



DE CONTROL DE LLAMADAS BASADO EN UNA RED BLUETOOTH

(Callings Control System Based in the Bluetooth Network)

M. A. Molina Vilchis*

Escuela Superior de Ingenier a Mec nica y El ctrica (ESIME) Culhuacan-IPN. M xico,
D.F. M xico

mamolinav@ipn.mx

E. Bracho Molina**

Universidad Aut noma Metropolitana (UAM) Unidad Iztapalapa. M xico, D.F. M xico

eleazarbm@hotmail.com

RESUMEN

Es com n que se provoquen interrupciones o interferencias por el uso indiscriminado de tel fonos celulares en recintos donde se realizan eventos acad micos, culturales o sociales; de ah  que surja la necesidad de controlar la recepci n o transmisi n de llamadas sin violar el derecho a la comunicaci n de las personas, y dado que los dispositivos de control de llamadas comerciales son costosos e inaccesibles para estas aplicaciones, se presenta en este art culo una aplicaci n basada en redes Bluetooth para este prop sito.

Palabras clave: Bluetooth, Ad hoc, Protocolo, MAC, PIN.

ABSTRACT

It is common interruptions or signal jamming occurs from careless use cell phones in academic, cultural or social halls, it comes a need to control reception or calling transmission, without infringe people communication laws, and due commercial calling control devices are inaccessible and expensive for these applications, because this article pretends to show an application based on Bluetooth networks.

Key words: Bluetooth, Ad hoc, Protocolo, MAC, PIN.

Introducci n

La importancia que han cobrado las Redes de  rea Personal, se basa en la popularidad de los dispositivos m viles tales como: los tel fonos celulares, PDA y computadoras port tiles; ya que permiten la comunicaci n eficiente en cualquier momento y lugar en un entorno personal, de tal manera que estos dispositivos se perciben como partes de un sistema de comunicaci n integral a lo que se le ha denominado computaci n pervasiva.



Este concepto novedoso se le atribuye a Mark Weiser, quien lo anticipó en sus escritos en 1988 cuando trabajaba para Xerox en el laboratorio de Palo Alto (PARC) en EUA.

Entre los dispositivos móviles inteligentes destacan los teléfonos móviles conocidos como teléfonos celulares que incorporan funcionalidades que hasta no hace mucho tiempo parecían futuristas, como la transmisión y recepción de archivos multimedia, juegos, reproducción de archivos MP3, correo electrónico, web, envío de mensajes de texto y fotografías, recepción de radio y televisión digitales, entre otros. Lo anterior ha hecho muy populares a estos dispositivos; tan sólo en México existen más de 65 millones de teléfonos celulares.

No obstante, el abuso indiscriminado de estos dispositivos ha creado serios problemas, sobre todo la recepción o transmisión de llamadas telefónicas en ambientes con restricciones de comunicación, pues generan interrupciones no deseadas. También en eventos culturales, ruedas de prensa, conferencias, presentaciones, entre otros, lo que ha provocado que existan prohibiciones de uso.

Para solucionar el problema anterior, en este artículo se presenta el desarrollo de una aplicación de una red Bluetooth para el control de comunicaciones telefónicas celulares en ambientes controlados. Se tratan aspectos técnicos de la red Bluetooth; se presenta el diseño de la aplicación, y se muestra su desarrollo con los resultados de las pruebas de operación.

Trabajo previo

La tecnología Bluetooth se ha incorporado de manera natural a estos dispositivos facilitando su interacción y comunicación, y se erige como el estándar de facto de las redes de área personal (PAN, Personal Area Network). Sin embargo, el origen de esta tecnología está relacionado estrechamente con las investigaciones en el campo de las comunicaciones inalámbricas (Cano J.C., 2008). Así, desde su origen en la década de los 70 hasta ahora, el interés por estas redes se ha visto progresivamente incrementado, convirtiéndola en la tecnología más popular de los últimos años.

Los recientes avances en la materia se han centrado en las redes Ad hoc. Este término hace referencia a redes flexibles inalámbricas sin infraestructura, donde los dispositivos se conectan de forma autónoma para establecer la comunicación y transferir información multimedia, sin que sea necesario usar equipos o dispositivos especializados, como puntos de acceso, repetidores o conmutadores.

Asimismo, deben poder adaptarse dinámicamente ante cambios continuos como: la posición de los dispositivos, la potencia de la señal, el tráfico de la red y la distribución de la carga. Siendo su principal reto la adaptación a los continuos e impredecibles cambios de topología.



Recientemente la tecnología Bluetooth se ha mostrado como una plataforma de soporte prometedora para estas redes, su principal ventaja es su habilidad para localizar, de forma transparente, dispositivos móviles cercanos, así como los servicios que ofrecen.

Bluetooth es el estándar impulsado por compañías líderes en comunicaciones tales como Ericsson, IBM, Intel, Nokia y Toshiba, entre otras; mismas que forman el Grupo de Interés Especial (Special Interest Group, SIG) Bluetooth desde mayo de 1998, (Muller- Nathan, 2002).

Definida dentro del estándar IEEE 802.15, referido a las redes inalámbricas tipo personal, la conexión inalámbrica Bluetooth opera en el rango de radiofrecuencia de los 2,4 GHz (2,400 a 2,485 GHz) (Mata y Ramírez, 2008), y no requiere licencia de uso en ningún lugar del mundo. Con una banda de guarda de 2 MHz a 3,5 MHz para cumplir con las regulaciones internacionales.

Transmite en espectro disperso, con salto de frecuencia, en dúplex y hasta 1,600 saltos/s. La señal salta entre 79 frecuencias en intervalos de 1 MHz para tener un alto grado de tolerancia a las interferencias y obtener comunicaciones robustas. Además establece comunicaciones punto a punto y multipunto, donde un dispositivo puede establecer de forma simultánea hasta siete canales de comunicación a la vez con un solo radio de cobertura.

Bluetooth transmite a una tasa de 1 Mbps en su funcionamiento básico, y de 2 a 3 Mbps en modo mejorado (Bluetooth 2.0). Utiliza modulación Gaussiana por Desplazamiento de Frecuencia (Gaussian Frequency Shift Keying, GFSK) para el modo básico, mientras que para el modo mejorado utiliza la Modulación Diferencial por Desplazamiento de Fase en Cuadratura (Defferential Quadrature Phase-Shift Keying, DQPSK) y la Modulación Diferencial por Desplazamiento de Fase (Defferential Phase-Shift Keying, 8-DPSK).

Para lograr que la comunicación sea en dúplex, divide el tiempo de transmisión en ranuras por medio de la técnica Dúplex por División de Tiempo (Time-Division Duplex, TDD). Asimismo logra una cobertura de 100 metros para los dispositivos de clase 1, de 20 metros para los de clase 2 y de 10 metros para los de clase 3, con potencias de, 1 mW, 10 mW y 100 mW respectivamente. Siendo los dispositivos de clase 3 los mayormente usados.

Bluetooth permite conectarse casi con cualquier dispositivo compatible que se halle en las proximidades; cuando los dispositivos se conectan, se forma una piconet, en ésta los dispositivos comparten el mismo canal y adquieren dos roles distintos: maestro o esclavo. En cada piconet solamente puede existir un maestro y un máximo de siete esclavos, estos últimos no pueden establecer enlaces entre sí, por lo que todo el tráfico es enviado al maestro.



Para que el dispositivo maestro logre establecer la comunicaci n, en principio, deber  ejecutar el protocolo de descubrimiento.  ste env a una se al en multidifusi n para solicitar, b sicamente, a los dispositivos cercanos, su direcci n MAC (Medium Access Control) y el PIN (Personal Identifier Number). De esta manera logra registrar a los dispositivos que se encuentran en estado activo, sin importar el servicio que est n prestando.

Se han propuesto diversas aplicaciones basadas en Bluetooth para dar soluci n a diferentes problemas del quehacer humano. As  en (Pece, 2008), se presenta el desarrollo de un sistema de informaci n contextual para terminales m viles, el cual sirve como mecanismo de localizaci n en ambientes tales como: museos, monumentos, entre otros; como apoyo a la red de transporte p blico en Italia, para llevar a cabo tareas de diagn stico inal mbrico y mantenimiento preventivo en los autobuses (Torino, 2008).

Otra aplicaci n interesante es una alarma como bot n de p nico, que env a un mensaje de texto desde un tel fono m vil a un n mero de emergencia, en caso de que el portador del m vil sufra alg n percance en la v a p blica (Kell, 2008).

Mientras que en el campo de la medicina, en el Centro Noruego para la Telemedicina, se ha desarrollado una aplicaci n que hace posible vigilar el nivel de glucosa en la sangre de manera remota por medio de un tel fono celular. Esta aplicaci n est  dirigida al auto cuidado de la diabetes en pacientes menores de edad. Otras aplicaciones se centran en el mantenimiento y operaci n de estaciones de bombeo de agua en  reas urbanas para el control del agua.

No obstante, se han desarrollado muy pocas aplicaciones para el monitoreo, identificaci n y rastreo de dispositivos m viles y sus servicios, uno de ellos es el BlueSweep de AirMagnet Inc. que facilita a los usuarios de dispositivos m viles localizarlos e identificar los servicios en los que se encuentran trabajando en tiempo real, con la finalidad de identificar las posibles amenazas de seguridad y llevar un registro de incidentes.

Dise o de la aplicaci n

A. Antecedentes del problema:

Entre las soluciones propuestas para resolver el problema del control de tel fonos celulares en ambientes con restricciones de uso, se pueden mencionar los sistemas de bloqueo de llamadas que obstruyen la recepci n en un radio de 30 metros aproximadamente, y que son utilizados principalmente para los sistemas penitenciarios.

Entre los equipos que se ofertan se puede citar el Radio Capsule SRC-300 (Gonz lez y Palacios, 2008), y el Portable Palm Phone Jammer (Alegr a, 2007), que



cuentan con un alcance aproximado de 10 a 50 metros, los cuales pueden trabajar en frecuencias de 800 a 900 MHz y de 1930 a 1990 MHz, con un consumo de 5 volts. No obstante, esta tecnolog a es muy costosa y est  limitada a estos centros de reclusi n, por lo que para otros fines es inaccesible.

Si partimos del hecho de que muchos de los tel fonos celulares incorporan la tecnolog a Bluetooth, entonces es posible plantear una soluci n para el problema que nos ocupa, por medio de una aplicaci n que permita detectar o identificar distintas marcas de tel fonos celulares activos y sus servicios, de una forma r pida, autom tica y en un radio de hasta 100 metros, utilizando su propio protocolo de descubrimiento de dispositivos.

Respetando el derecho que todos los seres humanos tienen para comunicarse, esta aplicaci n no debe contraponerse a este derecho, por lo que se establece como requisito en su dise o las siguientes caracter sticas:

1. Una vez detectados, se transmite una aplicaci n que al ser aceptada por el usuario, despliega un men  de texto en donde se pueden visualizar diferentes opciones.
2. La consola de control inicia la comunicaci n por medio del env o de mensajes de texto a los tel fonos celulares que est n dentro de su alcance.
3. Los tel fonos celulares deben enviar su identificaci n, es decir su direcci n MAC, para que pueda descargarse la aplicaci n desde la consola.

B. Contexto de operaci n:

El sistema propuesto utiliza una consola de control para ser instalada, ya sea en la cabina de un auditorio, sala de prensa, sal n de convenciones o cualquier otro sitio que requiera el control de llamadas.

Esta consola es una computadora de escritorio o port til, configurada como maestro, que env a se ales de descubrimiento de los tel fonos celulares con un alcance de hasta 40 metros (Toshiba, 2008). Si el recinto sobrepasa esta distancia, es posible usar equipos repetidores para establecer varias conexiones Bluetooth simult neas a fin de alcanzar hasta 200 metros.

C. Interfaz de usuario:

Los requisitos para el dise o de la interfaz de la aplicaci n del lado de la consola son: que cuente con la tecnolog a Bluetooth, con una tarjeta inal brica y que el sistema operativo sea compatible con el software utilizado.

La interfaz de los teléfonos celulares debe ser diseñada de tal manera que sea compatible con las características de despliegue y operación de la mayoría de las marcas de teléfonos, es decir, en modo de texto y con selección por menú, utilizando los botones de desplazamiento convencionales. Ver Fig. 1.



Fig.1. Interfaz del Usuario.

D. Diseño de los objetos del sistema:

El diseño de aplicaciones para teléfonos celulares es con base a la programación orientada a objetos (POO), de esta manera la aplicación se basa en dos clases: Listener y Auditorio. La primera es esencial para el funcionamiento de la tecnología Bluetooth y la segunda es la aplicación en sí.

La función de Listener, propia de la tecnología Bluetooth, es detectar de forma automática los teléfonos celulares activos, y confirmar su presencia para luego mostrarlos en la pantalla. Además de la reservación de los servicios de operación de la consola como dispositivo maestro.

Esta clase tiene como atributo Vector Disp_Encontrados para poder realizar la comunicación automáticamente, y utiliza los métodos relacionados con los servicios como son: 1. deviceDiscovered, que realiza la función de detectar los teléfonos celulares con tecnología Bluetooth; 2. serviceSearchCompleted, que recibe una afirmación a la búsqueda de servicios; y 3. servicesDiscovered, para confirmar la localización de servicios y mostrar en pantalla (Borches y Juzgado, 2008).

La clase llamada Auditorio es usada para ejecutar la aplicación, en ésta se incluyen los atributos y servicios que tienen que ver con la operación y el despliegue de mensajes presentados en la interfaz del usuario del teléfono celular.

E. Funcionamiento esperado:

En la Fig.2., se muestra el protocolo para la instalación de la aplicación. La computadora de control inicia la comunicación enviando un mensaje en multidifusión

(Req_Inquiry) a los teléfonos celulares para su descubrimiento y con ello identifica su nombre y dirección física. En consecuencia, los teléfonos celulares envían el mensaje Res_MAC_Add como respuesta al requerimiento, remitiendo su dirección física y nombre. A continuación la consola de control envía el mensaje Req_DSP para decidir con qué dispositivo va a establecer la conexión y el servicio que se dispone a usar.

El teléfono celular expide la Autorización del PIN, para que el dispositivo responda con los diferentes servicios que puede ofrecer. Una vez recibido el PIN se abre la conexión con el mensaje Abrir Conexión. Esto permitirá enviar la aplicación al dispositivo seleccionado. Finalmente sólo es necesario que el usuario instale la aplicación y haga uso de ella.

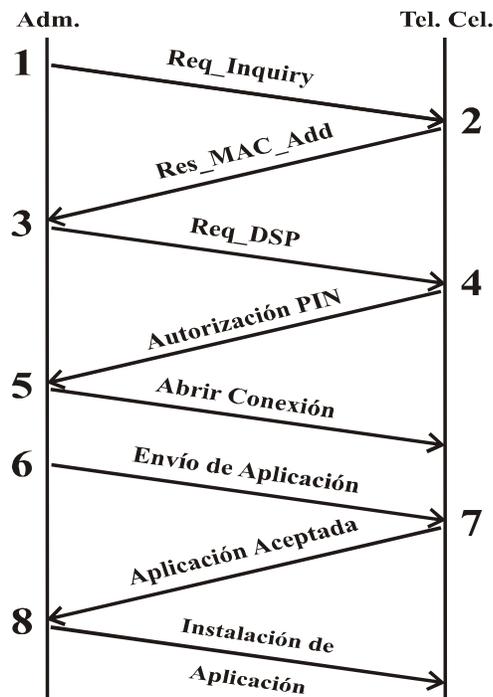


Fig. 2. Protocolo para la instalación de la aplicación.

Desarrollo de la aplicación

A. Plataforma de desarrollo:

El desarrollo de aplicaciones para teléfonos celulares se basa en la plataforma Java con: Java 2 Micro Edition (J2ME), que utiliza un subconjunto reducido de componentes Java SE, máquinas virtuales y la interfaz de programación de aplicaciones (Application Programming Interface, API), Sun Microsystems, Java JDK (Java Development Kit) para aplicaciones de teléfonos celulares y NetBeans.



B. Funciones:

Para el desarrollo de la aplicaci n se definieron tres funciones: 1. Auditorio constructor, 2. Inicializes the application y 3. Performs an action assigned to the Mobile Device; mostradas en la Fig.3.

```
* The Auditorio constructor.
*/
public Auditorio() { //Constructor
}
//<editor-fold defaultstate="collapsed"desc="Generated Method: initialize">
/**
 * Initalizes the application.
 */
private void initialize() { //Comenzar a Ejecutar el Programa
    // write pre-initialize user code here
}
//</editor-fold>

//<editor-fold defaultstate="collapsed"desc="Generated Method: startMIDlet">
/**
 * Performs an action assigned to the Mobile Device - MIDlet Started point.
 */
public void startMIDlet() { //Ejecuci n de Aplicaci n
    // write pre-action user code here
    switchDisplayable(getAlertaBienvenida(), getMenu());
    // Muestra la alerta Bienvenido y despu s el Men  en Pantalla
}
//</editor-fold>
```

Fig.3. Funciones de Inicializaci n.

Adicionalmente se definieron otras funciones: switchDisplayable para el despliegue de mensajes en la pantalla del celular, ver Fig.4.; commandAction que especifica la aplicaci n que se env a a los dispositivos esclavos, ver Fig.5.; y exitMIDlet mostrada en la Fig.6, para la salida definitiva de la aplicaci n.

```
public void switchDisplayable(Alert alert, Displayable nextDisplayable) {
    // write pre-switch user code here
    Display display = getDisplay(); //Se asigna la variable de tipo Display
    if (alert == null) { //Se compara si Alert es de tipo nulo
        display.setCurrent(nextDisplayable); // Cambia de pantalla
    } else { // Si no se cumple la condici n vuelve a cambiar de pantalla
        display.setCurrent(alert, nextDisplayable);
    }
    // write post-switch user code here
}
//</editor-fold>
```

Fig.4. Funci n switchDisplayable().



```
public void commandAction(Command command, Displayable displayable) {
    if (displayable == Alerta) {
        if (command == ComandoRegresaAlerta) {
            switchDisplayable(null, getMenu());
        } else if (command == ComandoSalirAlerta) {
            exitMIDlet();
        }
    } else if (displayable == Menu) {
        if (command == ComandoContinuar) {
            int Seleccion;
            Seleccion = GrupoElementos.getSelectedIndex();

            //La variable Seleccion contiene el  ndice seleccionado del grupo de
            elementos switch (Seleccion)
            {
                //Selecciona la opci n alerta.
                case 0:
                {
                    switchDisplayable(null, getAlerta());
                    // Funci n de NetBeans que permite vibrar al seleccionar la opci n
                    javax.microedition.lcdui.Display.getDisplay(this).vibrate(1000);
                    break;
                }
                //Selecciona la opci n Recomendaci n.
                case 1:
                {
                    switchDisplayable(null, getRecomendacion());
                    // Funci n de NetBeans que permite vibrar al seleccionar la opci n
                    javax.microedition.lcdui.Display.getDisplay(this).vibrate(1000);
                    break;
                }
                //Selecciona la opci n Precauci n.
                case 2:
                {
                    switchDisplayable(null, getPrecauci n());
                    // Funci n de NetBeans que permite vibrar al seleccionar la opci n
                    javax.microedition.lcdui.Display.getDisplay(this).vibrate(1000);
                    break;
                }
                //Selecciona la opci n Salir.
                case 3:
                {
                    exitMIDlet();
                }
            }
        } else if (command == ComandoSalir) {
            exitMIDlet();
        }
    } else if (displayable == Precauci n) {
        if (command == ComandoRegresaPrecaucion) {
            switchDisplayable(null, getMenu());
        } else if (command == ComandoSalePrecaucion) {
            exitMIDlet();
        }
    } else if (displayable == Recomendacion) {
        if (command == ComandoRegresaRecomendacion) {
            switchDisplayable(null, getMenu());
        } else if (command == ComandoSalirRecomendacion) {
            exitMIDlet();
        }
    }
}
//</editor-fold>
```

Fig.5. Funci n commandAction().

```
/**
 * Exits MIDlet.
 */
public void exitMIDlet() { //Sale del MIDlet
    switchDisplayable (null, null); //Coloca en nulo el switch
    destroyApp(true); //Recibe verdadero
    notifyDestroyed(); //Notifica la destrucción de la aplicación
}
/**
```

Fig.6. Función exitMIDlet().

En la Fig. 7. se especifican las Funciones para Pausar y Salir del MIDlet, con tres opciones: 1. Para el reinicio de la aplicación, después de que ha sido pausada cuando alguna acción del teléfono tiene prioridad, como por ejemplo, la entrada de una llamada telefónica o un mensaje de texto. 2. Pausa del MIDlet, da un valor verdadero para que siempre que suceda alguna de las acciones antes mencionadas, automáticamente realice le pausa y no exista conflicto con la acción prioritaria. 3. Destrucción de la aplicación. Es decir, borra la aplicación en el dispositivo.

```
public void startApp() { // Reinicia la aplicación pausada
    if (midletPaused) { resumeMIDlet ();}
    else {
        initialize ();
        startMIDlet ();
    }
    midletPaused = false;
}
public void pauseApp() { //Pausa la aplicación cuando es necesario
    midletPaused = true;
}
/**
 * Hace el llamado para poner fin al MIDlet.
 */
public void destroyApp(boolean unconditional) {
}
}
```

Fig. 7. Funciones para Pausar y Salir del MIDlet.

C. Pruebas:

Para efectos de prueba, se utilizó un auditorio común durante la realización de un evento cultural. El recinto fue de un área de 213 m² y se propusieron tres tipos de prueba: 1. Descubrimiento, 2. Transmisión y 3. Compatibilidad.

1. Prueba de Descubrimiento: Las primeras pruebas de alcance y descubrimiento tienen como finalidad que la consola de control realice la detección y reconocimiento

de diferentes marcas de teléfonos celulares. Para iniciar la búsqueda, la consola envía una señal en multidifusión de localización, a la que los teléfonos celulares deben responder con su dirección física (MAC) y su NIP. Ver fig. 8.

2. Pruebas de Transmisión: A continuación la consola hace la transferencia de la aplicación directamente a los teléfonos reconocidos en modo punto a punto. Sin embargo, hay que considerar que la transferencia puede ser interrumpida por diferentes causas: 1. El rechazo de la aplicación por parte del usuario, 2. Pérdida en la conexión debido a ese rechazo, 3. Insuficiencia del alcance de la señal, o 4. Interferencias con otras señales de radio de otro tipo de dispositivos. Para esos casos, la aplicación mostró en pantalla un mensaje de error.

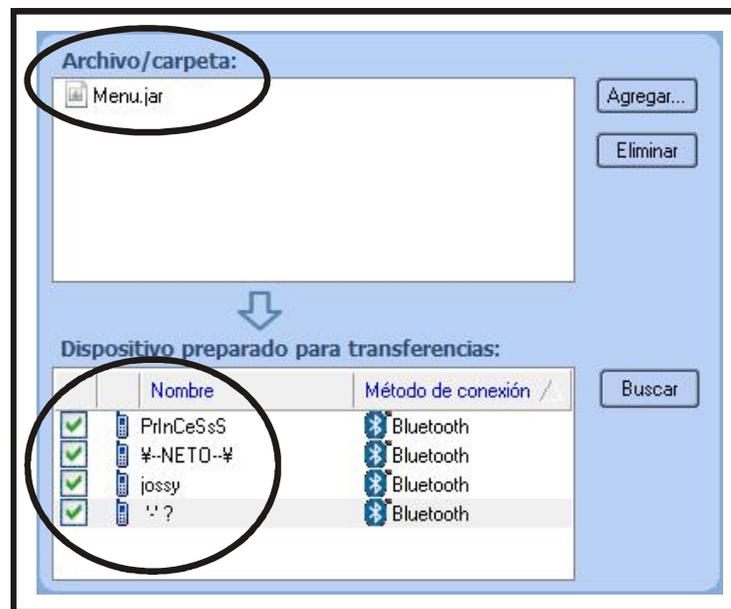


Fig.8 Teléfonos Celulares Seleccionados para la Transmisión de la Aplicación.

3. Pruebas de Compatibilidad: Tomando en cuenta que cada fabricante tiene instalados diferentes sistemas operativos, plataformas y versiones de Bluetooth. Con esta prueba se verificó si la aplicación es compatible con los diferentes fabricantes de teléfonos celulares. En este caso se probaron diez marcas distintas, entre ellas: Sony Ericsson, Nokia, LG, Samsung y Motorola, obteniendo resultados satisfactorios en la ejecución de la aplicación, para la mayoría de los casos.

Resultados

Para la obtención de resultados de las pruebas se empleó el programa SysNucleo USBTrace. Los resultados obtenidos de la primera prueba se muestran en la Fig.9., donde se pueden observar del lado izquierdo de la imagen los dispositivos

detectados, mientras a la derecha se muestran los detalles del registro, entre ellos, el más importante para nuestros propósitos es el estado de la conexión.

Seq	Type	Time	Request	I/O	En...	Device Object	IRP	Status
0	STA...	0.000000	START OF LOG					
1	URB	1.505278	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x8455CE...	STATUS_SUCCESS
2	URB	1.505282	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x8455CE...	STATUS_SUCCESS
3	URB	1.505313	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x8455CE...	STATUS_PENDING
4	URB	1.545280	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x8455CE...	STATUS_SUCCESS
5	URB	1.545310	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x8447BB...	STATUS_SUCCESS
6	URB	1.545319	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x8447BB...	STATUS_SUCCESS
7	URB	1.545381	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x8447BB...	STATUS_PENDING
8	URB	1.930284	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x8447BB...	STATUS_SUCCESS
9	URB	1.930332	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x8457CE...	STATUS_SUCCESS
10	URB	1.930348	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x8457CE...	STATUS_SUCCESS
11	URB	1.930409	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x8457CE...	STATUS_PENDING
12	URB	2.960261	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x8457CE...	STATUS_SUCCESS
13	URB	2.960469	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x84AA6...	STATUS_SUCCESS
14	URB	2.960477	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x84AA6...	STATUS_SUCCESS
15	URB	2.960542	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x84AA6...	STATUS_PENDING
16	URB	3.009848	CLASS_DEVICE	OUT	0	\Device\USBPD...	0x8457CE...	STATUS_SUCCESS

Fig.9. Resultados de la prueba de detección.

Los resultados de la segunda prueba se muestran en la Fig.10., en ésta se observa el éxito o interrupción de la transferencia. Los teléfonos celulares que no pudieron recibir la aplicación fueron los modelos SGH-J700 de Samsung, MG300 y MG800c de LG.

Seq	Type	Time	Request	I/O	En...	Device Object	IRP	Status	Buffer Snippet
0	STA...	0.000000	START OF LOG						
1	URB	0.904112	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x84AB6...	STATUS_SUCCESS	
2	URB	0.904121	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x84AB6...	STATUS_SUCCESS	
3	URB	0.904186	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x84AB6...	STATUS_PENDING	
4	URB	0.904205	BULK_OR_INTERRUPT_TRA...	OUT	82	\Device\USBPD...	0x8D768E...	STATUS_SUCCESS	
5	URB	0.904212	BULK_OR_INTERRUPT_TRA...	OUT	82	\Device\USBPD...	0x8D768E...	STATUS_SUCCESS	
6	URB	0.904263	BULK_OR_INTERRUPT_TRA...	IN	82	\Device\USBPD...	0x8D768E...	STATUS_PENDING	
7	URB	0.909360	CLASS_DEVICE	OUT	0	\Device\USBPD...	0x844D3E...	STATUS_SUCCESS	0C 08 02 2A 00
8	URB	0.909364	CLASS_DEVICE	OUT	0	\Device\USBPD...	0x844D3E...	STATUS_SUCCESS	0C 08 02 2A 00
9	URB	0.909401	CONTROL_TRANSFER	IN	0	\Device\USBPD...	0x844D3E...	STATUS_PENDING	
10	URB	0.914100	CONTROL_TRANSFER	IN	0	\Device\USBPD...	0x844D3E...	STATUS_SUCCESS	
11	URB	0.917072	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x84AB6...	STATUS_SUCCESS	0E 08 01 0C 08 00 2A ...
12	URB	0.917098	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x84AB6...	STATUS_SUCCESS	
13	URB	0.917105	BULK_OR_INTERRUPT_TRA...	OUT	81	\Device\USBPD...	0x84AB6...	STATUS_SUCCESS	
14	URB	0.917146	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x84AB6...	STATUS_PENDING	
15	URB	0.924966	CLASS_DEVICE	OUT	0	\Device\USBPD...	0x8454EC...	STATUS_SUCCESS	0D 08 04 2A 00 05 00
16	URB	0.924979	CLASS_DEVICE	OUT	0	\Device\USBPD...	0x8454EC...	STATUS_SUCCESS	0D 08 04 2A 00 05 00
17	URB	0.925100	CONTROL_TRANSFER	IN	0	\Device\USBPD...	0x8454EC...	STATUS_PENDING	
18	URB	0.933085	BULK_OR_INTERRUPT_TRA...	IN	81	\Device\USBPD...	0x84AB6...	STATUS_SUCCESS	1B 03 2A 00 05

Fig.10. Resultados de la transferencia de la aplicación.



Conclusiones

Los tel fonos celulares han facilitado en gran medida nuestra vida diaria, pero tambi n se han convertido en un problema, dado el abuso que se ha hecho de estos, sobre todo en aquellos ambientes donde est  restringido su uso. Las soluciones en hardware inhiben la recepci n y transmisi n de llamadas, y atentan contra el derecho a la comunicaci n.

Una alternativa es una aplicaci n de software basada en la tecnolog a Bluetooth, para la detecci n de tel fonos celulares activos y sus servicios, es decir, que no s lo detecte la actividad en la recepci n y transmisi n de llamadas, sino en otras facilidades que requieren de autorizaci n, y que adem s pueda adecuarse a las restricciones impuestas en distintos ambientes. Siendo la compatibilidad con distintas marcas de tel fonos celulares, el costo y la ubicuidad esenciales en su dise o.

Como trabajo a futuro esta aplicaci n puede extenderse a otras restricciones que pudieran estar relacionadas con el uso de c maras fotogr ficas transmisi n de mensajes de texto o grabaciones de videos sin autorizaci n.

Agradecimientos

MAMV agradece el apoyo econ mico recibido del programa COFAA del IPN y la ayuda recibida por la SIP al proyecto de investigaci n 20090277.

Referencias Bibliogr ficas

- ALEGR A, A. (2007). **Bloquea la se al de celular con Portable Palm Jammer**. Documento en l nea. Disponible en: <http://tecnoday.net/2007/09/29/bloquea-la-se al-de-celular-con-portable-pal-phone-jammer>. Consultado en marzo de 2008.
- BORCHES JUZGADO, C. (2008). **Java 2 Microedition**. Documento en l nea. Disponible en: www.it.uc3m.es/celeste/.../bluetooth/EstudioTecnologico1_0.pdf. Consultado en 2008, de Soporte Bluetooth, Versi n 1.0, Universidad Carlos III de Madrid.
- CANO J.C. (2008). **Redes inal mbricas Ad Hoc como tecnolog a de soporte para la computaci n ubicua**. Documento en l nea. Disponible en: <http://www.grc.upv.es/calafate/download/main-novatica.pdf>. Consultado el 20 de octubre de 2009, de Departamento de Inform tica de Sistemas y Computadores, Universidad Polit cnica de Valencia.
- GONZ LEZ PALACIOS, G. (2008). **Bloqueo de Se ales Celulares, Inform tica Jur dica**. Documento en l nea. Disponible en: <http://www.bloqueocelular.blogspot.com>. Consultado en marzo de 2008, de Universidad Aut noma Latinoamericana, Facultad de Derecho.



PECE, J. P. (2008). **Sistema de Informaci n Contextual para Terminales M viles Basado en Tecnolog a Bluetooth**. Documento en l nea. Disponible en: http://w3.iec.csic.es/URSI/articulos_modernos/articulos_gandia_2005/articulos/NC1/394.pdf. Departamento de Electr nica y Sistemas , Universidad de A. Coru a. Consultado en marzo de 2009.

TORINO, C. R. (2008). **ATM Spa-Reference Application**. Documento en l nea. Disponible en: <http://www.spanish.bluetooth.com/NR/rdonlyres/23.5EC2BC-644F-4806-85ED-DD1666FEF43B/0/atm.pdf>. Consultado el 16 de octubre de 2008.

KELL, O. (2008). **Securecom Technologies Ltd**. Documento en l nea. Disponible en: <http://www.spanish.bluetooth.com/NR/rdonlyres/EC33B6D5-A831-4D3B-886B-48EDEF-4660D8/0/securecom.pdf>. Consultado en abril de 2008, de Head of business development Bluetooth panic button potential to save lives.

MATA RAM REZ, M. (2008). **Gestiopolis**. Documento en l nea. Disponible en: http://www.gestiopolis.com/canales_8/gar/tecnologia-bluetooth.htm. Consultado el 10 de marzo de 2009, de Tecnolog a Bluetooth.

MULLER NATHAN, J. (2002). **Tecnolog a Bluetooth**. Madrid: Interamericana de Espa a.

Otros documentos

JAVA. (S.F.). **Conozca m s sobre la tecnolog a Java**. Documento en l nea. Disponible en: <http://java.com/es/about/>. Consultado en octubre de 2008.

JAVA. (2008). **P gina oficial de Java para Windows**. Documento en l nea. Disponible en: http://java.com/es/download/windows_ie.jsp?locate=es&host=java.com:80&bhop=1. Consultado en octubre de 2008.

NETBEANS. (S.F.). **NetBeans IDE 61**. Documento en l nea. Disponible en: <http://www.netbeans.org/index.html>. Consultado en octubre de 2008.

NST. (2008). **The Norwegian Center for Telemedicine**. Documento en l nea. Disponible en: <http://spanish.bluetooth.com/NR/rdonlyres/BA25A6D5-15F0-437F-84CI-AABA58818BFD/0/Diabetes.pdf>. Consultado en diciembre de 2008.

SYSNUCLES. (2008). **USB Trace Tour: User Interface**. Documento en l nea. Disponible en: http://www.sysnucleus.com/usbtrace_tour.html. Consultado el octubre de 2008.

SYSNUCLEUS. (2008). **Viewing captured data**. Documento en l nea. Disponible en: http://www.sysnucleus.com/usbtrace_tour3.html. Consultado en octubre de 2008.

TOSHIBA. (2008). **Manual de usuario**, serie A210-A215.