

Autorizada la entrega del proyecto al alumno:

JAVIER A. CUENCA RETANA

El Director del Proyecto

DAVID CONTRERAS BÁRCENA

Fdo.: Fecha: /9/2010

Vº Bº del Coordinador de Proyectos

DAVID CONTRERAS BÁRCENA

Fdo.: Fecha: /9/2010



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO TÉCNICO EN INFORMÁTICA DE SISTEMAS

PROYECTO FIN DE CARRERA
SISTEMA DE POSICIONAMIENTO
EN INTERIORES MEDIANTE
BALIZAS BLUETOOTH

AUTOR: JAVIER A. CUENCA RETANA

MADRID, Septiembre 2010

AGRADECIMIENTOS

Gracias a mis Padres, a los que se lo debo todo.

Sin ellos nada de esto habría sido posible.

*Gracias mis Hermanos, por sus
consejos, paciencia y comprensión.*

*Gracias a mis Amigos, por estar siempre ahí
cuando los necesito y encima soportarme.*

*Gracias todos los Profesores y Compañeros que
he tenido en mi vida, en especial a los del ICAI.*

*Gracias a Mónica, por estar siempre a mi lado,
apoyarme, esperarme y ser tan especial.*

A todos, muchas gracias.

RESUMEN

El presente proyecto se ha planteado como la búsqueda de una solución al problema que plantean los sistemas de posicionamiento en interiores.

Enfocado desde la perspectiva ofrecida por los cada vez más extendidos entornos inteligentes, aporta un nuevo método que integra totalmente el uso de teléfonos móviles en esos entornos.

Se realiza un estudio sobre la tecnología Bluetooth, proponiendo como balizas pasivas de bajo coste auriculares Bluetooth. En este caso particular se utiliza este tipo de dispositivo gracias a su reducido precio, pudiéndose utilizar cualquier otro dispositivo que integre la tecnología Bluetooth y sea capaz de cumplir con los servicios de descubrimiento.

Este proyecto tiene un carácter innovador al plantear una lógica de posicionamiento mediante un sistema no distribuido, presente de forma activa en un dispositivo móvil localizador, mediante el desarrollo de un MIDlet a través del estudio de la tecnología Java ME. El sistema obtenido (denominado BlueBrowse), mediante los datos de las balizas presentes en un archivo que forma parte de la aplicación Java, filtra qué dispositivos Bluetooth debe encontrar, y muestra en tiempo real información de baliza localizada y su posición exacta en el entorno.

Finalmente se describe la transferencia tecnológica del sistema efectuada mediante la realización de pruebas lógicas y físicas, cuyo éxito reflejan la viabilidad del método aportado.

ABSTRACT

This project has been conceived as the search for a solution to the problem posed by indoor positioning systems.

Approached from the perspective offered by the increasingly widespread intelligent environments, provides a new method that fully integrates mobile phone use in these environments.

It shows a study on Bluetooth technology, proposing as a low-cost passive tags Bluetooth headset. In this particular case, this type of device is used due to its reduced price, but it can be used any other device that integrates Bluetooth technology, capable of complying with discovery services.

This innovative project is to raise a logic of positioning through a non-distributed system, actively present in a mobile device locator, by developing a MIDlet through the study of Java ME technology. The resulting system called BlueBrowse, by using the data from the tags present in a file included on the Java application, filters what Bluetooth devices must be found and shows in real-time the beacon's information and his exact position in the vicinity.

Finally, it describes the technological transfer of the system completed by logical and physical tests, whose success reflects this project's viability.

ÍNDICE

1.	INTRODUCCIÓN.....	2
1.1	MOTIVACIÓN.....	2
1.2	OBJETIVOS.....	3
1.3	METODOLOGÍA.....	4
1.4	RECURSOS UTILIZADOS.....	6
1.4.1	Software.....	6
1.4.2	Hardware.....	7
2.	ESTADO DEL ARTE.....	9
2.1.	ARQUITECTURAS DISTRIBUIDAS.....	9
2.1.1	<i>Ejemplos distribuidos.....</i>	<i>9</i>
2.1.1.1	El sistema Bat.....	10
2.1.1.2	Symeo.....	11
2.1.1.3	Cricket.....	12
2.1.1.4	iHAP.....	13
2.2	ARQUITECTURAS CENTRALIZADAS.....	14
2.2.1	<i>Ejemplos centralizados.....</i>	<i>14</i>
2.2.1.1	Método I.....	15
2.2.1.2	Método II.....	16
2.2.1.3	Método III.....	17
2.2.1.4	Método IV.....	18
2.3	MÉTODOS DE CÁLCULO DE LA POSICIÓN.....	19
2.3.1	<i>División del espacio en celdas.....</i>	<i>19</i>
2.3.2	<i>Multilateración de magnitudes.....</i>	<i>19</i>
3.	ESTUDIO DE TECNOLOGÍAS.....	21
3.1	INTRODUCCIÓN.....	21
3.2	BLUETOOTH.....	22
3.2.1	<i>Arquitectura.....</i>	<i>23</i>
3.2.2	<i>El modelo cliente-servidor.....</i>	<i>24</i>
3.3	JAVA ME.....	25
3.3.1	<i>Configuraciones.....</i>	<i>26</i>
3.3.2	<i>Perfiles.....</i>	<i>27</i>
3.3.3	<i>JSR-82.....</i>	<i>28</i>
3.3.4	<i>Descubrimiento de dispositivos Bluetooth en Java ME.....</i>	<i>30</i>

4. PROPUESTA DE ENTORNO INTELIGENTE	33
4.1 INTRODUCCIÓN.....	33
4.2 ENTORNO INTELIGENTE SIMULADO.	34
4.3 LAS BALIZAS (ELEMENTO PASIVO).....	36
4.4 EL DISPOSITIVO MÓVIL (ELEMENTO ACTIVO).	40
5. DESARROLLO DEL SISTEMA.....	42
5.1 ANÁLISIS DE REQUISITOS.....	42
5.2 DISEÑO E IMPLEMENTACIÓN.	44
5.2.1 Módulo principal “MainMIDlet”	46
5.2.2 Módulo lector “FileReader”	47
5.2.3 Clase “PDI”	49
5.2.4 Módulo buscador “BTFinder”	50
5.2.5 Módulo pantalla inicial “FirstDisplay”	52
5.2.6 Módulo pantalla de información “InfoDisplay”	53
6. RESULTADOS OBTENIDOS.....	55
6.1 PRUEBAS LÓGICAS.	55
6.2 PRUEBAS FÍSICAS.....	60
7. CONCLUSIONES Y TRABAJOS FUTUROS.....	63
8. BIBLIOGRAFÍA.....	66
ANEXO A - MANUAL DE INSTALACIÓN.	69
ANEXO B - MANUAL DE EJECUCIÓN.	71
ANEXO C - PLANIFICACIÓN.....	76
ANEXO D - VALORACIÓN ECONÓMICA.....	78

1. INTRODUCCIÓN

1. Introducción.

1.1 Motivación.

La evolución y progreso del mundo de las nuevas tecnologías está dirigida hacia los denominados entornos inteligentes. A partir de la integración de los conceptos de tecnología, informática y el entorno en el que vivimos, se ha originado una nueva y única idea en la cual no se podrá diferenciar los elementos que la forman.

Estos llamados “ambientes inteligentes” detectan nuestra presencia e interaccionan con nosotros en función de nuestro comportamiento, a través de gestos, voz, movimiento y contexto. En la actualidad, sus posibilidades de aplicación son enormes por lo que no serán pocos los campos que se verán beneficiados en un futuro próximo.

El posicionamiento de interiores es uno de estos ámbitos, pues aún no existen técnicas aceptadas y extendidas para la resolución de este problema.

Así mismo, en el presente tampoco son muchas las aplicaciones que permitan la interacción espontánea entre los usuarios de dispositivos móviles y los entornos inteligentes.

Es en estos dos hechos donde reside la principal motivación para la realización de este proyecto: la implementación de una alternativa al posicionamiento en espacios cerrados, mediante el desarrollo de una aplicación para dispositivos móviles que admita la interacción con un entorno inteligente realizado ad hoc, formado por balizas pasivas.

1.2 Objetivos.

Son varias las actividades necesarias para llevar a cabo adecuadamente este sistema. La documentación es una de ellas, pues es indispensable obtener unas nociones óptimas sobre los conceptos que se van manejar. Seguidamente se procederá a la aplicación práctica de dichos conocimientos para obtener un sistema que, finalmente pueda ser probado en el mundo real mediante una singular transferencia tecnológica. Por tanto, los objetivos marcados para llevar a cabo este proyecto son los siguientes:

- ❖ Estudio del estado del arte sobre los distintos tipos de posicionamiento en interiores existentes en la actualidad, y los entornos inteligentes que utilizan Bluetooth.
- ❖ Estudio sobre la tecnología Bluetooth.
- ❖ Estudio sobre J2ME y su manera de trabajar con Bluetooth.
- ❖ Desarrollo de una aplicación para móviles que permita posicionamiento en interiores mediante balizas pasivas.
- ❖ Proposición un entorno inteligente en el cual se integre la aplicación a desarrollar.
- ❖ Efectuar la implementación del sistema en un caso práctico, ejecutando la aplicación en un dispositivo móvil dentro de un entorno inteligente real.

1.3 Metodología.

Desde un comienzo se barajó la posibilidad de realizar el proyecto desde el enfoque de un paradigma de desarrollo ágil del software. Estos procesos ágiles de desarrollo se caracterizan por promover iteraciones a lo largo de todo el ciclo de vida del proyecto.

- De todas las metodologías ágiles existentes, se optó por seguir la programación extrema, aunque no fue posible en todos sus aspectos debido al carácter individual de la realización de un proyecto de fin de carrera. Las características compartidas por metodología adoptada con la programación extrema original [PROG10] fueron las siguientes:
- Desarrollo iterativo e incremental mediante la realización de pequeñas mejoras, unas tras otras.
- Realización de pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- Frecuente integración del equipo de programación con el cliente o usuario.
- Corrección de todos los errores antes de añadir nueva funcionalidad.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y capacidad de mantenimiento pero sin modificar su comportamiento.
- Simplicidad en el código: es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Evidentemente quedaban descartadas características como la programación por parejas y la propiedad del código compartida

A continuación se describen cada una de las fases en las que se dividió el ciclo de vida del software de nuestro sistema:

1. Fase de la exploración: En esta fase, se plantearon los requisitos de la aplicación. Al mismo tiempo se procedió a familiarizarse con las herramientas, tecnologías y prácticas que se utilizaron durante la realización del proyecto.
2. Fase de planeamiento: se estimó la duración del desarrollo de cada módulo del sistema, incluyendo varias iteraciones hasta lograr una buena versión de cada uno de ellos. Una vez completado el módulo era añadido al conjuntos comprobando su funcionamiento en conjunto.
3. Fase de producción: Completada la primera versión completa del sistema se procedió a probar y comprobar su funcionamiento. Los nuevos cambios que fueron encontrados fueron añadiéndose tras cada comprobación.

1.4 Recursos utilizados.

Los recursos requeridos para la realización e implementación de este proyecto han sido los descritos a continuación.

1.4.1 Software.

Java™ Platform Micro Edition Software Development KIT 3.0: herramienta gratuita de Java, esencial para el desarrollo de aplicaciones destinadas a móviles como es el caso de este proyecto.

Sony Ericsson SDK 2.5.0.6 for the Java™ ME Platform (Emulator): Emulador gratuito de terminales Sony Ericsson. De fácil instalación y uso, fue fundamental para la realización de pruebas lógicas del sistema.

NetBeans IDE 6.8: Entorno de desarrollo integrado de código abierto, ideado para escribir, compilar, depurar y ejecutar programas en Java y otros lenguajes de programación.

Sistema operativo MS Windows Vista™: Debido al carácter universitario de este trabajo, y a pesar de su incompatibilidad causada por las limitaciones que ofrece para trabajar con cierto complementos informáticos actualmente no compatibles con este sistema operativo, se optó por su utilización pudiéndose realizar este proyecto en perfectas condiciones.

Microsoft Office 2007: Herramienta ofimática destinada a la redacción y documentación de la memoria de este proyecto.

1.4.2 Hardware.

- Portátil de gama media Acer Aspire 5100.

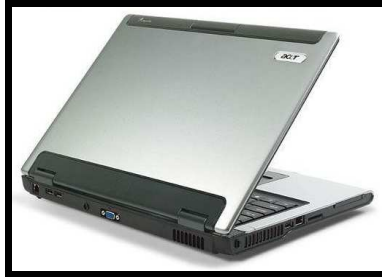


Figura 1 - Portátil de gama media.

- Adaptador USB Bluetooth para la transmisión de datos desde el PC al dispositivo móvil.



Figura 2 - Adaptador USB Bluetooth.

- Dispositivos móviles de gama media y baja, con Bluetooth integrado para su uso como balizas (Figura 3), y con soporte para aplicaciones Java para soportar y ejecutar la aplicación (Nokia 2730 Classic, Figura 4).



Figura 3 - móviles-baliza



Figura 4- Nokia 2730

2. ESTADO DEL ARTE.

2. Estado del Arte.

El posicionamiento de interiores (IPS o LPS) es una tecnología de posicionamiento complementaria y en ocasiones alternativa al GPS, especialmente en zonas donde su señal no alcanza o es débil, como en complejos subterráneos, el interior de construcciones o entre edificios de gran altura. Este capítulo de la memoria está dedicado a servir como toma de contacto con las distintas soluciones existentes en la actualidad para este problema. Dichas técnicas se pueden clasificar según la arquitectura en la que estén basadas sea distribuida o centralizada.

2.1. Arquitecturas distribuidas.

La primera está basada en el cálculo de la posición del dispositivo a localizar, mediante el cómputo de su posición en un servidor externo al entorno de posicionamiento (elemento activo). Para estimar la posición exacta, dicho equipo utiliza datos recibidos desde balizas (o puntos de acceso) repartidas en la infraestructura, cuya misión es detectar las señales procedentes del dispositivo móvil a localizar. La lógica de posicionamiento es externa al dispositivo móvil mientras que la presentación de datos varía dependiendo de cada sistema: unos requieren el envío de la posición exacta al dispositivo móvil, y otros simplemente su presentación en otro terminal.

2.1.1 .Ejemplos distribuidos.

La gran mayoría de soluciones aportadas en la actualidad al posicionamiento en interiores utilizan una arquitectura distribuida. Estos sistemas están basados en Ultrasonidos, Radio Frecuencia, Infrarrojos o la combinación de estas tecnologías. No obstante los más efectivos son los basados en radiofrecuencia, pues al contrario que el resto no requieren de una línea directa de visión entre emisor y receptor. Esta tecnología, presenta la ventaja de atravesar la mayor parte del mobiliario presente en interiores, así como no verse afectada en exceso su señal debido al movimiento de las personas.

A continuación se exponen algunos de los ejemplos conocidos de este tipo de sistemas.

2.1.1.1 El sistema Bat.

Pionero en el desarrollo de un sistema de posicionamiento por ultrasonidos de banda ancha, consiste en dispositivos móviles identificados unívocamente que transmiten una señal ultrasónica recibida por receptores fijos montados en el techo de la estancia (Figura 5). Este sistema desarrollado por la Universidad de Cambridge [CAMB10], mide la distancia entre cada emisor y receptor en función del tiempo de “vuelo” de la señal ultrasónica, calculando de este modo la posición del emisor mediante multilateración (Figura 6) con una precisión de 2 cm.

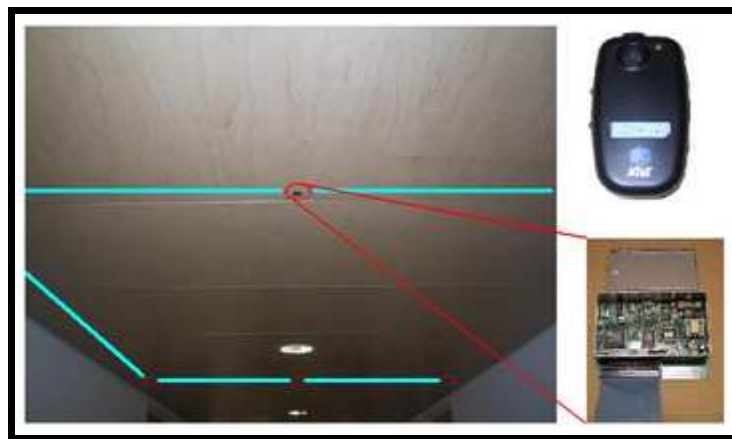


Figura 5 - Emisor localizado en el techo.

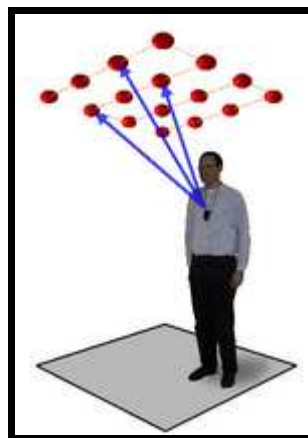


Figura 6 - Multilateración de la señal.

2.1.1.2 Symeo.

La empresa *Symeo GmbH* [SYME10] ofrece un sistema de posicionamiento en 2 dimensiones basado en la medición de distancias (Figura 7) a partir de las diferencia del tiempo de llegada entre antenas que emiten señales en la banda ISM (5.8 GHz). Mediante la multilateración se pueden determinar posiciones entre antenas fijas y móviles (Figura 8), llegando a alcanzar un error en el posicionamiento de 5 cm.

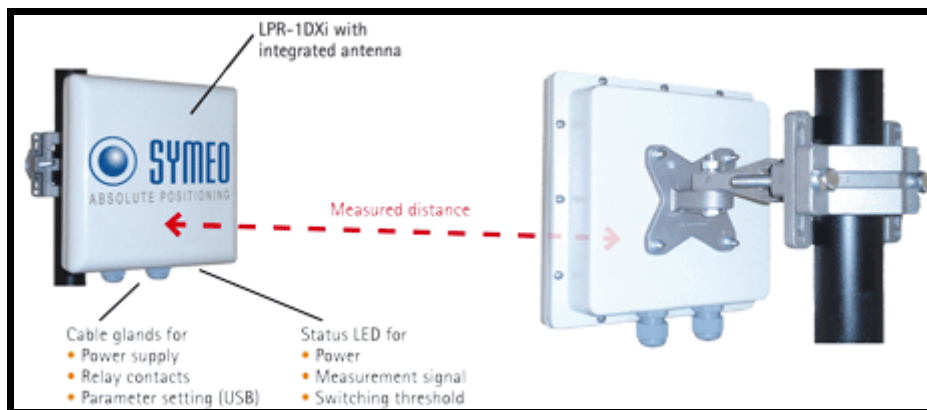


Figura 7 - Multilateración basada en la medición de distancias.



Figura 8 - Distintos tipos de antenas del sistema Symeo.

2.1.1.3 Cricket.

Desarrollado por el MIT [MIT10], este modelo posiciona mediante la combinación de radio frecuencia y ultrasonidos (Figura 9). Consiste en situar emisores en posiciones determinadas que emiten simultáneamente ultrasonidos y señales de radio frecuencia con un número de identificación único (Figura 10). Debido a que las señales de radio frecuencia viajan unas 10⁶ veces más rápido que los ultrasonidos, el receptor que es el elemento móvil, utiliza la diferencia de tiempo entre el inicio de la transmisión de la señal de radio frecuencia y el pulso ultrasónico correspondiente, para calcular la distancia que le separa del emisor o baliza. Con varias de estas distancias conocidas se puede determinar la posición exacta (consiguiendo una precisión de 1 a 2 cm) mediante multilateración.

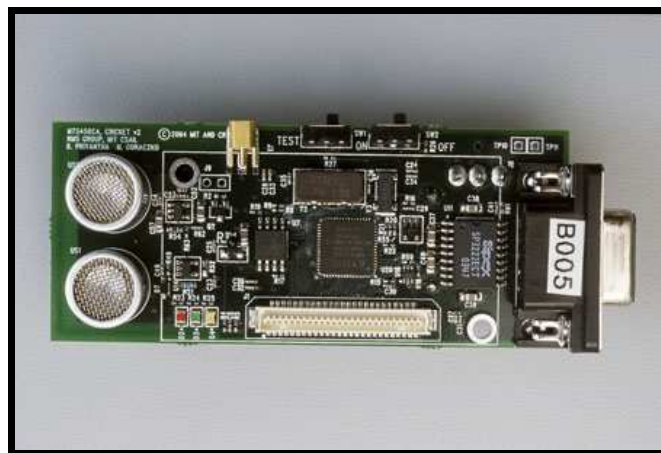


Figura 9 - Emisor de radio frecuencia y ultrasonidos.

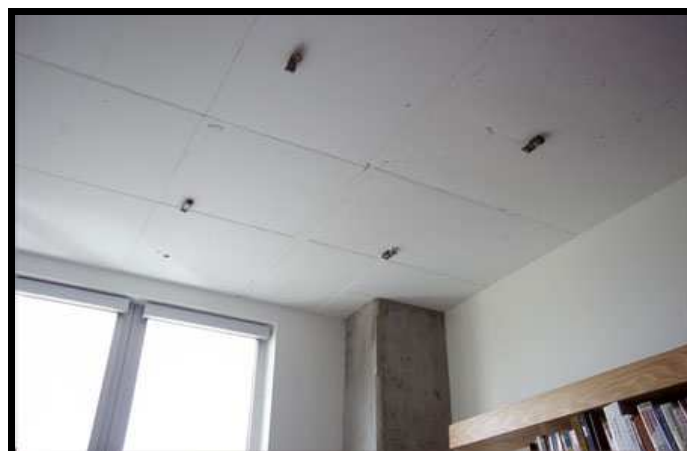


Figura 10 - Disposición de los emisores en el techo.

2.1.1.4 iHAP.

Este sistema domótico es capaz de ubicar en cada momento en qué habitación se encuentra un usuario portador de un dispositivo Bluetooth, dentro de una vivienda [IVAN05]. Los puntos de acceso miden la intensidad con la que reciben la señal Bluetooth además del dato de la intensidad con el que está emitiendo el dispositivo. Mediante la trilateración de la intensidad un servidor es capaz de calcular la localización del usuario y mostrarle contenidos multimedia en cada habitación del hogar según vaya cambiando de estancia.



Figura 11 - Disposición del sistema iHAP en el interior de una vivienda

2.2 Arquitecturas Centralizadas.

La segunda técnica, que hace uso de una arquitectura centralizada, es la utilizada en este proyecto. En este caso es el dispositivo móvil el elemento activo, pues la lógica de posicionamiento se encuentra y es llevada a cabo exclusivamente en dicho dispositivo. Mediante la identificación de las señales procedentes de las balizas (elementos pasivos) es capaz de estimar su posición exacta indoor. Dicha estimación, será más exacta dependiendo de una serie de factores que se expondrán y analizarán profundamente en el capítulo 4. del presente trabajo.

2.2.1 Ejemplos centralizados.

En general, los ejemplos basados en arquitecturas centralizadas hacen uso de la tecnología Bluetooth, pues es la más adecuada al encontrarse integrada en gran parte de los dispositivos electrónicos de uso doméstico. Estos dispositivos actuando como balizas, ayudan a situarse al dispositivo móvil en el entorno, al ir detectándolas mientras se desplaza por él, utilizando el método de división del espacio en celdas citado con anterioridad. Su rango de alcance, determinado por la potencia de transmisión, los agrupa en 3 clases distintas como se aprecia en la siguiente tabla.

	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Rango (aproximado)
Clase 1	100 mW	20 dBm	~ 100 metros
Clase 2	2,5 mW	4 dBm	~ 25 metros
Clase 3	1mW	0 dBm	~ 1 metro

Tabla 1 - Clasificación de dispositivos Bluetooth según potencia de emisión.

La dificultad por tanto, reside en encontrar un dispositivo emisor de bajo coste cuyo rango de alcance pueda ser variable, y conseguir de este modo que las señales de las balizas no se solapen. A continuación se exponen los distintos métodos conocidos de construcción de balizas.

2.2.1.1 Método I.

La primera opción es utilizar un adaptador USB Bluetooth (o dongle) conectado a un PC. Para conseguir que el rango de alcance sea corto se aísla el dongle con papel de aluminio. Es el método más sencillo pero también el que presenta más desventajas:

- ❖ El coste: la necesidad de utilizar un PC para inicializar la baliza y darle alimentación eléctrica, implica la utilización de un recurso de alto coste. Aun en el caso de que sea antiguo, existe la necesidad de que soporten los drivers del dongle, algo que puede ser difícil en estos equipos.
- ❖ La manejabilidad: es evidente la falta de esta característica pues incluso el uso de un PC portátil implicaría que su posicionamiento como baliza dentro del entorno fuera algo aparatoso.
- ❖ El alcance de la señal: Es muy difícil conseguir apantallar uniformemente la señal, pues aunque se aísla bien el dongle, la señal Bluetooth puede escapar a plena potencia ha travesando el interior del PC por el paso inevitable del puerto USB, como del modo ilustrado por la siguiente figura.

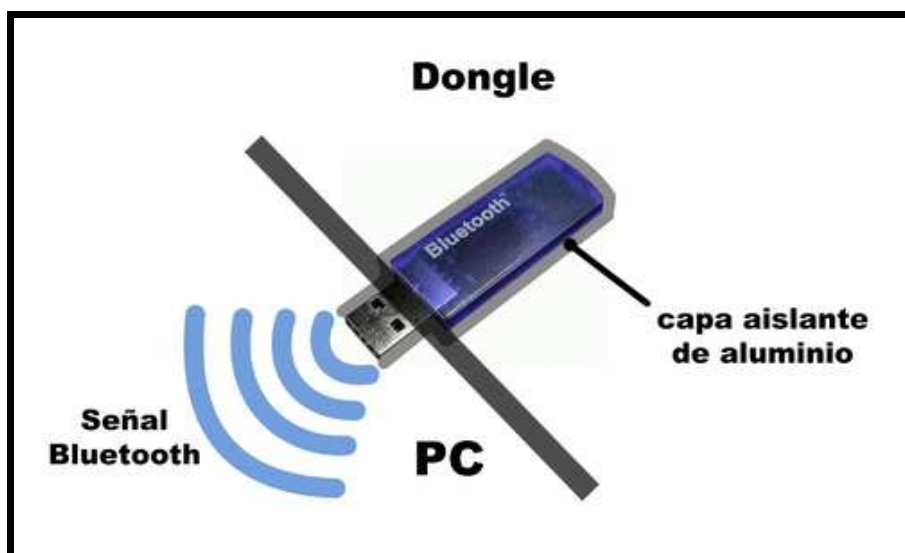


Figura 12 - Apantallamiento de la señal de un dongle Bluetooth

2.2.1.2 Método II.

El segundo método, basado en el trabajo realizado por Albert Huang del MIT [ALBE05], consiste en el uso de un dongle Bluetooth, un USB hub con adaptador de Corriente alterna y papel de aluminio aislando el conjunto (Figura 13). Para poder inicializar el dongle y permitir que emita su mac, se conecta al USB hub conectado previamente a un PC. Posteriormente se desconecta el hub del PC y como la alimentación del dispositivo está asegurada por su adaptador de corriente, se obtiene una baliza de corto alcance totalmente operativa. La única desventaja de este modo de obtención de balizas es que una pérdida momentánea de alimentación ocasionaría que la baliza tuviera que ser reiniciada mediante una nueva conexión al PC.



Figura 13 - Material necesario para la obtención de balizas del Metodo II.

2.2.1.3 Método III.

El tercer método aportado por Keneth Cheung, Stephen Intille y Kent Larson también del MIT [KENE06], consiste en la sustitución del dongle por un auricular Bluetooth para teléfono móvil de los disponibles actualmente (Figura 14). Estos dispositivos al encenderlos entran en un estado de espera sin emitir su mac, pero responden con ella si un dispositivo de su alrededor realiza alguna inquiry.



Figura 14 - Auriculares Bluetooth para teléfono móvil.

Al poseer baterías propias no necesitan estar alimentados continuamente, y apantallar su señal de radio frecuencia se consigue al introducirlos en un caja de forrada con dos capas de papel de aluminio (10 m), consiguiendo un radio de acción de 2 - 3 metros. El inconveniente presentado por este modelo es que pasado un tiempo prudencial el auricular pasa de estado de espera a apagarse, al no vincularse con ningún dispositivo móvil. Este problema fuerza a que cada cierto tiempo se deba reiniciar el auricular, es decir una desventaja similar a la del trabajo de Albert Huang.

2.2.1.4 Método IV.

La cuarta solución ha sido hallada durante la realización de este proyecto y consiste en la simple utilización de teléfonos móviles con Bluetooth operativo que se encuentren fuera de servicio por cualquier motivo (su antigüedad, desperfectos, poca duración de la batería, etc.). Activando su visibilidad Bluetooth pueden funcionar durante horas sin cambiar de estado o apagarse, y pueden ser recargados mediante su adaptador o sencillamente funcionar conectados permanentemente a la corriente eléctrica. Para apantallar nada más simple que cubrirlos de papel de aluminio de 30 micras consiguiendo un rango de 2 a 3 metros. En son en definitiva, la baliza perfecta para nuestro entorno inteligente.



Figura 15 - Algunas de las balizas utilizadas en la realización de este proyecto.

2.3 Métodos de cálculo de la posición.

En cuanto a los métodos empleados para el cálculo de la posición de un dispositivo móvil en interiores, con independencia al tipo de arquitectura y tecnología empleada son las siguientes:

2.3.1 División del espacio en celdas.

Identificando celdas con puntos de acceso, se asigna a cada punto un espacio de igual tamaño al del su área de cobertura. De este modo se identifica en que celda esta el dispositivo móvil. Esta es la técnica empleada en el presente proyecto.

2.3.2 Multilateración de magnitudes.

La trilateración es un método matemático para determinar las posiciones relativas de objetos mediante la geometría de triángulos. A partir de las localizaciones conocidas de dos puntos de referencia, y una magnitud medida entre el sujeto y cada uno de dichos puntos, se puede estimar su posición en el plano. Para determinar de forma única y precisa la localización relativa de dicho punto se necesitan generalmente al menos 3 puntos de referencia. Las magnitudes medidas por los puntos de acceso, y utilizadas para realizar el cálculo, son las siguientes

- ❖ Mediante el ángulo de llegada de la señal procedente del dispositivo emisor
- ❖ Mediante la intensidad de la señal medida procedente del dispositivo emisor.
- ❖ Mediante el tiempo de llegada de los paquetes o señales enviados entre el emisor y el punto de acceso.

3. ESTUDIO DE TECNOLOGÍAS

3. Estudio de Tecnologías.

3.1 Introducción.

En la actualidad, el aumento del uso de internet en la sociedad y la necesidad de estar conectado desde cualquier lugar y momento ha propiciado un gran avance en la informática móvil y las comunicaciones. El acceso a la información utilizando medios sin cables no es reciente, pero dicha necesidad de comunicación ha originado en especial, un crecimiento en el sector de las tecnologías inalámbricas sin precedentes [TIMO08]. De todas ellas, las denominadas de “corto alcance” pueden usar indistintamente Radio Frecuencia o infrarrojos empleando franjas del espectro de frecuencia cuya explotación normalmente no requiere licencia (es gratuito).

De entre todos los estándares existentes hay tres (figura tal) que destacan por ser los más extendidos: los infrarrojos de la Asociación de Datos por infrarrojos (IrDA®), las redes de área local inalámbricas (WLAN o también conocidas como IEEE 802.11 y sus variantes) y la tecnología inalámbrica Bluetooth.

	Posición	Baliza detectada	Tiempo de detección
Tipo de conexión	Infrarrojos, línea de visión mediante un haz estrecho	Propagación esférica del espectro	Propagación esférica del espectro
Espectro	Óptico 850 - 900 nm	Radio frecuencia 2.4 GHz (5 GHz para 802.11a/n)	Radio frecuencia 2.4 GHz
Potencia de transmisión	40 - 500 mW/s	100 mW	10 - 100 mW
Tasa máxima de transmisión de datos	9600bps - 16 Mbps	11 Mbps (54 Mbps para 802.11a, 802.11g)	3 Mbps
Rango de alcance	1 metro	100 metros	10 - 100 metros
Nº de dispositivos soportados	2	Conexión a través de un punto de acceso	8 (activos), 200 (pasivos)
Canales de voz	No	No	Si
Tipo de direccionamiento	Id física de 32 bit ⁴	mac de 48 bit	mac de 48 bit

Tabla 2 - Comparativa de tecnologías de corto alcance

3.2 Bluetooth.

Como se explica en el Capítulo 2.2, este proyecto se desarrollará empleando la tecnología Bluetooth al ser la que mejor se ajusta a sus necesidades y objetivos. Algunas de sus características son las siguientes:

- La tecnología inalámbrica Bluetooth está orientada a aplicaciones de voz y datos (sin grandes necesidades de ancho de banda).
- Funciona en la banda de frecuencia de 2.4 GHz, que no precisa de ninguna licencia.
- Establece un radio de acción de 10 o 100 metros dependiendo de la clase del dispositivo. Su uso es adecuado cuando puede haber dos o más dispositivos en un área reducida. La máxima velocidad de transmisión es de 3 Mbps.
- Los objetos sólidos no suponen ningún obstáculo, no es necesario que los dispositivos estén situados en la misma línea de visión ya que se transmite en todas direcciones.
- Simplifica el descubrimiento y configuración de los dispositivos, sin un control explícito de direcciones de red, permisos u otros aspectos típicos de redes tradicionales.
- La seguridad siempre ha sido una de las prioridades en el desarrollo de la tecnología Bluetooth y continúa siéndolo. La especificación Bluetooth ofrece tres modos de seguridad.
- El coste de los chips Bluetooth es inferior a tres dólares estadounidenses. Es una tecnología muy extendida: se utiliza en gran número de productos tales como teléfonos móviles, PDAs, impresoras, módems, auriculares, ratones...

3.2.1 Arquitectura.

La siguiente ilustración muestra la pila de protocolos específicos para el uso de la tecnología inalámbrica Bluetooth. Los protocolos de color más oscuro son a los que se puede tener acceso mediante los perfiles definidos con la especificación JSR-82. En el caso que nos ocupa, trabajaremos con el protocolo SDP (Service Discovery Protocol) que define el descubrimiento de dispositivos y sus servicios [SERG03]. Para tener acceso a dicho protocolo manejaremos por tanto el perfil básico Bluetooth SDAP (Service Discovery Application Profile) que describe las operaciones fundamentales necesarias para trabajar con dicho protocolo.

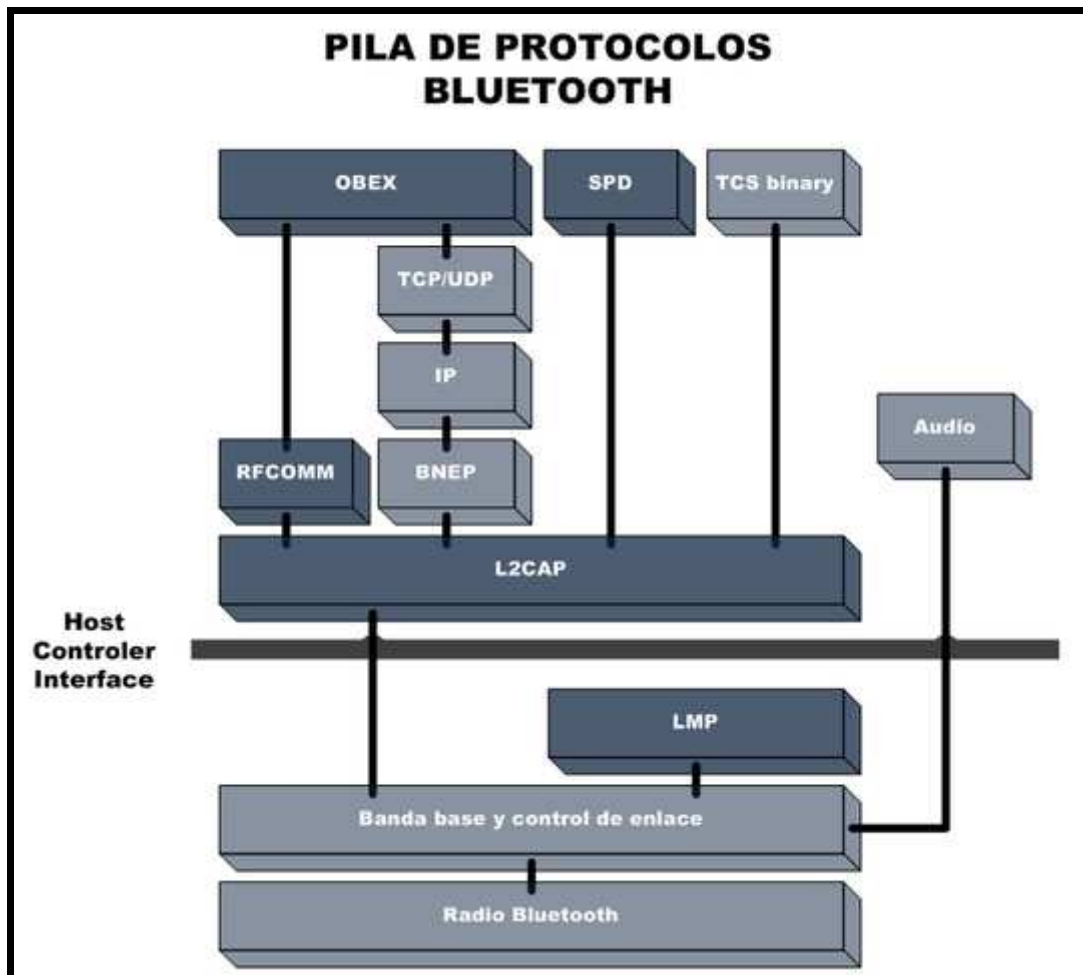


Figura 16 - Pila de protocolos Bluetooth.

3.2.2 El modelo cliente-servidor.

En esta sección se expone de forma general el modelo cliente-servidor utilizado para establecer la comunicación en dispositivos Bluetooth. Un servicio es una aplicación que actúa como servidor proporcionando asistencia a los dispositivos clientes a través de la comunicación Bluetooth. Estos servicios están a disposición de los dispositivos clientes remotos mediante la definición de un registro de servicio (en adelante *service record*) que describe el servicio y la adición de ese servicio a la base de datos de descubrimiento de servicios (SDDb) del dispositivo local.

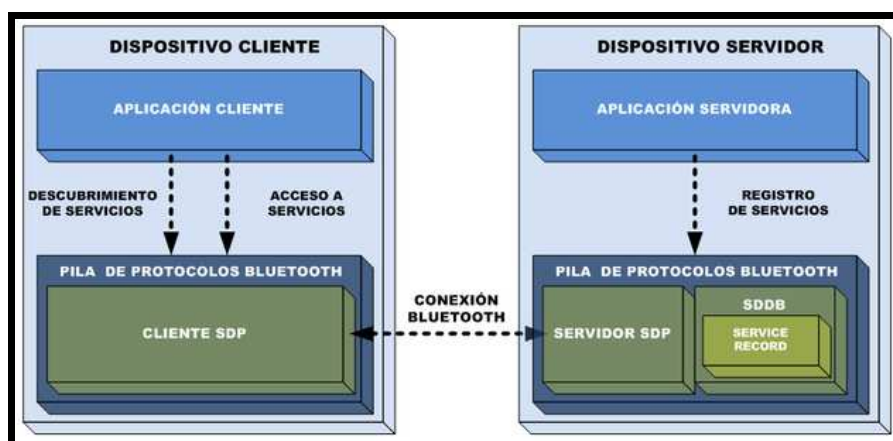


Figura 17 - Modelo cliente-servidor Bluetooth.

El SDP, como se mencionó anteriormente, es un protocolo Bluetooth para descubrir los servicios prestados por un dispositivo. En el instante que un servidor de aplicaciones registra un servicio en la SDDb, la pila Bluetooth proporciona un servidor SDP, el cual mantiene la base de datos de los servicios registrados. Los servicios de descubrimiento de los dispositivos clientes utilizan SDP para consultar al servidor de dicho protocolo por el registro de algún servicio de interés. En dicho instante el *service record* proporciona suficiente información para permitir a un cliente SDP conectar con el Servicio Bluetooth presente en el dispositivo servidor.

Después de registrar un *service record* en el SDDb, la aplicación servidora espera a que alguna aplicación cliente inicie contacto con el servidor y de este modo darle acceso al servicio. La aplicación cliente y la servidora entonces establecen una conexión Bluetooth para llevar a cabo sus tareas.

3.3 Java ME

La plataforma Java ME (anteriormente conocido como J2ME), permite disfrutar del uso, posibilidades y beneficios de la tecnología Java (Portabilidad del código, la programación orientada a objetos, y un rápido ciclo de desarrollo del código) en dispositivos de recursos limitados como teléfonos móviles, PDA, navegadores GPS, Smartphones y demás aparatos de consumo, aprovechando las capacidades nativas de cada dispositivo. Para ello su arquitectura define configuraciones, perfiles y paquetes opcionales para permitir la modularidad y personalización según el dispositivo.

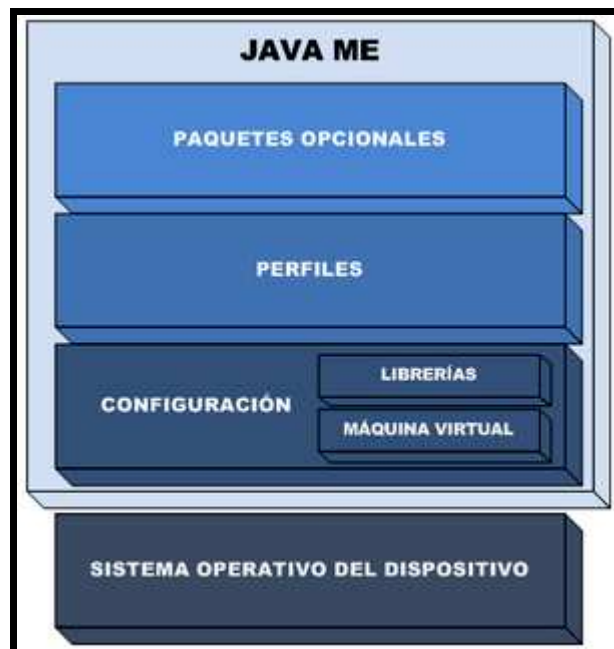


Figura 18 - Arquitectura Java ME.

3.3.1 Configuraciones.

La maquina virtual de Java (JVM) interpreta los byte codes generados al compilar los programas java. Un programa java puede ejecutarse en cualquier dispositivo que disponga de una maquina virtual adecuada y un conjunto adecuado de clases Java. La configuración define la funcionalidad mínima necesaria para un grupo particular o categoría de dispositivos. Este proyecto hace uso de la más pequeña de las configuraciones, la “Connected, Limited Device Configuration” en adelante CLDC, utilizada en dispositivos de gama baja con procesadores lentos, memoria limitada y que soportan conexiones de red intermitentemente. Esta configuración incluye una KVM (Kylobyte Virtual Machine) en vez de la habitual JVM, debido al tamaño pequeño de memoria de que disponen los dispositivos. En la siguiente tabla se muestran los paquetes que incluye esta configuración [JAVA10].

Paquetes del CLCD 1.1	
java.io	Proporciona clases y paquetes estándar de E/S. Es un subconjunto de J2SE.
java.lang	Proporciona clases e interfaces de la máquina virtual. Es un subconjunto de J2SE
java.util	Contiene la colección de clases, interfaces y utilidades estándar. Es un subconjunto de J2SE.
javax.microedition.io	Clases e interfaces de conexión genérica CLDC

Tabla 3 -Paquetes incluidos en la configuración CLCD 1.1

3.3.2 Perfiles

Los perfiles añaden funcionalidad y las APIs necesarias para completar un entorno de ejecución totalmente operativo para cada tipo de dispositivo. El MIDP fue el primer perfil creado y diseñado para teléfonos móviles y similares. Combinado con la configuración CLDC aporta funcionalidad en la gestión de sus interfaces, las comunicaciones y el almacenamiento de datos persistente, dentro de sus aplicaciones denominadas MIDlets. Trabajaremos con el perfil MIDP 2.1 en nuestro sistema.

Paquetes del perfil MID	
javax.microedition.lcdui	La API del interfaz de usuario. Proporciona un conjunto de características para la implementación de las interfaces de usuario de las aplicaciones MIDP
javax.microedition.rms	El Mobile Information Device Profile provee de mecanismos para el almacenamiento persistente de datos en el dispositivo, y su posterior recuperación
javax.microedition.MIDlet	El paquete MIDlet define las aplicaciones del perfil MIDP y la interacción entre la aplicación y el entorno en el que esta se ejecuta.
javax.microedition.io	Clases e interfaces conexión genérica
java.io	Proporciona clases para la entrada y salida a través de flujos de datos.
java.lang	Proporciona clases que son fundamentales para el lenguaje de programación Java
java.util	Clases e interfaces de utilidades estándar

Tabla 4 - Paquetes incluidos en el perfil MID 2.1

3.3.3 JSR-82

Desarrollada por la Java Community process como JSR-82 (java Specification Request 82), es la especificación de la plataforma J2ME para las APIs (Interfaces de programación de aplicaciones) que proporcionan a los Java MIDlets la capacidad de manejar Bluetooth en los dispositivos en los que están instalados.

Se encuentra dividida en dos partes, el paquete javax.bluetooth y el paquete javax.obex. En este proyecto trabaja exclusivamente con el primer paquete pues es que permite acceder al perfil Bluetooth SDAP del dispositivo Bluetooth local y mediante el uso del protocolo SDP iniciar la búsqueda de los dispositivos del entorno. A continuación describimos las interfaces, clases y excepciones forman el paquete javax.bluetooth [PEDR04].

Interfaces	
DiscoveryListener	Este interfaz permite a las aplicaciones especificar un evento en el <i>listener</i> que reaccione ante eventos de búsqueda. También se utiliza para encontrar dispositivos.
L2CAPConnection	Este interfaz representa las conexiones L2CAP.
L2CAPConnectionNotifier	Este interfaz es usado por los servidores L2CAP para escuchar conexiones de clientes.
ServiceRecord	Este interfaz define el Service Record de Bluetooth que contiene los pares (atributo ID, valor).

Tabla 5 - Interfaces pertenecientes a la especificación JSR-82

Clases	
DataElement	Esta clase contiene varios tipos de datos que un atributo de servicio Bluetooth puede usar. También proporciona una interfaz que permite construir y recuperar valores de dicho atributo de servicio
DeviceClass	Esta clase define los valores del tipo de dispositivo y los tipos de servicios de un dispositivo
DiscoveryAgent	Esta clase provee métodos para descubrir dispositivos y servicios.
LocalDevice	Esta clase provee acceso y control sobre el dispositivo local Bluetooth.
RemoteDevice	Esta clase representa al dispositivo Bluetooth remoto.
UUID	Esta clase encapsula enteros sin signo que pueden ser de 16, 32 ó 128 bits de longitud. Estos enteros se usan como un identificador universal cuyo valor representa un atributo del servicio.

Tabla 6 - Clases pertenecientes a la especificación JSR-82

Excepciones	
BluetoothConnectionException	Esta excepción se lanza cuando una conexión Bluetooth (RFCOMM o L2CAP) no puede ser establecida satisfactoriamente.
BluetoothStateException	Esta excepción ocurre cuando un dispositivo no puede atender una petición que normalmente atendería por culpa de las características de la conexión radio
ServiceRegistrationException	Esta excepción se lanza cuando se intenta añadir o modificar un ServiceRecord en la SDDb y se produce algún error.

Tabla 7 - Excepciones incluidas en la especificación JSR-82

3.3.4 Descubrimiento de dispositivos Bluetooth en Java ME.

La especificación JSR-82 ofrece la capacidad de descubrir dispositivos y servicios remotos mediante la clase `DiscoveryAgent`, y el interfaz `DiscoveryListener`. Cada dispositivo dispone de un solo objeto `DiscoveryAgent`, el cual proporciona métodos para iniciar la búsqueda de dispositivos y servicios remotos. Dicho objeto `DiscoveryAgent` es invocado por el dispositivo local mediante el método `LocalDevice.getDiscoveryAgent()` (Código 1).

```
try {  
    LocalDevice ld = LocalDevice.getLocalDevice();  
    DiscoveryAgent da = ld.getDiscoveryAgent();  
} catch (BluetoothStateException bse) { }
```

Código 1 - Invocación de los objetos `LocalDevice` y el `DiscoveryAgent`.

La implementación de estas instrucciones debe capturar las posibles excepciones `BluetoothStateException` lanzadas por `LocalDevice.getLocalDevice()`. Una vez es obtenido el `LocalDevice`, la llamada a `getDiscoveryAgent()` devuelve el `DiscoveryAgent` asociado a éste.

```
public class Listener implements DiscoveryListener {  
    public void deviceDiscovered(RemoteDevice rd, DeviceClass clase) [...]  
    public void inquiryCompleted(int completado) [...]  
    public void servicesDiscovered(int transID, ServiceRecord[] servRecord) [...]  
    public void serviceSearchCompleted(int transID, int respCode) [...]  
}
```

Código 2 - Interfaz `DiscoveryListener` y sus distintos métodos.

Por otro lado, el interfaz `DiscoveryListener` (Código 2) es utilizado por el `DiscoveryAgent` para devolver los dispositivos y servicios descubiertos tan pronto como son encontrados. También proporciona una serie de eventos que entre otros, notifican cuando la búsqueda de dispositivos o servicios ha sido completada.

Una vez disponibles todos los elementos anteriores se puede proceder a iniciar una búsqueda, mediante el método `startInquiry()` que toma como argumentos el tipo de búsqueda y la implementación el interfaz `DiscoveryListener`.

```
try {  
    da.startInquiry(DiscoveryAgent.GIAC, new Listener());  
} catch (BluetoothStateException bse) { }
```

Código 3 - Inicio de la búsqueda de dispositivos Bluetooth.

El tipo de búsqueda General Inquiry Access Code (GIAC) localiza todos los dispositivos ubicados en su zona, estén en modo general o limited. El modo Limited Inquiry Access Code (LIAC) se utiliza cuando se necesita una comunicación exclusiva entre dos dispositivos.

El método `startInquiry()` (Código 3) lanza una `BluetoothStateException` si la búsqueda no se ha iniciado debido a que el dispositivo está en un estado que no le permite completar la siguiente búsqueda. Esto puede ocurrir si el dispositivo ya está llevando a cabo una búsqueda o está ya conectado a otro dispositivo.

Iniciada la búsqueda, el evento `deviceDiscovered()` del `DiscoveryListener` proporcionará dos objetos cada vez que un dispositivo Bluetooth remoto sea encontrado: `RemoteDevice` y `DeviceClass`. El primero proporciona entre otros datos, la dirección del dispositivo y su friendly name, mientras que el segundo informa del tipo de dispositivo del que se trata.

Para cancelar la búsqueda se utiliza el método `cancelInquiry()` pasándole como argumento el objeto `DiscoveryListener` empleado cuando esta fue iniciada. Este método devuelve `true` si la búsqueda se ha cancelado y `false` en caso contrario.

La búsqueda finaliza en el momento en el que el evento `inquiryCompleted()` del `DiscoveryListener` devuelve la razón de su finalización como argumento al método. Dicho argumento puede ser `DiscoveryListener.INQUIRY_COMPLETED` si se completa la búsqueda con normalidad. Este evento se ejecuta sin que el método `cancelInquiry()` tenga lugar.

4. PROPUESTA DE **ENTORNO INTELIGENTE**

4. Propuesta de entorno inteligente.

4.1 Introducción

Este capítulo está dedicado al estudio y proposición del entorno inteligente sobre el que se basó el desarrollo de la aplicación. De este modo, al término de dicho desarrollo y mediante la adecuación de un entorno real de condiciones similares al propuesto, se realizaron casos prácticos

Como ya se indicó anteriormente, está basado en una arquitectura centralizada utilizando el método de división en celdas a través de balizas pasivas Bluetooth. Primero se definen algunos aspectos de interés referentes al entorno simulado, seguidamente las balizas o elementos pasivos y a finalmente el dispositivo móvil utilizado como elemento activo.

4.2 Entorno inteligente simulado.

El entorno inteligente simulado fue un museo o galería de arte. La exposición albergada por dicho espacio público estuvo compuesta por cuatro cuadros (en adelante denominados también PDIs, esto es, Puntos De Interés); tres de ellos eran de tamaño considerable (3 x 2 metros) y un tercero de tamaño más reducido.(0,5 x 0,5 metros). Los aspectos tenidos en cuenta para el desarrollo del entorno fueron los siguientes:

Espacio físico: el espacio físico real con el que se contó para la simulación del entorno inteligente fue el proporcionado por un garaje (Figura 19). Las columnas sirvieron como enclave de los PDI, la distancia entre ellas era de 8 metros aproximadamente.

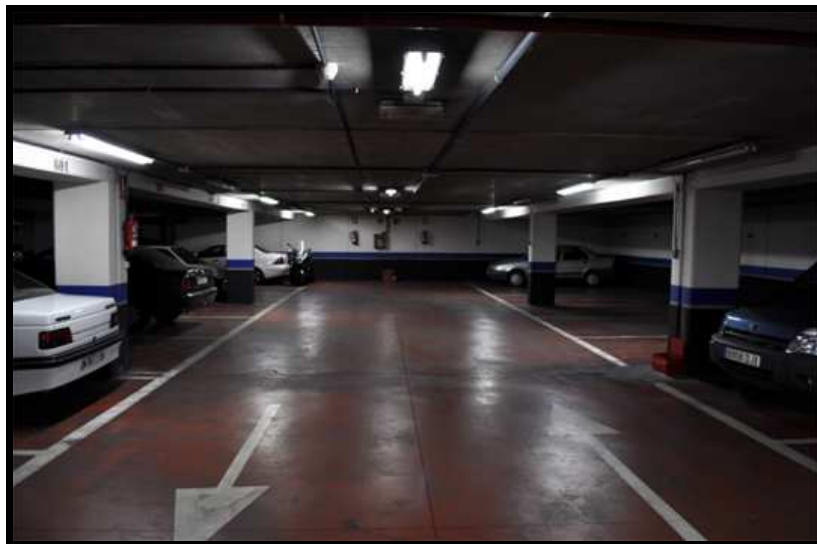


Figura 19 - Garaje propuesto y utilizado para la realización de pruebas físicas.

División del espacio en sectores: La diferencia de tamaño entre cuadros, requirió también una división en celdas o sectores del espacio desigual. Es evidente que la distancia necesaria para contemplar un objeto de gran tamaño no es el mismo que el de uno de medidas más reducidas. En este caso se precisó de un sector, y por tanto baliza, cuyo rango de alcance abarcara un máximo de 1 metro de distancia alrededor del PDI. Para el resto de PDIs se estableció que cada sector abarcara un radio máximo de 4 metros (Figura 20).

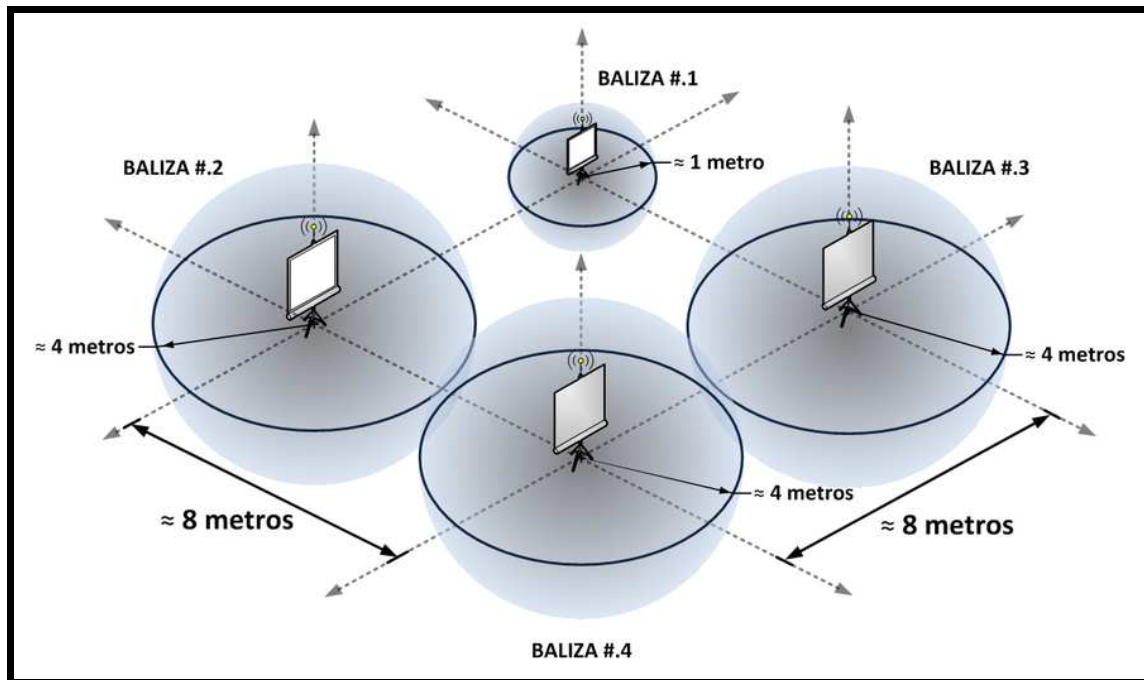


Figura 20 - Disposición de las balizas en el entorno propuesto.

Velocidad del dispositivo móvil: Dado el carácter cultural del entorno elegido, la velocidad del individuo que portaba el dispositivo móvil no debía ser excesivamente rápida, por lo que se tomó como válida una velocidad de 2 km/h (0,55 metros por segundo). Esta velocidad junto a la frecuencia de búsqueda del MIDlet, establecida en 1 segundo, permitió visualizar la información de cada PDI (Punto de Interés) sin problemas ni interferencias.

Calidad de la señal: Los materiales que componen las edificaciones, como el hormigón armado, las líneas de la instalación eléctrica, aislantes, aparatos electrónicos y demás componentes de una construcción normal, provocan que los campos de emisión de las balizas sean asimétricos [KENE06]. El espacio en este caso era amplio y sin obstáculos entre balizas, por lo que las señales Bluetooth no se vieron demasiado afectadas. Como apunte, aclarar que estos campos tampoco se ven excesivamente afectados por el movimiento causado por la gente presente en ellos.

4.3 Las Balizas (elemento pasivo)

Para construir nuestro entorno se necesitaron balizas Bluetooth en estado detectable (Discoverable mode) que respondieran al mensaje de búsqueda del MIDlet. La principal dificultad residió en encontrar un dispositivo de bajo coste que actuara como baliza emitiendo su Bluetooth address, pues son inexistentes en el mercado. Además hubo que añadir la necesidad de que las balizas tuvieran un alcance de pocos metros, pues el solapamiento de señales dificultaría el posicionamiento. El método de obtención de las balizas de bajo coste fue el reciclaje de teléfonos móviles con funcionalidad Bluetooth siguiendo punto 2.2.1.4 *Método IV* expuesto en este trabajo. Se dispuso de cuatro teléfonos móviles con dicha funcionalidad, de los cuales 3 estaban fuera de servicio (incluso sin tarjeta sim,), y un tercero (Samsung GT-S5600) que fue la excepción pues aun estaba en uso y operativo.

Para obtener el apantallamiento de la señal de radiofrecuencia, y por tanto conseguir que en un espacio indoor no se solaparan las señales de los cuatro, se envolvió cada terminal en una o varias capas de aluminio de 10 μm (10^{-5} metros), procurando no aislar completamente el interior para no apantallar totalmente la señal de radiofrecuencia. En la siguiente Figura vemos un ejemplo de aislamiento.



Figura 21 - Ejemplo de baliza aislada por una capa de 10 μm de aluminio

La estimación de los rangos para cada tipo de aislamiento, se realizó mediante una búsqueda realizada por el terminal móvil que albergó posteriormente la aplicación desarrollada. Cuando el terminal detectaba la baliza se tomaba como rango de alcance la distancia que los separaba, obteniéndose los rangos de la siguiente tabla.

	Aislamiento I	Aislamiento II	Aislamiento III	Aislamiento IV
Sony Ericsson K510i	≈ 25 metros	≈ 8,3 metros	≈ 1 metro	0 metros
Nokia 5000d-2	≈ 10 metros	≈ 2,1 metros	≈ 0,3 metros	0 metros
Nokia 2630	≈ 10 metros	≈ 0,4 metros	0 metros	0 metros
Samsung GT-S5600	≈ 20 metros	≈ 3 metros	≈ 0,1 metros	0 metros

Tabla 8 - Rangos obtenidos con los distintos tipos de aislamiento

El aislamiento I corresponde al rango de alcance medido sin aislar el dispositivo. Los aislamientos II, III y IV corresponden al aislamiento de cada dispositivo por 1, 2 y 3 capas de aluminio respectivamente.

	Aislamiento I	Aislamiento II	Aislamiento III	Aislamiento IV
Grosor	0 μm	10 μm	20 μm	30 μm

Tabla 9 - Equivalencia aislamiento-grosor

El análisis gráfico de los datos presente en la Figura 22, muestra las distintas posibilidades ofrecidas por este método. En las distancias menores a 5 metros el aislamiento I fue el más eficaz en la mayor parte de terminales a excepción del Sony Ericsson. No obstante, dicho teléfono fue el único que aun mantenía una señal disponible y aceptable con un aislamiento III, pudiendo conseguir un posicionamiento más preciso.

En la Figura 23 se efectuó una aproximación logarítmica sobre los datos de la Tabla 8, para apreciar de manera continua y más precisa como variaba el rango de alcance conforme aumentaba el grosor de el aislamiento de aluminio aplicado a las balizas.

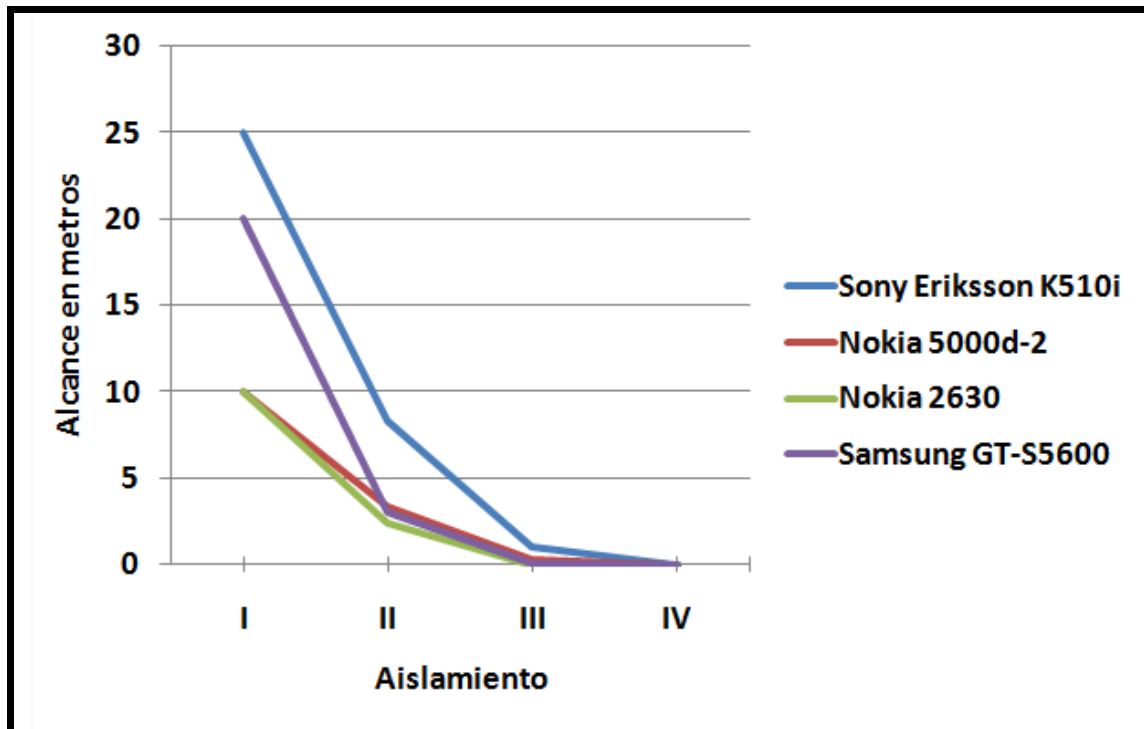


Figura 22 - Medidas obtenidas con los distintos tipos de aislamiento para cada baliza.

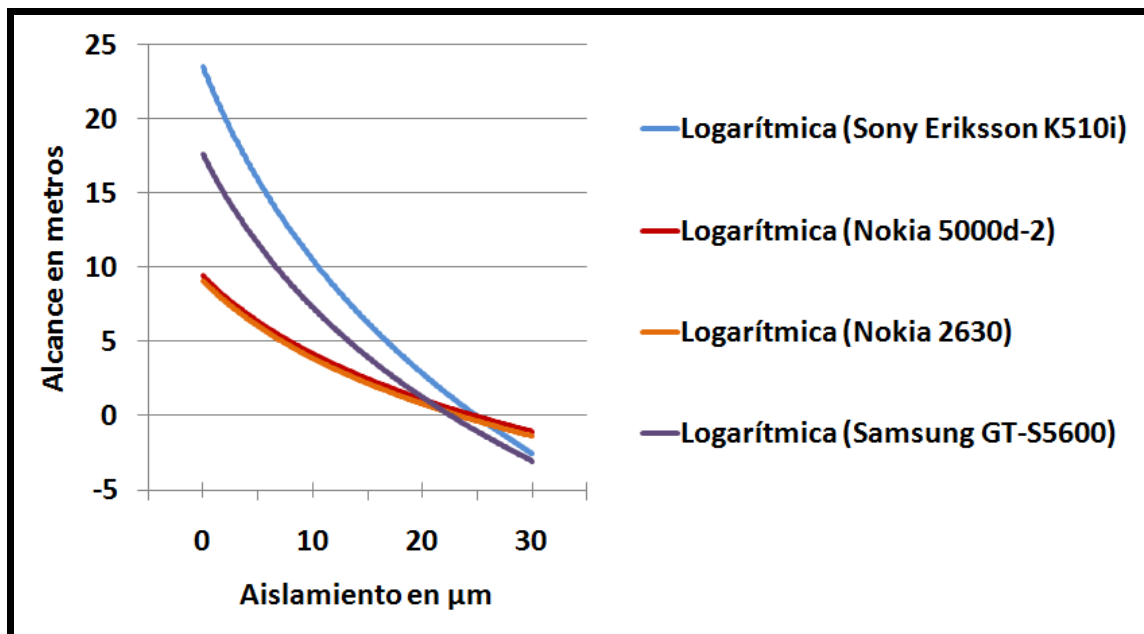


Figura 23 - Aproximación logarítmica de los resultados obtenidos.

Se concluyó que la versatilidad ofrecida por el terminal Sony Ericsson, lo convertía en la baliza ideal, pues brindaba un rango de alcance parametrizable mediante su aislamiento de 1 a 10 metros aproximadamente.

Por tanto, a la vista de dichos resultados, se optó por asignar cada baliza a un PDI del siguiente modo:

	Baliza #.1	Baliza #.2	Baliza #.3	Baliza #.4
Modelo	Sony Ericsson K510i	Nokia 5000d-2	Nokia 2630	Samsung GT-S5600
mac	001620708206	0021AAA090EA	001E3B2305B5	002567732674
Alcance	≈ 1 metro	≈ 4 metros	≈ 4 metros	≈ 4 metros
Aislamiento	20 µm	10 µm	10 µm	10 µm
Cuadro	0,5 x 0,5 metros	3 x 2 metros	3 x 2 metros	3 x 2 metros

Tabla 10 - Características finales de las balizas

4.4 El dispositivo móvil (elemento activo).

La elección del dispositivo móvil destinado a albergar y ejecutar el MIDlet desarrollado no fue en un principio relevante. Se buscó un dispositivo de gama baja que admitiera la ejecución programas J2ME con al menos una configuración CLCD 1.1 y un perfil MIDP 2.0.



Figura 24 - Sony Ericsson K510i.

El terminal móvil que se utilizó en primer término, un Sony Ericsson K510i [SONY10] como el de Figura 24, no ejecutaba en las primeras pruebas físicas la aplicación como se esperaba debido a un problema que explicaremos en el capítulo siguiente. Debido a esta incidencia optamos por cambiar de fabricante, eligiendo Nokia que ofrece un gran soporte para el desarrollo de aplicaciones Java para todo tipo de dispositivos. El modelo sobre el que se ejecuta finalmente la aplicación es un Nokia 2730 Classic [NOKI10] con soporte para una configuración CLDC 1.1 y MIDP 2.1 que podemos observar en la siguiente Figura.



Figura 25 - Nokia 2730 Classic.

5. DESARROLLO DEL SISTEMA

5. Desarrollo del Sistema

5.1 Análisis de requisitos.

El programa ubicado en el dispositivo móvil deberá cumplir con los siguientes requisitos obtenidos tras un análisis exhaustivo:

Requisito 1 - Acceso a un fichero de datos: La aplicación deberá disponer de un fichero de extensión .txt o similar, cuyo contenido será toda la información sobre los PDI necesaria para su detección, así como los datos a presentar en el momento en que estos sean detectados.

Requisito 2 - Búsqueda de dispositivos: La aplicación deberá iniciar una búsqueda de los dispositivos Bluetooth disponibles en su entorno y mantenerla activa ininterrumpidamente.

Requisito 3 - Detección de dispositivos: En el momento en que un dispositivo sea detectado la aplicación deberá mostrar toda la información que encuentre a su disposición sobre dicho dispositivo.

Requisito 4 - Identificación de los dispositivos: La aplicación deberá filtrar los dispositivos detectados, esto es, la detección de dispositivos ajenos a la aplicación (no incluidos en el fichero de datos) no influirá en el correcto funcionamiento de la misma.

Requisito 5 - Optimización del código: La aplicación deberá dedicar el mínimo tiempo posible en el proceso de identificación de un dispositivo Bluetooth, debido al más que probable alto número de dispositivos que se pueden encontrar disponibles en cualquier espacio público.

Requisito 6 - Presentación de datos: La aplicación deberá informar en tiempo real del estado de ejecución de la aplicación a través de la pantalla del dispositivo móvil.

Requisito 7 - Interfaz aplicación-usuario: El usuario podrá controlar en todo momento la ejecución de la aplicación mediante las teclas de selección y desplazamiento del dispositivo móvil.

Requisito 8 - Portabilidad: La aplicación deberá poder ejecutarse en la gran mayoría de dispositivos móviles disponibles en el mercado actual.

Requisito 9 - Uso de recursos del dispositivo: La ejecución de la aplicación no interferirá en el correcto funcionamiento del dispositivo móvil ocupando grandes cantidades de memoria en ejecución o mediante un alto uso del tiempo de procesador. El tamaño de la aplicación tampoco deberá ocupar mucho espacio de almacenamiento del dispositivo.

5.2 Diseño e implementación.

La aplicación BlueBrowse ha sido desarrollada mediante módulos, cada uno de ellos con una función específica siguiendo el paradigma de orientación a objetos de la programación en Java. Dicho paradigma de programación permitió que el desarrollo iterativo fuera más sencillo, pues facilitó la adición de capacidades a la aplicación de forma incremental. Una vez instalada, descargada y en estado de ejecución muestra al usuario información sobre los PDI y su posición exacta en la instalación en el instante en el que su señal Bluetooth es detectada. En las siguientes ilustraciones se muestra la interacción entre los distintos módulos y elementos que forman el sistema (Figura 26), y a continuación el diagrama de flujo de la aplicación (Figura 27).

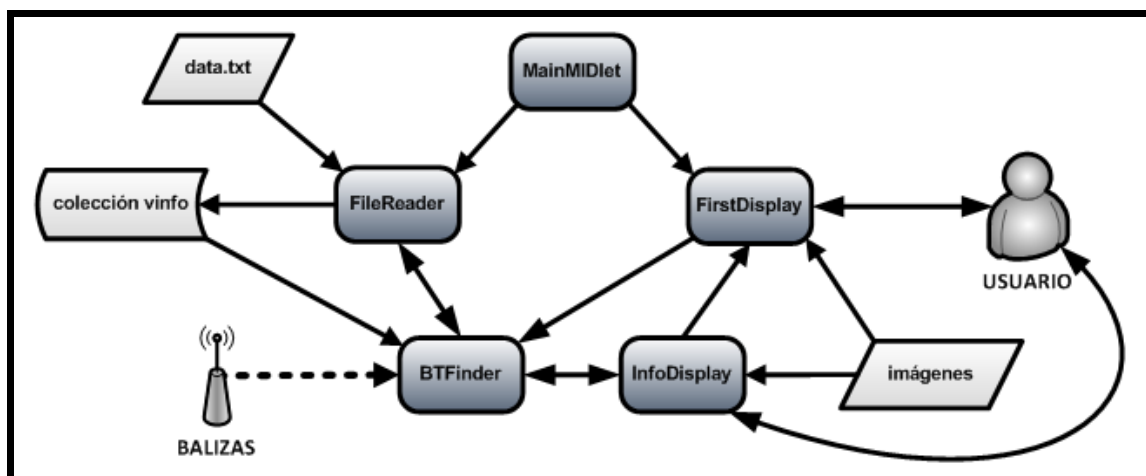


Figura 26 - Interacción entre los distintos módulos y elementos que componen el sistema.

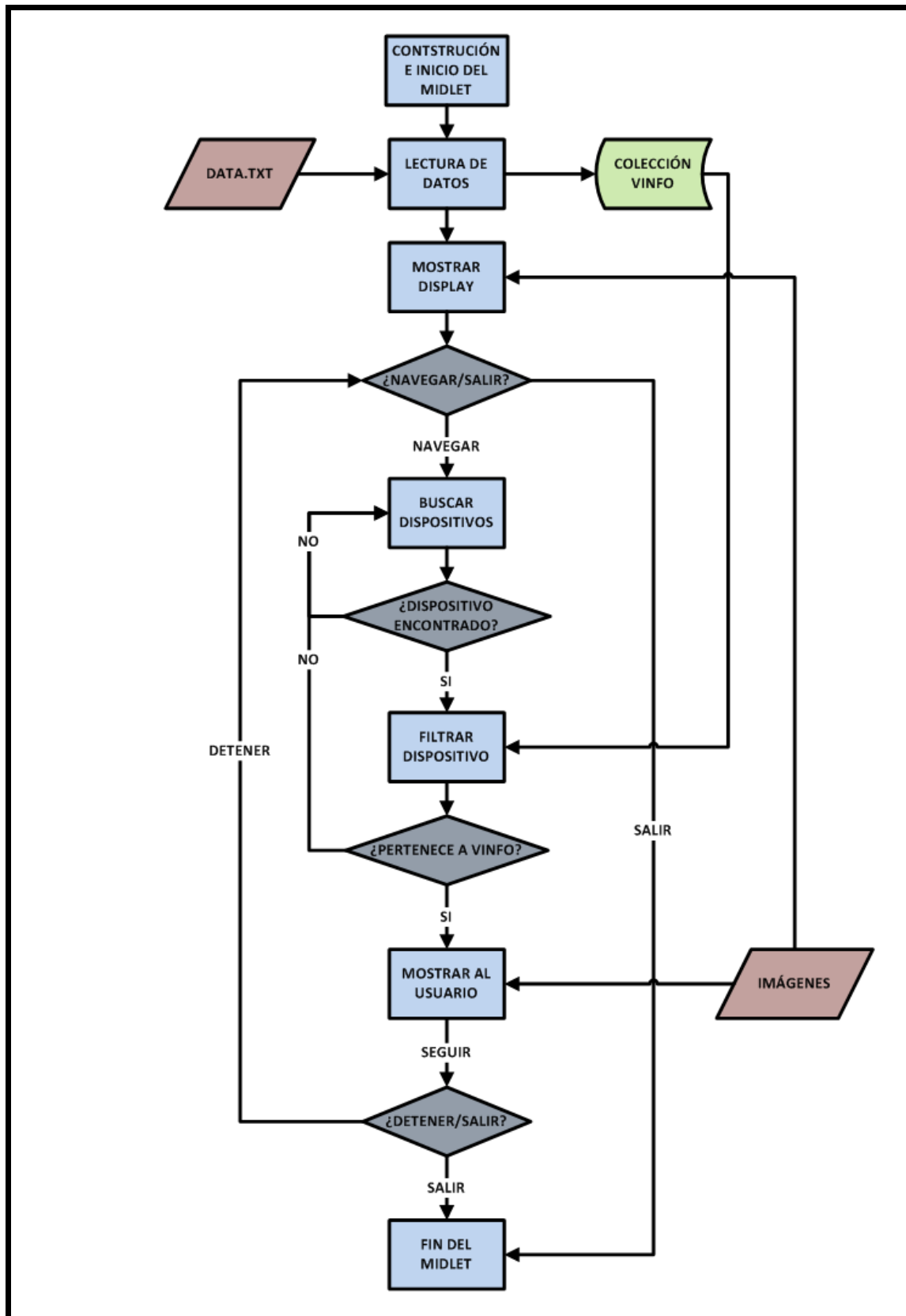


Figura 27 - Diagrama de flujo de la aplicación.

5.2.1 Módulo principal “MainMIDlet”

Este módulo es la parte principal de la aplicación. Como todos los MIDlets en él están implementados los métodos de activación, pausa y destrucción. Una vez iniciada la ejecución, se encarga de crear la pantalla inicial o de presentación mediante una instancia del módulo FirstDisplay así como del módulo lector FileReader (Código 4). Mediante la invocación de unos de los métodos de este último, al cual pasa como argumento el nombre del archivo, inicia la lectura del los PDI y toda su información.

```
//Constructor
public MainMIDlet() {
    pantalla = Display.getDisplay(this);
    fr = new FileReader(this);
    fd = new FirstDisplay(this);
    pantalla.setCurrent(fd);
}

public void startApp() {
    fr.readData(archivo);
}
```

Código 4 - Constructor del MainMIDlet y posterior lectura de datos

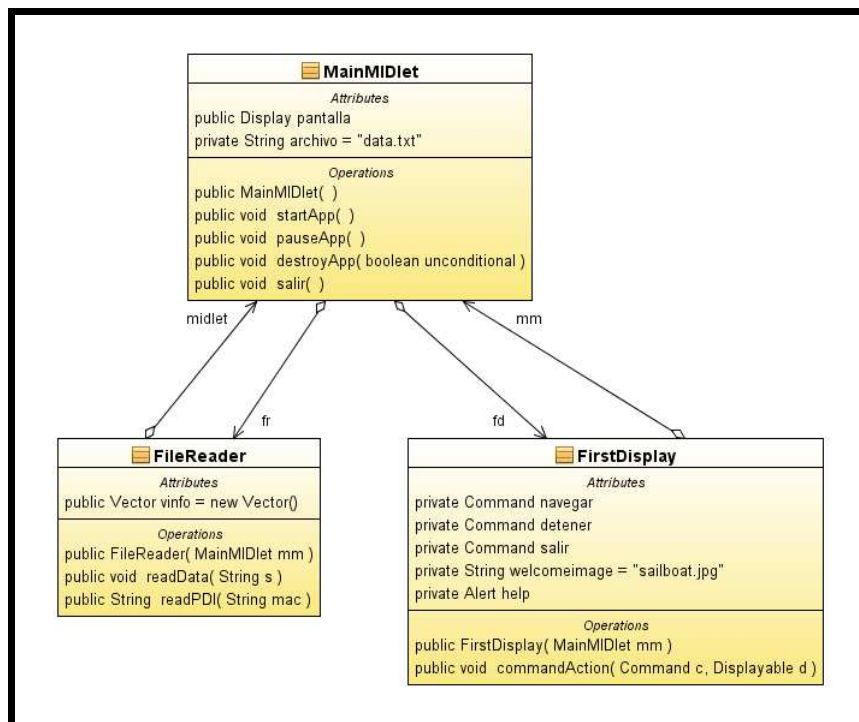


Figura 28 - Diagrama de clase de MainMIDlet y su relación con otros módulos.

5.2.2 Módulo lector “FileReader”

Este módulo implementa los métodos que manejan toda de la información referente a los PDI. Mediante el método `readData()` (Código 5) almacena toda la información del fichero pasado como argumento en un Vector cuyos elementos son objetos del tipo PDI.

En un principio se valoró que la aplicación pudiera renovar el archivo `data.txt` mediante un método que añadiera dicho recurso al archivo `.jar` ya instalado en el dispositivo móvil. Pero se desestimó al ser imposible modificar los contenidos del `jar` una vez instalado. La única opción por tanto sería descargar un nuevo MIDlet actualizado con nuevos datos sobre las balizas [HISP10].

Así pues, el archivo `data.txt` está formado por varias entradas (Figura 29), cada una correspondiente a toda la información de un PDI. No hay PDI repetidos, cada uno está referido unívocamente por su `mac` o “Bluetooth Address”. Por tanto el contenido de una de las entradas está conformado por los siguientes campos: la `mac` asignada al PDI (cadena 12 caracteres de tamaño fijo), el nombre (cadena de 30 caracteres también de tamaño fijo) y su información de interés (cadena de longitud variable).

```
try {
    StringBuffer sbinfo = new StringBuffer();
    int chrinfo = 0;
    String readmac = null;
    String readname = null;
    while ((chrinfo = isinfo.read()) != -1 ) {
        if (chrinfo != 95) {
            if(chrinfo!= 126){
                if (chrinfo != 124) {
                    sbinfo.append((char)chrinfo);
                } else {
                    readmac = sbinfo.toString();
                    sbinfo.delete(0,sbinfo.length());
                }
            }
        }
    }
}
```

Código 5 - Lectura del archivo `data.txt`.

Señalar que el contenido archivo se ajusta al código de representación de caracteres del formato ASCII, para favorecer su lectura, utilizando como caracteres especiales indicadores de la separación entre campos los siguientes:

- ‘|’ (ASCII 124) separador del campo mac y el campo nombre.
- ‘~’ (ASCII 126) separador del campo nombre y el campo texto.
- ‘_’ (ASCII 95) carácter de fin de entrada o PDI.

Por consiguiente, la estructura del archivo data.txt es similar a la siguiente:

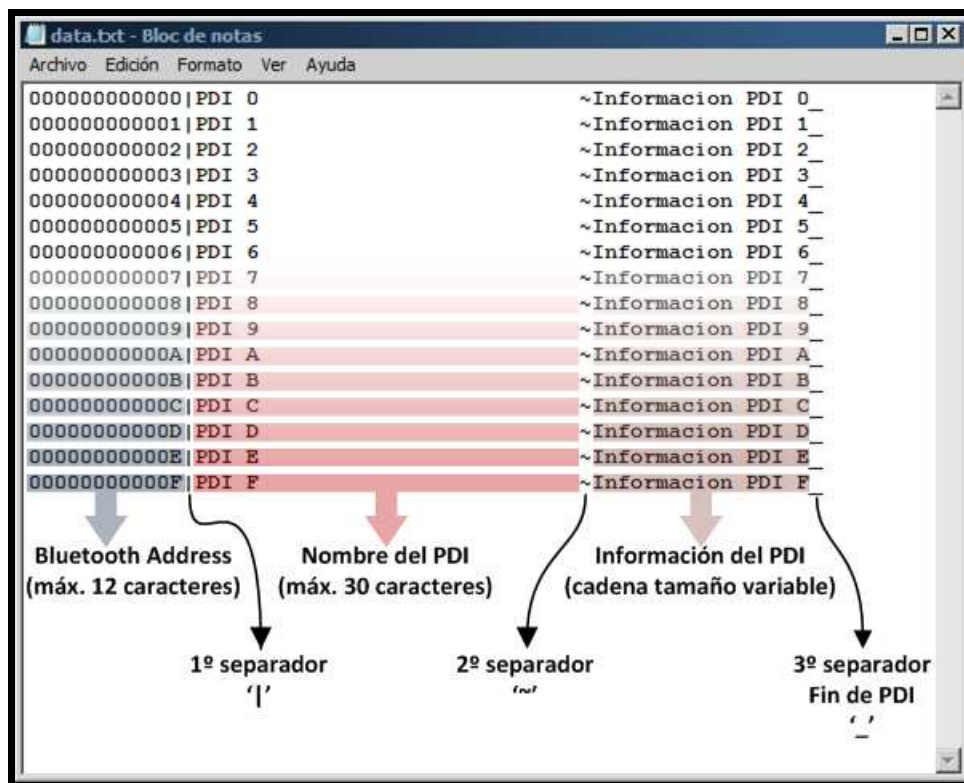


Figura 29 - Estructura del fichero de datos data.txt.

El método readPDI() tiene como misión devolver un objeto String con el nombre y el texto de interés referente al PDI, a partir otro objeto String que es una mac pasada como argumento de la función. Si no se encontrara dicha mac en El Vector de PDI almacenado, este método devuelve un String vacío "". Este método es invocado por el módulo BTFinder.

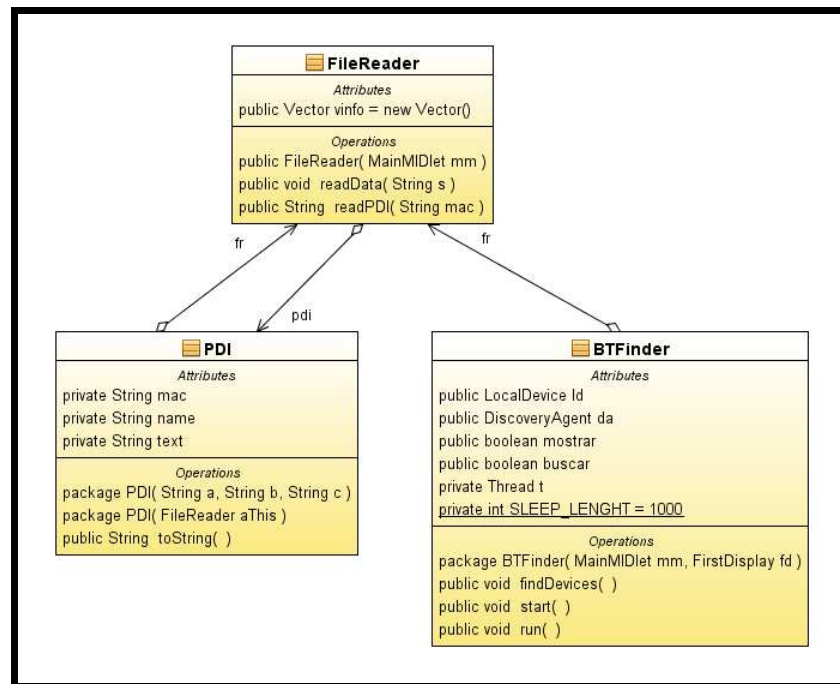


Figura 30 - Diagrama de clase de FileReader y su relación con otros módulos.

5.2.3 Clase "PDI"

Esta clase define los objetos que forman el Vector de módulo FileReader. Cada PDI está formado por tres Objetos String equivalentes a la mac, el nombre y el texto o información de interés sobre el punto posicionado por una baliza. Reemplaza el método toString() añadiendo espacios en blanco " " entre los campos.

```

public class PDI {

    private String mac;
    private String name;
    private String text;

    PDI( String a, String b, String c ) {
        mac = a;
        name = b;
        text = c;
    }
}

```

Código 6 - Clase PDI.

5.2.4 Módulo buscador “BTFinder”

Es el módulo encargado de la detección de los dispositivos Bluetooth cercanos al dispositivo móvil. Proporciona acceso al dispositivo Bluetooth local mediante el objeto `LocalDevice` para, posteriormente mediante el objeto `DiscoveryAgent` encapsular la funcionalidad del perfil Bluetooth SDAP que provee de los métodos necesarios para iniciar la búsqueda. Finalmente, El interfaz `DiscoveryListener` define las acciones a realizar cada vez que encuentre un dispositivo en su radio de alcance.

```
public void run() {  
    while(buscar) {  
        findDevices();  
        try {  
            Thread.sleep(SLEEP_LENGTH);  
        } catch (InterruptedException ie) {}  
    }  
}
```

Código 7 - método run() del Thread buscador de dispositivos Bluetooth.

Implementa a su vez, un Thread mediante el cual se consigue que la búsqueda se realice según un tiempo fijo definido de 1 segundo (Código 7), intervalo en el cual se validan los dispositivos detectados. Si el dispositivo es un PDI conocido y es posible mostrar la información, se da orden de ofrecerla por pantalla al usuario y se cancela la búsqueda de dispositivos. Este aspecto del programa presento una dificultad añadida durante su desarrollo, que en un principio parecía insalvable. Era esencial poder cambiar el tiempo de búsqueda de dispositivos que dedicaba el dispositivo local Bluetooth del móvil, lo cual se puede realizar mediante el acceso al Centro de Control Bluetooth o BCC responsable de controlar dicho dispositivo [PEDR04]. Desafortunadamente la especificación JSR-82 deja la implementación del BCC en manos de los proveedores de los terminales, no pudiendo ser cambiados por el usuario programador. La única solución posible entonces fue intentar iniciar e interrumpir la búsqueda según un intervalo de tiempo definido por nosotros, haciendo caso omiso a los 10 segundos de media que se obtenían al medir la duración de la búsqueda de dispositivos [ELEC10]. Durante las pruebas lógicas simuladas en el ordenador, se obtuvieron buenos resultados pues el método aplicado funcionaba correctamente, pero al trasladar la aplicación al

dispositivo móvil y realizar pruebas físicas no se apreció cambio alguno, pues la búsqueda de dispositivos mantenía una duración de 10 segundos aproximadamente, ignorando el código desarrollado.

```
public void deviceDiscovered(RemoteDevice rd, DeviceClass clase) {
    String readpdi = fr.readPDI(rd.getBluetoothAddress());
    if (!readpdi.equals("") & mostrar) {
        id.update(rd.getBluetoothAddress(), readpdi);
    }
    da.cancelInquiry(this);
}
```

Código 8 - método deviceDiscovered y cancelación de la búsqueda.

Tras diversas pruebas y bastante trabajo en un terminal Sony Ericsson K510i, se optó por cambiar a un dispositivo Nokia 2730 Classic. Fue entonces cuando se encontró la solución: todos los dispositivos Nokia mantenían también un tiempo de búsqueda aproximado de 10 segundos, sin embargo en ellos si es posible detener e iniciar búsquedas programadas sin tener que esperar al vencimiento del tiempo fijado por el proveedor (Código 8).

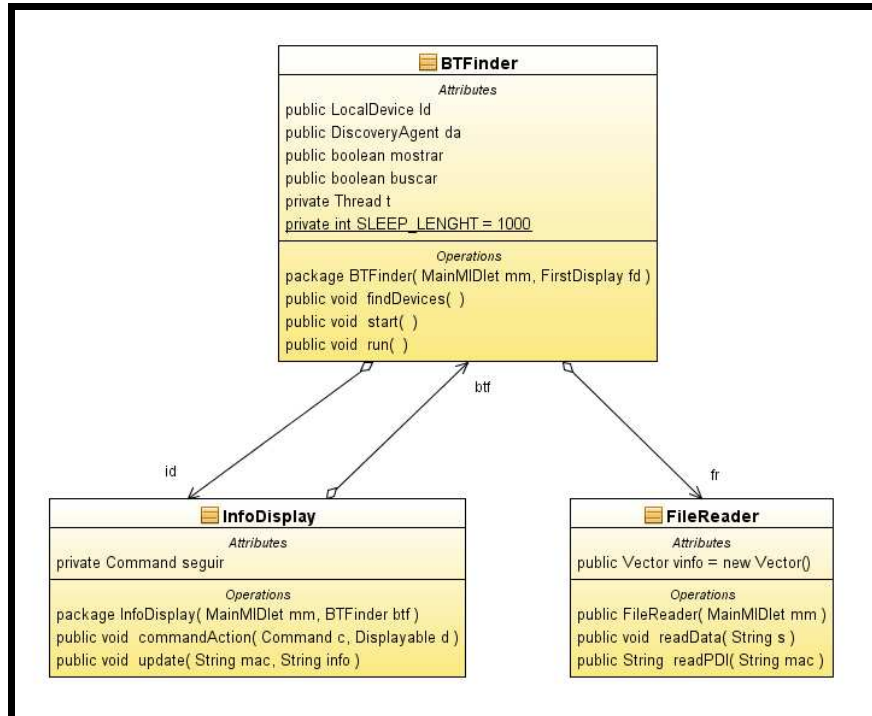


Figura 31 - Diagrama de clase de BTFinder y su relación con otros módulos.

5.2.5 Módulo pantalla inicial “FirstDisplay”

Este módulo presenta la pantalla de inicio de la aplicación en el dispositivo móvil, informando del estado de la búsqueda (por medio de un ticker) y manejando la ejecución del MIDlet a través de la acción realizada por el usuario mediante las teclas de selección como se observa en el Código 9.

```
public void commandAction (Command c, Displayable d) {

    if (c.getLabel().equals("Navegar")) {
        btbf.start();
        d.removeCommand(c);
        ticker.setString("Localizando Puntos de Interes");
        d.setTicker(ticker);
        d.addCommand(detener);
    } else if (c.getLabel().equals("Detener")){
        ticker.setString("Pulse Navegar para reiniciar la busqueda");
        btbf.buscar=false;
        btbf.mostrar = false;
        d.removeCommand(c);
        d.addCommand(navegar);
    } else{
        mm.salir();
    }
}
```

Código 9 - Manejo de las órdenes dadas por el usuario.

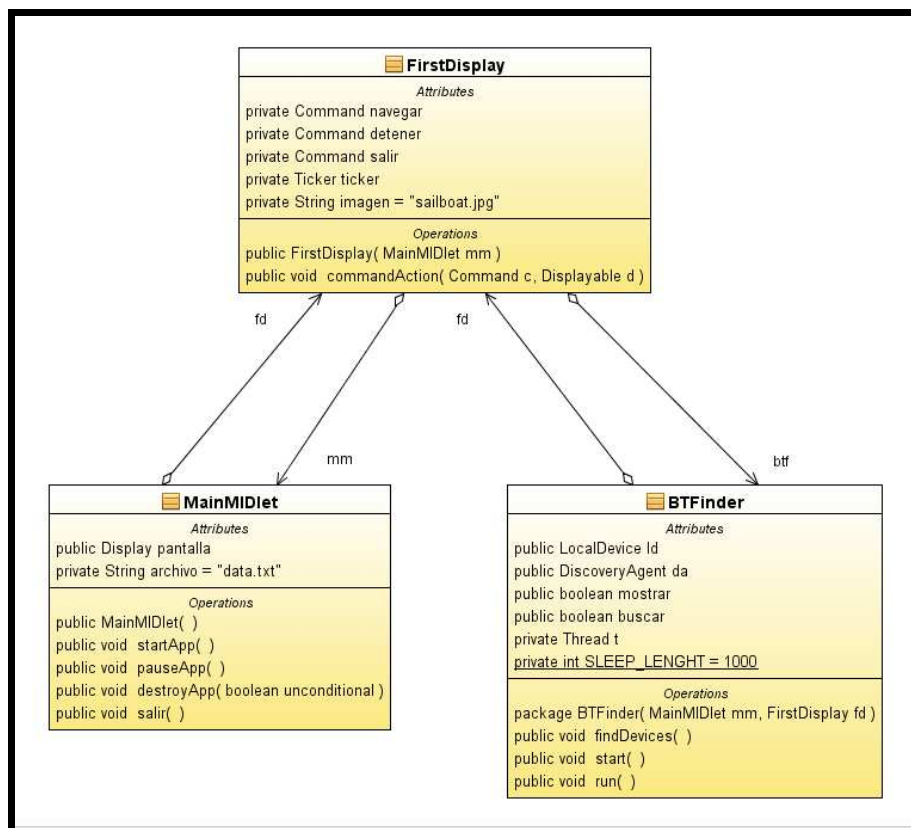


Figura 32 - Diagrama de clase de FirstDisplay y su relación con otros módulos.

5.2.6 Módulo pantalla de información “InfoDisplay”

El último módulo tiene como función la presentación de los datos del PDI detectado al usuario, permitiéndole su visionado mediante las teclas de desplazamiento o continuar con la navegación. Además de mostrar la información correspondiente al PDI visualiza dos imágenes: La primera es el aspecto del PDI y la segunda un grafico en el que se detalla la posición del PDI (y por tanto la del dispositivo móvil en dicho instante) dentro de la instalación. Estas dos imágenes están incluidas en el archivo .jar de la aplicación y se identifican mediante la mac de cada baliza y su extensión. Por ejemplo la baliza de mac “000000000000” tendrá asociada la imagen informativa “000000000000.jpg” y la imagen de posición “000000000000.jpeg” (Código 10).

```
this.setTitle(info.substring(0, 30));
InputStream isimage = getClass().getResourceAsStream(mac + ".jpg");
try {
    this.append(Image.createImage(isimage));
} catch (IOException ioe) {}
this.append(info.substring(30, info.length()));
InputStream issite = getClass().getResourceAsStream(mac + ".jpeg");
try {
    this.append(Image.createImage(issite));
} catch (IOException ioe) {}
mm.pantalla.setCurrent(this);
```

Código 10 - Inserción de imágenes en la aplicación.

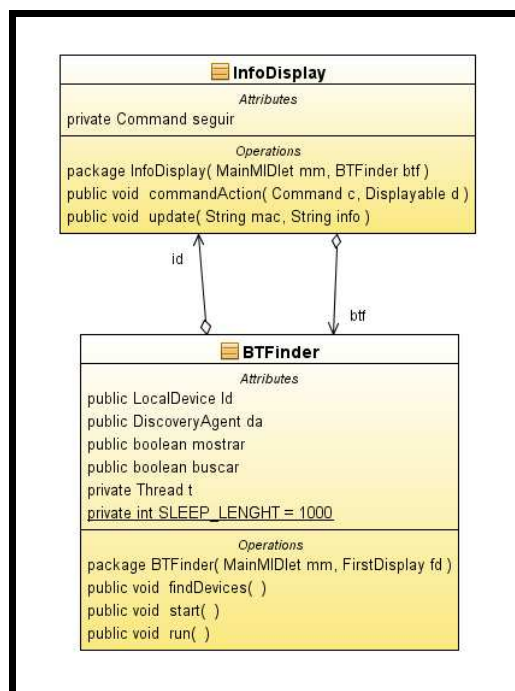


Figura 33 - Diagrama de clase de InfoDisplay y su relación con otros módulos.

6. RESULTADOS OBTENIDOS

6. Resultados obtenidos.

6.1 Pruebas lógicas.

Las pruebas lógicas del sistema se fueron realizando continuamente durante el proyecto, al tiempo que se incorporaban nuevas funcionalidades al sistema como corresponde en la programación orientada a objetos. Gracias al emulador Sony Ericsson SDK 2.5.0.6 fue sencillo comprobar el correcto funcionamiento, pues permitió simular las distintas pruebas requeridas mediante una instalación, manejo y configuración (Figura 34) realmente sencillos.

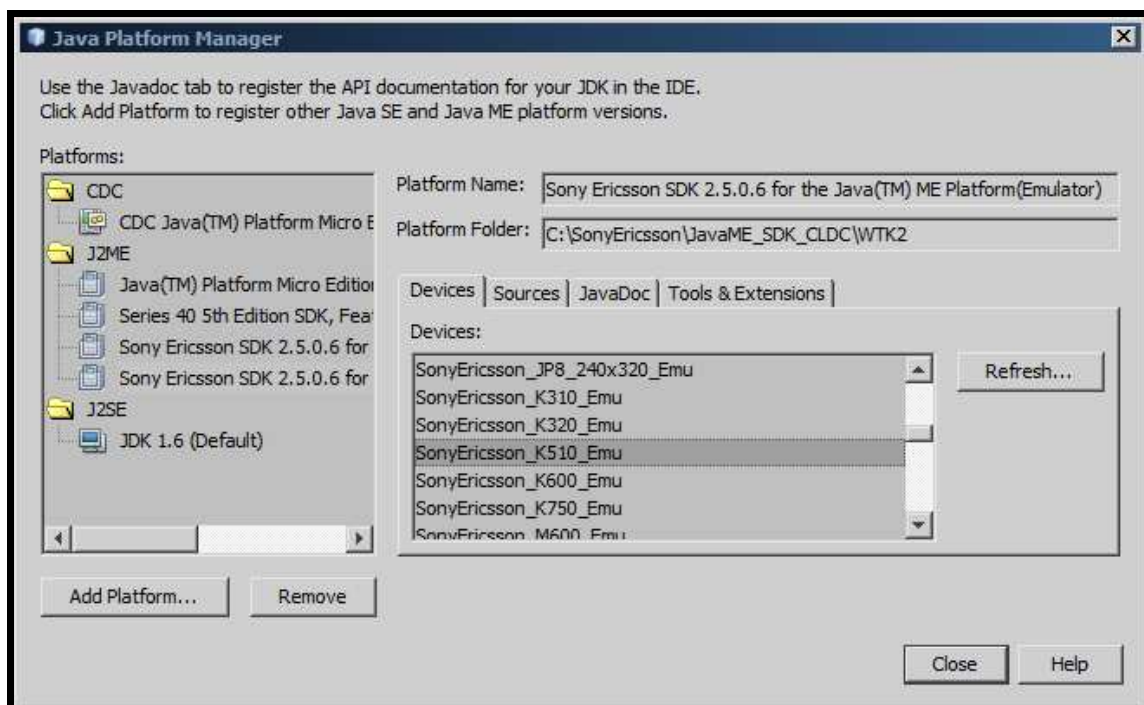


Figura 34 - Configuración del Emulador Sony Ericsson SDK 2.5.0.6.

La prueba lógica descrita a continuación servirá para verificar el correcto funcionamiento y como ejemplo de prueba, mediante la emulación de dos balizas (Figuras 35 y 36) y de un dispositivo móvil con la aplicación BlueBrowse (Figura 37).



Figura 35- Baliza 1



Figura 36 - Baliza 2

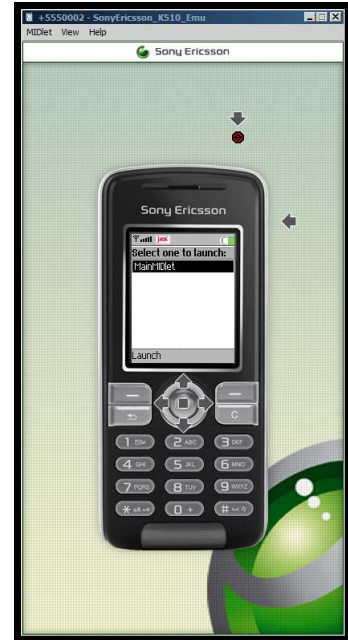


Figura 37 - BlueBrowse

Una vez emulados, procedemos a lanzar las respectivas aplicaciones obteniendo así las balizase (Figuras 38 y 39) iniciando el sistema de posicionamiento (Figura 40).



Figura 38 - Baliza 1



Figura 39 - Baliza 2



Figura 40 - BlueBrowse

Observando la salida por consola (Figura 41) se comprueba como la aplicación se ha iniciado con éxito leyendo tras tomar control del dispositivo local Bluetooth y ha almacenando el contenido del fichero data.txt. Posteriormente la pulsación por parte del usuario de la tecla de navegación provoca que tras breves instantes, sea localizada la baliza conocida. Sólo la mac de una de las balizas, la "0123456789AF", está incluida dentro de nuestro fichero de datos, por tanto el MIDlet tendrá que filtrar ambas mac y mostrar sólo información de esta como se advierte en la consola.

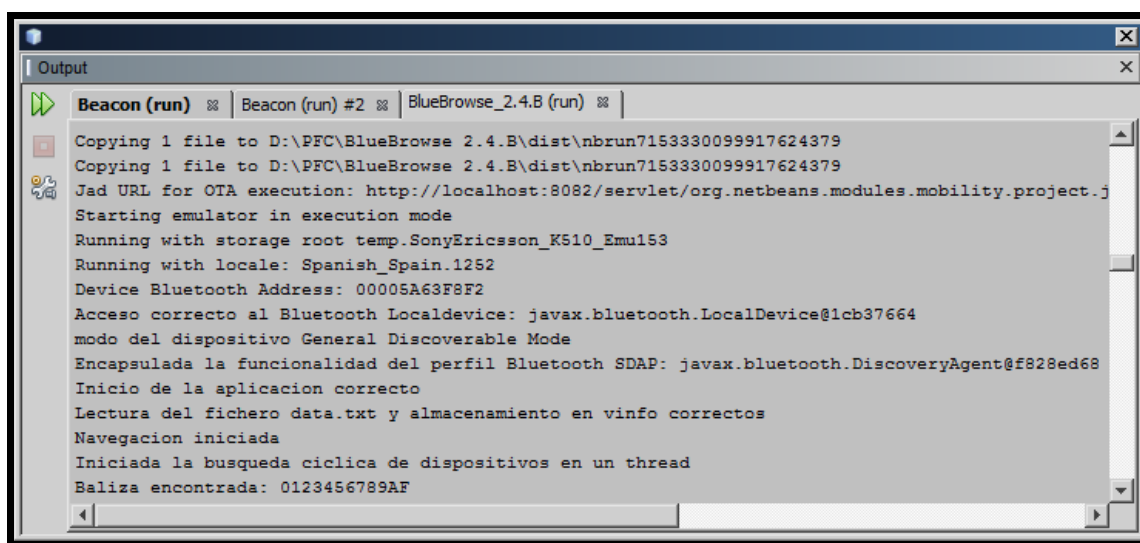


Figura 41 - Inicio de la ejecución y toma de control del dispositivo local Bluetooth

De forma simultánea y como es de esperar, en la pantalla del dispositivo móvil se muestra toda la información necesaria sobre la baliza: sus datos como PDI: la imagen del cuadro y su texto correspondiente, así como su posición exacta dentro del museo. En los instantes previos al descubrimiento un ticker avisa del estado de la búsqueda (Figura 42) , y posteriormente es presentada la información obtenida a partir de la mac detectada: una imagen del cuadro, el texto informativo sobre este y la imagen con la localización exacta en la sala del museo (Figuras 43, 44 y 45 respectivamente).



Figura 42 - Ticker informativo.



Figura 44 - Aspecto del PDI.



Figura 43 - Información del PDI.



Figura 45 - Localización en el entorno.

En todo momento la ejecución está controlada por el visitante pudiendo continuar con la navegación y detenerla en caso de que desee dejar de navegar sin abandonar la aplicación, así como reiniciarla. En la Figura 46 se puede comprobar cómo después de detener la navegación y apagar la baliza conocida, la aplicación sigue detectando la baliza "0000000DECAF" pero no muestra información alguna. Finalmente (Figura 47) termina con la ejecución al pulsar la tecla "Salir".

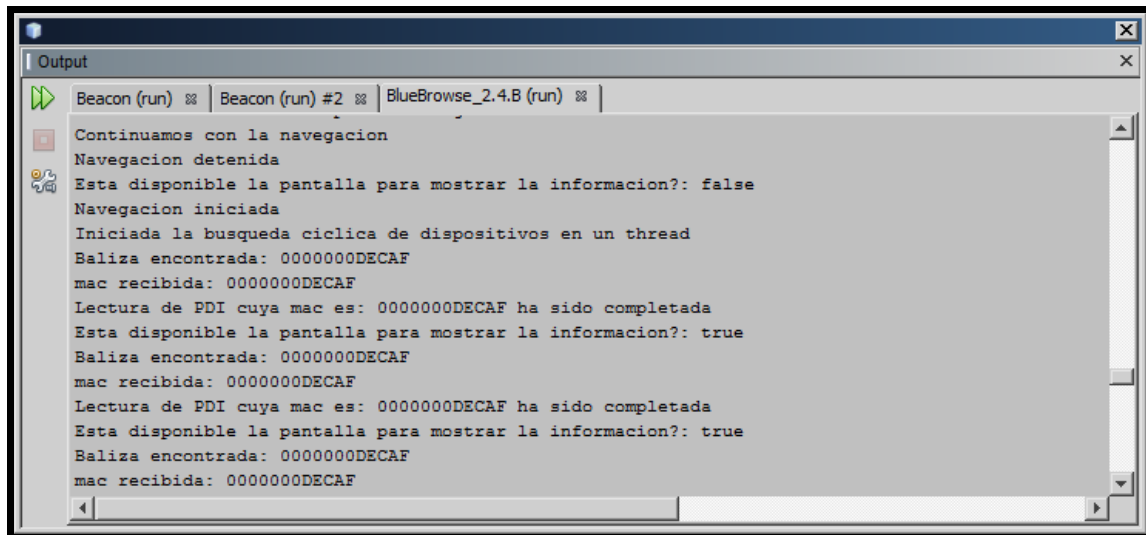


Figura 46 - Parada y reinicio de la aplicación tras la desactivación de una baliza conocida.

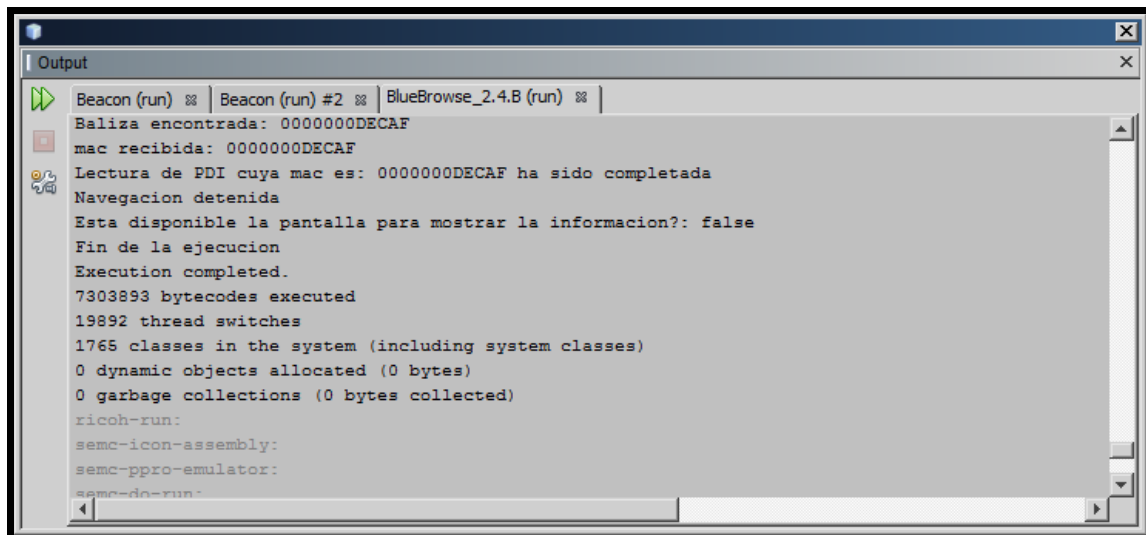


Figura 47 - Fin de la ejecución.

6.2 Pruebas físicas.

Para la realización de las pruebas físicas se dispuso de un entorno con las mismas características descritas y propuestas en el Capítulo 4 de esta memoria. Se procedió midiendo el tiempo que tardaba el MIDlet en detectar la baliza, estando el terminal móvil en una posición dentro del rango de alcance de la misma. Para cada baliza y su PDI correspondiente se realizaron cuatro mediciones de dicho tiempo. En la Figura 48 se aprecia las posiciones en la que se tomaron los tiempos

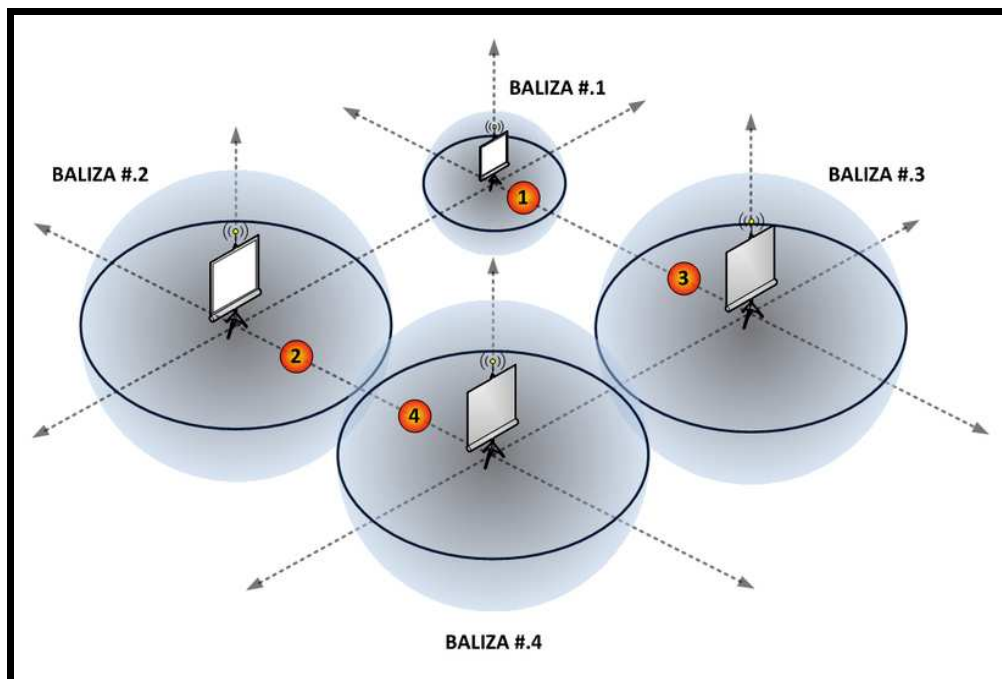


Figura 48 - Posiciones de medición de los tiempos de detección.

Apuntar también la inclusión de algunos dispositivos Bluetooth conocidos y desconocidos, no pertenecientes al conjunto de balizas a detectar durante la realización de las pruebas, forzando a la aplicación a filtrar los dispositivos detectados por el terminal móvil.

Friendly name	mac
T9	090926AC15A5
Paco	60D0A9058268
Nokia 2720a-2	0CDDEFC447C4

Tabla 11 - Dispositivos ajenos al sistema presentes en las pruebas

Los tiempos registrados se pueden observar a continuación en la siguiente tabla:

	1ª medición	2ª medición	3ª medición	4ª medición	Media	Desv. Típica
Baliza #.1	0,6	0,8	0,6	0,7	0,68	0,10
Baliza #.2	2,0	0,4	1,5	2,3	1,55	0,83
Baliza #.3	2,9	2,8	3,1	3,5	3,08	0,31
Baliza #.4	1,2	1,1	0,5	1,3	1,03	0,36

Tabla 12 - Tiempos tomados en las mediciones

La impresión de dichos datos en forma gráfica (Figura 49) proporcionó una mejor lectura de los resultados conseguidos. De nuevo fue el terminal Sony Ericsson el más destacado entre las demás balizas al ser el que mejor media y desviación típica consiguió en el tiempo de detección, con medidas siempre por debajo de 1 segundo. Este dato junto con la alta capacidad de parametrización de su rango de alcance lo convirtió finalmente en la baliza perfecta para el tipo de sistemas de posicionamiento, en interiores que definió este trabajo.

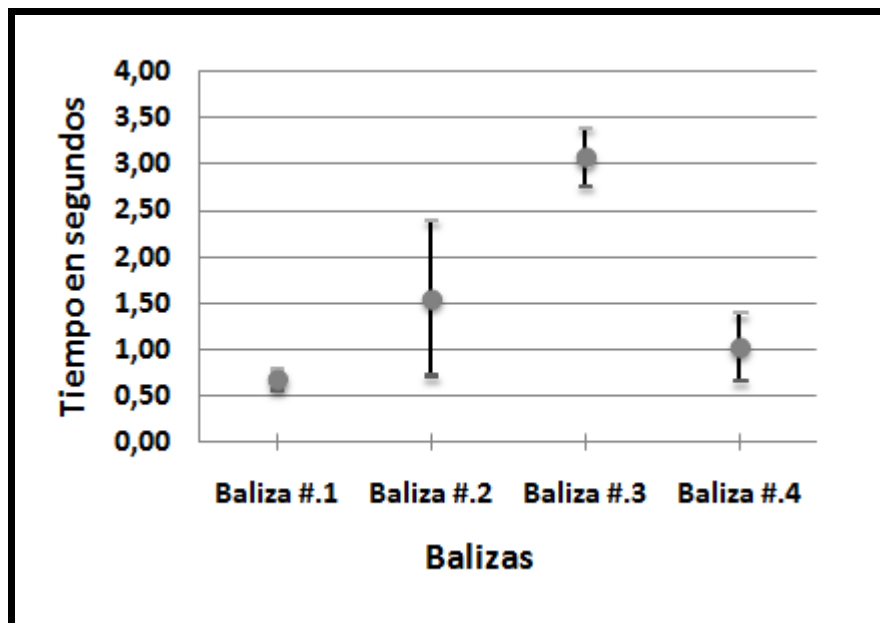


Figura 49 - Gráfica estadística de los tiempos medidos.

Como apunte final y a la vista de los resultados esperados y obtenidos, el sistema funcionó correctamente por lo que pudimos concluir que el desarrollo y los objetivos marcados al inicio de este proyecto fueron alcanzados con éxito.

7. CONCLUSIONES Y

TRABAJOS FUTUROS

7. Conclusiones y trabajos futuros.

Tras la realización de este proyecto, no nos queda duda sobre la futura importancia que los llamados entornos inteligentes tendrán en el futuro más próximo. Aunque su avance aun es tímido, poco a poco van apareciendo nuevos conceptos en dicho campo como el de la Figura 50. Su implantación y uso serán imprescindibles el día de mañana, al igual que lo es actualmente la necesidad de estar conectado al mundo en cualquier momento y lugar. El sistema desarrollado en este proyecto, demuestran que con medios sencillos, y sobre todo buenas e innovadoras ideas dicho avance está asegurado.

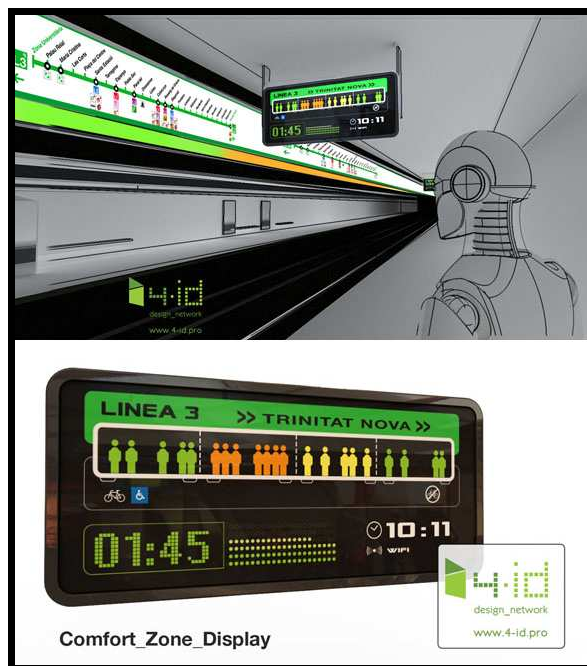


Figura 50 - Nuevos conceptos de entornos inteligentes.

En cuanto al posicionamiento en interiores, cabe señalar que son muchas las soluciones propuestas, pero pocas las que podemos encontrar fuera del ambiente universitario implementadas en la actualidad. La aplicación BlueBrowse ha dejado patente ser una posible solución, por lo que mediante un sistema profesional realizado ad hoc, con balizas diseñadas específicamente para ello y las posibilidades que brindan los actuales Smartphones, los resultados serían más que satisfactorios.

Un sistema de dichas características podría servir a usuarios discapacitados como apoyo en la realización de actividades cotidianas. Mediante el desarrollo de un interfaz auditivo, el uso de auriculares, y un algoritmo de navegación (Dijkstra, por ejemplo) se les podría facilitar el desplazamiento en metro. El usuario invidente indicaría a través del interfaz su destino dentro de la estación, y mediante la detección de las balizas se le guiaría por el camino a seguir, dándole aviso de obstáculos tales como escaleras mecánicas, ascensores, obras de mantenimiento, salidas, etc.

La tecnología J2ME sería una inmejorable plataforma para proyectarla. Aunque durante la gestación del sistema se apreció que el juego de herramientas en determinados momento podía resultar escaso, como en el caso de las colecciones de objetos, ciertamente el uso de esta plataforma y la programación mediante su IDE NetBeans han otorgado un gran dinamismo al desarrollo del trabajo.

En cuanto a la tecnología Bluetooth, su posición de estándar de facto global como protocolo de comunicación inalámbrica entre dispositivos no parece estar asegurada: en la actualidad el Bluetooth SIG ha anunciado la adopción formal de la especificación 4.0. Sólo queda apuntar que sería interesante que en un futuro se diera por parte de las compañías proveedoras de los terminales móviles, la posibilidad al usuario programador dar acceso algunas características de la pila Bluetooth de los dispositivos, pues facilitaría considerablemente su labor.

8. BIBLIOGRAFÍA

8. Bibliografía.

[ALBE05] Albert Huang. Artículo - The use of Bluetooth in linux and location aware Computing. Año 2005.

[IVAN05] Iván Marsá Maestre, Miriam Machuca, Andrés Navarro y Juan R. Velasco. Artículo - Un enfoque práctico para la localización de usuarios mediante Bluetooth en entornos domóticos. Año 2005.

[KENE06] Kenneth C. Cheung, Stephen S. Intille, and Kent Larson. Artículo - An Inexpensive Bluetooth-Based Indoor Positioning Hack. Año 2006.

[NIGE09] Nigel Davies, Adrian Friday, Peter Newman, Sarah Rutledge and Oliver Storz. Artículo - Using Bluetooth Device Names to Support Interaction in Smart Environments. Año 2009.

[PEDR04] Pedro Daniel Borches Juzgado. Java 2 Micro Edition: Soporte Bluetooth. Año 2004.

[SERG03] Sergio Gálvez Rojas y Luís Ortega Díaz. Java a tope: J2ME (Java 2 Micro Edition) edición electrónica. Año 2003.

[TIMO08] Timothy J. Thompson, Paul J. Kline and C. Bala Kumar. Bluetooth application programming with the Java APIs essentials edition. Morgan Kaufmann publishers. Año 2008.

Referencias de Internet citadas:

[4-ID10] 4-ID blog. 4-ID creative network meeting point. Comfort Zone Display. <http://4-id.org/blog4id/?p=163> visitada: 2/5/2010.

[BLUE10] Sourcing gate. Especificaciones del auricular Bluetooth Bluedio T9. <http://www.sourcinggate.com> visitada: 25/3/2010.

[CAMB10] University of Cambridge. Computer Laboratory Digital Technology Group. The Bat System. <http://www.cl.cam.ac.uk/research/dtg/research/wiki/BatSystem> visitada: 8/3/2010.

[ELEC10] electroSystems. Un Blog para todos... Medir tiempo de ejecución en Java. <http://electrosystemss.blogspot.com> visitada: 23/4/2010.

[HISP10]. JavaHispano. Foro J2ME. Añadir archivos de recursos a un MIDlet ya instalado. <http://www.javahispano.org> visitada: 17/5/2010.

[JAVA10] Oracle. For Java Developers. <http://www.oracle.com/technetwork/java/index.html> visitada: 15/1/2010.

[MIT10] Massachusetts Institute of Technology. The Cricket Indoor Location System. <http://cricket.csail.mit.edu/#overview> visitada 13/3/2010.

[NOKI10] Nokia connecting people. Especificaciones del teléfono móvil Nokia 2730 Classic. <http://www.forum.nokia.com> visitada: 25/4/2010.

[PROG10] Wikipedia. Programación extrema. [http://es.wikipedia.org/wiki/Programaci3n extrema](http://es.wikipedia.org/wiki/Programaci3n_extrema) visitada: 4/1/2010.

[SONY10] Sony Ericsson. Especificaciones del Sony Ericsson K510i. <http://www.sonyericsson.com> visitada: 2/2/2010

[SYME10] Symeo Absolute positioning. LPR-1D Distance measurement and anti-collision. <http://www.symeo.com> visitada: 13/3/2010.

[TRIL10] Wikipedia. Trilateración <http://es.wikipedia.org/wiki/Trilateraci3n> visitada: 15/1/2010.

ANEXOS

Anexo A - Manual de instalación.

La instalación de la aplicación BlueBrowse en dispositivos móviles cuyas características cumplan con los requisitos necesarios para su ejecución, es muy sencilla. Una vez compilada, preverificada y empaquetada la aplicación BlueBrowse 2.4 se obtiene un archivo de extensión .jar en la carpeta dist del proyecto. Este archivo BlueBrowse_2.4.jar contiene todos los recursos que usa la aplicación. Mediante la conexión de un dongle Bluetooth al PC y la activación de la opción Bluetooth en el terminal móvil, seremos capaces de enviar e instalar en el dispositivo la aplicación.

1. Envío del .archivo BlueBrowse.jar a un móvil mediante un dongle Bluetooth.

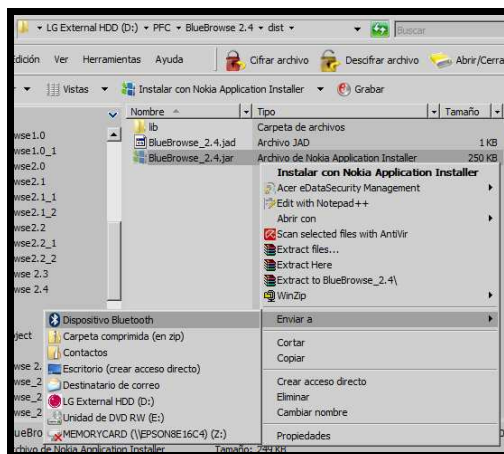


Figura 51 - Inicio del envío.

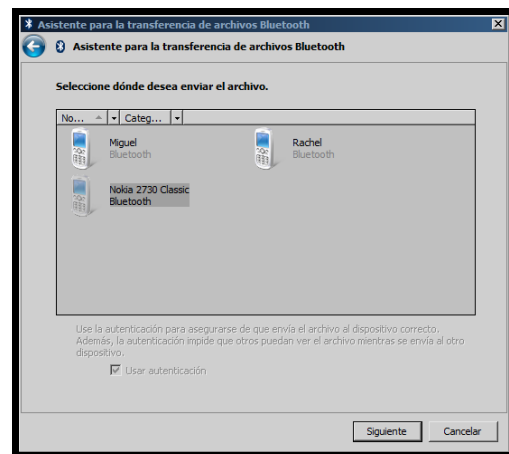


Figura 52 - Selección del destino.

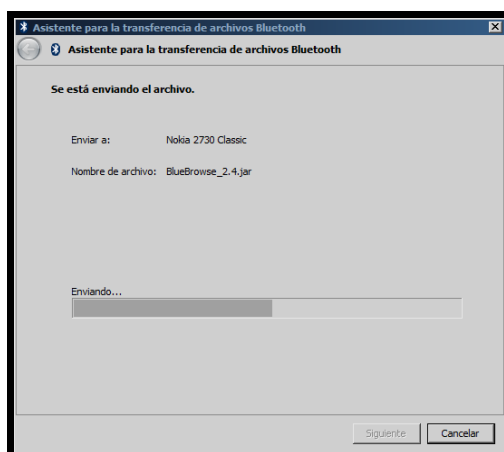


Figura 53 - Enviando la aplicación.

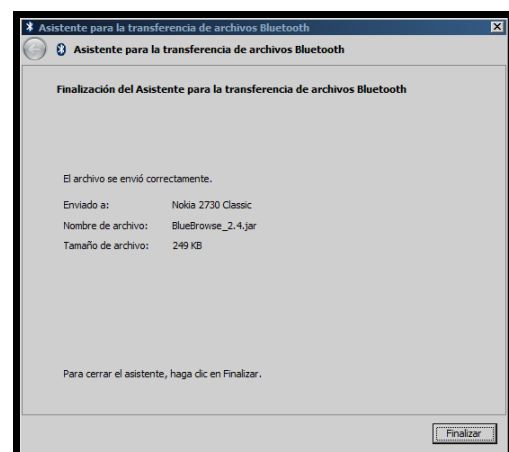


Figura 54 - Envío completado

2. Recepción e instalación en el dispositivo móvil. Por defecto la aplicación BlueBrowse se guarda en la subcarpeta “Juegos” de las aplicaciones del terminal, pudiendo ser reubicada en cualquier otra carpeta según el criterio del usuario



Figura 55 - Recepción de la aplicación.



Figura 56 - Recepción finalizada.

Anexo B - Manual de ejecución.

La ejecución es totalmente intuitiva, de fácil manejo y muestra en tiempo real el estado en el que se encuentra.

1. Comienzo de la aplicación. Una vez recibida e instalada la aplicación BlueBrowse, se puede proceder a su ejecución. Como se apuntó en el Anexo A, en la Figura 58 se observa como la aplicación ha sido reubicada en la subcarpeta “Colección” de la carpeta “Aplicaciones” del terminal móvil.



Figura 57 - Nokia 2730 Classic.



Figura 58 - Aplicación BlueBrowse.

2. Pulsando la tecla de selección “Abrir” se inicia el programa. Si el dispositivo Bluetooth del teléfono no está disponible da la opción de activarlo (Figura 59) con antelación al inicio del MIDlet. Si está habilitado, después de un brevísimo tiempo de carga (Figura 70), aparecerá antes nosotros la pantalla de bienvenida (Figura 71).



Figura 59 - Activación Bluetooth.



Figura 60 - Carga de la aplicación.



Figura 61 - Aplicación BlueBrowse 2.4.

3. Pulsando la tecla de selección “Navegar” comienza la Búsqueda de balizas (Figura 62). En todo momento un “ticker” nos da aviso del estado en el que se encuentra la ejecución (Figura 63).



Figura 62 - Pantalla de inicio.



Figura 63 - Localizando PDIs.

4. En el momento que sea detectado un punto de interés gracias a su baliza, será presentada toda la información archivada en el fichero data.txt presente en el .jar, además de la posición exacta dentro del museo. Las siguientes Figuras (64 - 68) se muestra la detección de dos balizas distintas. Toda la información es accesible mediante las teclas de scroll:



Figura 64 - Aspecto del PDI localizado.

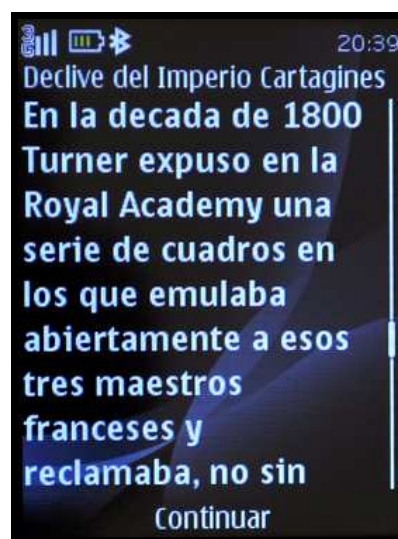


Figura 65 - Información del PDI localizado



Figura 66 - Localización del PDI detectado.



Figura 67 - Aspecto de otro PDI localizado.



Figura 68 - Localización del otro PDI detectado.

5. Si el usuario desea interrumpir por algún motivo la navegación sin abandonar la aplicación (Figura 69), puede pausar la ejecución pulsando la tecla “Detener”.

Por último en la Figura 70 se aprecia como al pulsar la tecla “Salir” se da por finalizada la ejecución de la aplicación BlueBrowse.



Figura 69 - BlueBrowse detenido.

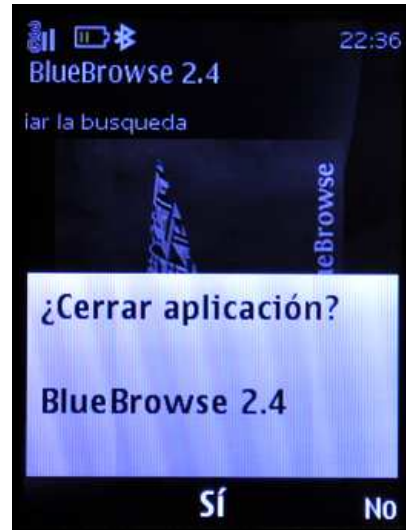


Figura 70 - Fin de la aplicación.

ANEXO C - Planificación.

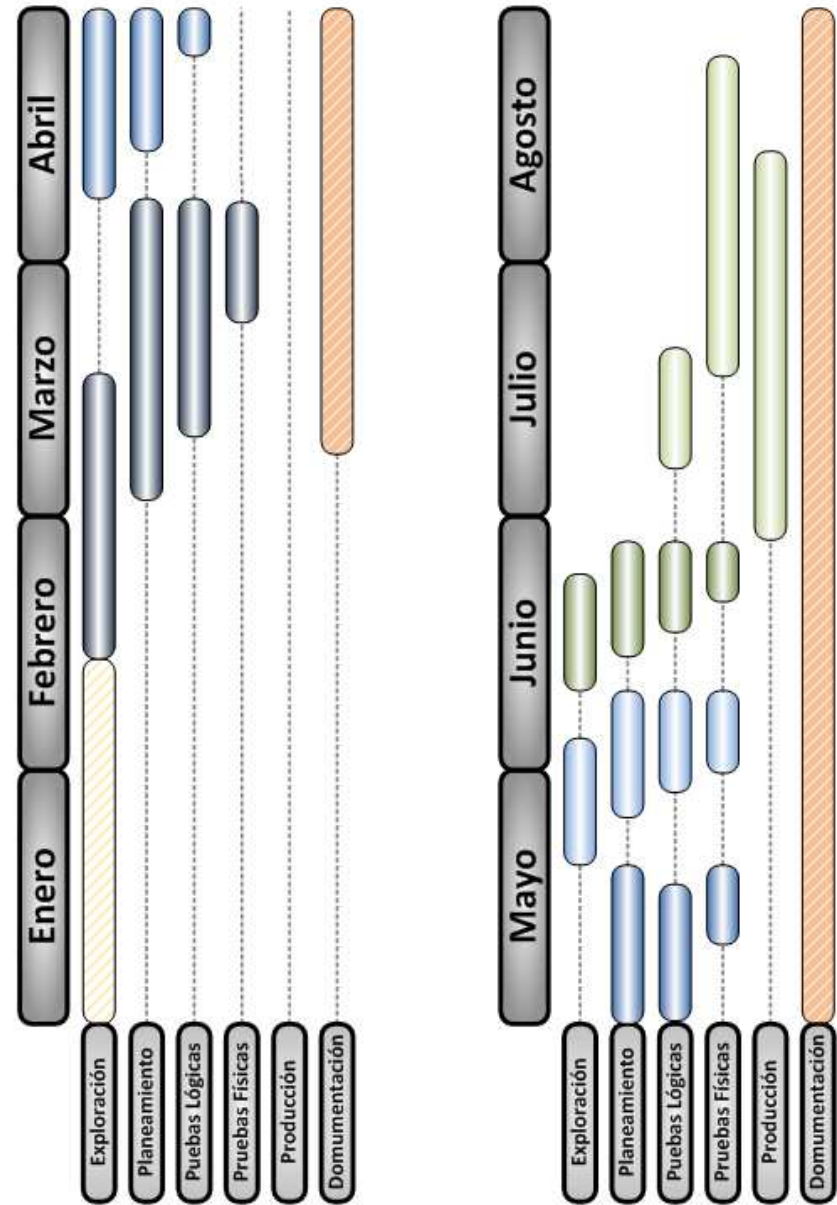
Este proyecto se elaboró durante el curso 09/10, entre los meses de Enero y Agosto. La metodología utilizada, perteneciente a las llamadas metodologías de desarrollo software ágiles, promovió la siguiente planificación cuyas actividades se definen a continuación:

- Exploración: planteamiento de los requisitos necesarios para la realización de cada módulo o tarea.
- Planeamiento: desarrollo de cada uno de los módulos software que conforman la aplicación final.
- Pruebas lógicas: control el funcionamiento de cada módulo o del producto final mediante simulación en PC.
- Pruebas físicas: control el funcionamiento de cada módulo o del producto final mediante su implementación en un teléfono móvil.
- Documentación de cada una de las tareas que se van realizando.

Por otro lado, las tareas o módulos desarrollados fueron las siguientes:

8. Adquisición de conocimiento:
9. BTFinder: módulo software.
10. FileReader + PDI: módulos software.
11. FirstDisplay: módulo software.
12. InfoDisplay: módulo software.
13. BlueBrowse: producto software final.
14. Redacción de la memoria: realización de este documento

Planificación



ANEXO D - Valoración económica.

El presupuesto final de este proyecto referido a continuación ha sido desglosado tras una valoración económica real en tres partes: costes de desarrollo, costes tecnológicos y costes de implantación.

Los primeros engloban todo el material utilizado en su realización, tanto hardware como software.

Recurso	Tipo	Precio
Acer Aspire 5100	Hardware	700 €
Adaptador USB Bluetooth	Hardware	7,02 €
Auricular Bluetooth Bluedio T9*	Hardware	3,52 €
Java™ ME Software Development KIT 3.0	Software	0 €
MS Office Professional 2007	Software	614,14 €
MS Windows Vista™	Software	175 €
Netbeans 6.8 IDE	Software	0 €
Nokia 2630*	Hardware	63 €
Nokia 2730 Classic*	Hardware	89 €
Nokia 5000d-2*	Hardware	98 €
Papel de Aluminio	Hardware	1,30 €
Samsung GT-S5600	Hardware	203,48 €
Sony Ericsson K510i	Hardware	89 €
Sony Ericsson SDK 2.5.0.6	Software	0 €
TOTAL		2.036,44 €

* Producto prestado o de segunda mano.

Tabla 13 - Coste del material utilizado en el proyecto.

Los costes de desarrollo incluyen el salario de los trabajadores necesarios para la correcta finalización del sistema.

	Horas	Coste/Hora	Salario
Director de Proyecto	17	80,00 €/hora	1360,00 €
Jefe de Proyecto	30	75,00 €/hora	2250,00 €
Analista	75	55,00 €/hora	4125,00 €
Programador	100	50,00 €/hora	5000,00 €
Diseñador	60	60,00 €/hora	3600,00 €
TOTAL			16.335,00 €

Tabla 14 - Coste de desarrollo del proyecto.

Finalmente los costes de implantación recogen el gasto de una supuesta implantación de la aplicación en un espacio público englobando costes tecnológicos y de desarrollo. Requiriendo dicho espacio un total de 20 balizas Bluetooth para el correcto posicionamiento, se podría utilizar como balizas pasivas auriculares BT. El coste final

Recursos Materiales	Tipo	Unidades	Precio Unidad	Precio Total
Acer Aspire 5100	Hardware	1	700 €	700 €
Adaptador USB Bluetooth	Hardware	1	7,02€	7,02€
Auricular Bluetooth Bluedio T9*	Hardware	20	3,52 €	70,4 €
Java™ ME Software Development KIT 3.0	Software	1	0 €	0 €
MS Office Professional 2007	Software	1	614,14 €	614,14 €
MS Windows Vista™	Software	1	175 €	175 €
Netbeans 6.8 IDE	Software	1	0 €	0 €
Papel de Aluminio	Hardware	1	1,30 €	1,30 €
TOTAL				1.561,84 €

Recursos Humanos	Horas	Coste/Hora	Salario
Director de Proyecto	17	80,00 €/hora	1360,00 €
Jefe de Proyecto	30	75,00 €/hora	2250,00 €
Analista	75	55,00 €/hora	4125,00 €
Programador	100	50,00 €/hora	5000,00 €
Diseñador	60	60,00 €/hora	3600,00 €
Técnico instalador	24	35,00 €/hora	840,00 €
TOTAL			17.175,00 €

Tipo de coste	Precio
Recursos Materiales	1.561,84 €
Recursos Humanos	17.175,00 €
TOTAL COSTE IMPLANTACIÓN	18.736,84 €

Tabla 15 - Costes de implantación para una solución que requiera 20 balizas.