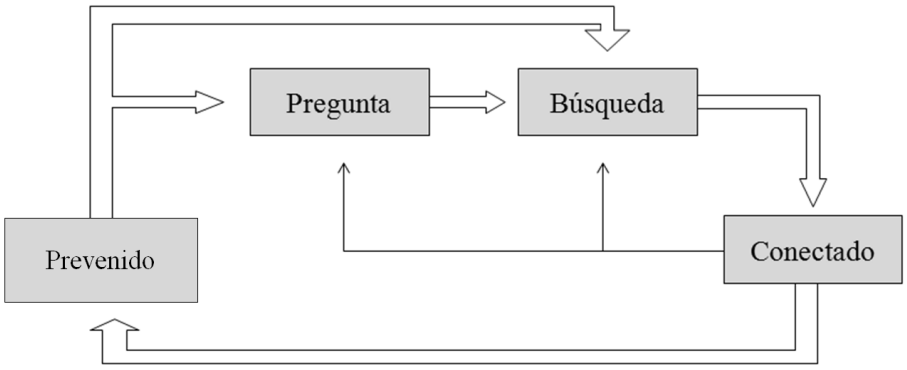

Figura 2.9 Procedimientos de acceso.

Proceso Page (Búsqueda)

Durante este proceso un dispositivo invita a otro a unirse a una *piconet* utilizando un mensaje *page*. El dispositivo que emite el mensaje *page* es el potencial maestro de la *piconet*, y el dispositivo que escucha (o escanea) los *pages* es el potencial esclavo de la *piconet*. Un maestro selecciona una frecuencia nueva con la cual transmitir los mensajes *page* cada 312,5 µseg. Durante el *page scan*, un dispositivo escanea los mensajes de búsqueda que hayan sido transmitidos, seleccionando una frecuencia nueva de escaneo cada 1,28 segundos.

Es de notar que el maestro cambia de frecuencia a una tasa más alta que el esclavo. El esclavo utiliza su propio reloj para manejar su FSM (*Frequency-Selection Module*) que se encarga de seleccionar la próxima frecuencia a ser utilizada. El maestro estima el valor del reloj del esclavo basándose en la última comunicación entre ambos dispositivos. Y en el peor de los casos, el maestro utilizará su propio reloj.

Durante la operación *page*, se utiliza una secuencia de saltos de búsqueda. Para generar dicha secuencia, el maestro y el esclavo usan los 28 bits menos significativos de la dirección del esclavo como entrada de sus respectivos FSM. Para cada dispositivo, esta secuencia de saltos de búsqueda se define por una secuencia periódica compuesta por 32 (o en su defecto 16) frecuencias distribuidas uniformemente sobre las 79 (o en su defecto 23) canales de frecuencia de la banda de los 2.4 GHz.

Proceso Inquiry (Pregunta)

El propósito de un dispositivo de utilizar el proceso *inquiry* se debe a la necesidad de recolectar información sobre otros dispositivos Bluetooth cercanos. Principalmente involucra obtener elementos fundamentales tales como: la BD_ ADDR y el valor del reloj del otro dispositivo. Este estado *inquiry* está compuesto por varios subestados que podrán ser asumidos por potenciales maestros y esclavos. El estado *inquiry* será asumido por un potencial maestro y los estados de *inquiry scan* e *inquiry response* serán asumidos por potenciales esclavos.

Al igual que en el proceso *page*, un dispositivo que se encuentre indagando la información de otros dispositivos cercanos, va a seleccionar una nueva frecuencia con la cual transmitir un paquete *inquiry* cada 312 µseg. Los dispositivos que reciben estos paquetes deben estar en el estado *inquiry scan*, y seleccionar una nueva frecuencia para escanear cada 1,28 segundos. Es de notar que el potencial maestro cambia de frecuencias a una tasa más alta que el potencial esclavo. Ambos dispositivos utilizan su propio reloj para manejar su FSM.

Existe un código GIAC (*General Inquiry Access Code*, Código de Acceso a Pregunta General) para preguntar por dispositivos en general, y una serie de códigos de acceso de pregunta dedicados DIAC (*Dedicated Inquiry Access Code*, Código de Acceso a Pregunta Dedicado) para preguntar por dispositivos en específico.

Para el proceso *inquiry*, el maestro escucha las diferentes respuestas, pero nada más lee una y continúa escaneando por otras respuestas. En el caso de que exista contienda entre

diferentes dispositivos, éstos, al no recibir respuesta del maestro, esperan un número aleatorio de ranuras de tiempo y se mantienen a la escucha de un nuevo mensaje *inquiry* que provenga del maestro. El mensaje de respuesta consiste en un paquete FHS que tiene los parámetros del dispositivo.

Se utiliza una secuencia de saltos de pregunta. Para generar dicha secuencia, los dispositivos utilizan los 28 bits menos significativos de la dirección de pregunta. Utilizan el GIAC como la dirección de entrada para sus respectivos FSM. La secuencia de saltos de pregunta es una secuencia periódica bien definida compuesta por 32 (o en su defecto 16) frecuencias distribuidas uniformemente sobre los 79 (o en su defecto 23) canales de frecuencia de la banda de los 2.4 GHz.

Estados y Subestados Bluetooth

La Figura 2.10 muestra todos los posibles estados, subestados, y transiciones que puede asumir un dispositivo Bluetooth.

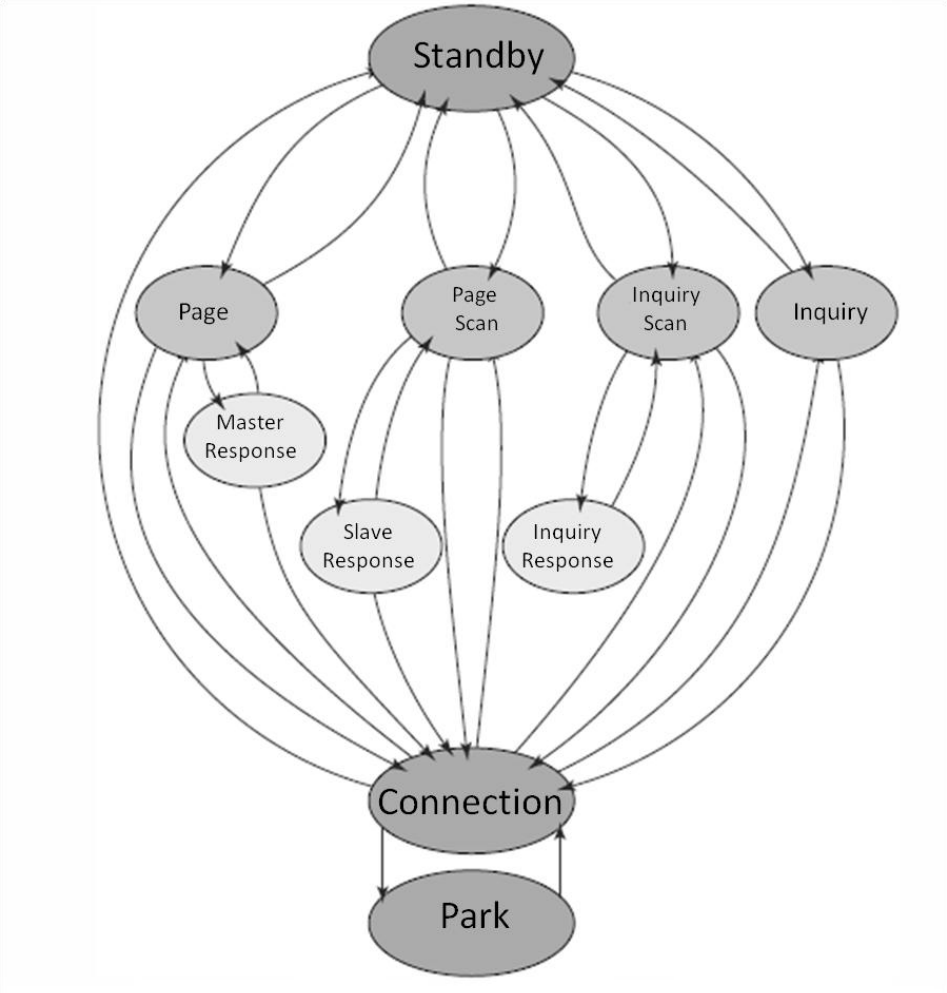

Figura 2.10 Diagrama de estados Bluetooth [4].

Audio Bluetooth

No existe un grupo de trabajo para el audio per se dentro del SIG. Sin embargo ha sido una parte heredada de las comunicaciones inalámbricas de Bluetooth desde su concepción y siempre ha estado integrado como parte fundamental en el diseño de la pila de protocolos. El audio (de voz o de otro tipo) es transportado sobre un enlace SCO de la capa de Banda Base.

Mientras que el audio de Bluetooth no necesariamente es de uso exclusivo para el transporte de voz, el diseño está optimizado para el transporte de la misma. El sonido tiende a ser continuo durante períodos de tiempo y por lo tanto isócrono, o de tiempo limitado. La tasa de transmisión para el tráfico de audio Bluetooth está fijado en 64 Kbps, suficiente para una conversación normal. Si bien no se opone la comunicación de otros medios de audio (por ejemplo, música) a través de un enlace de audio Bluetooth, el diseño no se basa en el tráfico de audio, sino que claramente se centra en el tráfico de voz.

Se utilizan dos esquemas de codificación para el audio de Bluetooth, como se resume en la Tabla 2.5. El primero es el esquema PCM (*Pulse Codec Modulation*, Modulación de Pulsos Codificados) donde se aplican cualquiera de los dos tipos de compresión logarítmica (llamados A-law y μ-law). El segundo esquema es CVSD (*Continuous Variable Slope Delta*, Codificación de Pendiente Continua de Variable Delta). Las características de las conversaciones de voz normal, que tienen una continuidad más predecible que la del audio en general, hacen de CVSD más eficiente. También habitualmente es más tolerante a los errores de comunicación. Por lo tanto CVSD, en general, es el método más efectivo y eficiente para las comunicaciones de audio de Bluetooth.

Codecs de voz	
Logarítmico de 8 bits	A-law
	μ-law
Lineal	CVSD

Tabla 2.5 Esquema de codificación de voz.

2.2.3. Link Management Protocol (LMP)

LMP es el protocolo responsable de establecer y controlar los enlaces entre los dispositivos Bluetooth. Se encarga de la negociación del tamaño de los paquetes de la Banda Base; y de los aspectos de seguridad como la autenticación y cifrado, al generar, intercambiar, y comprobar las llaves de cifrado. LMP también controla los modos de consumo de energía y los ciclos de trabajo de los dispositivos de radio Bluetooth, al igual que los estados de conexión de una unidad Bluetooth en una *piconet*.

Los mensajes LMP son filtrados e interpretados por el gestor de enlace del lado del receptor, por lo tanto estos nunca son replicados a las capas superiores. Los mensajes LMP no transportan datos de aplicación, solo información de control,- y tienen una mayor prioridad que los datos de usuario. Si un gestor de enlace necesita enviar un mensaje, este no va a ser retrasado por el tráfico L2CAP. Adicionalmente, los mensajes LMP no son reconocidos

explícitamente, ya que el canal lógico provee un enlace suficientemente confiable, haciendo de los reconocimientos una tarea innecesaria.

Formato del Paquete LMP

Las PDU LC son siempre enviadas como paquetes de una única ranura de tiempo, y la cabecera de datos tiene una longitud de un byte, siendo los dos bits menos significativos (campo L_CH) los que determinan el canal lógico. El formato de la cabecera puede ser observado en la Figura 2.11. El bit del campo flujo siempre tiene un valor de '1' y es ignorado en el extremo receptor.

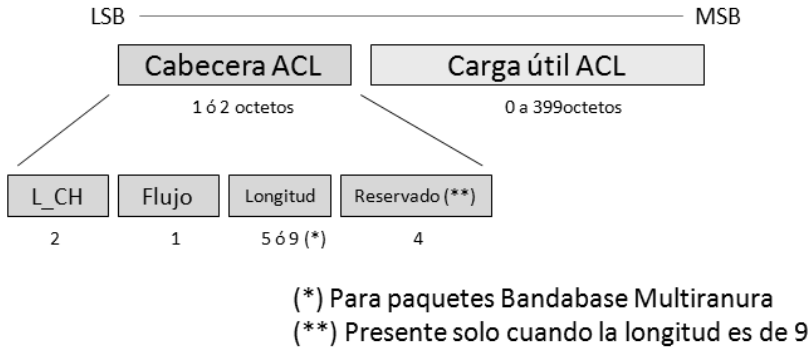

Figura 2.11 Formato de cabecera LMP.

Las LMP_PDU que se muestra en la Figura 2.12 son transportadas en la carga útil de los paquetes ACL cuyo campo L_CH tiene un valor binario de 'b11'. Las LMP_PDU son transmitidas sobre una única ranura de tiempo sobre un paquete DM1 o sobre un paquete DV. Las LMP_PDU tienen alta prioridad, y si es necesario pueden adelantarse incluso a una transmisión SCO para transmitir información de control a otro dispositivo. La Tabla 2.6 resume el formato de los paquetes LMP_PDU.

Campo	Tamaño	Descripción
ID de transacción	1 bit	'b0': corresponde a una transacción iniciada por el maestro.
		'b1': corresponde a una transacción iniciada por el esclavo.
opCode	7 bits	Identifica la LMP_PDU y el tipo de contenido que transporta.
Carga útil	0-17 bytes	Si un enlace SCO está presente usando paquetes HV1 y la longitud del campo de datos es menor de 9 bytes, las PDU se pueden transmitir en paquetes DV. En caso contrario, se usarán paquetes DM1.

Tabla 2.6 Formato de paquete LMP.

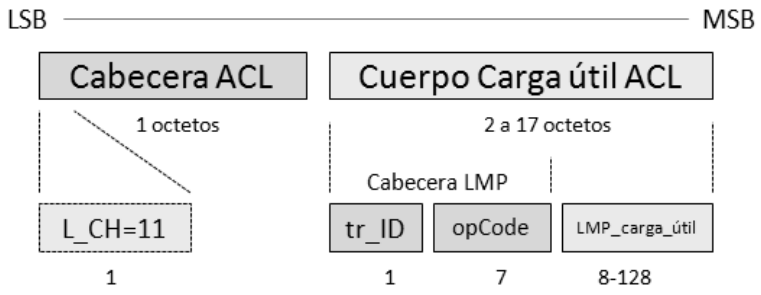

Figura 2.12 Formato paquete LMP.

Procedimientos LMP

LMP utiliza una gran variedad de procedimientos relacionados con la gestión de las *piconets* y la seguridad. Cada procedimiento específico posee sus propias PDU, que son intercambiadas entre las dos entidades LC tal como se describe y se representa en la Figura 2.13 donde las líneas continuas representan paquetes obligatorios y las punteadas paquetes opcionales:

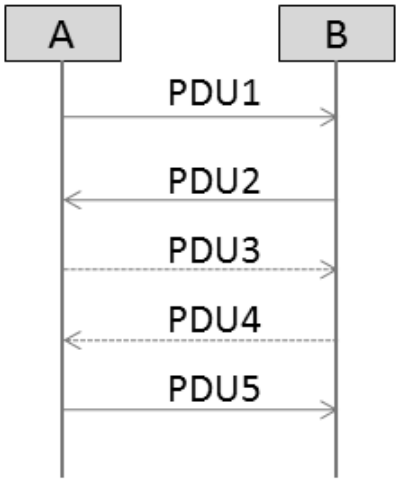

Figura 2.13 Procedimientos LMP.

En total, LMP utiliza 27 procedimientos, donde hay 16 procedimientos obligatorios (que deben ser implementados) y 11 procedimientos opcionales, cuya descripción y especificación es bastante extensa, por ello solo serán listados a continuación:

Procedimientos Obligatorios:

- Respuesta general.
- Autenticación.
- *Pairing* (o emparejamiento, proceso mediante el cual dos dispositivos intercambian una clave privada usada para el cifrado).
- Cambio de la clave de enlace.
- Cambio de la clave de enlace actual.
- Petición del *offset* del reloj.
- Información de petición de nombre.
- Desconexión.
- Calidad de servicio.
- Control de la versión LMP. Información de las características soportadas de paquetes multi-ranura.
- Supervisión del enlace.
- Establecimiento de conexión.
- Modo de prueba.
- Manejo de errores.

Procedimientos Opcionales:

- Cifrado.
- Información del desplazamiento de la ranura de tiempo.
- Petición de información de temporización.
- Petición de cambio de rol maestro-esclavo.
- Petición de modo *hold*.
- Petición de modo *sniff*.
- Petición de modo *park*.
- Control de energía.
- Cambio de canal impulsado por la calidad (Entre DM y DH).
- Establecimiento de enlace SCO.
- Petición de esquema de búsqueda.

Establecimiento de una Conexión

Cuando una unidad de búsqueda desea crear una conexión que involucre las capas superiores del LM, este envía un PDU LMP_host_connection_req. Cuando el otro extremo recibe este mensaje, el dispositivo es informado sobre la conexión entrante. El dispositivo remoto puede aceptar o rechazar esta petición de conexión enviando un PDU LMP_accepted o un PDU LMP_not_accepted respectivamente.

Por otro lado, si el esclavo necesita cambiar de rol con el maestro, este envía un PDU LMP_slot_offset y un PDU LMP_switch_req una vez haya recibido el PDU LMP_host_connection_req. Si el cambio de rol falla, el LM debe continuar con la creación de la conexión a menos que no la pueda soportar debido a falta de recursos, en tal caso la conexión debe ser terminada con el siguiente PDU LMP_detach, con un código de error (*other end*

terminated connection: low resources, el otro extremo terminó la conexión: bajos recursos). Cuando el cambio de roles es completado satisfactoriamente, el antiguo esclavo va a responder con un PDU LMP_accepted o un PDU LMP_not_accepted para el PDU LMP_host_connection_req (con un ID de transacción con el valor de 0).

Si la unidad de búsqueda recibe un PDU LMP_not_accepted en respuesta a un PDU LMP_host_connection_req este debe desconectar el enlace inmediatamente.

2.2.4. Logical Link Control and Adaptation Protocol (L2CAP)

El principal objetivo de la capa L2CAP es ocultar las particularidades de los protocolos de transporte de las capas inferiores a las capas superiores. De este modo, un gran número de protocolos de transporte y aplicaciones de las capas superiores que ya han sido desarrollados pueden utilizar los enlaces Bluetooth, con pocas, o ninguna modificación.

La capa L2CAP solo utiliza enlaces ACL. Los paquetes de esta capa, conocidos como L2CAP_PDUs, son transportados sobre los BB_PDUs ACL cuyo campo de la cabecera L_CH contiene el valor ‘b10’, que denota el inicio de un L2CAP_PDU, o ‘b01’, que denota la continuación de un L2CAP_PDU. A pesar de que los L2CAP_PDUs están cercanamente asociadas con las BB_PDUs ACL, los conceptos de los protocolos de transporte de las capas inferiores como: maestro, esclavo, secuencias de saltos de frecuencia, reloj nativo, y muchos otros son transparentes para la capa L2CAP. Las capas inferiores de transporte proveen al equivalente de una interfaz de paquetes para L2CAP sobre la cual L2CAP envía y recibe los datos y los mensajes de control, pero L2CAP y las capas superiores ignoran el funcionamiento de dichos protocolos inferiores de transporte.

La capa L2CAP soporta la multiplexación de los protocolos de las capas superiores, debido a que el protocolo Banda Base no soporta ningún campo de tipo que identifique las capas altas de protocolos. L2CAP es capaz de distinguir entre los protocolos de la capa superior SDP, RFCOMM y TCS. Además, facilita la segmentación y reensamblado de los paquetes de gran tamaño de las capas superiores para que se ajusten a los paquetes más pequeños utilizados en la Banda Base. Por otro lado, la capa L2CAP informa a las capas superiores del tamaño máximo del paquete que su par L2CAP en el otro dispositivo puede manejar. Por lo tanto es responsabilidad de las capas superiores fragmentar su información en paquetes que no violen dicho tamaño.

Adicionalmente, la capa L2CAP soporta el intercambio de información de calidad de servicio, con la cual ayuda a controlar los recursos de transmisión de forma tal que soporte el QoS esperado. Finalmente, la capa L2CAP también provee un grupo de abstracción a las capas superiores, esto permite asignar grupos de direcciones de protocolos de las capas superiores dentro de una *piconet* sin exponer dicho concepto a las mencionadas capas.

La capa L2CAP asume que las características de las capas inferiores proporcionan un canal de comunicaciones full-duplex que entrega los L2CAP_PDUs de forma ordenada. L2CAP por sí mismo no provee de mecanismos para asegurar una transmisión confiable de sus PDUs.

Sin embargo, se basa en el proceso de retransmisión que provee la capa de Banda Base que soporta un canal de comunicaciones lo suficientemente confiable para las capas superiores.

Paquetes Usados por L2CAP

L2CAP es un protocolo basado en paquetes pero sigue un modelo de comunicaciones basado en canales. Un canal representa un flujo de datos entre dos entidades L2CAP de dos dispositivos remotos. Los canales pueden ser orientados a conexión o no orientados a conexión. Los puntos finales de los canales están identificados por los identificadores de canal CID (*Channel Identifier*) cuyo tamaño es de dos octetos. Los canales identificados con un valor de CID de 0x0002 son no orientados a conexión, y por otro lado los canales de señalización (con CIDs 0x0001 y 0x0005) son orientados a conexión. Todos los canales con CID asignados dinámicamente son orientados a conexión. Todos los campos deben estar ordenados de la forma LSB **[4]**.

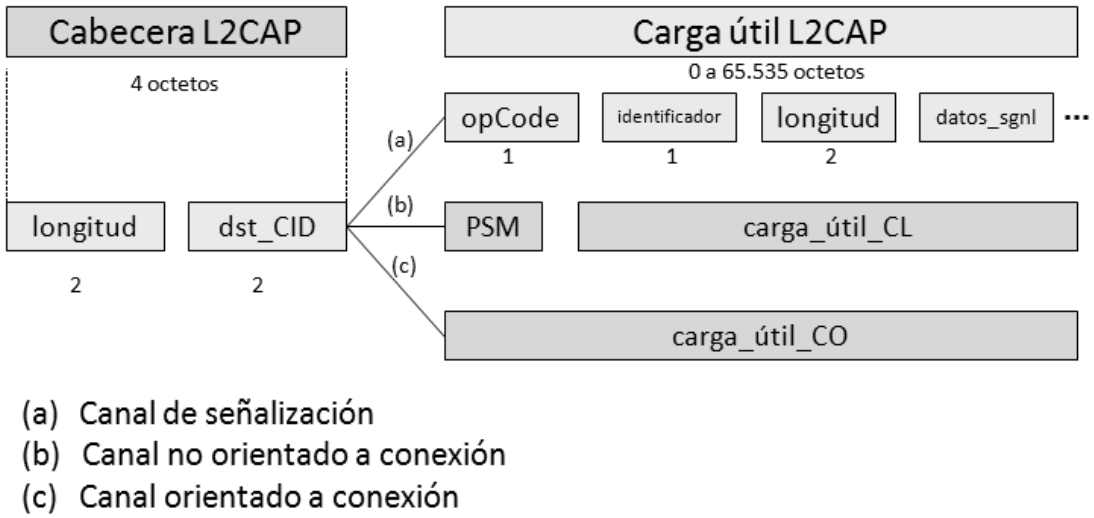

Figura 2.14 Formato paquete L2CAP.

Como se muestra en la Figura 2.14, la carga útil de los paquetes de señalización contienen los siguientes campos:

- Un campo opCode de un octeto que identifica los tipo de datos de señalización.
- Un campo de identificación de un octeto usado para coincidir las respuestas con las solicitudes.
- Un campo de dos bytes que contiene el tamaño del campo de datos.
- Y por último los datos de señalización.

También se muestra el formato de los paquetes L2CAP no orientados a conexión, el cual posee un campo PSM (*Protocol/Service Multiplexer*, Multiplexor de Protocolo/Servicio) utilizado para soportar el protocolo de multiplexación en los canales L2CAP. Para los canales orientados

a conexión este campo PSM está incluido en los datos de señalización que se envía en la solicitud de conexión utilizando los paquetes de señalización L2CAP.

2.2.5. Service Discovery Protocol (SDP)

El descubrimiento de servicios es un proceso mediante el cual los dispositivos pueden localizar y recopilar información y en última instancia, hacer uso de otros servicios de la red. En las redes tradicionales como las LAN (*Local Area Network*), estos servicios pueden estar configurados estáticamente y gestionados por un administrador de red. En estos ambientes, el administrador o el usuario final realizan la configuración necesaria para que un participante de la red pueda utilizar otro servicio de algún miembro de la red.

Para lograr los objetivos de simplicidad y versatilidad, los dispositivos y las redes a las cuales se incorporen, deben funcionar una vez que estos empiezan a trabajar. Es decir, que los aparatos involucrados dentro de la red deben ser auto-configurables. Y esto se refiere a que los dispositivos de red y servicios, simplemente se descubran mutuamente, negocien lo que deban hacer y con cuáles dispositivos necesitan colaborar sin ninguna intervención del usuario.

Los protocolos para la detección de servicios pueden ayudar a lograr tal auto-configuración. Dado que gran parte de la comunicación entre los dispositivos que hacen uso de Bluetooth es del tipo punto a punto o de naturaleza ad-hoc, el SIG determinó que un protocolo de descubrimiento de servicios en la pila de protocolos podría ser de gran valor. El protocolo resultante, conocido como SDP (*Service Discovery Protocol*, Protocolo de Descubrimiento de Servicios), es un componente central de casi todos los perfiles y casos de uso en dicha tecnología.

Como ya es evidente en otras capas de la pila de protocolos, se han adoptado protocolos ya existentes. Sin embargo, en este caso, el SIG desarrolló su propio protocolo único y optimizado para la comunicación inalámbrica Bluetooth en lugar de adoptar otro protocolo de descubrimiento de servicios de la industria **[5]**. En consecuencia se tiene un protocolo que debe poseer las siguientes características **[4]**:

1. Proporcionar a los clientes la capacidad de buscar servicios basados en atributos específicos.
2. Permitir que los servicios sean descubiertos basándose en la clase de servicio.
3. Realizar una navegación a través de los servicios sin necesidad de tener conocimientos de las características específicas de dichos servicios.
4. Proporcionar el medio de descubrir nuevos servicios que vayan a estar disponibles cuando los dispositivos entren en un área de proximidad RF (radiofrecuencia) con un dispositivo cliente.
5. Proporcionar un mecanismo para determinar cuando un servicio pasa al estado de indisponibilidad cuando los dispositivos abandonen esa área RF de proximidad de un dispositivo cliente.
6. Suministrar los servicios, clases de servicio y atributos de servicios para conseguir una identificación de forma única.
7. Permitir a un cliente en un dispositivo descubrir un servicio en otro dispositivo sin tener que consultar un tercer dispositivo.

8. SDP debe ser apropiado para usarse en dispositivos de complejidad limitada.
9. Suministrar un mecanismo para descubrir información de forma gradual sobre los servicios que ofrezca un dispositivo. Así, se pretende minimizar la cantidad de datos que deben ser intercambiados para que un cliente pueda determinar si un determinado servicio no le es necesario.
10. Soportar agentes intermediarios que mejoren la velocidad o eficiencia del proceso de descubrimiento.
11. Debe ser independiente del transporte.
12. Debe funcionar mientras usa L2CAP como su protocolo de transporte.
13. Debe permitir el descubrimiento y uso de servicios que proporcionen acceso a otros protocolos de descubrimiento de servicios.
14. Debe soportar la creación y definición de nuevos servicios sin requerir registrarse con una autoridad central.

Formato de una PDU SDP

Cada PDU SDP como se muestra en la Figura 2.15 está compuesta de una cabecera seguida de parámetros específicos de la PDU. La cabecera contiene tres campos: un ID de PDU, un ID de transacción y la longitud de los parámetros.

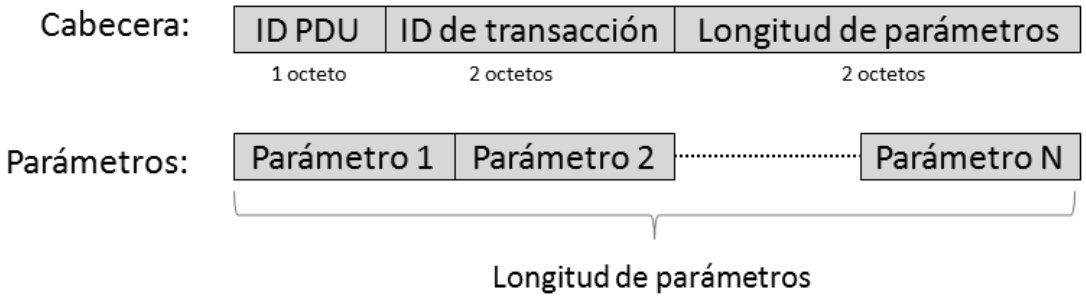

Figura 2.15 Formato paquete SDP.

- El campo ID de PDU identifica el tipo de PDU. Por ejemplo si es de petición o respuesta.
- El campo ID de transacción es utilizado para identificar unívocamente las PDU de petición y también para hacer coincidir las PDU de respuesta con dichas peticiones.
- El campo longitud de los parámetros específica el tamaño (en bytes) de todos los parámetros que contiene la PDU.

2.2.6. RFCOMM

RFCOMM es un protocolo simple de transporte, el cual provee emulación de los puertos seriales sobre el protocolo L2CAP. El protocolo se basa en el estándar ETSI TS 07.10 **[7]**. Soporta hasta 60 conexiones simultáneas entre dos dispositivos Bluetooth, aunque este número también es específico a cada implementación. El protocolo emula los 9 circuitos de una interfaz RS-232 tal como se muestra en la Tabla 2.7.

Pin	Nombre de circuito
102	*Signal Common* (Señal Común)
103	*Transmit Data* (TD, Transmisión de Datos)
104	*Received Data* (RD, Datos Recibidos)
105	*Request to Send* (RTS, Solicitud de Envío)
106	*Clear to Send* (CTS, Listo para Enviar)
107	*Data Set Ready* (DSR, Conjunto de datos Listos)
108	*Data Terminal Ready* (DTR, Terminal de Datos Preparado)
109	*Data Carrier Detect* (CD, Detectados Datos de Portadora)
125	*Ring Indicador* (RI, Indicador de Timbre)

Tabla 2.7 Circuitos RS-232 emulados en RFCOMM.

Básicamente, existen dos tipos de dispositivos:

- **Tipo 1:** Se trata de puntos terminales de comunicación, como las computadoras y las impresoras.
- **Tipo 2:** Son aquellos que forman parte de un segmento de comunicación, como por ejemplo, los módems.

RFCOMM no hace distinción entre ambos tipos, pero el acomodarse a ellos tiene sus consecuencias en el protocolo. Por lo tanto, la transferencia de información entre dos entidades RFCOMM se define tanto para los dispositivos tipo 1 y 2. Una parte de la información sólo se necesitará para el segundo tipo, mientras que otra se pretende que sea usada por ambos. Debido a que un dispositivo no es consciente del tipo del otro dispositivo en el camino de comunicación, cada uno debe pasar toda la información disponible especificada por el protocolo.

2.2.7. Telephony Control Protocol (TCS)

Este protocolo se encuentra basado en la recomendación ITU-T Q.931 **[12]** y realiza las siguientes funciones:

- **Control de llamada:** Señalización para el establecimiento y liberación de las llamadas de voz y datos entre dispositivos Bluetooth.
- **Gestión de grupos:** Señalización para facilitar el manejo de grupos de dispositivos Bluetooth.
- **TCS no orientado a conexión:** Intercambio de información de señalización que no esté relacionada con llamadas entrantes.

El protocolo TCS puede implementar un conjunto de comandos AT. Y dado que dichos comandos AT han sido diseñados para ser transportados a través de enlaces seriales, los dispositivos Bluetooth utilizan el protocolo RFCOMM para enviar y recibir señales de control basadas en el conjunto de comandos AT.

El conjunto de comandos AT pueden ser utilizados para soportar aplicaciones heredadas. Además de este protocolo de control, denominado TCS-AT, los grupos de trabajo de Bluetooth

han desarrollado un protocolo adicional de control de telefonía basado en paquetes de señalización, llamado TCS-BIN (donde BIN significa *Binary Encoding of Information*, Codificación Binaria de Información), que se ejecuta directamente en la parte superior del protocolo L2CAP. Dicho protocolo soporta las funciones normales de control de telefonía, tales como el inicio y finalización de llamadas, detección de tonos de llamadas, aceptar llamadas entrantes, entre otras. A diferencia de TCS-AT, TCS-BIN soporta comunicaciones punto-multipunto, de este modo se permite, por ejemplo, que una estación base inalámbrica pueda pasar una señal de una llamada entrante a varios auriculares inalámbricos asociados con dicha estación base.

2.2.8. Host Controller Interface (HCI)

La HCI (*Host Controller Interface,* Controlador de Interfaz de Host) proporciona una interfaz de comandos para el controlador de la Banda Base, el gestor de enlaces, y acceso a los parámetros de configuración. Dicha interfaz facilita un método uniforme de acceso a las capacidades de la Banda Base de Bluetooth.

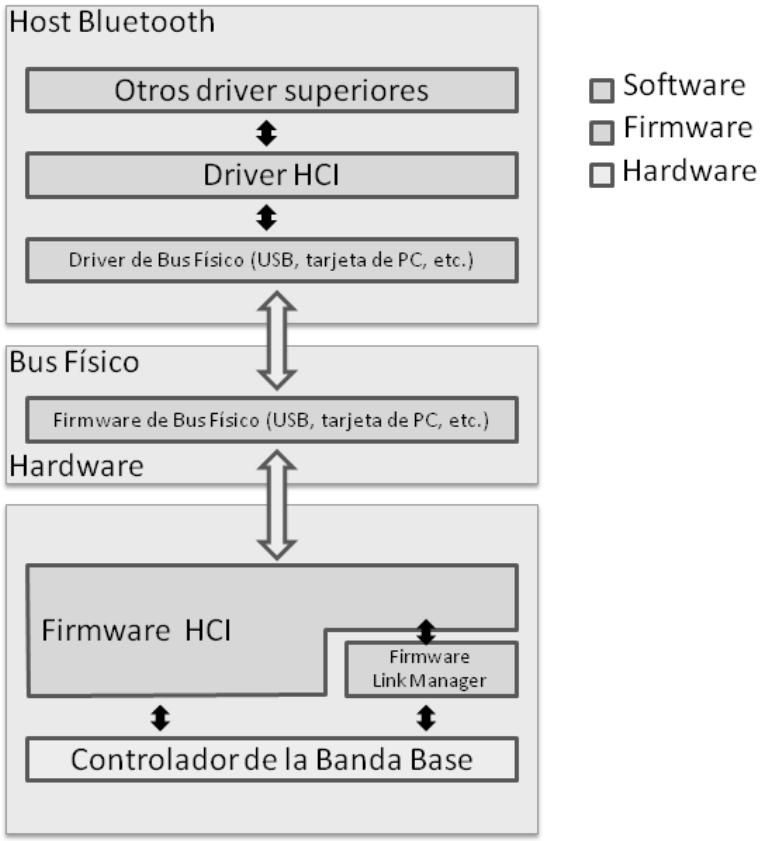

Figura 2.16 Capas inferiores de software HCI.

La Figura 2.16 proporciona una visión general de las capas inferiores de software. El firmware HCI implementa los comandos hardware HCI mediante el acceso a los comandos de gestión a nivel de enlace, a los registros de hardware de estado, registros de control y de

eventos. Varias capas pueden existir entre el controlador de HCI en el sistema host y el firmware HCI del dispositivo Bluetooth. Estas capas intermedias, ofrecen la posibilidad de transferencia de datos sin la necesidad del conocimiento íntimo de los mismos.

HCI es usado para aislar la Banda Base de Bluetooth y el administrador de enlaces de un protocolo de transporte tal como el RS-232 o USB (*Universal Serial Bus*, Bus Serial Universal). Esto permite una interfaz estándar para el hardware de Bluetooth. Un manejador de dispositivos HCI en el host es usado para interactuar con una aplicación Bluetooth con el protocolo de transporte. Actualmente existen tres mecanismos de transporte soportados: USB, RS-232 y el UART (*Universal Asynchronous Receiver-Transmitter*, Transmisor/Receptor Asíncrono Universal). Utilizando HCI, una aplicación Bluetooth puede acceder al hardware de Bluetooth sin el conocimiento de la capa de transporte u otros detalles de implementación del hardware.

2.3. Protocolos Adoptados por Bluetooth

Para soportar un mayor número de aplicaciones, la tecnología Bluetooth ha adoptado una serie de protocolos de comunicación de la industria, que se describen a continuación.

2.3.1. PPP (Point-to-Point Protocol, Protocolo Punto a Punto)

El protocolo Punto a Punto (PPP), definido en el RFC 1661 **[22]** proporciona un método estándar para transportar datagramas multiprotocolo sobre enlaces punto a punto. Y en la tecnología Bluetooth, está diseñado para correr sobre el protocolo RFCOMM.

2.3.2. TCP/UDP/IP

La implementación de TCP, UDP e IP en dispositivos Bluetooth permite la comunicación con otro dispositivo que se encuentre conectado a Internet. Además el dispositivo Bluetooth podría ser usado como un puente a Internet.

2.3.3. OBEX (Object Exchange, Protocolo de Intercambio de Objetos)

OBEX es un protocolo de capa de sesión desarrollado originalmente por la asociación de datos por infrarrojos (IrDA) con el nombre de IrOBEX. Su principal objetivo es soportar el intercambio de objetos de una manera sencilla y espontánea. El protocolo OBEX se puede utilizar para intercambiar todo tipo de objeto como archivos, fotos, entradas de calendario (vCal) y tarjetas de presentación (vCard). Como tal, el protocolo OBEX proporciona la misma funcionalidad básica que proporciona el Protocolo de Transferencia de Hipertexto (HTTP). Como HTTP, OBEX se basa en el modelo cliente-servidor y es independiente del mecanismo de transporte **[18]**.

2.3.4. WAP

El propósito de WAP (*Wireless Application Protocol*, Protocolo de Aplicación Inalámbrica) es proporcionar las especificaciones necesarias para las aplicaciones que funcionan en

dispositivos móviles conectados a través de redes de comunicación inalámbrica. Define un conjunto de protocolos: de transporte, de sesión, y de aplicación. Los cuales se aprecian en la Figura 2.17 **[13]**.

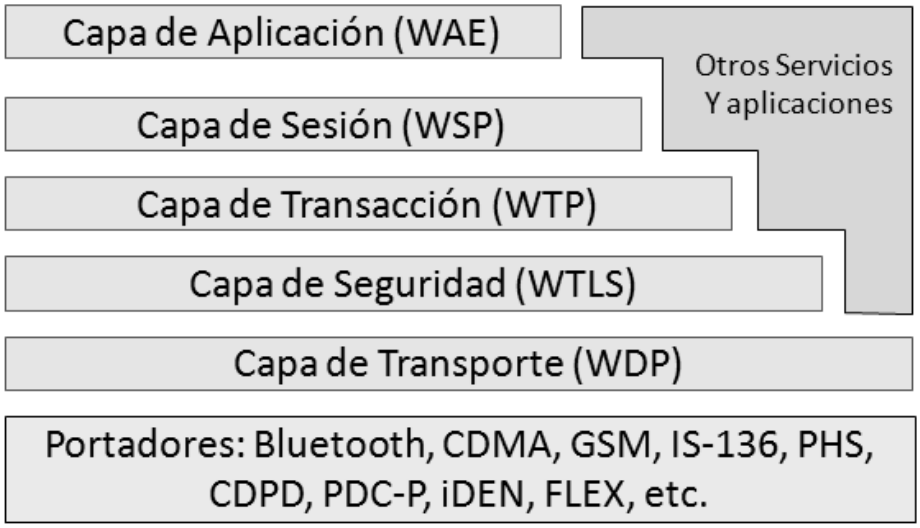

Figura 2.17 Pila de protocolos WAP.

WAP está pensado para dispositivos que poseen pantallas reducidas, por lo tanto el contenido debe ser entregado en un formato "libre de adornos". Adicionalmente, las limitaciones de ancho de banda de los servicios celulares de hoy en día significan que el contenido debe ser optimizado para ser entregado a los dispositivos de mano. Para obtener información de este modo, los sitios web están construidos con una versión más ligera del Lenguaje de Marcado de Hipertexto (HTML) llamado Lenguaje de Marcado Inalámbrico (WML). La idea de Bluetooth de adoptar WAP se basa en querer reutilizar las aplicaciones software desarrollados para el WAE (*Wireless Application Environment*, Ambiente de Aplicación WAP), entre las que se incluye navegadores WML que puedan interactuar con aplicaciones en el PC **[18]**.

2.4. Perfiles Bluetooth

Los perfiles han sido desarrollados con el fin de describir cómo se realizan las implementaciones de los modelos de uso, los cuales describen una serie de escenarios donde Bluetooth opera. Un perfil puede ser descrito como un corte vertical a través de la pila de protocolos. En él se definen las opciones de cada protocolo que son obligatorias para el perfil. También define los rangos de parámetros para cada protocolo. El concepto de perfil se utiliza para disminuir el riesgo de problemas de interoperabilidad entre productos de diferentes fabricantes.

Existen cuatro perfiles generales que son ampliamente utilizados por todo tipo de aplicaciones, y que son:

- GAP (*Generic Application Profile*, Perfil de Aplicación Genérica).
- SPP (*Serial Port Profile*, Perfil de Puerto Serial).
- SDAP (*Service Discovery Application Profile*, Perfil de Aplicación de Descubrimiento de Servicio).
- GOEP (*Generic Object Exchange Profile*, Perfil de Intercambio de Objeto Genérico).

A continuación se realizará una breve descripción de cada uno de ellos.

2.4.1. GAP

El propósito de este perfil es describir el uso de las capas más bajas de la pila de protocolos Bluetooth (Banda Base y LMP), aunque también se incluye la descripción de los aspectos de seguridad necesarios y las capas de protocolos más altas (L2CAP, RFCOMM, OBEX, etc.). Se trata del perfil más importante de todos los definidos por el SIG ya que ha de ser implementado prácticamente por la totalidad de los dispositivos Bluetooth sea cual sea su aplicación.

2.4.2. SPP

El perfil de puerto serial (SPP, *Serial Port Profile*) define los requisitos necesarios para los dispositivos Bluetooth para la creación y emulación de las conexiones a través de cables seriales utilizando RFCOMM. Los requisitos se expresan en términos de los servicios prestados a las aplicaciones, y mediante la definición de las características y procedimientos que se requieren para la interoperabilidad entre dispositivos Bluetooth.

2.4.3. SDAP

Este perfil define las características y procedimientos para que una aplicación en un dispositivo Bluetooth pueda descubrir los servicios registrados en otros dispositivos Bluetooth y de este modo recuperar cualquier información deseada, pertinente a estos servicios que esté disponible.

2.5. Seguridad en Bluetooth

La tecnología inalámbrica Bluetooth ofrece comunicaciones punto a punto en un ambiente de corto alcance. A fin de proporcionar confidencialidad, el sistema provee medidas de seguridad tanto en la capa de aplicación, como en la capa de enlace.

Esto significa que en cada dispositivo, las rutinas de autenticación y de cifrado se aplican de la misma manera. Cuatro entidades diferentes se utilizan para mantener la seguridad en la capa de enlace: la primera es la dirección del dispositivo Bluetooth, seguida de dos claves secretas, y por último un número pseudo-aleatorio que se regenera en cada nueva transacción. Las cuatro entidades y sus tamaños se resumen en la Tabla 2.8.

Entidad	Tamaño
BD_ADDR	48 bits
Llave privada de usuario conocida como PIN, autenticación	128 bits
Lave privada de usuario, cifrado con longitud configurable	8-128 bits
RAND	128 bits

Tabla 2.8 Entidades utilizadas para mantener la seguridad en la capa de enlace.

La BD_ADDR se puede obtener a través de las interacciones del usuario, o automáticamente a través de una rutina de *inquiry*.

Las claves secretas son derivadas durante el proceso de inicialización y nunca son reveladas. La clave de cifrado se deriva de la clave de autenticación durante el proceso de autenticación. Para el algoritmo de autenticación, el tamaño de la clave es siempre de 128 bits. Para el algoritmo de cifrado, el tamaño de la clave puede variar entre 1 y 16 octetos (8 a 128 bits).

La clave de cifrado es completamente diferente a la clave de autenticación (aunque esta última se utiliza cuando se crea la primera). Cada vez que se activa el cifrado, una nueva clave de cifrado se genera. Por lo tanto, la vida útil de la clave de cifrado no se corresponde necesariamente con la vida útil de la clave de autenticación.

Se prevé que las claves de autenticación son más estáticas en su naturaleza que las claves de cifrado una vez estas son establecidas, la aplicación particular que se ejecuta en el dispositivo decide cuándo cambiarlas. Para subrayar la importancia fundamental de las claves de autenticación a un enlace específico, se refiere a menudo a esta como la clave de enlace.

El RAND es un número pseudo-aleatorios que se puede derivar de un proceso aleatorio o pseudo-aleatorio en el dispositivo. Esto no es un parámetro estático y se cambia con frecuencia.

3. Escaneo Bluetooth (Bluetooth Scanning)

Debido a la masificación que ha tenido la tecnología Bluetooth, hoy en día es común que los dispositivos móviles como teléfonos celulares, teléfonos inteligentes, tabletas, PDAs, computadoras portátiles, entre otros, incluyan dentro de sus funcionalidades básicas la mencionada tecnología.

Dentro de las características de Bluetooth se encuentra la posibilidad que tiene cada dispositivo de recolectar información de otros dispositivos Bluetooth que se encuentren en un radio de alcance cercano. Esta información puede resultar bastante útil para diversos fines. En este capítulo se aborda este proceso, conocido como escaneo Bluetooth.

3.1. Escaneo Bluetooth

El escaneo Bluetooth involucra una serie de subprocesos que permiten recolectar información de los dispositivos dentro del radio de alcance del dispositivo que actúe como escáner Bluetooth, manipular los datos relevantes que se generen de dicho escaneo y almacenarlos para su posterior uso. A continuación se describen estos subprocesos que en conjunto permiten llevar a cabo un escaneo Bluetooth.

3.1.1. Proceso Inquiry de la Bandabase

Durante el proceso de descubrimiento de otros dispositivos Bluetooth se emplea el proceso *inquiry* que se explicó en detalle en el capítulo anterior, el cual está compuesto por un subestado *inquiry* que es asumido por el potencial maestro de una *piconet* y dos subestados, *inquiry scan* e *inquiry response* que son asumidos por los potenciales esclavos de una *piconet*.

Este proceso posee algunas desventajas. En primer lugar, toma mucho tiempo y produce un alto consumo de recursos si se usa regularmente. Y en segundo lugar, es bastante ineficiente para transmitir datos de forma simultánea. Sin embargo puede resultar bastante útil la utilización de este método para recolectar información de los dispositivos vecinos Bluetooth.

3.1.2. Descubrimientos de Dispositivos con HCI

La especificación de Bluetooth en la sección HCI define una interfaz de comandos y eventos asociados al descubrimiento de dispositivos que se encuentren en un área cercana **[4]**. Estos comandos y eventos asociados al descubrimiento de dispositivos se describen a continuación:

- **Comando *inquiry* (*inquiry command*):** Este comando permite que el dispositivo Bluetooth entre en modo *inquiry*. Este modo es usado para descubrir otros dispositivos Bluetooth cercanos.
- **Evento resultado de *inquiry* (*inquiry result event*):** Este evento indica que uno o múltiples dispositivos Bluetooth han respondido al proceso *inquiry* actual.

- **Evento resultado de *inquiry* con RSSI (*inquiry result with RSSI event*):** Este evento indica que uno o múltiples dispositivos Bluetooth han respondido al proceso *inquiry* actual incluyendo la información RSSI.
- **Comando cancelación de *inquiry* (*inquiry cancel command*):** Este comando trae como consecuencia la detención del proceso *inquiry* actual.
- **Evento *inquiry* completado (*inquiry complete event*):** Este evento indica que el proceso *inquiry* actual ha concluido.
- **Comando de *inquiry* periódico (*periodic inquiry mode command*):** Este comando es usado para configurar al dispositivo Bluetooth para que efectúe el proceso *inquiry* de forma automática y periódicamente.
- **Comando de salida *inquiry* periódico (*exit periodic inquiry mode command*):** Este comando es utilizado para salir del modo *inquiry* periódico.
- **Comando de lectura de actividad de escaneo *inquiry (read inquiry scan activity command*):** Este comando permite leer los valores de los parámetros de configuración para los intervalos de escaneo *inquiry*, como también la ventana de escaneo *inquiry*. El intervalo de escaneo *inquiry* define la cantidad de tiempo entre dos escaneos consecutivos *inquiry*. Y la ventana de escaneo *inquiry* define la duración del escaneo *inquiry*.
- **Comando de escritura de actividad de escaneo *inquiry* (*write inquiry scan activity command*):** Este comando permite escribir los valores de los parámetros de configuración para los intervalos de escaneo *inquiry*, como también la ventana de escaneo *inquiry*. El intervalo de escaneo *inquiry* define la cantidad de tiempo entre dos escaneos consecutivos *inquiry*. La ventana de escaneo *inquiry* define la duración del escaneo *inquiry*.
- **Comando de lectura de tipo de escaneo *inquiry* (*read inquiry scan type command*):** Este comando es usado para leer los parámetros de configuración de los tipos de escaneo *inquiry* del dispositivo. Estos parámetros pueden establecer que el escaneo *inquiry* sea de forma normal o de barrido entrelazado.
- **Comando de escritura de tipo de escaneo *inquiry* (*write inquiry scan type command*):** Este comando es usado para escribir los parámetros de configuración de los tipos de escaneo *inquiry* del dispositivo. Estos parámetros pueden establecer que el escaneo *inquiry* sea de forma normal o de barrido entrelazado.
- **Comando de lectura de modo *inquiry* (*read inquiry mode command*):** Este comando es utilizado para leer el parámetro de configuración del modo *inquiry* del dispositivo Bluetooth local.
- **Comando de escritura de modo *inquiry* (*read inquiry mode command*):** Este comando es utilizado para escribir el parámetro de configuración del modo *inquiry* del dispositivo Bluetooth local.

A continuación se explica con más detalle los comandos *inquiry*, *periodic inquiry*, y el evento de *inquiry result*, todos claves en el escaneo de dispositivos Bluetooth.

Comando Inquiry

Este comando va a provocar que el dispositivo Bluetooth entre en un modo *inquiry*. Este modo es usado para descubrir otros dispositivos Bluetooth. El parámetro de entrada LAP contiene el LAP de donde el código de acceso *inquiry* va a ser derivado cuando se realice el proceso *inquiry*. El parámetro de *Inquiry_Length* especifica el tiempo de duración del modo

inquiry. Y el último parámetro *Num_Responses* especifica el número de respuestas que podrán ser recibidas antes de que el modo *inquiry* concluya. Cuando el proceso *inquiry* termina, el controlador envía un evento de *inquiry* completado al host indicando que el proceso ha terminado. Los parámetros de este evento contendrán un resumen de los resultados obtenidos del proceso *inquiry*, donde se reportan el número de dispositivos Bluetooth que respondieron. Es de notar que cuando un dispositivo Bluetooth responde a un mensaje de pregunta, un evento *inquiry result* es generado para notificar al host del descubrimiento. La Tabla 3.1 muestra un resumen del comando.

Comando	OCF	Parámetros	Parámetros de retorno
HCI_Inquiry	0x0001	LAP, Inquiry_Length, Num_Responses	n/a

Tabla 3.1 Resumen comando *HCI_Inquiry*.

En la Tabla 3.2 muestra un resumen del parámetro LAP:

Valor	Descripción
0x9E8B00 0X9E8B3F	El LAP de donde el código de acceso *inquiry* va a ser derivado cuando se realice el proceso de *inquiry.*

Tabla 3.2 Resumen parámetro LAP del comando *HCI_Inquiry*.

En la Tabla 3.3 se muestra un resumen del parámetro Inquiry_Lenght:

Valor	Descripción
N = 0xXX	Especifica el máximo tiempo de duración del proceso *inquiry.* Tamaño: 1 octeto Rango: 0x01 – 0x30 Tiempo = N * 1,28 s Rango: 1,28 – 61,44 s

Tabla 3.3 Resumen parámetro *Inquiry_Lenght* del comando *HCI_Inquiry*.

En la Tabla 3.4 se muestra un resumen del parámetro Num_Responses:

Valor	Descripción
0x00	Número ilimitado de respuestas.
0xXX	Máximo número de respuestas que podrán ser recibidas antes de que el modo *inquiry* concluya. Rango: 0x01 – 0xFF

Tabla 3.4 Resumen parámetro *Num_Responses* del Comando *HCI_Inquiry*.

Comando Inquiry Periódico

El comando *Periodic_Inquiry_Mode* es usado para configurar al dispositivo Bluetooth para que efectúe el proceso *inquiry* de forma automática y periódicamente. Los parámetros *Max_Period_Length* y *Min_Period_Length* define el rango de tiempo entre dos preguntas consecutivas, desde el inicio de una pregunta hasta el comienzo de la próxima pregunta. El controlador va a utilizar este rango para determinar un nuevo tiempo aleatorio entre dos

preguntas consecutivas para cada pregunta. El parámetro de entrada LAP contiene el LAP de donde el código de acceso de pregunta va a ser derivado cuando se realice el proceso de pregunta. El parámetro de *Inquiry_Length* especifica el tiempo de duración del modo de *inquiry*. El último parámetro, *Num_Responses,* especifica el número de respuestas que podrán ser recibidas antes de que el modo de pregunta concluya. Este comando es completado cuando el proceso de pregunta ha sido iniciado por el dispositivo Bluetooth, un evento de comando completado es enviado del controlador al host. Cuando todo el proceso *inquiry* periódico es completado, el controlador va a enviar un evento de pregunta completada al host indicando que el último proceso de pregunta periódica ha terminado. Cuando un dispositivo Bluetooth responde a un mensaje de pregunta, un evento de resultado de pregunta es generado para notificar al host del descubrimiento. Se debe hacer notar la existencia de la siguiente relación:

Max_Period_Length > Min_Period_Length > Inquiry_Length

La Tabla 3.5 muestra un resumen del comando.

Comando	OCF	Parámetros	Parámetros de Retorno
HCI_Periodic_Inquiry_Mode	0x0003	Max_Period_Length, Min_Period_Length, LAP, Inquiry_Length, Num_Responses	*state*

Tabla 3.5 Comando *HCI_Periodic_Inquiry_Mode*.

En la Tabla 3.6 se muestra un resumen del parámetro *Max_Period_Length*:

Valor	Descripción
N = 0xXXXX	Máxima cantidad de tiempo entre dos preguntas consecutivas. Tamaño: 2 octetos Rango: 0x03 – 0xFFFF Tiempo = N * 1.28 s Rango: 3.84 – 83884.8 s 0.0 – 23.3 horas

Tabla 3.6 Parámetro *Max_Period_Length* del comando *HCI_Periodic_Inquiry_Mode*.

En la Tabla 3.7 se muestra un resumen del parámetro *Min_Period_Length*:

Valor	Descripción
N = 0xXXXX	Mínima cantidad de tiempo entre dos preguntas consecutivas. Tamaño: 2 octetos Rango: 0x03 – 0xFFFF Tiempo = N * 1.28 s Rango: 3.84 – 83884.8 s 0.0 – 23.3 horas

Tabla 3.7 Parámetro *Min_Period_Length* del comando *HCI_Periodic_Inquiry_Mode*.

En la Tabla 3.8 se muestra un resumen del parámetro LAP:

Valor	Descripción
0x9E8B00 0X9E8B3F	El LAP de donde el código de acceso *inquiry* va a ser derivado cuando se realice el proceso *inquiry*

Tabla 3.8 Parámetro LAP del Comando *HCI_Periodic_Inquiry_Mode*.

En la Tabla 3.9 se muestra un resumen del parámetro *Inquiry_Length*:

Valor	Descripción
N = 0xXX	Especifica el máximo tiempo de duración del proceso de pregunta. Tamaño: 1 octeto Rango: 0x01 – 0x30 Tiempo = N * 1,28 s Rango: 1,28 – 61,44 s

Tabla 3.9 Parámetro *Inquiry_Length* del comando *HCI_Periodic_Inquiry_Mode*.

En la Tabla 3.10 se muestra un resumen del parámetro *Num_Responses*:

Valor	Descripción
0x00	Número ilimitado de respuestas.
0xXX	Máximo número de respuestas que podrán ser recibidas antes de que el modo de pregunta concluya. Rango: 0x01 – 0xFF

Tabla 3.10 Parámetro *Num_Responses* del comando *HCI_Periodic_Inquiry_Mode*.

En la Tabla 3.11 se muestra un resumen del parámetro *state*:

Valor	Descripción
0x00	Comando de pregunta periódica exitoso.
0x01 – 0xFF	Falló comando de pregunta periódica. Código de error.

Tabla 3.11 Parámetro estado del comando *HCI_Periodic_Inquiry_Mode*.

Evento Resultado de Inquiry

Este evento indica que uno o múltiples dispositivos Bluetooth han respondido al proceso *inquiry* actual. Será enviado del controlador al host inmediatamente haya sido recibido la respuesta a una pregunta proveniente de un dispositivo remoto. El controlador puede encolar múltiples respuestas de *inquiry* en un solo evento de resultado de *inquiry*. En la Tabla 3.12 se muestra un resumen del evento.

Evento	Código de Evento	Parámetros de evento
Inquiry_Result	0x02	Num_Responses, BD_ADDR[i], Page_Scan_Repetition_Mode[i], Reserved[i], Reserved[i], Class_of_Device[i], Clock_Offset[i]

Tabla 3.12 Evento *Inquiry_Result*.

En la Tabla 3.13 se muestra un resumen del parámetro *Num_Responses*:

Valor	Descripción
0xXX	Número de respuestas de *inquiry.*

Tabla 3.13 Parámetro *Num_Responses* del evento *Inquiry_Result*.

En la Tabla 3.14 se muestra un resumen del parámetro *BD_ADDR[i]*:

Valor	Descripción
0xXXXXXXXXXXXX	La BD_ADDR de cada dispositivo que haya respondido.

Tabla 3.14 Parámetro BD_ADDR del evento *Inquiry_Result*.

En la Tabla 3.15 se muestra un resumen del parámetro *Page_Scan_Repetition_Mode*:

Valor	Descripción
0x00	R0
0x01	R1
0x02	R2
0x03 – 0xFF	Reservado

Tabla 3.15 15 Parámetro *Page_Scan_Repetition_Mode* del evento *Inquiry_Result*.

En la Tabla 3.16 se muestra un resumen del parámetro *reserved* (antiguamente *Page_Scan_Period_Mode* en la v1.1):

Valor	Descripción
0xXX	Reservado.

Tabla 3.16 Parámetro *reserved* 1 del evento *Inquiry_Result*.

En la Tabla 3.17 se muestra un resumen del parámetro *reserved* (antiguamente *Page_Scan_Mode* en la v1.1):

Valor	Descripción
0xXX	Reservado, debe ser dejado en 0x00.

Tabla 3.17 Parámetro *reserved* 2 del evento *Inquiry_Result*.

En la Tabla 3.18 se muestra un resumen del parámetro *Class_of_Device*:

Valor	Descripción
0xXXXXXX	Clase de dispositivo.

Tabla 3.18 Parámetro *Class_of_Device* del evento *Inquiry_Result*.

En la Tabla 3.19 se muestra un resumen del parámetro *Clock_Offset*:

Formato de Bit	Descripción
Bit 14-0	Bit 16-2 de CLKslave-CLKmaster.
Bit 15	Reservado.

Tabla 3.19 Parámetro *Clock_Offset* del evento *Inquiry_Result*.

3.1.3. Datos e Información que se Genera

Todo dispositivo Bluetooth es capaz de realizar descubrimiento de dispositivos, que le permiten recolectar información de otros dispositivos Bluetooth en un radio de 5 a 10 metros. Esta información incluye la BD_ADDR, el nombre del dispositivo, la clase del dispositivo, los servicios del dispositivo y opcionalmente información RSSI **[14]**.

La dirección del dispositivo Bluetooth como ya se explicó en el capítulo anterior es un identificador único del dispositivo que lo diferencia de cualquier otro dispositivo. El nombre del dispositivo es un campo que puede ser establecido por el usuario, por ejemplo: “Teléfono de Pedro”. La clase y servicios del dispositivo es un conjunto de tres enteros que corresponden a los servicios que ofrece el dispositivo y el tipo de dispositivo que ha sido descubierto, por ejemplo: “Teléfono Celular”, “Computadora Portátil”, “Computadora de Escritorio” [2].Por último la información RSSI nos provee información del nivel de potencia de la señal que existe con el dispositivo descubierto.

Estos datos pueden ser almacenados para producir información útil que puede ser utilizada en distintas aplicaciones en diferentes campos, los cuales se pueden clasificar de la siguiente forma: las redes sociales, el rastreo y localización bajo techo, la medición de flujo de personas, la computación ubicua, entre otros. Dependiendo del caso, se almacenaran los datos que sean necesarios y luego serán interpretados de acuerdo a las necesidades.

[2] *https://www.bluetooth.org/Technical/AssignedNumbers/baseband.htm*

4. Metodología y Herramientas de Hardware y Software

El presente capítulo explica en detalle la metodología y las diversas herramientas de software y hardware que fueron empleadas con el objetivo de desarrollar la herramienta para el escaneo y la distribución de anuncios usando Bluetooth. Las herramientas de hardware están conformadas por todos aquellos dispositivos físicos que se utilizaron para el desarrollo de este trabajo. En cuanto al software, fueron utilizadas varias herramientas de apoyo para la programación de la aplicación.

4.1. Metodología

Las metodologías tradicionales de desarrollo de software se basan en una serie de pasos secuenciales, tales como la definición de los requerimientos, análisis y diseño, desarrollo de la aplicación, pruebas y despliegue. Estas metodologías obligan a definir y documentar un conjunto estable de requisitos en el inicio de un proyecto. Hay muchas metodologías diferentes de este tipo, este trabajo se apoyó principalmente en el modelo de cascada que es descrito a continuación.

El modelo de cascada fue propuesta por Winston Royce en 1970 **[1]**. Este enfoque destaca una progresión de fases estructuradas y bien definidas. Cada fase consiste en un conjunto de actividades y entregables que se deben cumplir antes de que la siguiente fase pueda comenzar. La primera fase trata de definir lo que el sistema va a hacer. La segunda fase determina el diseño de la aplicación. La tercera etapa es donde los desarrolladores comienzan a escribir el código. La cuarta fase consiste en las pruebas del sistema. Por último, la fase final se centra en las tareas de implementación tales como capacitación y documentación. Sin embargo, en la práctica de la ingeniería, la cascada utiliza el término como un nombre genérico para todos los modelos de ingeniería de software secuencial. La Figura 4.1 muestra un ciclo de vida de cascada tradicional.

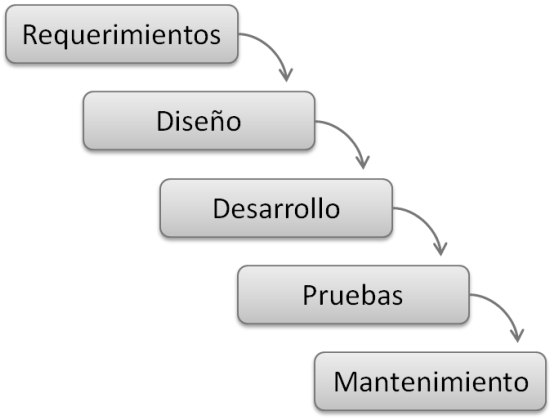

Figura 4.1 Modelo Cascada.

4.2. Herramientas de Hardware

Para la puesta en marcha de la herramienta para el escaneo y la distribución de anuncios usando Bluetooth fue necesaria la utilización de los siguientes componentes físicos:

Adaptador Bluetooth USB

El adaptador Bluetooth USB permite a una computadora de escritorio o laptop con puerto USB hacer uso de la tecnología Bluetooth. Para esta investigación se utilizaron varios adaptadores Bluetooth MSI 3X BToes 2.0, los cuales permitieron hacer uso de la tecnología Bluetooth para el escaneo de dispositivos y la distribución de anuncios a los dispositivos móviles.

Computadora Personal

Es un equipo computacional que en este caso sirvió para realizar las labores de escaneo y distribución de anuncios usando Bluetooth. Para tal fin, a este equipo se conecta uno de los adaptadores Bluetooth antes mencionado. También es necesario que este equipo tenga acceso al servidor por medio de la red y poder comunicarse con el componente que gestiona el funcionamiento de la herramienta.

Servidor

Es un equipo computacional que aloja el componente que gestiona la herramienta, la base de datos de la herramienta y la aplicación para la administración de la misma. Este equipo debe brindar acceso por medio de la red a los equipos que realicen las tareas de escaneo y distribución de anuncios usando Bluetooth.

4.3. Herramientas de Software

Para el desarrollo de la herramienta para el escaneo y la distribución de anuncios usando Bluetooth se hizo uso de varios componentes de software que facilitaron este trabajo. A continuación se describen estas herramientas:

Lenguaje Python

Es un lenguaje de programación interpretado que permite trabajar de una manera fácil y rápida. Se caracteriza por tener una sintaxis clara y fácil de leer, de utilizar un paradigma orientado a objetos, de ser modular, y de proveer manejo de excepciones y una gran cantidad de librerías para realizar virtualmente cualquier tarea **[21]**.

Puede ser ejecutado en una gran variedad de plataformas, tales como: Windows, Linux/Unix, Mac, y Amiga. También hay versiones que corren en .NET, la máquina virtual de Java y en los teléfonos celulares Nokia serie 60.

Es pertinente señalar que Python tiene una licencia de código abierto que permite su libre uso y distribución, incluso para uso comercial.

Biblioteca PyBluez

PyBluez es una librería para Python que permite a los desarrolladores hacer uso de la tecnología Bluetooth de una forma fácil y rápida. Ella permite trabajar tanto en la plataforma Windows como en la plataforma GNU/Linux **[20]**. Dentro del API de esta librería destacan las siguientes funciones:

- **advertise_service(sock, name, service_id, service_classes, profiles, provider, description):** Anunciar un servicio con el servidor local SDP.
- **discover_devices(duration, flush_cache, lookup_names):** Realiza el descubrimiento de dispositivos usando el primer recurso Bluetooth disponible. En la tabla 4.1 se describen cada uno de los parámetros que recibe.

Parámetro	Valor por defecto	Descripción
duration	8	Cantidad de tiempo en unidades de 1,28 segundos utilizados para realizar el descubrimiento de dispositivos.
flush_cache	*True*	Valor lógico que siendo *false* puede retornar dispositivos encontrados en descubrimientos anteriores.
lookup_names	*False*	Valor lógico que siendo *true* devolverá el nombre de los dispositivos encontrados.

Tabla 4.1 Parámetros función *discover_devices*.

- **find_service(name, uuid, address):** Busca los servicios SDP que coincidan con el criterio especificado. Si ningún criterio es especificado, retorna una lista de servicios de todos los dispositivos cercanos. Si más de un criterio es señalado, entonces el resultado debe concordar con todos los criterios. El resultado de la búsqueda será una lista de diccionarios con los objetos clave-valor que se describen en la tabla 4.2.

Clave	Valor
Host	La dirección Bluetooth de dispositivo que anuncia el servicio.
Name	El nombre del servicio que es anunciado.
Description	Una descripción del servicio que es anunciado.
Provider	El nombre de la persona u organización que provee el servicio.
Protocol	Puede ser RFCOMM, L2CAP o *none* si el protocolo no es especificado o UNKOWN si el protocolo es especificado pero no reconocido.
Port	El número PSM si el protocolo es L2CAP, el canal si el protocolo es RFCOMM o *None* si no es especificado.
Service-classes	Una lista de IDs de clase de servicio (UUID). Posiblemente vacio.
Profiles	Una lista de perfiles.
Service-id	El ID de servicio. *None* si no fue especificado.

Tabla 4.2 Lista de parámetros retornados por función *find_services*.

- **is_valid_address(address):** Retorna un valor lógico indicando si una dirección Bluetooth es válida o no.
- **is_valid_uuid(uuid):** Retorna un valor lógico indicando si un UUID es válido o no.
- **lookup_name(address, timeout):** Permite buscar el nombre de un dispositivo dada su dirección.
- **stop_advertising(sock):** Solicita al servidor SDP local que detenga el anuncio de un servicio asociado a un *socket* dado.

Biblioteca LightBlue

Es una librería multiplataforma que proporciona un API de Bluetooth para Python. Está disponible para MAC OS X, GNU/Linux y también en los teléfonos celulares Nokia serie 60. Provee un acceso simple a las operaciones de descubrimiento de dispositivos y servicios, creación de *sockets* RFCOMM y L2CAP, envío y recepción de archivos usando el protocolo OBEX, difusión de servicios RFCOMM y OBEX, y obtención de información del dispositivo local **[2]**. Dentro del API de esta librería destacan las siguientes funciones:

- **finddevices(getnames=True, length=10):** Permite realizar un descubrimiento de dispositivos y retorna una lista de los dispositivos encontrados en forma de tuplas con los siguientes valores: dirección, nombre y clase de dispositivo. Los parámetros que recibe esta función se describen en la tabla 4.3.

Parámetro	Valor por defecto	Descripción
getnames	*True*	Es del tipo lógico e indica al método que debe incluir el nombre de los dispositivos dentro de la lista retornada.
lenght	10	El número de segundos de duración utilizados para realizar la búsqueda de dispositivos.

Tabla 4.3 Parámetros función *finddevices*.

- **findservices(addr=None, name=None, servicetype=None):** Permite realizar un descubrimiento de servicios y retorna una lista con los servicios encontrados en forma de tuplas con los siguientes valores: dirección, puerto de servicio y nombre de servicio. Los parámetros que recibe esta función se describen en la tabla 4.4.

Parámetro	Valor por defecto	Descripción
addr	*None*	La dirección de un dispositivo, permite buscar los servicios de un dispositivo en especifico, si es *none* se buscará por los servicios de todos los dispositivos que se encuentren dentro del alcance del radio Bluetooth.
name	*None*	Permite especificar buscar por el nombre de un servicio en específico.
servicetype	*None*	Para buscar los servicios que coincidan con este parámetro, puede ser RFCOMM u OBEX.

Tabla 4.4 Parámetros función *findservices*.

- **finddevicename(address, usecache=True):** Permite conseguir el nombre de un dispositivo dada su dirección Bluetooth. Los parámetros que recibe esta función se

describen en la tabla 4.5.

Parámetro	Valor por defecto	Descripción
address	No aplica	La dirección del dispositivo que se requiere buscar el nombre.
usecache	*True*	Es del tipo lógico y si es *true* se utilizará la memoria cache local para conseguir el nombre del dispositivo si es posible de lo contrario el dispositivo remoto será contactado para solicitarle el nombre.

Tabla 4.5 Parámetros función *finddevicename*.

- **selectdevice():** Muestra una interfaz gráfica que permite al usuario final seleccionar un dispositivo de una lista de dispositivos descubiertos. Retorna el dispositivo seleccionado en forma de tupla de la siguiente forma (dirección, nombre, clase de dispositivo).
- **selectservice():** Muestra una interfaz gráfica que permite al usuario final seleccionar un servicio de una lista de dispositivos descubiertos y sus servicios. Retorna el servicio seleccionado en forma de tupla con los siguientes datos: dirección del dispositivo, puerto del servicio y nombre del servicio.
- **gethostaddr():** Retorna la dirección Bluetooth del dispositivo local.
- **gethostclass():** Retorna la clase del dispositivo del dispositivo local.
- **socket(proto=RFCOMM):** Retorna un nuevo objeto *socket*. Los parámetros que recibe esta función se describen en la tabla 4.6.

Parámetro	Valor por defecto	Descripción
proto	RFCOMM	El tipo de *socket* que va a ser creado, puede ser RFCOMM o L2CAP.

Tabla 4.6 Parámetros función *socket*.

- **advertise(name, sock, servicetype):** Permite anunciar un servicio con el nombre y el objeto *socket* dado. Los parámetros que recibe esta función se describen en la tabla 4.7.

Parámetro	Descripción
name	Nombre del servicio que va a ser anunciado.
sock	Objeto *socket* que va a manejar este servicio.
servicetype	El tipo de servicio que va a ser anunciado, puede ser RFCOMM u OBEX.

Tabla 4.7 Parámetros función *advertise*.

- **stopadvertise(sock):** Detiene el anuncio de un servicio dado un *socket*.
- **splitclass(classofdevice):** Divide la clase de un dispositivo, retorna una tupla de tres elemento con la clase mayor de servicio, clase mayor de dispositivo y clase menor del dispositivo. Los parámetros que recibe esta función se describen en la tabla 4.8.

Parámetro	Descripción
classofdevice	La clase del dispositivo.

Tabla 4.8 Parámetros función *splitclass*.

Modulo OBEX de LightBlue

Provee de una clase OBEX con funciones para el envío y recepción de archivos utilizando el protocolo OBEX **[2]**.

- **sendfile(address, channel, source):** Envía un archivo a un dispositivo remoto. Los parámetros que recibe esta función se describen en la tabla 4.9.

Parámetro	Descripción
address	Dirección Bluetooth del dispositivo remoto.
cannel	El canal RFCOMM del servicio remoto OBEX.
source	El archivo que va ser enviado.

Tabla 4.9 Parámetros función *sendfile*.

- **recvfile(sock, dest):** Recibir un archivo utilizando el protocolo OBEX. Los parámetros que recibe esta función se describen en la tabla 4.10.

Parámetro	Descripción
sock	El *socket* del servidor donde el archivo va a ser recibido.
dest	El archivo donde va a ser recibido el archivo.

Tabla 4.10 Parámetros función *recvfile*.

Manejador de Base de Datos MySQL

Es un sistema manejador de base de datos relacional, multihilo y multiusuario ampliamente utilizado en la actualidad en gran variedad de aplicaciones. Tiene un licenciamiento dual el cual se ofrece bajo GNU/GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar una licencia específica que les permita tal fin **[19]**.

Servidor HTTP Apache

Es un servidor HTTP de código abierto diseñado para funcionar en los sistemas operativos modernos como UNIX, Microsoft Windows, MacOS/X y Netware. El objetivo de este proyecto es proveer de un servidor seguro, eficiente y extensible que ofrezca servicios HTTP haciendo uso del estándar de este protocolo. Apache ha sido el servidor web más popular en Internet desde Abril de 1996 **[23]**.

Lenguaje PHP

Es un lenguaje interpretado diseñado originalmente para la creación de páginas web dinámicas. Gran parte de su sintaxis es tomado de otros lenguajes como C, Java, Perl y también posee algunas características propias del mismo lenguaje PHP. Es un lenguaje multiparadigma, es decir, se pueden utilizar distintos enfoques de programación como imperativo, orientado a objetos, procedural y/o reflexivo **[25]**.

Hoy en día se encuentra en su versión 5.4.x. Es de resaltar su gran cantidad de bibliotecas que facilitan el trabajo del programador. También posee una comunidad de usuarios bastante

grande que está dispuesta a colaborar en la resolución de cualquier duda relacionada al lenguaje.

Framework CodeIgniter

Es un poderoso framework para PHP, pensado para los programadores que necesitan una herramienta simple y elegante para crear aplicaciones web. Disminuye los tiempos de programación ya que simplifica la realización de muchas tareas a la hora de programar. Hace uso del patrón MVC y es bastante flexible y fácil de aprender **[6]**.

Extensión Imagick

Es una extensión para PHP con el fin de crear y modificar imágenes utilizando el API de ImageMagick. Este API es un conjunto de software para crear, editar y componer imágenes de mapas de bit. Permite leer, convertir y escribir imágenes en gran variedad de formatos (más de 100) incluidos DPX, EXR, JPEG, GIF, JPEG-2000, PDF, PhotoCD, PNG, Postscript, SVG y TIFF. Los derechos pertenecen a ImageMagick Studio LLC, una organización sin fines de lucro dedicada al desarrollo de soluciones de software de imágenes para su libre uso **[25]**.

Biblioteca JQuery

JQuery es una biblioteca JavaScript rápida y concisa que simplifica la manipulación del documento HTML, el manejo de eventos, animación y la interacción AJAX para un rápido desarrollo web **[24]**.

Complemento JQGrid

JQGrid es un complemento para JQuery que proporciona soluciones para la representación y manipulación de datos tabulados en la web, haciendo uso de la tecnología AJAX. Como se utiliza del lado del cliente puede ser integrado con cualquier tecnología del lado del servidor como PHP, ASP, JSP, entre otras **[26]**.

Biblioteca Open Flash Chart

Esta es una librería que permite generar gráficos estadísticos en objetos Flash utilizando JSON para crear la gráfica. Del lado del servidor soporta gran variedad de lenguajes, entre los cuales destaca PHP **[9]**.

5. Análisis y Diseño de la Aplicación

El presente capitulo describe el proceso de análisis que se llevó a cabo para el desarrollo de una herramienta para el escaneo y la distribución de anuncios usando Bluetooth dentro de la Facultad de Ciencias de la UCV, la cual se denomina BlueCiens. Se abarca el análisis de los requerimientos y el modelado de la aplicación. De este modo se consigue entender cuáles son los objetivos que se buscan al desarrollar esta herramienta y los procedimientos realizados para lograr dichos objetivos.

5.1. Análisis de los Requerimientos

Para desarrollar una herramienta que soporte el escaneo Bluetooth y la distribución de anuncios en la Facultad de Ciencias de la UCV se concibió un sistema con una arquitectura cliente/servidor, tal como se muestra en la Figura 5.1. Los clientes, llamados *Puntos de Acceso*, tendrán la tarea de recolectar datos y enviar anuncios a los distintos dispositivos Bluetooth. Por otro lado el servidor, denominado *Manager*, se encargará de coordinar la actuación de los *Puntos de Acceso* y registrar los datos que estos hayan escaneado. También se requiere de una aplicación que permita la administración de dicha herramienta, la cual será el contacto directo del usuario con el sistema.

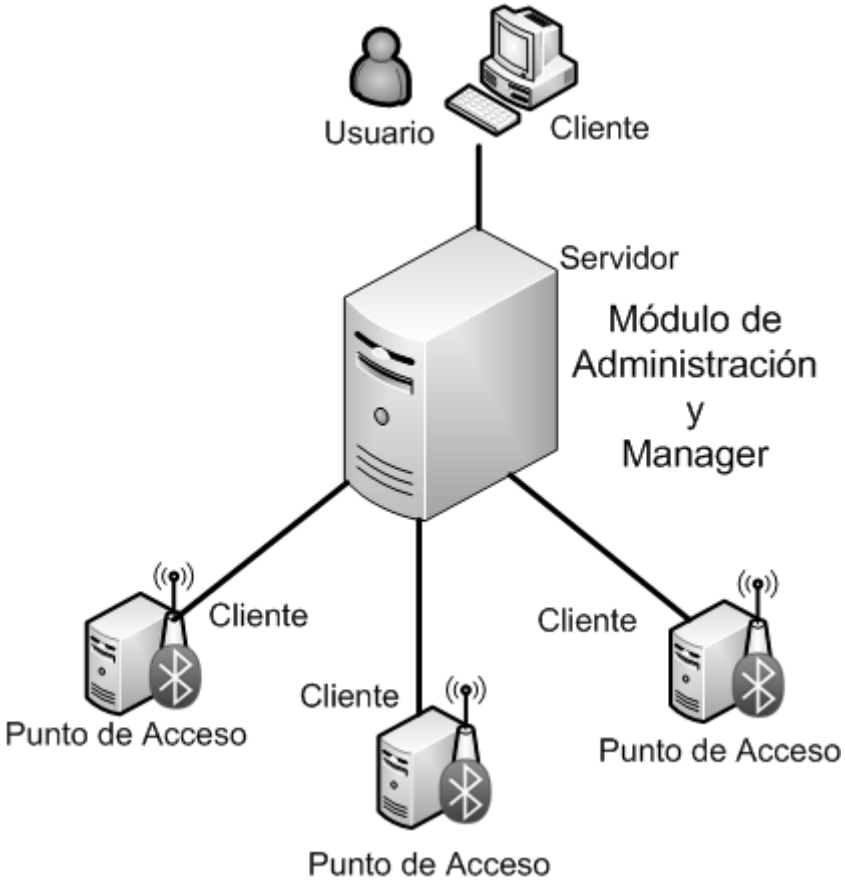

Figura 5.1 Arquitectura de herramienta BlueCiens.

5.1.1. Requerimientos Funcionales

Los requerimientos funcionales definen qué hace un sistema. Es decir, describen cómo reaccionará ese sistema ante una entrada en particular. A continuación se describen los requerimientos funcionales para la herramienta de escaneo y distribución de anuncios usando Bluetooth. Dichos requerimientos están separados en tres componentes, que interactúan para lograr los objetivos de la herramienta.

Cliente: Puntos de Acceso

Los *Puntos de Acceso* tienen como principal objetivo recolectar los datos de los dispositivos Bluetooth, asimismo, tienen la tarea de enviar los anuncios haciendo uso de dicha tecnología.

- **Recolectar datos de los dispositivos Bluetooth:** Cumpliendo con los parámetros que le indique el *Manager*, uno de los cuales puede ser el período de tiempo para realizar la recolección, se deben recolectar los datos de los dispositivos Bluetooth que se encuentren dentro del radio de alcance del *Punto de Acceso*. A medida que se recolectan dichos datos, estos deben ser enviados al *Manager* que se encargará de registrarlos.
- **Enviar anuncios a los dispositivos Bluetooth:** Siguiendo los parámetros indicados por el *Manager*, como la descripción del anuncio y el período de tiempo durante el cual se desea entregar, se deben enviar dichos anuncios a los distintos dispositivos utilizando la tecnología Bluetooth.

Servidor: Manager

El *Manager* tiene la tarea de coordinar la actuación de los *Puntos de Acceso* y registrar los datos recolectados por los mismos.

- **Coordinar *Puntos de Acceso* Bluetooth:** El servidor tiene la tarea de administrar las conexiones con los diferentes *Puntos de Acceso*, y debe informar a cada uno de estos cuándo tienen que realizar alguna recolección de datos Bluetooth, y de igual forma, cuándo tienen algún anuncio que deben distribuir a los distintos dispositivos Bluetooth que se encuentren dentro de su rango de alcance.
- **Registrar datos Bluetooth:** Otra tarea pertinente al servidor es el registro de los datos de los diferentes dispositivos Bluetooth que hayan sido recolectados por los *Puntos de Acceso*.

Aplicación de Administración

Es necesario tener una interfaz que permita la administración de las tareas de escaneo de dispositivos Bluetooth y la distribución de anuncios utilizando Bluetooth.

- **Administrar escaneos:** Se requiere una interfaz que permita al usuario crear y visualizar las tareas de escaneo, permitiendo definir el rango de tiempo durante el cual

se desea realizar un escaneo de dispositivos Bluetooth y cuáles *Puntos de Acceso* serán utilizados para tal fin.

- **Visualizar datos recolectados:** Se desea poder visualizar los datos de los dispositivos que han sido escaneados, también se quiere un mecanismo que permita visualizar gráficos estadísticos de estos datos.
- **Administrar distribución de anuncios:** Se requiere una interfaz que permita al usuario crear y visualizar los anuncios que se quieren enviar utilizando Bluetooth. Debe permitir definir el rango de tiempo durante el cual se desea enviar un anuncio a los distintos dispositivos Bluetooth y cuáles *Puntos de Acceso* serán utilizados para esta tarea.

5.1.2. Requerimientos no Funcionales

Los requerimientos no funcionales definen los atributos de un sistema cuando trabaja. Incluyen una descripción completa de los niveles requeridos por dicho sistema en cuanto a eficiencia, confiabilidad, seguridad, facilidad de mantenimiento, portabilidad, entre otros. A continuación se describen los requerimientos no funcionales para la herramienta de escaneo y distribución de anuncios usando Bluetooth.

Seguridad

- La interfaz administrativa debe estar protegida con un sistema de autenticación, para permitir que solo los usuarios registrados puedan ingresar y hacer uso del sistema.

Rendimiento

- La herramienta de distribución de anuncios no debe convertirse en un generador de "*spam*" que provoque el rechazo de las personas que reciban los anuncios.
- Tanto los *Puntos de Acceso* que se ejecutan en los clientes, así como el *Manager* que se ejecuta en el servidor, deben hacerlo en un segundo plano (en forma de demonio) y consumir la menor cantidad posible de recursos de procesamiento, memoria y ancho de banda.

Usabilidad

- Se debe conservar la simplicidad de la interfaz administrativa.
- La interfaz administrativa debe ser intuitiva y fácil de usar.
- Deben existir mecanismos de registro de actividades (*logs*) del *Manager* y los *Puntos de Acceso* que permitan analizar el comportamiento de la herramienta.

5.2. Modelado de la Aplicación

Una vez definidos los requerimientos de la aplicación se procede al modelado de dichas especificaciones. De este modo se define formalmente el comportamiento del sistema.

El modelado de una aplicación se realiza para comprender el sistema que se va a desarrollar. Para modelar, analizar y diseñar sistemas orientados a objetos se utiliza un

conjunto de herramientas llamadas UML (*Unified Modeling Language*, Lenguaje Unificado de Modelado) **[10]**.

Entre las herramientas que proporciona UML se encuentran: el diagrama de casos de usos, el diagrama de clases, el diagrama de estados, el diagrama de secuencias, entre otros. Para el modelado de la herramienta de escaneo y distribución de anuncios usando Bluetooth se emplearán los diagramas de casos de usos, el diagrama de secuencia, además del diagrama entidad-relación que aunque no pertenece a UML permite modelar la base de datos de la aplicación.

5.2.1. Diagrama de Casos de Uso

Los casos de uso forman parte del análisis del sistema, de forma que al ser parte del análisis ayudan a describir qué es lo que el sistema debe hacer. Los casos de uso definen qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

Casos de Uso, Nivel 0

En este caso de uso se modela el sistema a nivel general, con sus respectivos actores. Tal como se muestra en la Figura 5.2.

Usuario Administrador 1: Actor que representa al administrador del módulo *Manager*, debe ser un usuario experto que tenga dominio en el funcionamiento de la herramienta.

Usuario Administrador 2: Actor que representa al administrador del módulo Punto *de Acceso*, debe ser un usuario experto que tenga dominio en el funcionamiento de la herramienta.

Usuario Administrador 3: Actor que representa al usuario administrador del módulo de administración, no debe ser necesariamente un usuario experto que tenga dominio en el funcionamiento de la herramienta.

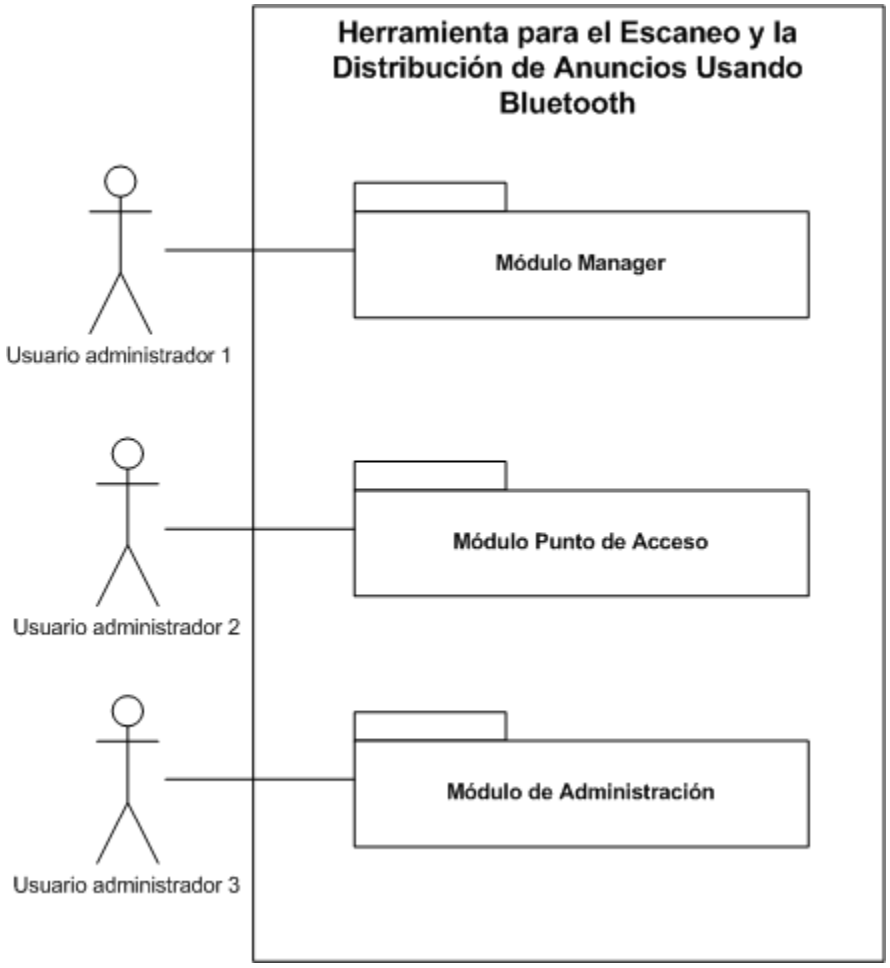

Figura 5.2 Casos de uso, nivel 0.

Casos de Uso, Nivel 1

En este nivel el caso de uso refleja en términos generales la interacción que tiene el usuario con el sistema tal como se muestra en las Figuras que van de la 5.3 a la 5.5. Se puede observar cómo el usuario puede interactuar con todas las opciones del sistema. La descripción de los casos de uso del nivel 1 se muestra en las Tablas que van de la 5.1 a la 5.11.

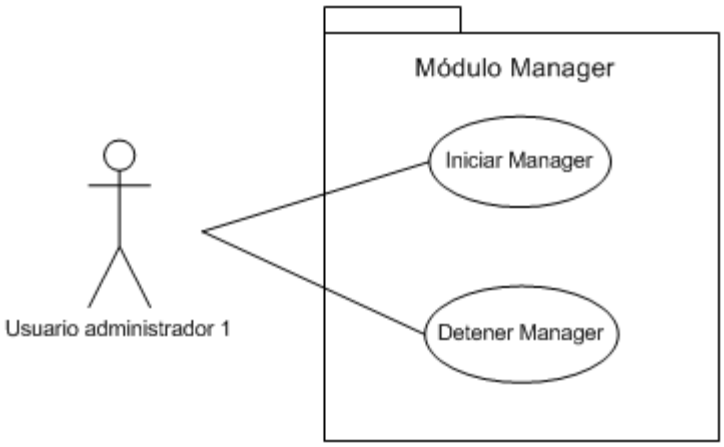

Figura 5.3 Casos de uso, nivel 1, módulo *Manager*.

Caso de uso	Iniciar *Manager.*
Actor	Usuario administrador 1.
Descripción	Da inicio al demonio *Manager* en el servidor, el cual atenderá a todos los *Puntos de Acceso* que se conecten con este.
Flujo Básico	a) El usuario da la orden para iniciar el demonio *Manager*. b) Se crea el *socket* que se mantendrá a la escucha de cualquier petición proveniente de los *Puntos de Acceso*.
Pre condiciones	Ninguna.
Post condiciones	Se pueden establecer las conexiones con los *Puntos de Acceso*.

Tabla 5.1 Iniciar *Manager*.

Caso de uso	Detener *Manager.*
Actor	Usuario administrador 1.
Descripción	Detiene el demonio *Manager* en el servidor.
Flujo Básico	a) El usuario da la orden para detener el *Manager*. b) Termina cualquier conexión existente con los *Puntos de Acceso*. c) Detiene el demonio. d) Se detiene el demonio *Manager*.
Pre condiciones	Debe estar en ejecución el demonio del *Manager*.
Post condiciones	Se termina la ejecución del demonio, ya no se podrán recibir conexiones de los *Puntos de Acceso*.

Tabla 5.2 Detener *Manager*.

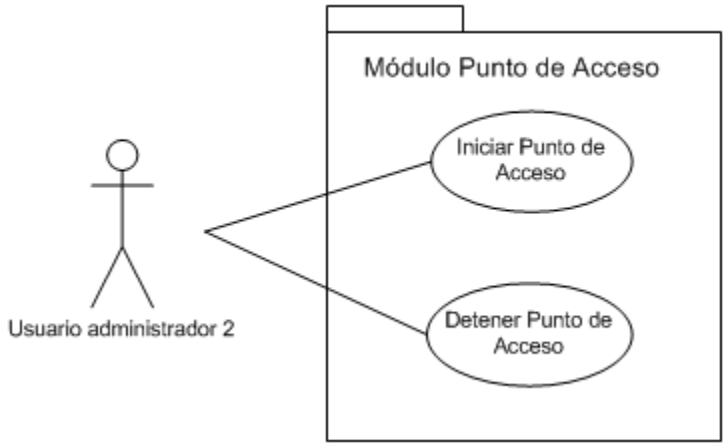

Figura 5.4 Casos de uso, nivel 1, módulo *Punto de Acceso*.

Caso de uso	Iniciar *Punto de Acceso.*
Actor	Usuario administrador 2.
Descripción	Da inicio al demonio del *Punto de Acceso* en el cliente, de este modo se podrán recolectar los datos de los dispositivos Bluetooth y entregar los anuncios utilizando dicha tecnología según los parámetros proporcionados por el servidor *Manager*.
Flujo Básico	a) El usuario da la orden para iniciar el demonio del *Punto de Acceso*. b) Se establece la conexión con el servidor *Manager*. c) Se envían los datos al servidor del *Punto de Acceso* (Dirección IP, Nombre y Descripción). d) El demonio se mantiene a la espera de cualquier tarea de escaneo o de distribución de anuncios que deba realizar. e) Si se recibe alguna tarea, se realiza y se envían los datos obtenidos al servidor *Manager*. f) El demonio se mantiene activo realizando los pasos (d) y (e).
Pre condiciones	El usuario debe haber configurado el *Punto de Acceso*. Debe estar activo y a la espera de peticiones el servidor *Manager*.
Post condiciones	El *Punto de Acceso* queda activo y listo para realizar tareas de escaneo y distribución de anuncios utilizando la tecnología Bluetooth.

Tabla 5.3 Iniciar *Punto de Acceso.*

Caso de uso	Detener *Punto de Acceso.*
Actor	Usuario administrador 2.
Descripción	Detiene el demonio del *Punto de Acceso* en el cliente.
Flujo Básico	a) El usuario da la orden para detener el *Punto de Acceso*. b) Se termina cualquier tarea que se esté realizando de escaneo y de envío de anuncios. c) Se detiene el demonio del *Punto de Acceso*.
Pre condiciones	El demonio del *Punto de Acceso* debe estar activo.
Post condiciones	Se termina la ejecución del *Punto de Acceso*.

Tabla 5.4 Detener *Punto de Acceso.*

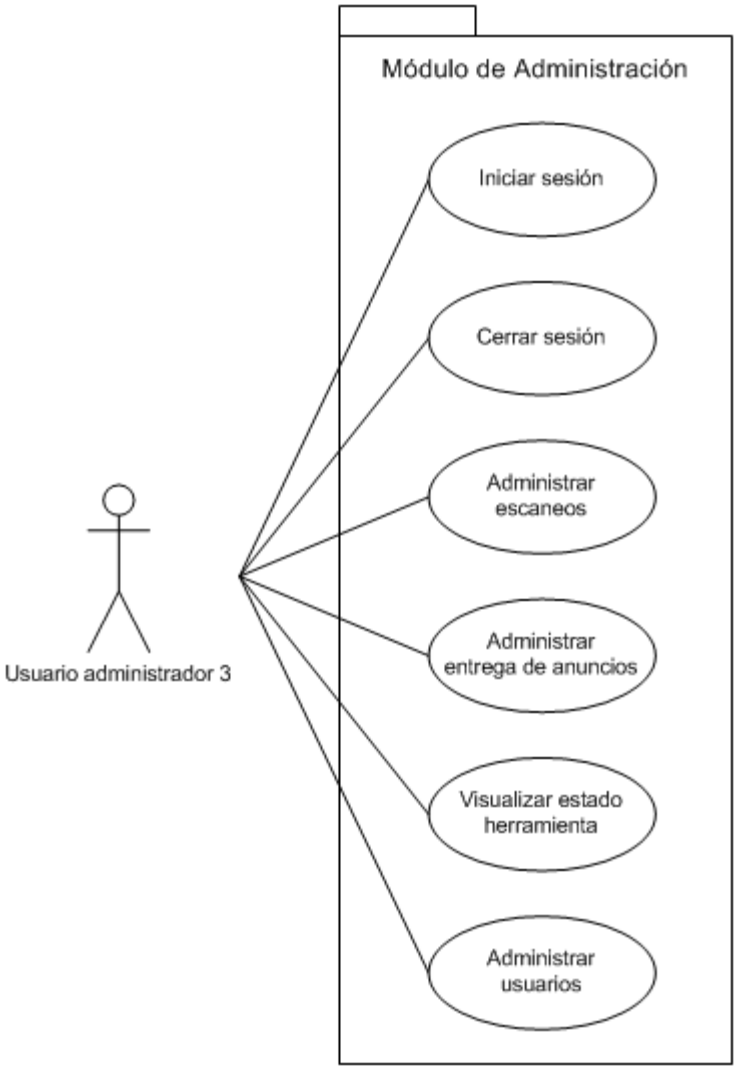

Figura 5.5 Casos de uso, nivel 1, módulo de administración.

Caso de uso	Iniciar sesión.
Actor	Usuario administrador 3.
Descripción	Permite autenticar y autorizar al usuario para que haga uso de la herramienta de administración de la aplicación.
Flujo Básico	a) Se solicita el usuario y la clave para ingresar al sistema. b) El sistema autentica y autoriza el ingreso al sistema. c) Se muestra un mensaje al usuario indicando el resultado proceso.
Pre condiciones	Ninguna.
Post condiciones	Se crea una sesión de trabajo para el usuario, donde se tiene acceso al menú de administración de la aplicación.

Tabla 5.5 Iniciar sesión.

Caso de uso	Cerrar sesión.
Actor	Usuario administrador 3.
Descripción	Permite concluir la sesión de trabajo del usuario.
Flujo Básico	a) El usuario da la orden para cerrar la sesión. b) Se termina la sesión para ese usuario en la aplicación.
Pre condiciones	El usuario debe tener una sesión de trabajo activa.
Post condiciones	Se termina la sesión de trabajo del usuario, por tanto no se podrá hacer uso de las opciones de administración de la aplicación.

Tabla 5.6 Cerrar sesión.

Caso de uso	Administrar escaneos.
Actor	Usuario administrador 3.
Descripción	Permite al usuario administrar todas las tareas relacionadas al escaneo de dispositivos Bluetooth. Permite crear tareas de escaneo y ver los resultados de dichos escaneos.
Flujo Básico	a) Se selecciona la opción que se requiera administrar relacionada al escaneo de dispositivos Bluetooth. b) Se hace uso de la opción seleccionada.
Pre condiciones	El usuario debe tener una sesión de trabajo activa.
Post condiciones	Ninguna.

Tabla 5.7 Administrar escaneos.

Caso de uso	Administrar entrega de anuncios.
Actor	Usuario administrador 3.
Descripción	Permite al usuario administrar todas las tareas relacionadas a la distribución de anuncios utilizando Bluetooth. Permite crear anuncios para ser distribuidos por los *Puntos de Acceso* usando Bluetooth.
Flujo Básico	a) Se selecciona la opción que se requiera administrar relacionada a la distribución de anuncios utilizando Bluetooth. b) Se hace uso de la opción seleccionada.
Pre condiciones	El usuario debe tener una sesión de trabajo activa.
Post condiciones	Ninguna.

Tabla 5.8 Administrar entrega de anuncios.

Caso de uso	Administrar usuarios.
Actor	Usuario administrador 3.
Descripción	Permite al usuario administrar los usuarios que hacen uso de la aplicación de administración. Permite crear, modificar y eliminar los usuarios del módulo de administración de la herramienta.
Flujo Básico	a) Se selecciona la opción que se requiera administrar relacionada a la administración de usuarios. b) Se hace uso de la opción seleccionada.
Pre condiciones	El usuario debe tener una sesión de trabajo activa.
Post condiciones	Ninguna.

Tabla 5.9 Administrar usuarios.

Caso de uso	Visualizar estado de la herramienta.
Actor	Usuario administrador 3.
Descripción	Permite al usuario ver el estado de los demonios *Manager* y *Puntos de Acceso*.
Flujo Básico	a) El usuario selecciona en el menú cual entidad desea conocer el estado. b) El sistema arroja cual es el estado de dicha entidad, indica si el demonio *Manager* esta activo o no y si los *Puntos de Acceso* están conectados o no.
Pre condiciones	El usuario debe tener una sesión de trabajo activa.
Post condiciones	Ninguna.

Tabla 5.10 Visualizar estado de la herramienta.

5.2.2. Diagrama de Secuencia

Un diagrama de secuencia muestra una interacción que se representa como la secuencia de mensajes entre las instancias de clases, componentes, subsistemas o actores. El tiempo fluye hacia abajo en el diagrama y muestra el flujo de control de un participante a otro **[16]**.

Cada diagrama de secuencia va ligado a un caso de uso. En la Figura 5.6 se muestra como el usuario administrador da inicio al demonio *Manager*. Se puede apreciar cómo después de dar inicio a la herramienta, la instancia de la clase *NetworkManager* permanece activa. Este objeto es el encargado de procesar las solicitudes provenientes de los múltiples *Puntos de Acceso*.

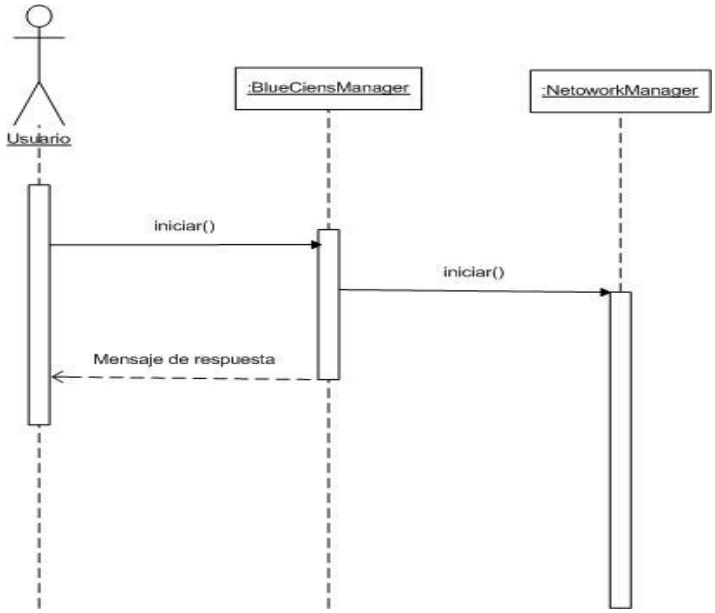

Figura 5.6 Diagrama de secuencia, iniciar *Manager*.

En la Figura 5.7 se puede apreciar como el usuario interactúa con la herramienta *Manager* para detenerla. Se observa como la instancia de la clase *NetworkManager* es detenida y destruida.

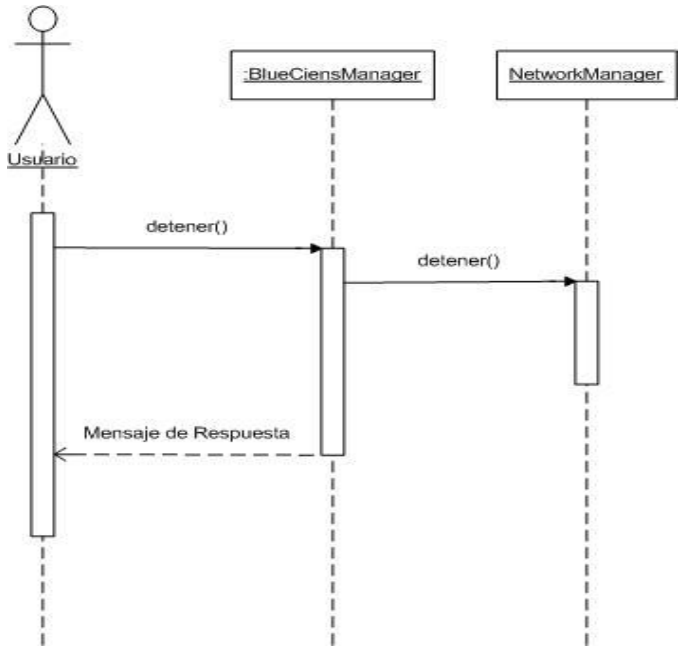

Figura 5.7 Diagrama de secuencia, detener *Manager*.

En la Figura 5.8 se puede observar cómo el usuario interactúa con la herramienta del *Punto de Acceso* para darle inicio. Se puede apreciar cómo la instancia de la clase *NetworkAP* queda activa, la cual se encargará de comunicarse con el *Manager*.

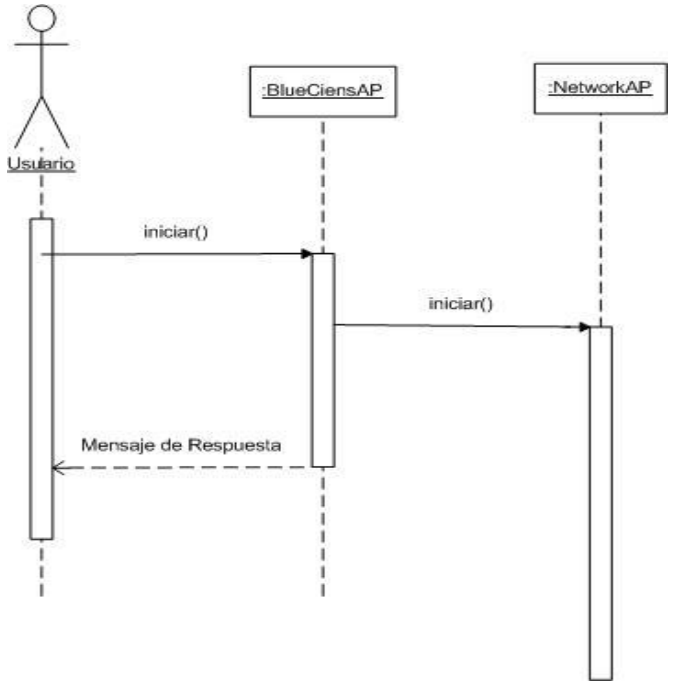

Figura 5.8 Diagrama de secuencia, iniciar *Punto de Acceso*.

En la Figura 5.9 se muestra cómo el usuario interactúa con la herramienta en el *Punto de Acceso* para detenerla. Se puede observar cómo la instancia de la clase *NetworkAP* es detenida y destruida.

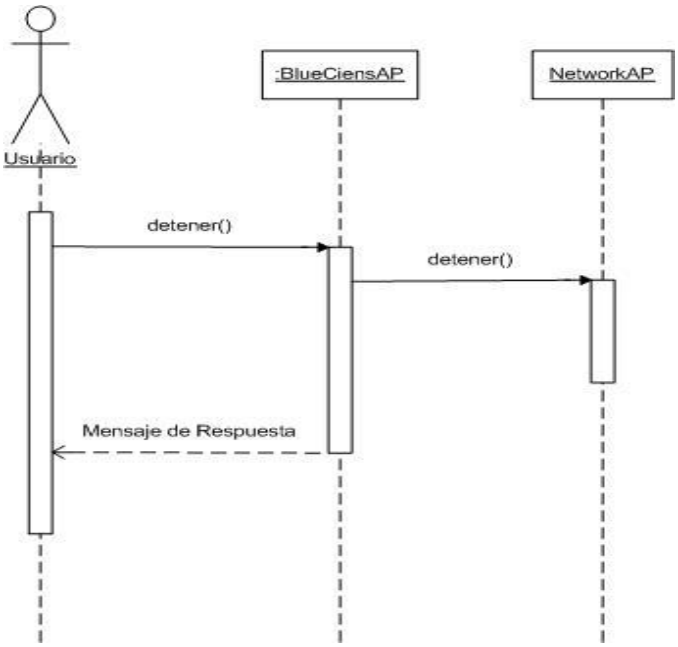

Figura 5.9 Diagrama de secuencia, detener *Punto de Acceso*.

En la Figura 5.10 se muestra cómo el usuario interactúa con el componente de administración de la herramienta. Se observa como el usuario hace uso de las distintas

opciones que se ofrecen, tales como: el escaneo, la distribución de anuncios y la visualización del estado de los *Puntos de Acceso* y del *Manager*.

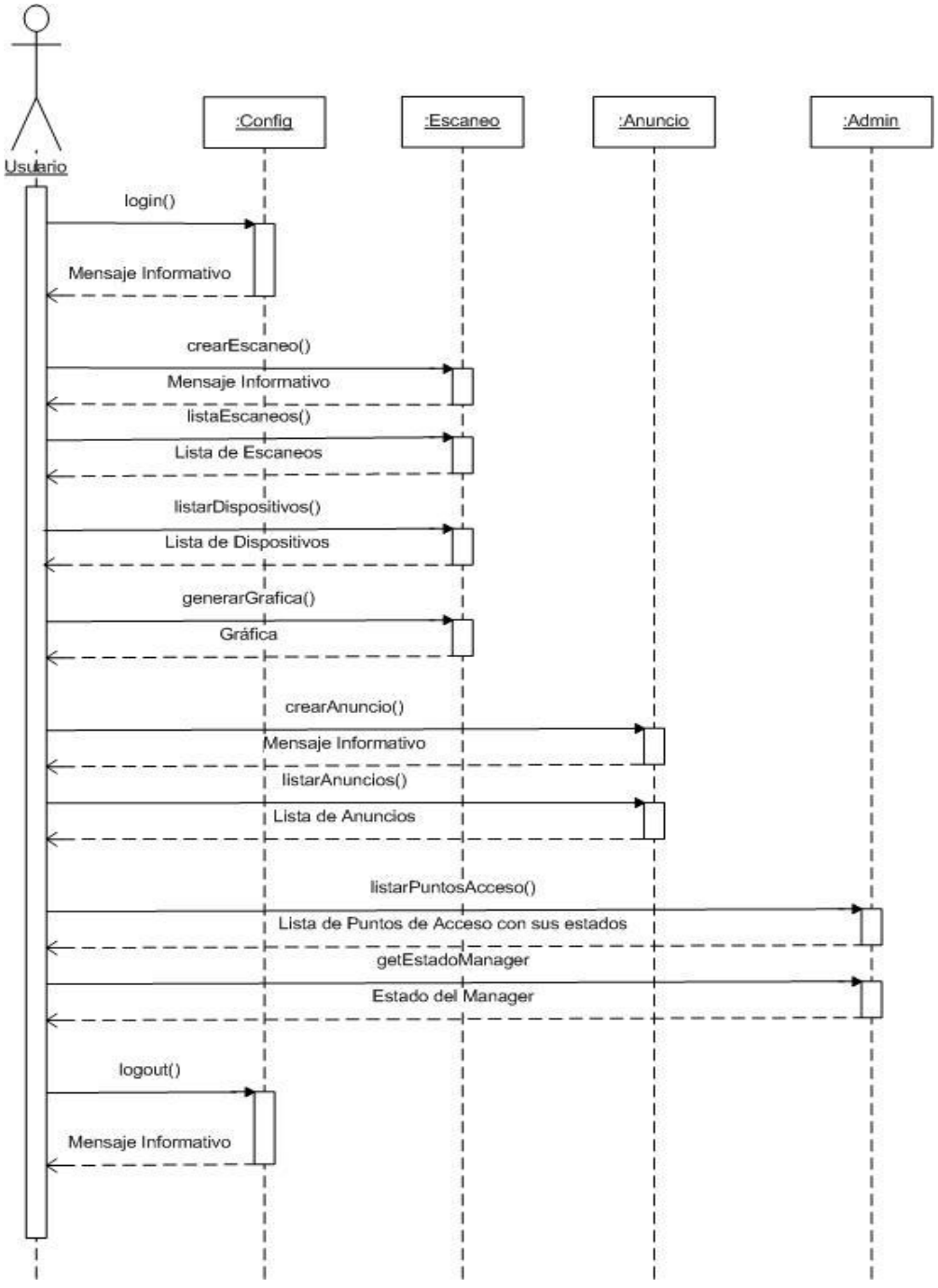

Figura 5.10 Diagrama de secuencia, administración de la herramienta.

5.2.3. Modelo Entidad-Relación

El modelo Entidad-Relación (E-R) es uno de los varios modelos conceptuales existentes para el diseño de bases de datos. El propósito de este modelo es simplificar el diseño de bases

de datos a partir de descripciones textuales de los requerimientos. Los elementos esenciales del modelo son las entidades, los atributos y las relaciones entre las entidades **[10]**.

En la Figura 5.11 se ilustra el modelo entidad-relación de la herramienta para el escaneo y la distribución de anuncios usando Bluetooth. Se utiliza el prefijo “bc_” para identificar las tablas que pertenecen a la herramienta BlueCiens.

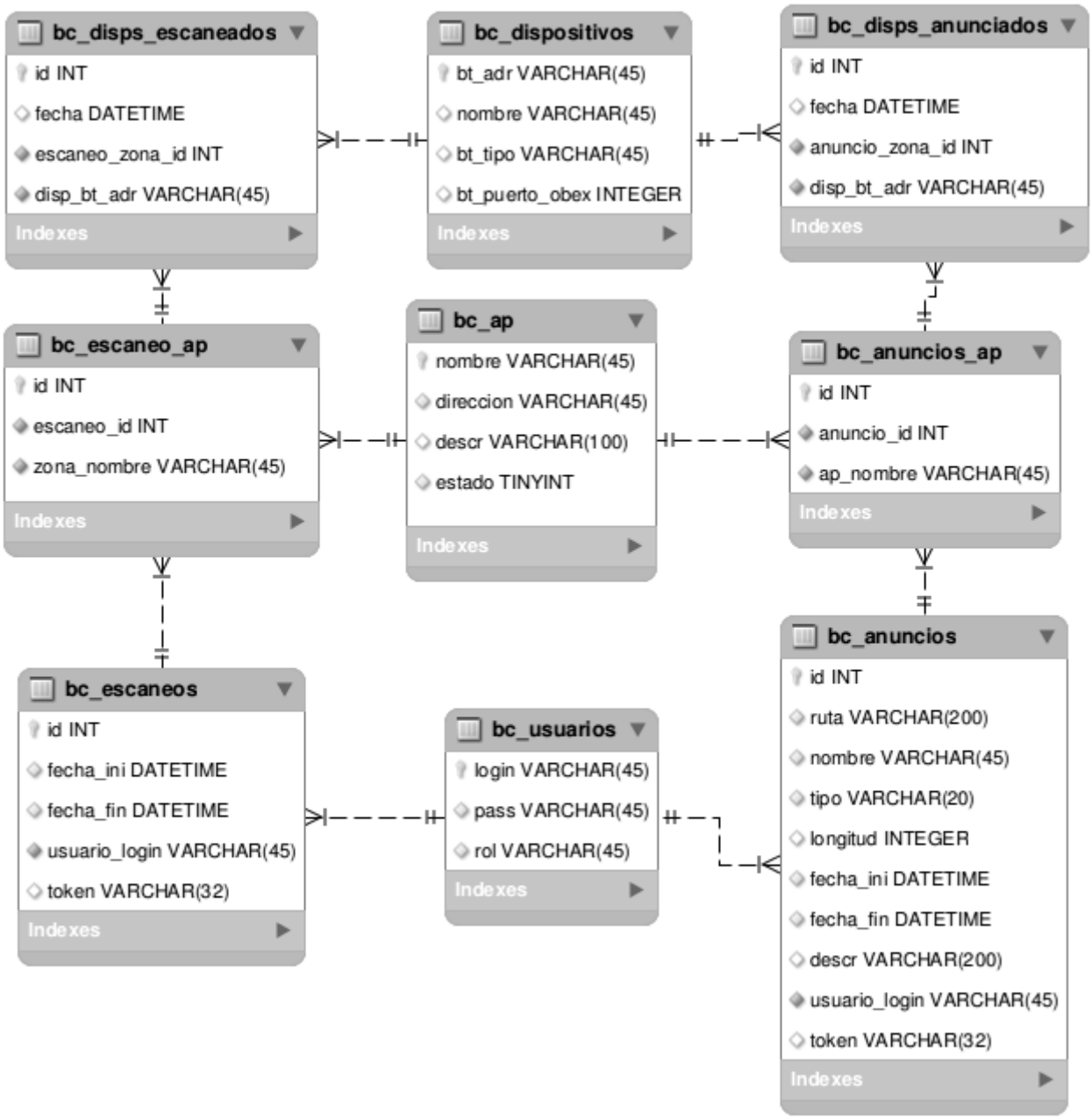

Figura 5.11 Modelo Entidad-Relación.

6. Implementación de la Aplicación

La implementación de la herramienta se basa en un modelo Cliente/Servidor. Los clientes, representados por los *Puntos de Acceso*, son los encargados de realizar las tareas de escaneo y distribución de anuncios utilizando la tecnología Bluetooth. Por otra parte el servidor, representado por la herramienta *Manager*, tiene como principal objetivo coordinar el trabajo que realizan los *Puntos de Acceso* y registrar los datos que estos generen. Del mismo modo, en el servidor también se encuentra la herramienta de administración, cuyo principal propósito es proveer de una interfaz sencilla que permita al usuario interactuar con la herramienta.

6.1. Arquitectura de la Herramienta de Escaneo y Distribución de Anuncios Usando Bluetooth

Seguidamente se presenta la topología utilizada para el desarrollo de esta herramienta. Se definen los componentes de hardware y software, como también las conexiones que unen a los mismos y permiten la comunicación entre ellos. De este modo se consigue comprender con más detalle el funcionamiento del sistema de una forma global.

En la Figura 6.1, se muestra el servidor como componente central, el cual cumple varias funciones. Una es alojar la herramienta *Manager* que controla los múltiples *Puntos de Acceso*, aquí también se encuentra la herramienta de administración que permite al usuario administrador interactuar con la herramienta. Asimismo, en este componente se ubica la base de datos donde se registra la información que es generada por los distintos *Puntos de Acceso* Bluetooth y los datos que permiten el funcionamiento de la herramienta, tales como usuarios, *Puntos de Acceso*, tareas de escaneo y anuncios.

Distribuidos en la periferia del servidor se ubican los *Puntos de Acceso* que se encargan de capturar los datos de los dispositivos Bluetooth que se encuentran dentro de su radio de alcance, así como la distribución de anuncios utilizando Bluetooth. Todas estas tareas son realizadas bajo las directrices de la herramienta *Manager*.

Desde cualquier computador personal provisto con un navegador web, el usuario administrador puede acceder a la herramienta de administración que está alojada en el servidor. Se utiliza el protocolo HTTP para comunicar el navegador web del usuario con el servidor. De este modo se simplifica el uso de la herramienta ya que los navegadores son aplicaciones estándares dentro de cualquier sistema operativo de la actualidad.

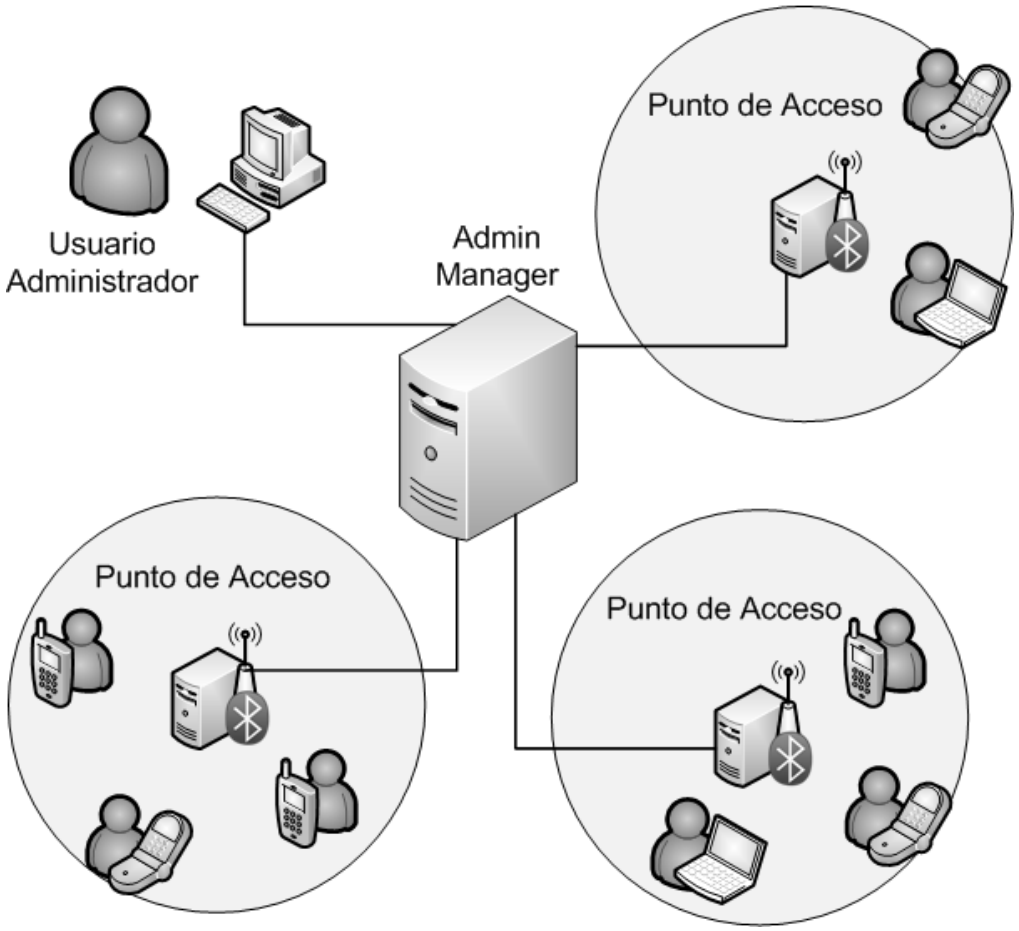

Figura 6.1 Arquitectura de la herramienta de escaneo y distribución de anuncios usando Bluetooth.

6.2. Implementación de las Funcionalidades

Como ya se explicó la herramienta BlueCiens se divide en tres componentes básicos, el *Manager* y el componente de administración, los cuales deben correr de forma conjunta sobre un servidor con plataforma GNU/Linux, y los *Puntos de Acceso* que corren en computadoras personales con una plataforma GNU/Linux.

Para entender de forma más precisa el funcionamiento de la herramienta BlueCiens en esta sección se explica en detalle la implementación de cada una de las funcionalidades de la herramienta.

Iniciar el Manager

El *Manager* tiene un carácter fundamental para el funcionamiento de los demás componentes de la herramienta, por lo tanto, es necesaria su puesta en marcha antes de iniciar cualquier *Punto de Acceso*. Este componente fue concebido como una aplicación que se ejecuta en segundo plano, también conocido como demonio. Esta decisión obedece a que el mismo debe permanecer en permanente escucha de los múltiples *Puntos de Acceso* de la

herramienta y atender sus solicitudes sin afectar el trabajo de los otros procesos que se estén ejecutando en el servidor.

Antes de dar inicio a este componente hay que configurar diferentes parámetros de funcionamiento, tales como la dirección IP y el puerto utilizado para escuchar las solicitudes provenientes de los *Puntos de Acceso*. Para llevar a cabo esta tarea existe un archivo de configuración ubicado en el directorio /etc/blueciensManager/, el cual utiliza una sintaxis sencilla y fácil de entender.

Una vez establecidos todos los parámetros como aparecen en la Figura 6.2 se puede dar inicio al demonio *Manager*, ejecutando el componente con el parámetro "*start*" tal como aparece en la Figura 6.3. Una vez iniciado será posible conectar cualquier *Punto de Acceso* con el servidor y de esta forma dar respuesta a sus solicitudes, las cuales se explican en detalle más adelante.

```
# [MANAGER_INFO]
# server_host:localhost
# server_port:9999
```

Figura 6.2 Configuración del *Manager*.

```
SERVER:~# python blueciensManager.py start
El Manager se ha iniciado
SERVER:~#
```

Figura 6.3 Iniciar *Manager*.

Detener el Manager

Para detener la ejecución del *Manager* simplemente se ejecuta la herramienta utilizando el parámetro "*stop*" tal como aparece en la Figura 6.4. Este comando interrumpe cualquier conexión existente con los *Puntos de Acceso*, y del mismo modo detiene la ejecución del demonio *Manager*.

```
SERVER:~# python blueciensManager.py stop
El Manager se ha detenido
SERVER:~#
```

Figura 6.4 Detener *Manager*.

El componente *Manager* también acepta otros dos parámetros, uno es el parámetro "*status*" que permite conocer el estado en que se encuentra el demonio (encendido o apagado), y de igual forma puede recibir el parámetro "*foreground*" que permite ejecutar la herramienta en primer plano y poder observar su comportamiento.

Iniciar un Punto de Acceso

El componente que se ejecuta en los *Puntos de Acceso* está concebido para ejecutarse en computadoras personales que pueden desempeñar otras tareas simultáneamente, unido a esto, es necesario que este componente se mantenga en contacto permanente con el servidor para permanecer informado de las tareas que se le asignen, por tales motivos, se decidió de igual forma implementar este componente como una aplicación del tipo demonio.

Antes de dar inicio a un *Punto de Acceso* se deben cumplir ciertas condiciones, primero es necesario que el servidor *Manager* este encendido y listo para atender cualquier solicitud. El siguiente paso es editar el archivo de configuración que se ubica en el directorio /etc/blueciensAP/ donde se especifican los siguientes parámetros:

- Nombre del *Punto de Acceso*, el cual debe ser único, ya que identifica unívocamente al *Punto de Acceso* dentro del sistema global.
- Descripción del *Punto de Acceso*, la cual brinda mayores detalles, tal como la ubicación del mismo o las características del equipo donde se encuentra.
- Dirección IP y puerto por el cual escucha el servidor *Manager*, para poder conectarse con el mismo.

Una vez configurados estos parámetros como aparecen en la Figura 6.5 se puede dar inicio al *Punto de Acceso* ejecutando el componente con el parámetro "*start*" tal como aparece en la Figura 6.6. De este modo se inicia el demonio, el cual establece una conexión con el servidor y da inicio a un ciclo permanente, donde cada cierto intervalo de tiempo se pregunta al servidor si hay tareas que el *Punto de Acceso* deba realizar, ya sea para escanear dispositivos o para entregar anuncios. El *Manager* consulta la base de datos y le envía una respuesta al *Punto de Acceso*. Si una tarea es suministrada, este la debe ejecutar y enviar los resultados al *Manager* que se encargará de registrar los mismos en la base de datos.

```
# [MANAGER_INFO]
# server_host:localhost
# server_port:9999
# [AP_INFO]
# ap_name:local1
# ap_descr:Servidor local
```

Figura 6.5 Configuración del *Punto de Acceso*.

```
CLIENT:~# python blueciensAP.py start
El Punto de Acceso se ha iniciado
CLIENT:~#
```

Figura 6.6 Iniciar *Punto de Acceso*.

La conexión con el servidor *Manager* se hace utilizando *sockets,* los cuales una vez establecidos sirven para intercambiar mensajes utilizando el lenguaje de marcado JSON. Los mensajes que puede enviar el *Punto de Acceso* al servidor son resumidos en la Tabla 6.1.

Solicitudes		Respuestas		Descripción
Mensaje	**Parámetros**	**Mensaje**	**Parámetros**	
“ap_data”	Dirección, descripción y nombre.	“rap_data”	Respuesta indicando recepción de datos.	Permite informar al *Manager* sobre los datos del *Punto de Acceso*.
“hello_scan”	No tiene.	“rhello_scan_data”	Id, fecha de inicio y fecha de finalización.	Pregunta al *Manager* si existe alguna tarea de escaneo pendiente para ese *Punto de Acceso*, de ser así se envían los datos necesarios al *Punto de Acceso* para que inicie el escaneo de dispositivos Bluetooth.
		“rhello_scan”	Respuesta indicando que no hay tareas de escaneo.	
“scan_data”	Fecha y dispositivos.	“rscan_data”	Respuesta indicando recepción de datos.	Permite enviar los datos de los dispositivos capturados por el *Punto de Acceso* al servidor.
“hello_anun”	No tiene.	“rhello_anun_data”	Id de los anuncios.	Pregunta al *Manager* si existe algún anuncio que deba entregar el *Punto de Acceso*. De ser así se envía una lista con los identificadores de dichos anuncios.
		“rhello_anun”	Respuesta indicando que no hay anuncios que deba entregar este *Punto de Acceso*.	
“get_anuncio”	Identificador del anuncio.	“anuncio”	Id, nombre, tipo de archivo, longitud de archivo, ruta, fecha de inicio, fecha de finalización y descripción.	Permite solicitar al servidor los datos de un anuncio dado su identificador, una vez recibidos los datos se envía el anuncio para ser almacenado temporalmente en el *Punto de Acceso*.
“disp_anunciados”	Lista de dispositivos con los anuncios entregados a dichos dispositivo.	“rdisp_anunciados”	Respuesta indicando recepción de datos.	Permite informar al servidor de los anuncios que han sido entregados a los dispositivos Bluetooth, con el propósito de evitar enviar un mismo anuncio más de una vez

				a un mismo dispositivo.
“get_anuncios _disp”	Lista con dispositivo.	“rget_anuncios_disp”	Lista de dispositivos con los anuncios entregados a dichos dispositivo.	Permite preguntar al servidor que anuncios han sido entregados a un dispositivo. También, con el propósito de evitar enviar un mismo anuncio más de una vez a un mismo dispositivo.

Tabla 6.1 Mensajes del *Punto de Acceso.*

Detener un Punto de Acceso

Para detener el *Punto de Acceso* se ejecuta el componente utilizando el parámetro “stop” tal como aparece en la Figura 6.7, el cual interrumpe la conexión con el servidor *Manager* y detiene la ejecución del demonio. Del mismo modo que sucede con el *Manager* también se tienen los parámetros “*status*” y “*foreground*” para conocer el estado del demonio y para poder ejecutar la herramienta en primer plano respectivamente.

```
CLIENT:~# python blueciensAP.py stop
El Punto de Acceso se ha detenido
CLIENT:~#
```

Figura 6.7 Detener *Punto de Acceso*.

Revisar los Registros (logs)

Tanto los componentes *Manager* como *Puntos de Acceso* llevan el registro de sus actividades en archivos ubicados en el directorio de *logs* del sistema operativo /var/log/. Para revisar estos registros simplemente se pueden abrir utilizando cualquier editor de textos y de tal forma se puede conocer con más detalle el comportamiento de dichas herramientas.

Administrar la Herramienta

Para administrar la herramienta se hace uso de un componente web, desde el cual se pueden administrar las distintas funcionalidades que ofrece la herramienta, tales como realizar el escaneo de dispositivos Bluetooth, visualizar los resultados generados por dichos escaneos y configurar la distribución de anuncios utilizando la tecnología Bluetooth. Para entender con más detalle cómo funciona este componente a continuación se describe la implementación de sus funcionalidades.

Control de Acceso al Componente de Administración

Para evitar el acceso de usuarios no autorizados a la herramienta, el primer paso para hacer uso del componente administrativo es autenticarse tal como se muestra en la Figura 6.8, se solicita un nombre de usuario y una contraseña para ingresar. Estos datos serán

comprobados y de ser satisfactorios se permitirá el acceso de este usuario al componente de administración.

Figura 6.8 Control de acceso de la herramienta.

Escaneo de Dispositivos Bluetooth

Dentro de este componente se puede definir el escaneo de dispositivos, visualizar los datos de los dispositivos capturados y generar gráficos estadísticos que proporcionen mayor información de los escaneos llevados a cabo. Para uso de estas funcionalidades se utiliza el menú situado en la parte izquierda de la interfaz de usuario en la sección “Escaneos”, tal como se muestra en la Figura 6.9.

Nueva Tarea de Escaneo

Esta opción permite generar una nueva tarea de escaneo, el usuario debe completar el formulario que se despliega para tal fin, como se muestra en la Figura 6.9. Se solicitan datos como la fecha de inicio, la fecha de finalización y los *Puntos de Acceso* destinados para realizar el escaneo. Una vez generado, esta tarea se registra en la base de datos para que eventualmente el *Manager* coordine su ejecución.

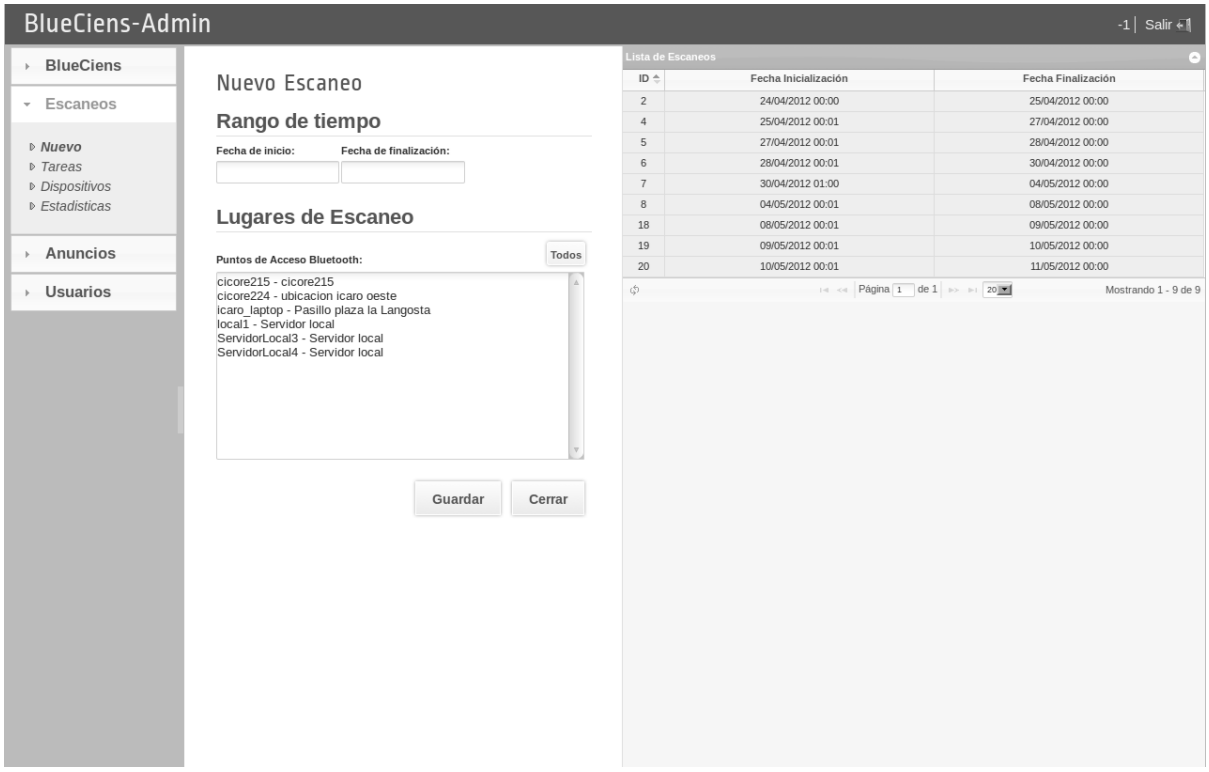

Figura 6.9 Nueva tarea de escaneo.

Lista de Tareas de Escaneo

Si el usuario desea ver cuales tareas de escaneo se han generado, esta opción consulta en la base de datos y lista todas las tareas de este tipo que se han creado en el sistema. De este modo se puede conocer los rangos de fecha en que están pautadas las capturas de los datos de los dispositivos Bluetooth, los *Puntos de Acceso* que realizaran esta labor y el usuario administrador que generó la tarea de escaneo tal como se muestra en la Figura 6.10.

Figura 6.10 Listar tareas de escaneo.

Dispositivos Escaneados

Esta opción permite consultar todos los dispositivos que han sido capturados en las labores de escaneo tal como se muestra en la Figura 6.11, se pueden ver datos como: la dirección Bluetooth del dispositivo, el nombre del dispositivo, la clase de dispositivo, el momento y el lugar donde fue capturado el dispositivo. De la misma manera se puede generar un archivo Excel con estos datos para manipularlos de otra forma, si así lo requiere el usuario. De igual modo se permite seleccionar un dispositivo si se desea conocer más detalles del mismo.

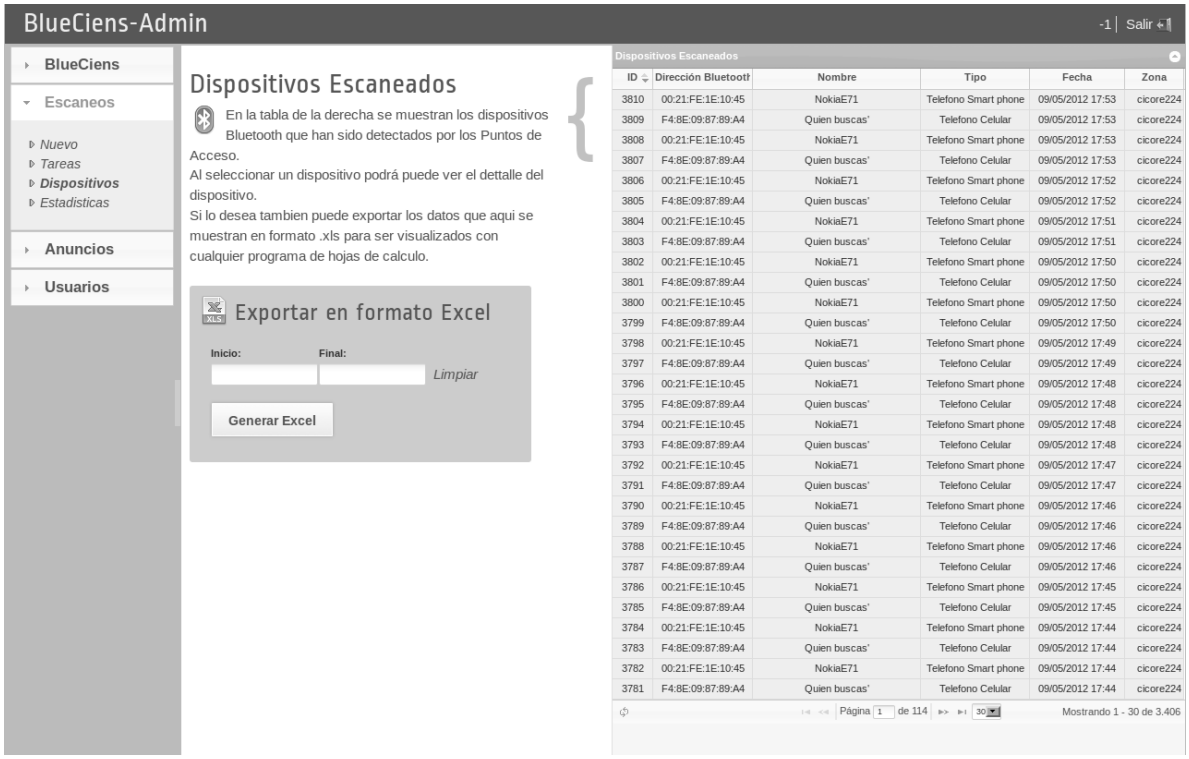

Figura 6.11 Lista de dispositivos escaneados.

Estadísticas de Dispositivos Escaneados

Para entender de una manera más sencilla los datos capturados, se pueden generar gráficas que ilustran el comportamiento de los dispositivos Bluetooth en los lugares donde han sido llevados a cabo los escaneos de dispositivos Bluetooth. En primer lugar se solicita que tipo de gráfica se requiere tal como se muestra en la Figura 6.12, para ello la herramienta permite decidir entre cuatro gráficas diferentes.

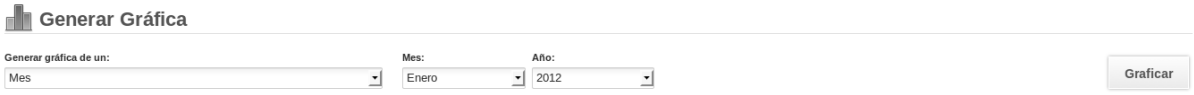

Figura 6.12 Generar gráfica estadística.

La primera opción es generar una gráfica por día, donde se puede observar cuantos dispositivos fueron capturados por hora durante un día. La siguiente opción es por mes, donde se puede observar cuantos dispositivos fueron capturados por día durante un mes. También otra alternativa es por año, donde se puede observar cuantos dispositivos se han capturado por mes durante un año. Por último, se puede generar una gráfica que muestra los intervalos de aparición de los dispositivos Bluetooth durante un día.

Una vez especificada la opción a graficar, se solicitan los datos correspondientes para hacer la consulta en la base de datos. Una vez obtenidos los datos necesarios se genera la

gráfica que es mostrada al usuario para su análisis tal como se muestra en la Figura 6.13. De este modo se puede obtener información bastante valiosa del uso de los dispositivos Bluetooth en los espacios donde haya sido desplegada la herramienta.

Figura 6.13 Gráfica estadística.

Distribución de Anuncios

La herramienta administrativa también permite generar los anuncios que se pueden distribuir a los dispositivos Bluetooth que soporten el protocolo OBEX. Del mismo modo se pueden visualizar los anuncios ya creados. Para acceder a estas funcionalidades también hacemos uso del menú situado en la parte izquierda de la interfaz de usuario en la sección “Anuncios”.

Nueva Distribución de Anuncio

Para crear un nuevo anuncio se utiliza un formulario como se muestra en la Figura 6.14, donde se le solicita al usuario datos tales como el nombre del anuncio, una breve descripción del mismo, el cuerpo del mensaje que se desea entregar, la fecha de inicio, la cual indica a partir de qué momento se desea empezar a hacer entrega del anuncio, la fecha de finalización, la cual pauta hasta qué momento se hará entrega de dicho anuncio, y por último los *Puntos de Acceso* que harán entrega de este anuncio.

Una vez especificados todos estos datos, serán guardados, y la herramienta hace uso del cuerpo del mensaje para generar una imagen GIF, la cual se utiliza para enviar a los dispositivos Bluetooth. La decisión de enviar el anuncio en este formato se debe a que puede

ser leído por una mayor cantidad de dispositivos. Una vez generado y registrado el anuncio, el *Manager* se encargará eventualmente de coordinar los *Puntos de Acceso* destinados para tal fin para la entrega del mencionado anuncio.

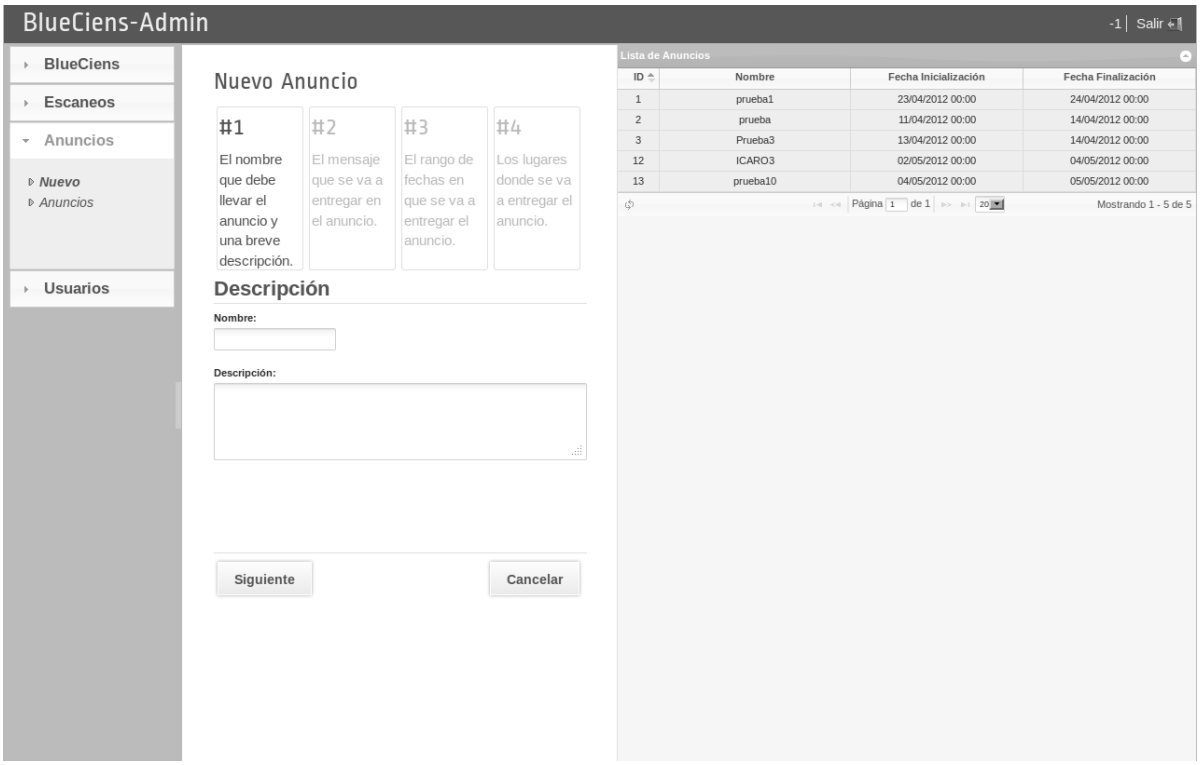

Figura 6.14 Nuevo anuncio.

Listar Anuncios

Se puede conseguir visualizar los anuncios ya creados en la herramienta como se muestra en la Figura 6.15, una consulta a la base de datos devolverá una lista con todos estos, indicando la fecha de inicio y finalización para su entrega como se muestra en la Figura 6.16. Al seleccionar un anuncio se puede observar con mas detalla el anuncio, también se puede conocer que usuario administrador generó el anuncio y a cuántos dispositivos Bluetooth le ha sido entregado.

Figura 6.15 Detalle de anuncio.

Figura 6.16 Lista de anuncios.

Configuración

Se pueden tener varios usuarios en la herramienta, por tal motivo es necesario un módulo que permita manejar este tema. Por tal motivo desde aquí se pueden crear nuevos usuarios y listar los que se encuentran registrados tal como muestra la Figura 6.17. Al crear un nuevo usuario un formulario solicita los datos del mismo, los cuales al ser guardados serán registrados en la base de datos de la herramienta. De este modo, el nuevo usuario podrá hacer uso del componente administrativo hasta donde sus permisos lo permitan. También se pueden visualizar todos los usuarios registrados en la herramienta. Al seleccionar alguno se pueden modificar sus permisos o inhabilitar al usuario si así se desea.

Figura 6.17 Lista de usuarios.

Estado de la Herramienta

Con esta funcionalidad se puede conocer el estado en que se encuentran los distintos componentes de la herramienta BlueCiens. Tal como se muestra en la Figura 6.18, se listan los *Puntos de Acceso* que han sido registrados y cuáles de estos se encuentran actualmente conectados con el servidor. Por otro lado en la Figura 6.19 se muestra el estado del *Manager* y si este se encuentra actualmente activo o no.

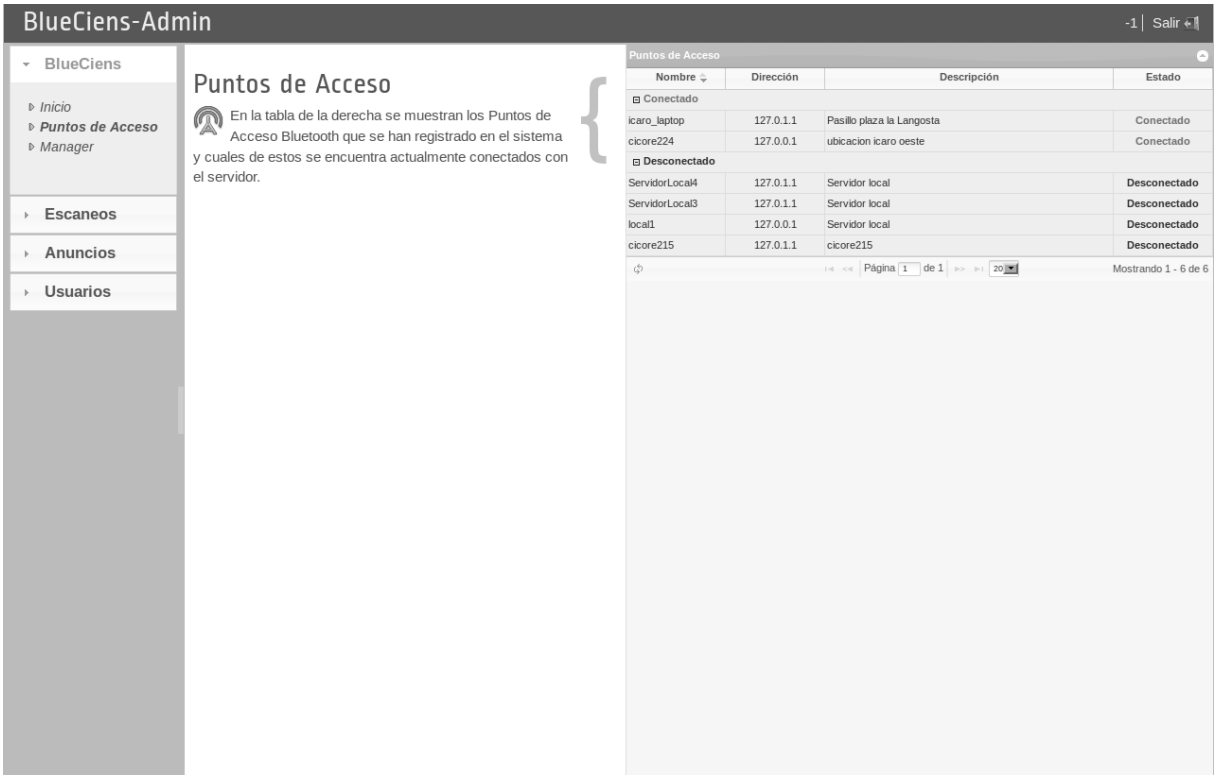

Figura 6.18 Estado de los *Puntos de Acceso*.

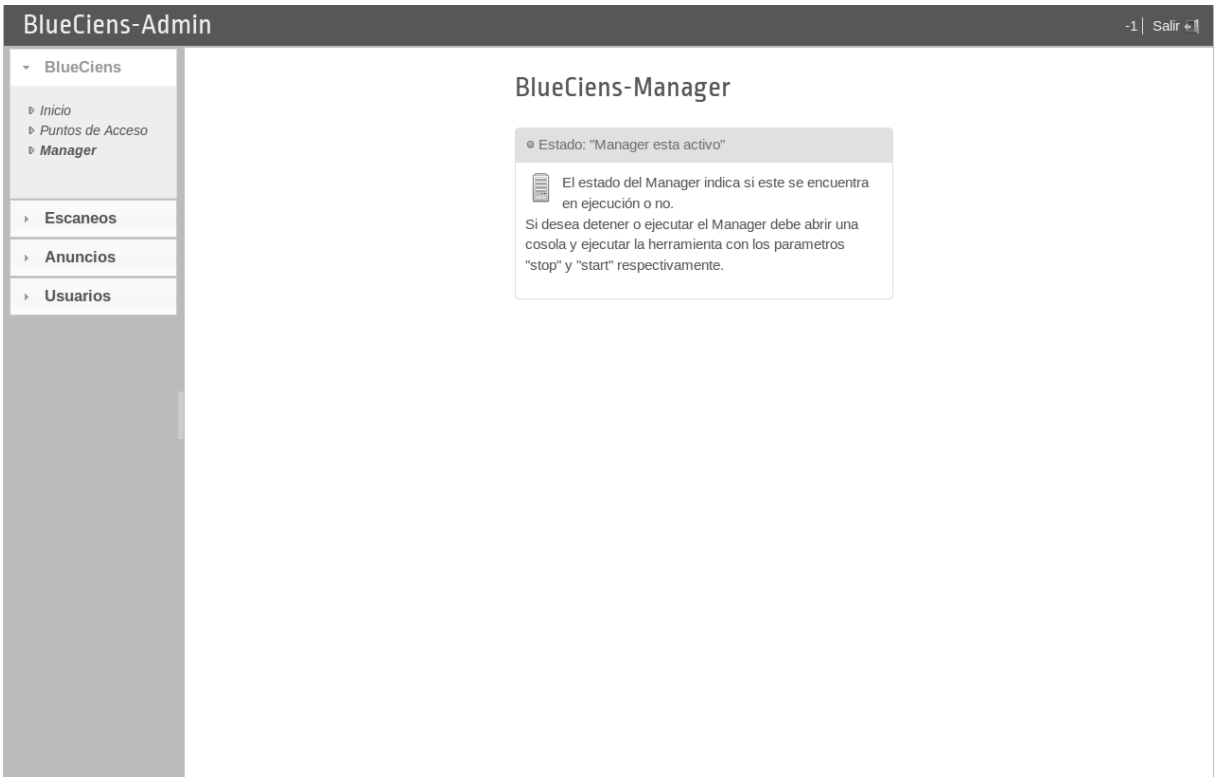

Figura 6.19 Estado del *Manager*.

7. Realización de Pruebas y Análisis de los Resultados

Una vez concluida la implementación de la herramienta es necesario transitar por un proceso de pruebas que permita comprobar que los requerimientos planteados en un principio han sido alcanzados. En consecuencia, en este capítulo se aborda este proceso, desde la definición del entorno de pruebas, su despliegue, la ejecución de los casos de prueba y el posterior análisis de los resultados obtenidos.

El objetivo fundamental de la herramienta como ya se ha mencionado es el escaneo y la distribución de anuncios usando Bluetooth. Por tal motivo, este proceso de pruebas va enfocado fundamentalmente en evaluar dichas funcionalidades y analizar los resultados que se generen.

7.1. Definición del Entorno de Pruebas

Para la realización de las pruebas sobre la herramienta de escaneo y distribución de anuncios es necesario recrear la arquitectura que se plantea en el capitulo anterior. De este modo se tiene un ambiente controlado que permite someter la aplicación a pruebas de funcionamiento y desempeño. Por lo tanto, se debe preparar un servidor que aloje el componente *Manager* y el componente de administración de la herramienta, del mismo modo, es necesario poner en marcha al menos un *Punto de Acceso*, que permita el escaneo y la distribución de anuncios utilizando Bluetooth, tal como se muestra en la Figura 7.1.

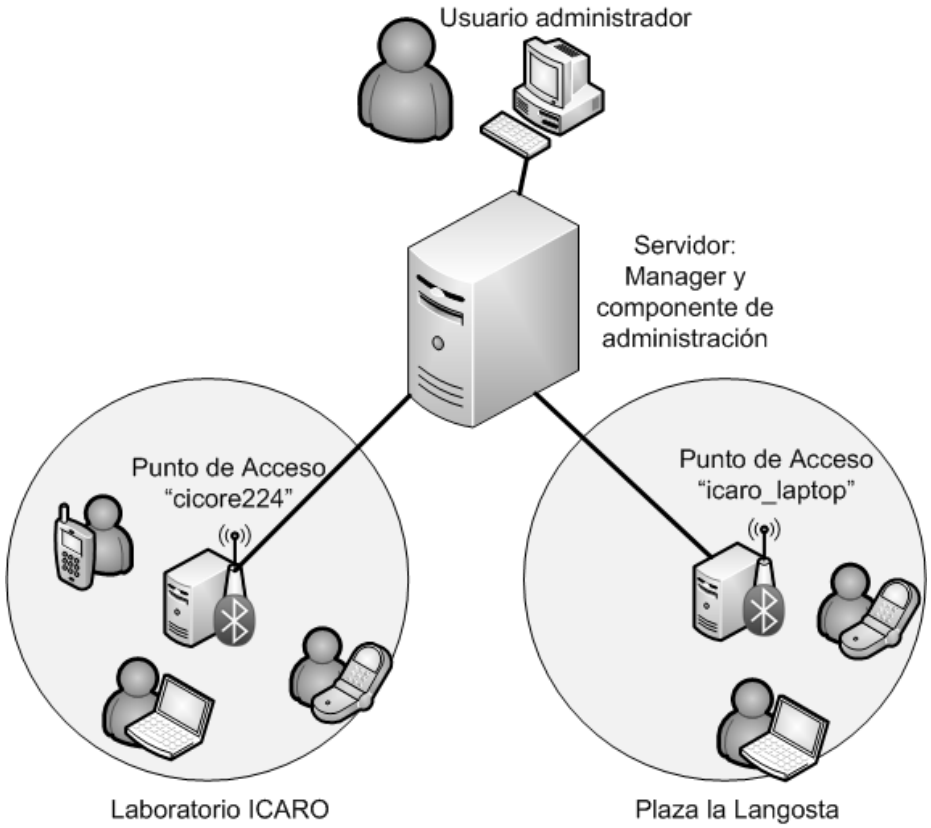

Figura 7.1 Entorno de pruebas

La definición de dicho entorno requiere una evaluación de los recursos con los que se dispone. Por lo tanto, en este caso se cuenta con los siguientes equipos:

- Una máquina virtual en un servidor Debian Squeeze, una distribución de GNU/Linux que basa sus principios en el software libre y está caracterizada por su estabilidad y robustez, la cual se adecua para funcionar como servidor de la herramienta.
- Para cumplir los roles de *Puntos de Acceso* se cuenta con una computadora personal y un computador portátil, ambos con el sistema operativo Ubuntu 10.04, una distribución GNU/Linux basada en Debian, así mismo los dos equipos igualmente disponen de adaptadores Bluetooth USB.
- También se cuenta con la interconexión de dichos componentes utilizando la red local del laboratorio ICARO.

Una vez definidos todos estos elementos necesarios para construir el entorno de pruebas, se puede proceder a la puesta en marcha de cada uno de estos componentes. De este modo se obtiene un entorno que permite ejecutar los casos de prueba y evaluar el desempeño de la herramienta, y del mismo modo el correcto cumplimiento de las funcionalidades que fueron trazadas en un principio.

7.2. Puesta en Marcha del Entorno de Pruebas

Para que cada uno de los componentes definidos en el escenario de pruebas se encuentre totalmente operativo fue necesario realizar ciertas configuraciones sobre los equipos para garantizar su funcionamiento. Por lo tanto, a continuación se describen los detalles de este proceso.

Para poner en marcha el servidor y todos sus componentes, fue necesario instalar y configurar las siguientes aplicaciones y herramientas:

- Python en su versión 2.7, que permite la ejecución del *Manager*.
- El paquete python-mysqldb que es la interfaz que permite acceder al manejador de base de datos MySQL desde cualquier aplicación desarrollada en Python.
- MySQL, manejador de base de datos que será utilizado para crear la base de datos.
- Apache con el lenguaje PHP 5.x, servidor web que alojará el componente de administración de la herramienta.

Una vez que se tienen todas estas aplicaciones trabajando correctamente, se procede a instalar los componentes *Manager* y de administración, asimismo, se procede a la creación de la base de datos utilizando el modelo planteado en uno de los capítulos anteriores. Una vez concluidos estos pasos se debe editar el archivo de configuración del *Manager*, en este momento se podrá ejecutar el demonio del *Manager*. De este modo se tiene totalmente operativo el servidor y solo resta la puesta en marcha de los *Puntos de Acceso* Bluetooth.

El *Punto de Acceso* número uno, cuyo nombre es “cicore224” está desplegado en una computadora personal, que para poder cumplir su rol utiliza un adaptador Bluetooth directamente conectado al puerto USB. Este equipo se encuentra ubicado dentro de la sala oeste del laboratorio ICARO. Esta localización permite interactuar con los dispositivos Bluetooth que se encuentren dentro de este recinto. Según mediciones se tiene un radio de alcance de hasta 5 metros en la cobertura de dicho *Punto de Acceso*.

El *Punto de Acceso* número dos, cuyo nombre es “icaro_laptop” se encuentra desplegado en una computadora portátil. La localización de este equipo es una oficina en la sala este del laboratorio ICARO. En esta ocasión el adaptador Bluetooth está conectado a través de una extensión USB de 10 metros de longitud, que permite colocar dicho adaptador fuera del laboratorio, específicamente sobre el pasillo adyacente a la plaza la Langosta de la Facultad de Ciencias de la UCV, de este modo se puede interactuar con los dispositivos Bluetooth en un ambiente donde existe una mayor concurrencia de personas y probablemente una mayor cantidad de dispositivos. Según mediciones se tiene un radio de alcance de hasta 5 metros en la cobertura de dicho *Punto de Acceso*.

Ambos *Puntos de Acceso* requieren la instalación y configuración de las siguientes aplicaciones y herramientas:

- Python en su versión 2.7, que permite la ejecución del *Punto de Acceso*.

- Las librerías PyBluez y Lightblue que permiten la utilización de las funcionalidades de Bluetooth desde las aplicaciones desarrolladas en Python.

Después que se tienen estas aplicaciones y herramientas en correcto funcionamiento se procede a la instalación del *Punto de Acceso*. Luego de haber concluido la instalación se procede a editar el archivo de configuración. De este modo ya se puede ejecutar el *Punto de Acceso*, es necesario que el *Manager* se encuentre activo y listo para recibir cualquier petición proveniente de los *Puntos de Acceso*.

De esta manera concluye la puesta en marcha del entorno de pruebas, el siguiente paso es definir y ejecutar los casos de prueba que arrojarán unos resultados que permitirán evaluar el correcto funcionamiento y desempeño de la aplicación.

7.3. Casos de Prueba

Una vez concluida la puesta en marcha del entorno de pruebas se deben definir varios casos de pruebas que vayan alineados con los requerimientos definidos inicialmente. Este paso es de suma importancia, ya que permite evaluar el cumplimiento de los objetivos que se han planteado. A continuación se definen estos casos de prueba.

Los requerimientos fundamentales de la aplicación son el escaneo y la distribución de anuncios usando Bluetooth. Por tal motivo se define un caso de prueba que permite evaluar el escaneo de dispositivos Bluetooth, y por otro lado, se define un caso de pruebas que permite evaluar la distribución de anuncios utilizando Bluetooth.

Escaneo de dispositivos Bluetooth

Este caso de prueba consiste en evaluar el proceso de captura de datos de los dispositivos Bluetooth. Para tal fin se debe crear una nueva tarea de escaneo desde el componente de administración de la herramienta. Para ello se debe definir el rango de tiempo durante el cual se va a llevar a cabo la prueba y los *Puntos de Acceso* que serán utilizados para esta prueba. En la Tabla 7.1 se muestran los parámetros seleccionados.

Fecha de inicio	7 de Mayo de 2012
Fecha de finalización	15 de Mayo de 2012
Puntos de Acceso	cicore224 e icaro_laptop

Tabla 7.1 Parámetros caso de prueba 1.

Los resultados esperados deben ser los datos de los dispositivos que hayan sido capturados por los *Puntos de Acceso* durante este rango de tiempo. Se espera conseguir gran cantidad de datos relacionados con los dispositivos Bluetooth de las personas que transiten por las áreas de cobertura de los *Puntos de Acceso.* Se puede mencionar que debido a la falta de control sobre la totalidad de los dispositivos que pueden ser escaneados por los *Puntos de Acceso*, se comprobará que se capturen los datos de un dispositivo Bluetooth sobre el que se tenga total dominio y se sabe que se encontrará dentro del área de pruebas. Para ello se hará uso de un teléfono celular Nokia C1-01 cuya dirección Bluetooth es F4:8E:09:87:89:A4. Por lo

tanto, se debe asegurar que este dispositivo sea registrado durante la ejecución de dicha prueba.

Distribución de Anuncios Usando Bluetooth

Este caso de pruebas consiste en evaluar la distribución de anuncios utilizando Bluetooth. Para conseguir este objetivo se debe generar al menos un anuncio utilizando la herramienta de administración. Se debe definir el nombre, el contenido, el rango de tiempo empleado para la entrega y los *Puntos de Acceso* encargados de la distribución de dicho anuncio. En la Tabla 7.2 se muestran dichos parámetros.

Nombre	BLUECIENS_PRUEBA2
Contenido	Activa tu Bluetooth y mantente informado
Fecha de inicio	18 de Mayo de 2012
Fecha de finalización	21 de mayo de 2012
Puntos de Acceso	cicore224 e icaro_laptop

Tabla 7.2 Parámetros caso de prueba 2.

Tal como sucede con la prueba anterior, en este caso tampoco se tiene control sobre la totalidad de los dispositivos a los que pueden llegar los anuncios utilizando Bluetooth. Por tal razón nos limitaremos a comprobar que los dispositivos a los que se tenga acceso reciban el anuncio que haya sido pautado para ser entregado utilizando Bluetooth. Para esta prueba también se utilizó el teléfono celular Nokia C1-01 cuya dirección Bluetooth es F4:8E:09:87:89:A4. En consecuencia, el teléfono celular debe recibir al menos una vez el anuncio de prueba, cuando este se encuentre del dentro del radio de cobertura de los *Puntos de Acceso*.

7.4. Análisis de resultados

La ejecución de los casos de prueba produjo una serie de resultados que son plasmados a continuación. Primeramente se exponen los resultados obtenidos tras la ejecución del caso de prueba para el escaneo de dispositivos Bluetooth y el correspondiente análisis de los mismos. De igual forma se hará lo mismo con el caso de pruebas para la distribución de anuncios usando Bluetooth.

Escaneo de dispositivos Bluetooth

Tal como se especificó anteriormente este caso de pruebas contempló un escaneo de dispositivos Bluetooth durante siete días utilizando los *Puntos de Acceso* “cicore224” e “icaro_laptop”, que cubren un radio de acción dentro del laboratorio ICARO y parte de la plaza la Langosta de la Facultad de Ciencias de la UCV, respectivamente.

Los resultados obtenidos son expuestos a continuación. Los datos que se muestran representan la cantidad de dispositivos escaneados durante el intervalo de tiempo que duró el caso de prueba. Al concluir dicho análisis general, se selecciona un día como muestra para

observar con más detalle los datos de los dispositivos escaneados, tales como los momentos del día en que fueron escaneados los dispositivos y otros datos de interés, como el nombre utilizado por los dispositivos para identificarse.

Es de notar que fuera del rango utilizado para realizar las pruebas también se obtuvieron resultados provenientes de pruebas realizadas para la puesta en marcha de la herramienta BlueCiens.

El primer conjunto de datos representa la cantidad de dispositivos escaneados por todos los *Puntos de Acceso* que conforman la herramienta BlueCiens durante el año 2012, tal como se aprecia en la Tabla 7.3 y en la Figura 7.2.

Meses del año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nob	Dic
Nº Dispositivos	0	0	0	23	64	0	0	0	0	0	0	0

Tabla 7.3 Cantidad de dispositivos escaneados por todos los *Puntos de Acceso* en 2012.

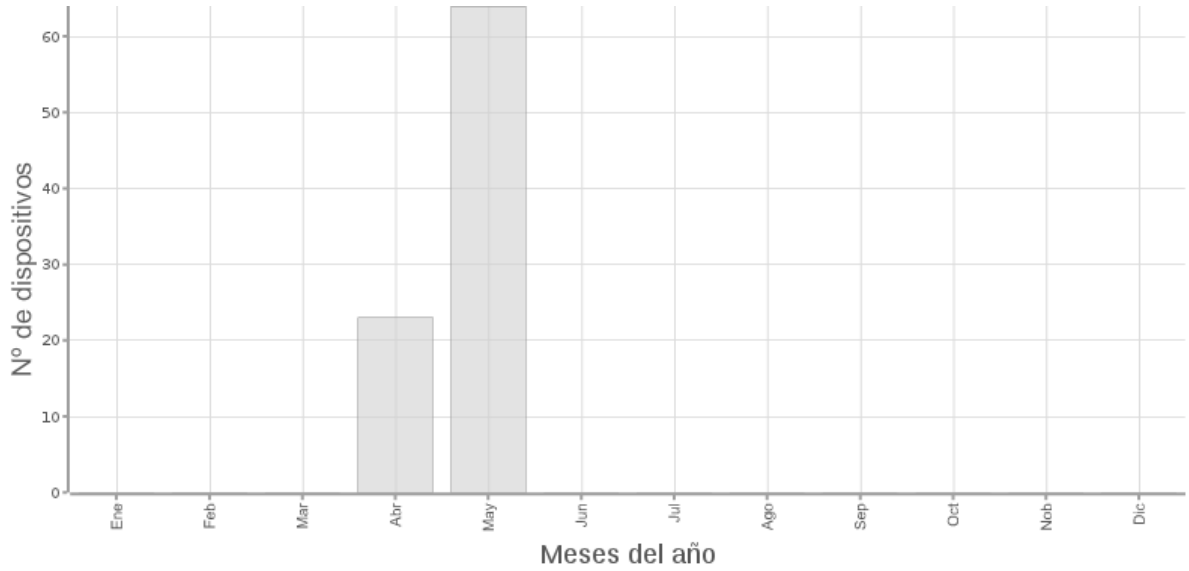

Figura 7.2 Cantidad de dispositivos escaneados por todos los *Puntos de Acceso* en 2012.

La gráfica anterior nos muestra un pico máximo de 64 dispositivos diferentes escaneados durante el mes de Mayo. Esto revela que la herramienta efectivamente está cumpliendo con uno de sus objetivos para la cual fue diseñada, ya que durante este período de evaluación se observa la gran cantidad de dispositivos registrados. Esto permite deducir que la herramienta estuvo en funcionamiento y realizando la tarea que le fue asignada para el escaneo de dispositivos Bluetooth.

El siguiente conjunto de datos representa la cantidad de dispositivos escaneados por el *Punto de Acceso* “icaro_laptop” durante el año 2012, tal como se muestra en la Tabla 7.4 y en la Figura 7.3.

Meses del año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nob	Dic
Nº Dispositivos	0	0	0	13	48	0	0	0	0	0	0	0

Tabla 7.4 Cantidad de dispositivos escaneados por "icaro_laptop" en 2012.

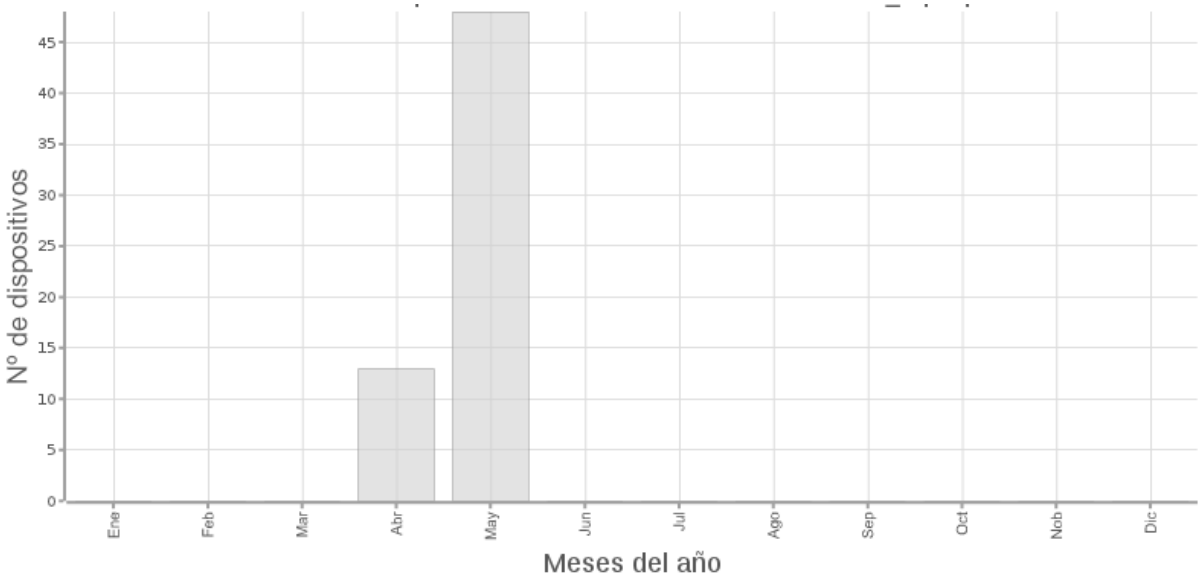

Figura 7.3 Cantidad de dispositivos escaneados por "icaro_laptop" en 2012.

Se puede apreciar en la gráfica como existe una relación con la primera Figura 7.2, ya que también existe un pico máximo de 48 dispositivos escaneados durante el mes de Mayo. Esto nos indica que este *Punto de Acceso* se mantuvo en funcionamiento y realizando la tarea que le fue asignada durante este período de evaluación.

Seguidamente se exponen los datos obtenidos por el *Punto de Acceso* “cicore224”. En la Tabla 7.5 y en la Figura 7.4 se aprecia el número de dispositivos escaneados por este *Punto de Acceso* durante el año 2012.

Meses del año	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nob	Dic
Nº Dispositivos	0	0	0	12	37	0	0	0	0	0	0	0

Tabla 7.5 Cantidad de dispositivos escaneados por "cicore224" en 2012.

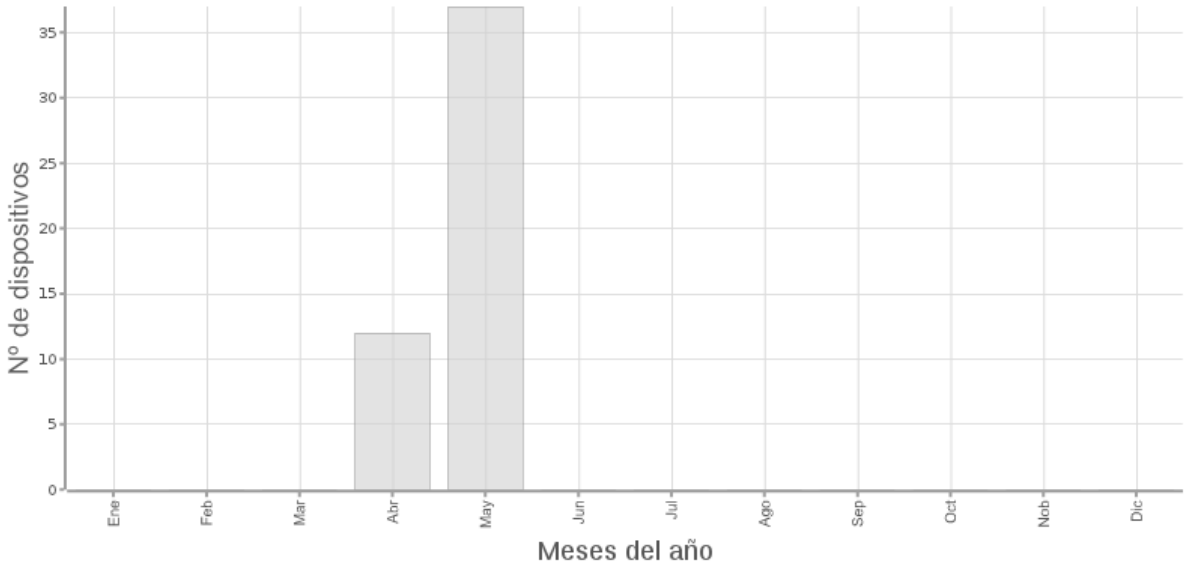

Figura 7.4 Cantidad de dispositivos escaneados por "cicore224" en 2012.

En la Figura 7.4 se observa de igual forma un pico máximo de 37 dispositivos escaneados durante el mes de Mayo lo cual revela el funcionamiento de este *Punto de Acceso* y un comportamiento similar al de su par, el *Punto de Acceso* “icaro_laptop”. Se puede notar aquí como el *Punto de Acceso* “icaro_laptop” ubicado en la plaza registra un mayor número de dispositivos que el *Punto de Acceso* “cicore224” ubicado dentro del laboratorio.

Una vez visto el comportamiento de la herramienta desde un punto de vista general, se observará ahora con más detalle que sucedió durante el mes de Mayo. El siguiente conjunto de datos plasmados en la Tabla 7.6 y la Figura 7.5 exhibe el número de dispositivos escaneados diariamente durante el mes de Mayo por la totalidad de los *Puntos de Acceso* utilizados por la herramienta BlueCiens.

Días del mes	Número de dispositivos
1	0
2	3
3	2
4	5
5	0
6	0
7	2
8	23
9	15
10	24
11	10
12	6
13	2
14	10
15	5

Tabla 7.6 Cantidad de dispositivos escaneado por todos los *Puntos de Acceso* durante Mayo de 2012.

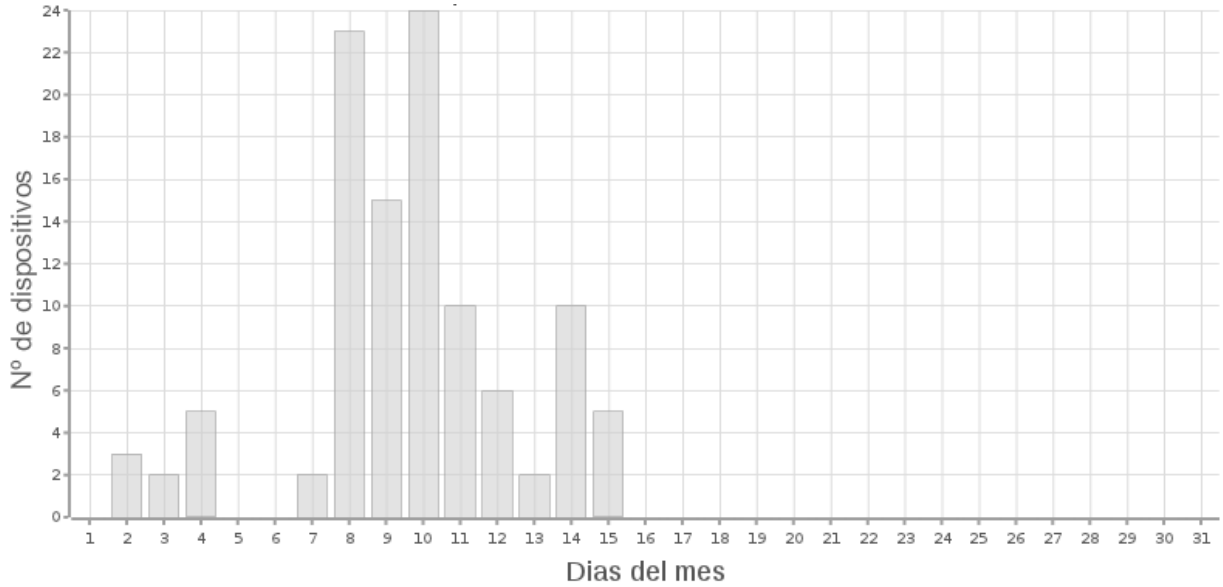

Figura 7.5 Cantidad de dispositivos escaneado por todos los *Puntos de Acceso* durante Mayo de 2012.

Se puede apreciar como la cantidad de dispositivos escaneados se concentra durante los días que se llevó a cabo la prueba, teniendo una menor magnitud el fin de semana, los días 12 y 13 de Mayo y el día 7 que se da inicio la prueba en horas de la tarde y no se obtienen resultados del día completo. Los días en que se observa más dispositivos escaneados son los días martes 8 y jueves 10 de Mayo con un pico máximo de 24 dispositivos escaneados.

A continuación en la Tabla 7.7 y la Figura 7.6 se muestran el conjunto de datos provenientes del escaneo de dispositivos Bluetooth durante el mes de Mayo por el *Punto de Acceso* “icaro_laptop”.

Días del mes	Número de dispositivos
1	0
2	1
3	2
4	5
5	0
6	0
7	1
8	15
9	14
10	17
11	7
12	5
13	1
14	6
15	2

Tabla 7.7 Cantidad de dispositivos escaneado por el *Punto de Acceso* "icaro_laptop" durante Mayo de 2012.

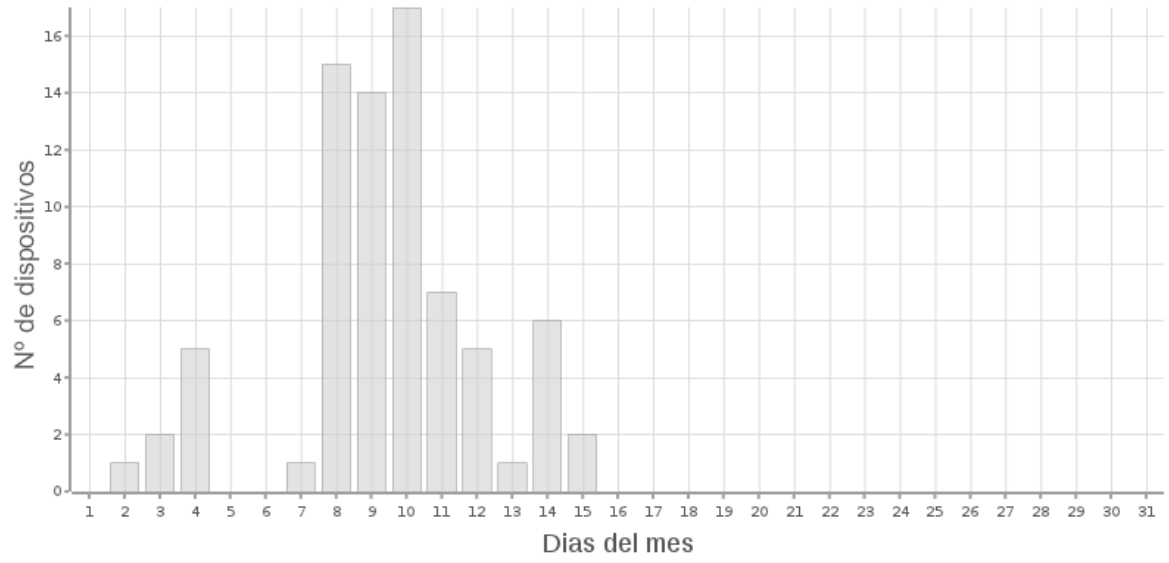

Figura 7.6 Cantidad de dispositivos escaneado por el *Punto de Acceso* "icaro_laptop" durante Mayo de 2012.

En los datos plasmados anteriormente en la Tabla 7.7 y en la Figura 7.6 se observa como existe una relación con los datos arrojados por el conjunto total de los *Puntos de Acceso*. Se puede apreciar como existe un mayor número de dispositivos escaneados durante los días de semana y el fin de semana disminuye notablemente la cantidad de dispositivos escaneados.

El siguiente conjunto de datos que se exponen en la Tabla 7.8 y la Figura 7.7 muestran los dispositivos escaneados por el *Punto de Acceso* “cicore224” durante el mes de Mayo.

Días del mes	Número de dispositivos
1	0
2	3
3	0
4	1
5	0
6	0
7	2
8	12
9	3
10	17
11	5
12	5
13	2
14	6
15	5

Tabla 7.8 Cantidad de dispositivos escaneado por el *Punto de Acceso* "cicore224" durante Mayo de 2012.

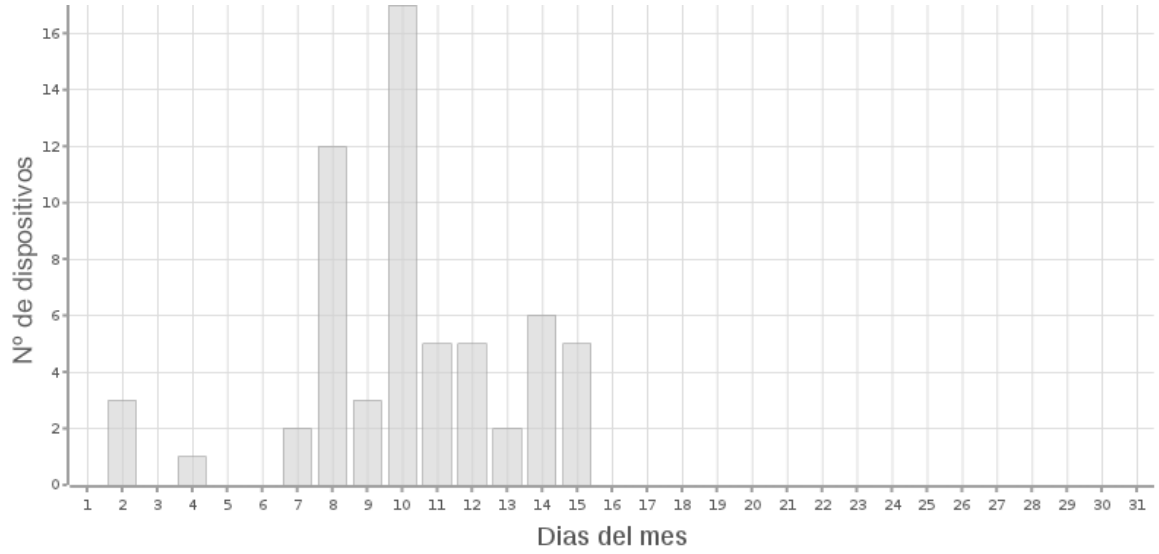

Figura 7.7 Cantidad de dispositivos escaneado por el *Punto de Acceso* "cicore224 " durante Mayo de 2012.

Se puede observar que la cantidad de dispositivos escaneados por el *Punto de Acceso* “cicore224” (ubicado dentro del laboratorio ICARO) en promedio es menor que los dispositivos escaneados por el *Punto de Acceso* “icaro_laptop” (ubicado en las adyacencias de la plaza la Langosta de la UCV). Por lo tanto se confirma que la cantidad de personas que transitan por el radio de alcance del *Punto de Acceso* Bluetooth es proporcional al número de dispositivos escaneados.

Una vez visto el comportamiento de la cantidad de dispositivos escaneados durante el mes de Mayo se procede a observar este comportamiento de forma más específica. Para tal propósito se seleccionó un día de muestra dentro de este período de pruebas para observar la cantidad de dispositivos escaneados a lo largo del día. El siguiente conjunto de datos que se revelan en la Tabla 7.9 y en la Figura 7.8 expone la cantidad de dispositivos escaneados durante el día 8 de Mayo por todos los *Puntos de Acceso* que componen la herramienta BlueCiens.

Horas del día	Número de dispositivos
00:00 - 00:59	0
01:00 - 01:59	0
02:00 - 02:59	0
03:00 - 03:59	0
04:00 - 04:59	0
05:00 - 05:59	0
06:00 - 06:59	0
07:00 - 07:59	0
08:00 - 08:59	1
09:00 - 09:59	2
10:00 - 10:59	5
11:00 - 11:59	5
12:00 - 12:59	7
13:00 - 13:59	1
14:00 - 14:59	5
15:00 - 15:59	4
16:00 - 16:59	2
17:00 - 17:59	2
18:00 - 18:59	3
19:00 - 19:59	1
20:00 - 20:59	1
21:00 - 21:59	1
22:00 - 22:59	1
23:00 - 23:59	1

Tabla 7.9 Cantidad de dispositivos escaneados por todos los *Puntos de Acceso* el día 8 de Mayo.

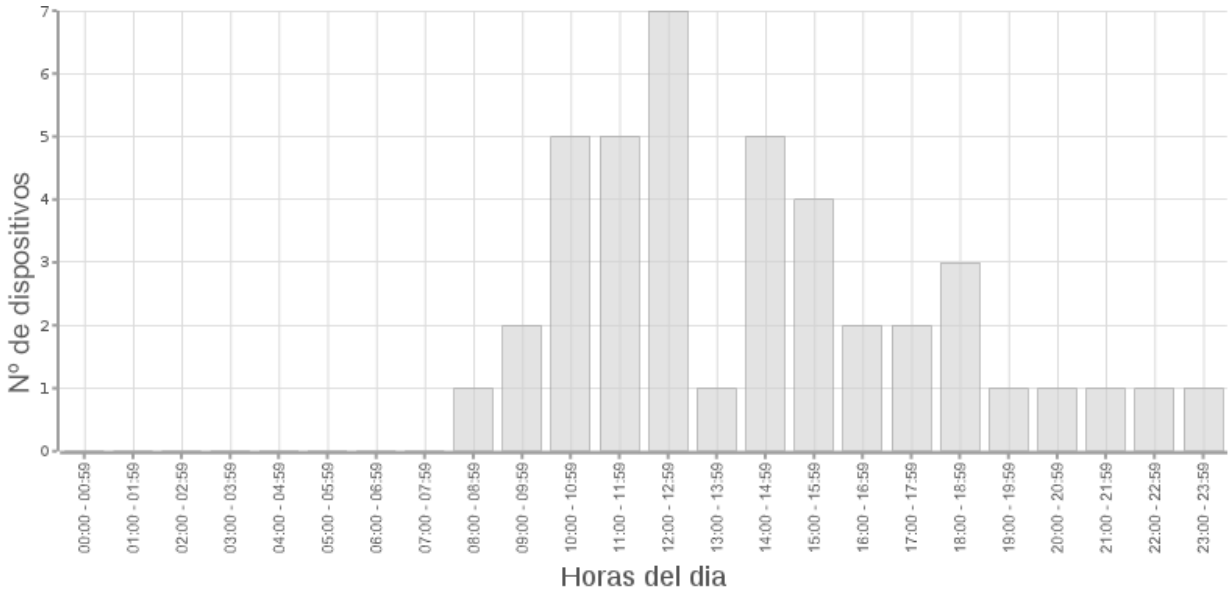

Figura 7.8 Cantidad de dispositivos escaneados por todos los *Puntos de Acceso* el día 8 de Mayo.

Tal como se puede distinguir el mayor número de dispositivos escaneados se concentra en las horas del mediodía, llegando a un pico máximo de 7 dispositivos escaneados durante las 12:00 y las 12:59. Estos resultados son coherentes debido a que estas horas son las de mayor afluencia de personas por las áreas donde se encuentran ubicados los *Puntos de Acceso*.

El siguiente conjunto de datos plasmados en la Tabla 7.10 y la Figura 7.9 muestra el número de dispositivos escaneados por el *Punto de Acceso* “cicore224” durante el día 8 de Mayo.

Horas del día	Número de dispositivos
00:00 - 00:59	0
01:00 - 01:59	0
02:00 - 02:59	0
03:00 - 03:59	0
04:00 - 04:59	0
05:00 - 05:59	0
06:00 - 06:59	0
07:00 - 07:59	0
08:00 - 08:59	0
09:00 - 09:59	1
10:00 - 10:59	1
11:00 - 11:59	1
12:00 - 12:59	5
13:00 - 13:59	0
14:00 - 14:59	1
15:00 - 15:59	1
16:00 - 16:59	2
17:00 - 17:59	1
18:00 - 18:59	3
19:00 - 19:59	1
20:00 - 20:59	1
21:00 - 21:59	1
22:00 - 22:59	1
23:00 - 23:59	1

Tabla 7.10 Cantidad de dispositivos escaneados por el *Punto de Acceso* "cicore224" el día 8 de Mayo.

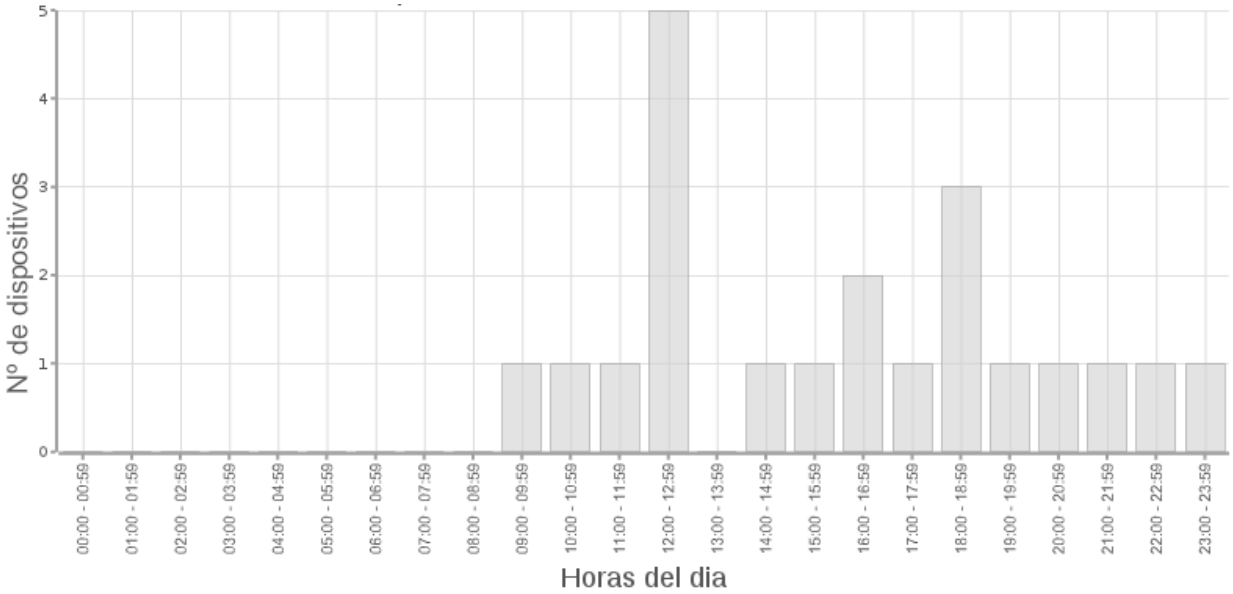

Figura 7.9 Cantidad de dispositivos escaneados por el *Punto de Acceso* "cicore224" el día 8 de Mayo.

Se puede observar de igual manera como el mayor número de dispositivos escaneados son en las horas del mediodía, y la menor cantidad de dispositivos son escaneados en la madrugada y en la noche. Cabe preguntarse por qué se capturaron dispositivos durante horas en la cual se puede pensar que no hay personas transitando por estas áreas. La respuesta a esta interrogante se debe a que durante este período de tiempo hubo dispositivos que fueron dejados sin intención en el lugar y por tal motivo fueron registrados por la herramienta.

A continuación se exponen en la Tabla 7.11 y la Figura 7.10 el conjunto de datos del número de dispositivos escaneados por el *Punto de Acceso* “icaro_laptop” durante el día 8 de Mayo.

Horas del día	Número de dispositivos
00:00 - 00:59	0
01:00 - 01:59	0
02:00 - 02:59	0
03:00 - 03:59	0
04:00 - 04:59	0
05:00 - 05:59	0
06:00 - 06:59	0
07:00 - 07:59	0
08:00 - 08:59	1
09:00 - 09:59	1
10:00 - 10:59	4
11:00 - 11:59	4
12:00 - 12:59	4
13:00 - 13:59	1
14:00 - 14:59	4
15:00 - 15:59	3
16:00 - 16:59	0
17:00 - 17:59	1
18:00 - 18:59	0
19:00 - 19:59	0
20:00 - 20:59	0
21:00 - 21:59	0
22:00 - 22:59	0
23:00 - 23:59	1

Tabla 7.11 Cantidad de dispositivos escaneados por el *Punto de Acceso* "icaro_laptop" el día 8 de Mayo.

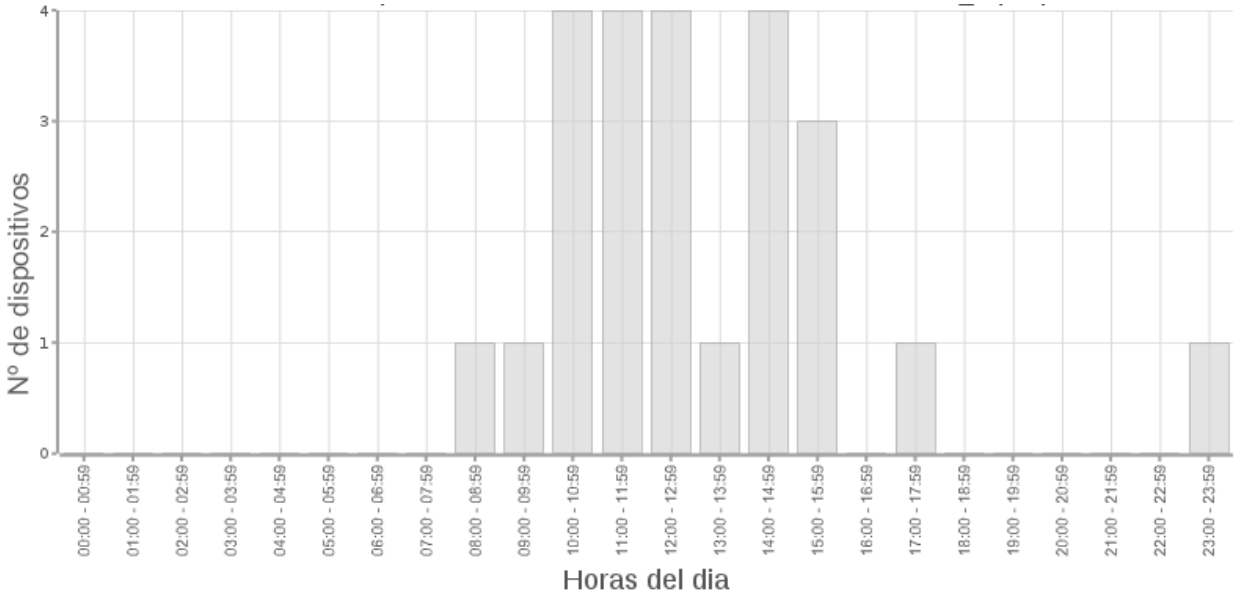

Figura 7.10 Cantidad de dispositivos escaneados por el *Punto de Acceso* "icaro_laptop" el día 8 de Mayo.

Se puede apreciar el mayor número de dispositivos escaneados durante la mañana en horas cercanas al mediodía y en la tarde. Un dispositivo fue escaneado entre las 23:00 y las 23:59, se puede pensar que es uno de los dispositivos dejados dentro del laboratorio capturado por el otro *Punto de Acceso*, y debido a la disminución del ruido radioeléctrico se logra escanear dicho dispositivo a una mayor distancia por el *Punto de Acceso* ubicado en la plaza la Langosta.

El siguiente conjunto de datos revela información bastante interesante. Se puede observar en detalle los momentos durante los cuales fueron escaneados los dispositivos por los *Puntos de Acceso*. Para empezar en la Tabla 7.12 y la Figura 7.11 se pueden ver los dispositivos escaneados por el *Punto de Acceso* “cicore224” durante el día 8 de Mayo.

Número de referencia	Dirección Bluetooth	Nombre del dispositivo
1	40:5F:BE:71:D6:EE	BlackBerry 9100
2	F4:8E:09:87:89:A4	Quien buscas'
3	E8:3E:B6:36:6C:E1	BlackBerry 9780
4	5C:4C:A9:E1:8F:9E	lore :)
5	00:24:90:5A:92:FB	3men2
6	F8:7B:7A:7B:4F:C0	Motorola EX115
7	3C:74:37:52:30:52	None
8	6C:D6:8A:32:3F:A2	None
9	18:14:56:BA:0A:2B	Pequitas
10	B4:B3:62:AC:B6:E2	dcd
11	00:25:66:8F:65:AE	None
12	00:21:FE:1E:10:45	NokiaE71

Tabla 7.12 Dispositivos escaneados por el *Punto de Acceso* "cicore224".

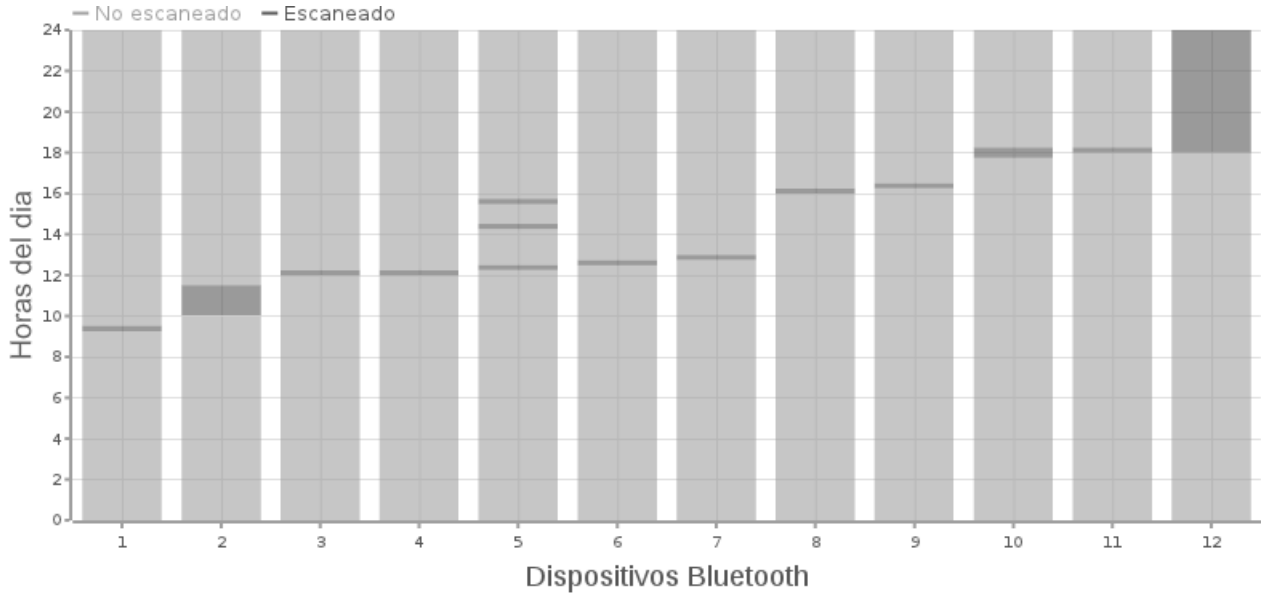

Figura 7.11 Dispositivos escaneados por el *Punto de Acceso* "cicore224".

Este conjunto de datos resulta interesante debido a que con ellos se puede saber en cuáles momentos se encontraba un dispositivo dentro del radio de alcance de un *Punto de Acceso*. De este modo se podría inferir en qué momento una persona, en este caso el dueño del dispositivo, transitó por este lugar. En este sentido se puede observar como aparece el dispositivo con la dirección F4:8E:09:87:89:A4 entre las 10:00 AM y las 12:00 M, el cual está siendo utilizado para realizar las pruebas de escaneo.

En la Tabla 7.13 y la Figura 7.12 se pueden observar los dispositivos escaneados por el *Punto de Acceso* “icaro_laptop” durante el día 8 de Mayo.

Número de referencia	Dirección Bluetooth	Nombre del dispositivo
1	C8:DF:7C:59:15:5C	Nokia 2690
2	4C:54:99:4D:4A:E4	El loco 100% :-(
3	14:74:11:95:B7:AE	miki!
4	40:4D:8E:64:2D:B2	Lito
5	00:23:B4:6B:3F:D6	Reiner
6	44:F4:59:40:1C:A5	GT-S3650
7	E8:3E:B6:36:6C:E1	BlackBerry 9780
8	F8:7B:7A:7B:4F:C0	Motorola EX115
9	C8:DF:7C:59:23:CB	Zoraida
10	03:C6:CB:60:66:01	QS200
11	5C:57:C8:75:2F:1D	@SURILI@
12	E0:A6:70:EC:50:B6	Nokia 2730 classic
13	00:1F:DE:50:39:3C	None
14	40:5F:BE:71:D6:EE	BlackBerry 9100
15	00:21:FE:1E:10:45	NokiaE71

Tabla 7.13 Dispositivos escaneados por el *Punto de Acceso* "icaro_laptop".

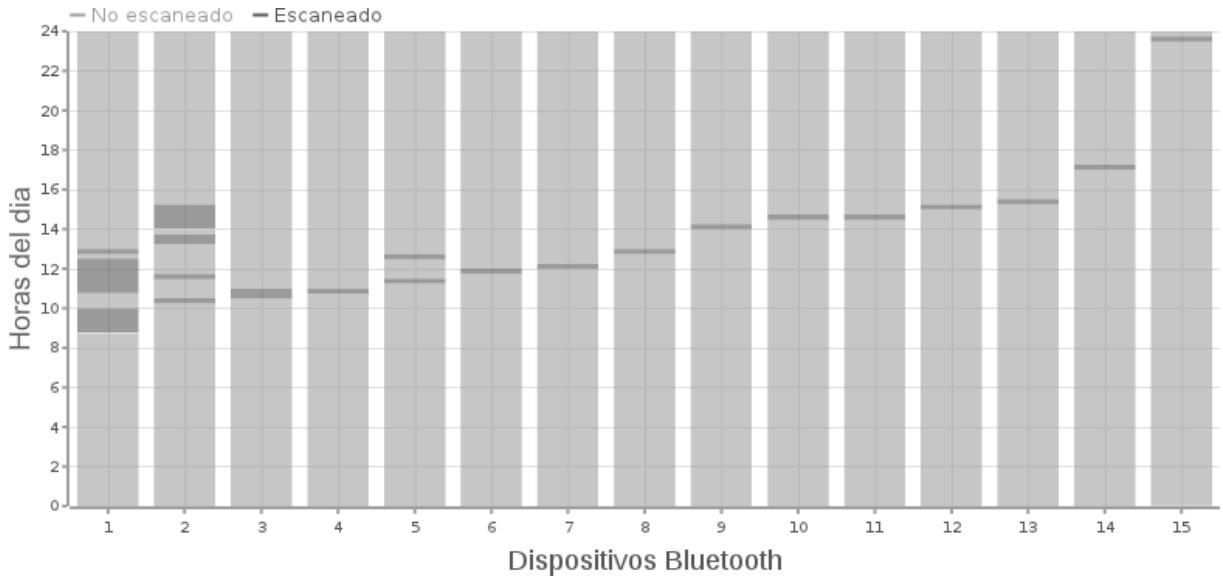

Figura 7.12 Dispositivos escaneados por el *Punto de Acceso* "icaro_laptop".

Se puede resaltar de la Figura 7.12 como aumenta el número de dispositivos escaneados en relación con el otro *Punto de Acceso*. Se puede ver también que la mayoría de los dispositivos fueron registrados en horas cercanas al mediodía. Del mismo modo, se puede mencionar como gran parte de los dispositivos capturados por los *Puntos de Acceso* conservan aún los nombres que traen configurados de fábrica, lo cual nos puede revelar qué tipo de dispositivo se está registrando.

Distribución de Anuncios Usando Bluetooth

Para este caso de pruebas se configuró en la herramienta la distribución de un anuncio durante 4 días. Este mensaje solo podrá ser entregado a aquellos dispositivos Bluetooth que dispongan del protocolo OBEX para el intercambio de archivos.

Tal como se explicó al momento de elaborar el diseño de este caso de pruebas, no se tiene control sobre todos los dispositivos Bluetooth que se encuentren dentro del radio de alcance de los *Puntos de Acceso* y puedan recibir el anuncio usando Bluetooth. Esta limitante ocasiona que solo se puede comprobar la recepción del anuncio en el dispositivo Bluetooth sobre el cual se tiene control.

Una vez iniciado el caso de pruebas se activó el dispositivo Bluetooth, un teléfono celular Nokia C1-01 con la dirección Bluetooth F4:8E:09:87:89:A4. Luego de unos minutos dentro del radio de alcance del *Punto de Acceso* “cicore224” se recibió en el teléfono celular una solicitud para aceptar un archivo proveniente del equipo “cicore224”. Al aceptar dicha solicitud se da inicio a la transferencia de un archivo del tipo imagen GIF llamado BLUECIENS_PRUEBA2.

Una vez completada la transferencia se puede abrir dicho archivo y se puede observar el anuncio enviado por el *Punto de Acceso*, tal como se muestra en la Figura 7.13 y en la Figura 7.14. Se puede apreciar una imagen GIF con 2 cuadros, el primero que muestra el logotipo de la herramienta BlueCiens y el segundo que contiene el cuerpo del mensaje que fue especificado anteriormente en el diseño del caso de pruebas.

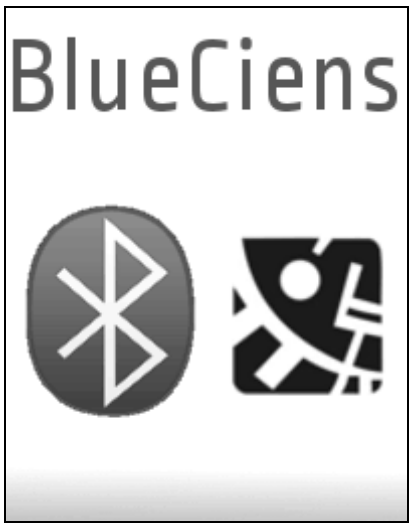

Figura 7.13 Portada de anuncio de BlueCiens.

Figura 7.14 Cuerpo de anuncio de BlueCiens.

Este anuncio fue recibido solo una vez por el dispositivo Bluetooth, lo cual indica que un mismo anuncio no será entregado más de una vez al mismo dispositivo, cumpliendo con uno de los requerimientos planteados inicialmente. Del mismo modo se puede destacar que según la herramienta BlueCiens se registró que este anuncio también fue entregado a 4 dispositivos más durante el lapso de tiempo que duró la prueba. En líneas generales el comportamiento de la herramienta cumplió con los objetivos planteados.

8. Conclusiones

Como se ha mencionado a lo largo de este trabajo, Bluetooth se ha convertido en una de las tecnologías preferidas de comunicaciones inalámbricas de corto alcance. Su bajo costo propicia su uso en un sinfín de dispositivos, lo cual trae consigo que estemos rodeados de aparatos que pueden comunicarse utilizando Bluetooth. Esto permite crear ambientes interconectados donde cada vez es más común que los dispositivos se comuniquen entre sí.

La técnica de escaneo Bluetooth permite recolectar datos de los dispositivos Bluetooth a fin de conocer las características y comportamientos de dichos dispositivos. A su vez la tecnología Bluetooth brinda la posibilidad de poder comunicarse por medio del envío de anuncios. Estas funcionalidades básicas permitieron plantearse el desarrollo de una herramienta para el escaneo y la distribución de anuncios usando Bluetooth dentro de la Facultad de Ciencias de la Universidad Central de Venezuela, la cual se denominó BlueCiens.

Con este propósito se concibió una herramienta que funciona a través de tres componentes: unos componentes llamados *Puntos de Acceso* que se encargan de realizar las labores de escaneo y distribución de anuncios; un componente central denominado *Manager* que coordina el trabajo de los *Puntos de Acceso*; y un componente que permite al usuario administrador hacer uso de la herramienta. Este modelo permitió el despliegue de la herramienta en un ámbito mayor, ya que solo basta con colocar un *Punto de Acceso* donde se desee utilizar la herramienta para extender el radio de acción de la misma.

A tal fin se implementó una metodología tradicional de desarrollo que hizo posible la evaluación del cumplimiento de los objetivos. Por otra parte se hizo una evaluación de las herramientas idóneas de hardware y software que facilitaran el desarrollo de este trabajo. De esta manera los resultados obtenidos permitieron llegar a algunas conclusiones de interés.

Al transitar por un período de pruebas se logró evaluar el cumplimiento de los requerimientos planteados. Estas pruebas se centraron en el análisis de las funcionalidades de escaneo y de distribución de anuncios usando Bluetooth.

Los resultados obtenidos en la prueba de escaneo revelaron que la herramienta utilizada logró recolectar datos de los dispositivos Bluetooth que se encontraban dentro del radio de alcance de la misma. Con esto se evidencia la posibilidad de obtener información relevante al mostrar las características y el comportamiento de los dispositivos Bluetooth en los lugares donde tuvo alcance dicha herramienta.

Por otra parte los resultados obtenidos en la prueba de distribución de anuncios demostraron que por medio de la herramienta se pueden enviar anuncios a los dispositivos Bluetooth que se encuentren dentro del radio de alcance de la misma, ofreciendo un medio alternativo para comunicarse con las personas que transitan dentro de la Facultad de Ciencias de la UCV.

8.1. Contribuciones

Este trabajo tiene como principal aporte el desarrollo de una herramienta para el escaneo y la distribución de anuncios utilizando Bluetooth dentro de la Facultad de Ciencias de la UCV.

El escaneo de dispositivos Bluetooth puede tener diversos fines, tales como su aplicación dentro de las redes sociales, el rastreo y la localización bajo techo, la medición del uso del espacio y muchos más. Por lo tanto tener una herramienta que facilite este trabajo resulta bastante ventajoso para emprender futuros trabajos en este campo.

La distribución de anuncios utilizando Bluetooth es una alternativa de comunicación a los medios tradicionales. La característica de proximidad que otorga esta tecnología permite mantener un enlace más estrecho con las personas que hacen vida dentro de un espacio, lo que conlleva a brindar información más precisa y específica de acuerdo al lugar donde se encuentre una persona.

Por tales razones, este trabajo pone a nuestro alcance poder hacer uso de estas funcionalidades dentro del ámbito que nos rodea para crear ambientes interconectados donde se aproveche al máximo las ventajas que nos brinda la tecnología.

8.2. Limitaciones

Para el desarrollo de la herramienta BlueCiens fue necesario enfrentar una serie de limitaciones que en algunos casos produjo retrasos y en otros obligaron a replantear la forma como era abordado el problema. A continuación se describen estas limitaciones:

- Una de las primeras limitaciones que se presentó fue la insuficiente cantidad de opciones de librerías de programación para el desarrollo de aplicaciones multiplataforma que faciliten el trabajo con la tecnología Bluetooth. Esto trajo como consecuencia buscar las librerías que ofrecieran las funcionalidades necesarias para desarrollar la herramienta, en especial para el escaneo y la distribución de anuncios. La simplicidad en el uso de las librerías también fue un factor clave para seleccionar la librería adecuada.
- Para un desempeño óptimo de la herramienta es necesario la existencia de una conexión permanente entre el *Manager* y los *Puntos de Acceso*. Cualquier interrupción de esta conexión puede detener las tareas de escaneo y distribución de anuncios que esté realizando un *Punto de Acceso*.
- No todos los dispositivos Bluetooth cuentan con el protocolo OBEX para el intercambio de archivos. Por tal motivo se limita el número de dispositivos a los cuales se le puede hacer entrega de los anuncios por esta vía.

8.3. Trabajos futuros

Con el fin de dar pie a la continuidad de este trabajo o al desarrollo de nuevos proyectos que se apoyen en éste para lograr sus objetivos, a continuación se plantean varias ideas que pueden ser llevadas a cabo.

Una de las características adicionales que pueden ser agregadas a la herramienta BlueCiens es el desarrollo de *Puntos de Acceso* que trabajen en la plataforma Windows, de esta forma se podrá tener un mayor alcance, y gracias a la naturaleza Cliente/Servidor de la herramienta podrán coexistir con *Puntos de Acceso* que se ejecuten en distintas plataformas.

Otra propuesta para dar continuidad a este trabajo tiene que ver con el desarrollo de terceras aplicaciones que se sustenten en la capacidad de escaneo que proporciona la herramienta BlueCiens. Los datos recolectados por esta herramienta pueden servir para distintos fines. Muchas aplicaciones de este tipo ya han sido desarrolladas en otras latitudes, sin embargo pueden ser implementadas en nuestro ámbito local.

Una de estas aplicaciones puede ser la localización de personas bajo techo, gracias a datos como el lugar donde se encuentra el *Punto de Acceso* que escanea un dispositivo Bluetooth, se puede inferir que el dueño del dispositivo escaneado se encuentra cerca de dicho lugar. Otra utilidad puede ser utilizar esta información para alimentar una red social, ya que se puede inferir que los dueños de dos dispositivos escaneados en un mismo lugar al mismo tiempo podrían conocerse y de este modo generar un posible vínculo social. De este modo nos damos cuenta que el universo de aplicaciones es bastante extenso, solo se necesita un poco de creatividad y utilizar las herramientas que ya se tienen a nuestro alcance.

En el ámbito de entrega de anuncios se puede extender más el proceso de creación del anuncio, el cual hasta el momento solo se puede insertar texto en el cuerpo del mensaje. Sin embargo queda la posibilidad de poder insertar imágenes en estos anuncios y hacer más atractivo este nuevo medio de comunicación dentro de la Facultad de Ciencias de la UCV.

Para terminar, hay que destacar que este trabajo puede servir como apoyo para crear ambientes interconectados que exploten al máximo los instrumentos tecnológicos que tenemos a nuestro alcance y así poder facilitar las tareas que realizamos a diario.

Referencias

[1] M. A. Awad (2005) *A Comparison between Agile and Traditional Software Development Methodologies*. The University of Western Australia.

[2] Bea, L. [Documento WWW]. Consultado Abril de 2012 en: http://lightblue.sourceforge.net/

[3] Bluetooth SIG. [Documento WWW]. Consultado Mayo 2011 en: http://www.bluetooth.org/

[4] Bluetooth SIG (30 de Junio de 2010) *Specification of Bluetooth System*. [Documento WWW]. Consultado en: http://www.bluetooth.org/

[5] Brent, M. (25 de Septiembre de 2000). *Bluetooth Revealed*.

[6] EllisLab, Inc. [Documento WWW]. Consultado Abril de 2012 en: http://codeigniter.com/

[7] ETSI. [Documento WWW]. Consultado Mayo de 2011: http:// http://www.etsi.org/

[8] García J. (2000). *Bluetooth.*

[9] Glazebrook, J. [Documento WWW]. Consultado Abril de 2012 en: http://teethgrinder.co.uk/open-flash-chart-2/

[10] Gracia, J. (7 de Mayo de 2005) *UML: Diagramas UML. ¿Qué es UML?* [Documento WWW]. Consultado Abril 2012 en: http://www.ingenierosoftware.com/analisisydiseno/uml.php

[11] Herrera, A. (04 de Octubre de 2004) *Modelamiento Básico de Datos con PostgreSQL*. [Documento WWW]. Consultado Abril de 2012 en: http://alvherre.cl/pgsql/modBasico/node3.html

[12] ITU. [Documento WWW]. Consultado Mayo 2011 en: http://www.itu.int/home/index.html

[13] Jormalainen S., Laine J. (30 de Noviembre de 1999) *Security in the WTLS* [Documento WWW]. Consultado en: http://www.tml.tkk.fi/Opinnot/Tik-110.501/1999/papers/wtls/

[14] Kostakos, V. *Using Bluetooth to capture passenger trips on public transport buses*. University of Madeira Human Computer Interaction Institute, Carnegie Mellon University.

[15] Longa N., Villavicencio C. (26 de Octubre de 2005) *Diseño e Implementación de un Punto de Acceso Informativo Inalámbrico Usando IrDA y Bluetooth.* Universidad Central de Venezuela.

[16] Microsoft. [Documento WWW]. Consultado Abril de 2012 en: http://msdn.microsoft.com/es-es/library/dd409377.aspx

[17] Nathan, E. y Alex (Sandy) Reality *Mining: Sensing Complex Social Systems*. Pentland, MIT Media Laboratory, Cambridge.

[18] Nathan, M. (8 de Septiembre de 2000). *Bluetooth Demystified*.

[19] Oracle Corporation. [Documento WWW]. Consultado Abril de 2012 en: http://www.mysql.com

[20] PyBluez Project. [Documento WWW]. consultado Abril de 2012 en: http://code.google.com/p/pybluez/

[21] Python Software Foundation. [Documento WWW]. Consultado Abril de 2012 en: http://www.python.org/

[22] Simpson, W. (Julio de 1994). *The Point-to-Point Protocol (PPP)*. RFC 1661.

[23] The Apache Software Foundation. [Documento WWW]. Consultado Abril de 2012 en: http://www.apache.org/

[24] The jQuery Foundation. [Documento WWW]. Consultado Abril de 2012 en: http://jquery.com/

[25] The PHP Group. [Documento WWW]. Consultado Abril de 2012 en: http://www.php.net/

[26] The Trirand Team. [Documento WWW]. Consultado Abril de 2012 en: http://www.trirand.com/blog/