

**DESARROLLO DE JUEGO MULTIJUGADOR BLUETOOTH PARA
DISPOSITIVOS MÓVILES CON J2ME**

RODIAN HERNANDO LAMUS BRAVO

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2006**

**DESARROLLO DE JUEGO MULTIJUGADOR BLUETOOTH PARA
DISPOSITIVOS MÓVILES CON J2ME.**

RODIAN HERNANDO LAMUS BRAVO.

**Proyecto de grado para optar al título de
Ingeniero de Sistemas e Informática**

**Director
Dr. SERGIO FERNANDO CATILLO CASTELBLANCO
Ingeniero de Sistemas, PhD**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERIAS FISICO MECANICAS
ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA
BUCARAMANGA
2006**

RESUMEN

TITULO

DESARROLLO DE JUEGO MULTIJUGADOR BLUETOOTH PARA DISPOSITIVOS MÓVILES CON J2ME*.

AUTORES

RODIAN HERNANDO LAMUS BRAVO**.

PALABRAS CLAVE

Computación Móvil, Bluetooth, Java 2 Micro Edition, Juego, Multijugador, Dispositivos móviles.

DESCRIPCIÓN

El Grupo de Ingeniería Telemática y Sistemas Inteligentes (GITSI) de la Escuela de Ingeniería de Sistemas e Informática (EISI) de la Universidad Industrial de Santander (UIS), consciente del surgimiento de nuevas tecnologías en el ámbito de la Computación Móvil, y de la necesidad de exploración e investigación de las mismas, pretende la generación de conocimiento que permita encontrar nuevas formas de aplicación de dichas tecnologías que den solución a problemas de la vida cotidiana ó que sencillamente faciliten el estilo de vida de las personas.

Este proyecto ha sido desarrollado con el propósito de llevar a cabo un estudio acerca de la tecnología Bluetooth y del entorno de desarrollo Java 2 Micro Edition (J2ME). El texto aborda las características técnicas de la especificación Bluetooth, además hace un estudio de la plataforma que ofrece Java para el desarrollo de aplicaciones para dispositivos móviles. Para tal efecto se diseño un juego multijugador, para el caso particular de dos jugadores, y basado en él, se desarrolló una aplicación para dispositivos móviles dotada de comunicación Bluetooth, utilizando J2ME como plataforma de desarrollo y Prototipado Evolutivo como metodología de desarrollo. La aplicación desarrollada permite al usuario, a través de un teléfono celular con soporte Java y conectividad Bluetooth, iniciar una partida del juego ó conectarse a una iniciada por otro usuario.

* Trabajo de Grado

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática.
Director Sergio Fernando Castillo Castelblanco.

ABSTRACT

TITLE

MULTIPLAYER BLUETOOTH GAME DEVELOPMENT FOR MOBILE DEVICES WITH J2ME*.

AUTHORS

RODIAN HERNANDO LAMUS BRAVO**.

KEY WORDS

Mobile Computing, Bluetooth, Java 2 Micro Edition, Game, Multiplayer, Mobile Devices.

DESCRIPTION

Mobile computing is one of the research topics of the Grupo de de Ingeniería Telemática y Sistemas Inteligentes (GITSI) at the Escuela de Ingeniería de Sistemas e Informática (EISI) at the Universidad Industrial de Santander (UIS). We want to generate knowledge that allow to find application's new ways of the new technologies, which resolve the problems of the daily life or just make easy the people life style.

This work studies both the Bluetooth Wireless Technology and the Java 2 Micro Edition (J2ME) development environment. The paper covers the technical characteristics of Bluetooth Specification, and the Java development platform for mobile devices applications. As a result, a multiplayer mobile game has been designed, in the two players' particular case. The game uses Bluetooth wireless technology like transmission means. Both the Evolutionary prototype Software development methodology and the J2ME development platform have been used in the carrying out of this project. The software allow to user with in a mobile phone with Java capability and Bluetooth connectivity, to start a match or to join to a started match by another user.

* Thesis

** Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director Sergio Fernando Castillo Castelblanco.

TABLA DE CONTENIDO.

INTRODUCCIÓN	1
1. PRESENTACION	3
1.1. DESCRIPCIÓN DEL DOCUMENTO	3
1.2. OBJETIVOS	4
1.2.1 OBJETIVO GENERAL	4
1.2.2 OBJETIVOS ESPECÍFICOS	5
1.3 JUSTIFICACIÓN	6
1.4 IMPACTO Y VIABILIDAD	8
1.4.1 IMPACTO	8
1.4.2 VIABILIDAD	8
1.5 SOPORTE	9
1.5.1 SOPORTE HARDWARE	9
1.5.2 SOPORTE SOFTWARE	10
1.5.2.1 <i>Java 2 JDK 1.5</i>	10
1.5.2.2 <i>Java 2 JRE 1.5</i>	10
1.5.2.3 <i>NetBeans IDE 5.0</i>	10
1.5.2.4 <i>NetBeans Mobility Pack 5.0</i>	11
1.5.2.5 <i>Nokia Developer's Suite 3.0 for J2ME</i>	12
1.5.2.6 <i>Nokia Connectivity Framework</i>	12
1.5.2.7 <i>Nokia PC Suite</i>	13
2. MARCO TEÓRICO	14
2.1. COMPUTACIÓN MÓVIL	14
2.1.1 INTRODUCCIÓN	14
2.1.2 CONCEPTUALIZACIÓN	15
2.1.3 VENTAJAS Y DESVENTAJAS	16
2.2 BLUETOOTH	17
2.2.1 INTRODUCCIÓN	17
2.2.2 HISTORIA	19
2.2.3 ARQUITECTURA	20
2.2.3.1 <i>Radio Bluetooth</i>	22
2.2.3.2 <i>Banda Base</i>	23
2.2.3.3 <i>Link Manager Protocol (LMP)</i>	23
2.2.3.4 <i>Host Controller Interface (HCI)</i>	23
2.2.3.5 <i>Audio Bluetooth</i>	24
2.2.3.6 <i>Logical Link Control and Adaption Protocol (L2CAP)</i>	24
2.2.3.7 <i>Service Discovery Protocol (SDP)</i>	24
2.2.3.8 <i>RFCOMM</i>	24

2.2.3.9	<i>Protocolos de Control de Telefonía</i>	25
2.2.3.10	<i>Protocolos Adoptados</i>	25
2.2.4	TOPOLOGÍA	25
2.2.5	DESCUBRIMIENTO DE DISPOSITIVOS BLUETOOTH	27
2.2.6	PERFILES BLUETOOTH	29
2.2.7	SEGURIDAD	32
2.2.7.1	<i>Modos de Seguridad</i>	32
2.2.7.2	<i>Autenticación</i>	33
2.2.7.3	<i>Encriptación</i>	35
2.2.7.4	<i>Autorización</i>	35
2.2.8	VENTAJAS Y DESVENTAJAS	36
2.2.9	APLICACIONES	37
2.3	JAVA 2 MICRO EDITION (J2ME)	38
2.3.1	INTRODUCCIÓN	38
2.3.2	ARQUITECTURA	39
2.3.2.1	<i>Sistema Operativo: Middleware</i>	39
2.3.2.2	<i>Máquina Virtual Java</i>	40
2.3.2.3	<i>Configuración</i>	40
2.3.2.4	<i>Perfil</i>	41
2.3.3	CONFIGURACIÓN CLDC	43
2.3.4	PERFIL MIDP	45
2.3.5	MIDLETS	46
2.3.6	PAQUETES OPCIONALES	48
2.4	APIS JAVA PARA LA TECNOLOGÍA BLUETOOTH	50
2.5	DESARROLLO DE JUEGOS PARA MÓVILES	52
3.	MARCO METODOLÓGICO	54
3.1	PROTOTIPADO EVOLUTIVO	55
3.1.1	CONCEPTO INICIAL	55
3.1.2	DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO INICIAL	56
3.1.3	REFINAMIENTO DEL PROTOTIPO HASTA SU ACEPTACIÓN	56
3.1.4	ENTREGA DEL PROTOTIPO FINAL	56
4.	DESARROLLO DEL PROYECTO	57
4.1.	BANDERCOL. JUEGO MULTIJUGADOR BLUETOOTH	57
4.1.1	OBJETIVO	57
4.1.2	PERSONAJES	57
4.1.3	ESCENARIOS	58
4.2	ANÁLISIS DE REQUERIMIENTOS	61
4.3	DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN	62
4.3.1	ASPECTOS GENERALES	62
4.3.2	SERVICIO BLUETOOTH DE JUEGO	63

4.3.3	DISEÑO DE LA APLICACIÓN	64
4.3.4	IMPLEMENTACIÓN DE LA APLICACIÓN.....	68
4.3.4.1	<i>Primer Prototipo</i>	69
4.3.4.2	<i>Segundo Prototipo</i>	71
4.3.4.3	<i>Tercer Prototipo</i>	73
4.3.4.4	<i>Prototipo Final</i>	74
5.	EVALUACIÓN	76
	CONCLUSIONES	77
	RECOMENDACIONES Y TRABAJOS A FUTURO	79
	REFERENCIAS	80
	BIBLIOGRAFÍA	82
	ANEXO A: MANUAL DE USUARIO	85

LISTA DE TABLAS

Tabla 1. Soporte Hardware.....	9
Tabla 2. Perfiles Bluetooth.....	29
Tabla 3. Librerías del perfil MIDP.....	46
Tabla 4. Tabla 4. Paquetes Opcionales para J2ME.....	48
Tabla 5. Objetos de los Escenarios	58
Tabla 6. Paquetes utilizados en el Desarrollo de la Aplicación.....	62
Tabla 7. Servicio Bluetooth de Juego	63

LISTA DE FIGURAS

Figura 1. Vista del NetBeans IDE 5.0	11
Figura 2. Vista del Nokia Developer's Suite 3.0 for J2ME™	12
Figura 3. Pila de Protocolos Bluetooth [4].....	21
Figura 4. Conexión Punto a Punto [5].....	26
Figura 5. Piconet [5].....	26
Figura 6. Scatternet	27
Figura 7. Service Discovery DataBase (SDDP) [5]	28
Figura 8. Entorno de Ejecución J2ME [8].....	39
Figura 9. Arquitectura J2ME [7]	42
Figura 10. Ciclo de Vida del MIDlet [8].....	47
Figura 11. Diagrama de Estados del MIDlet [8]	48
Figura 12. Mapa Actual de la Tecnología J2ME	50
Figura 13. Arquitectura JSR-082.....	51
Figura 14. Modelo de Prototipado Evolutivo [1]	55
Figura 15. Personajes de Juego	58
Figura 16. Escenario Completo.....	60
Figura 17. Diagrama de Casos de Uso.....	64
Figura 18. Diagrama de Actividades inicial de la Aplicación	66
Figura 19. Diagrama de Clases de Interfaz de Usuario	67
Figura 20. Diagrama de Clases del Juego.....	68
Figura 21. Menú Principal de la Aplicación	69
Figura 22. Formulario Opciones de Juego.....	70
Figura 23. Pantalla de Ayuda.....	70
Figura 24. Interfaz de Usuario del Servidor.	71
Figura 25. Interfaz de Usuario del Cliente.....	72
Figura 26. Tercer Prototipo - Visualización del Juego.....	74
Figura 27. Prototipo Final - Visualización del Juego	75
Figura 28. Run MIDP Application del Java ME SDK for CLDC de Sony Ericsson ..	86
Figura 29. Primera Pantalla de la Aplicación	88

INTRODUCCIÓN

Con el paso del tiempo, se ha hecho cada vez más común la presencia de la tecnología en la vida cotidiana de las personas, surgiendo como solución a muchos de los problemas que se presentan en las distintas disciplinas, como la medicina, las telecomunicaciones, la educación y la agricultura por sólo nombrar algunas, o simplemente haciendo más fácil el diario vivir. En menos de un siglo, los avances tecnológicos han tenido un gran impacto en la sociedad, transformándola y mejorando de varias maneras la calidad de vida.

Las telecomunicaciones han sido el sector en donde los aportes tecnológicos se han sentido con mayor intensidad. Los constantes adelantos en las industrias de la microelectrónica y de las comunicaciones móviles, han conseguido masificar la utilización de una nueva generación de sofisticados dispositivos móviles, que a su vez, hacen uso de un buen número de tecnologías inalámbricas que están en constante evolución.

De allí, que las tecnologías inalámbricas sean cada vez más populares en el mundo, no solamente porque prometen terminar con la creciente y omnipresente maraña de cables detrás de los escritorios, sino también, porque proveen acceso a información a través de dispositivos móviles en cualquier lugar y en cualquier momento.

Nuestro país no ha sido ajeno a este fenómeno. Aunque con cierto retraso con respecto a otros países de la región, el ingreso y uso de este tipo dispositivos y tecnologías ha crecido significativamente y hoy en día, no es privilegio de unos pocos como lo fue en un principio. Gradualmente, se están dando las condiciones para que en Colombia se masifique el uso de las tecnologías inalámbricas. De un

lado, hay un creciente ingreso de terminales de datos portátiles y personales, y del otro, la existencia de redes de sistemas de comunicaciones móviles.

El número de usuarios de teléfonos móviles se ha incrementado rápidamente. Esto da una gran oportunidad para la expansión y desarrollo de empresas en el mercado de dispositivos móviles. Los juegos inalámbricos en línea están en su apogeo. Sin las limitaciones de los juegos hechos para PC, los juegos inalámbricos en línea pueden ejecutarse en cualquier momento y en cualquier lugar usando dispositivos móviles. La disponibilidad de redes de comunicación móvil incrementa las capacidades de los dispositivos móviles.

Nuevas tecnologías como *Bluetooth* y *J2ME*¹ facilitan el desarrollo y despliegue de juegos en línea a través de dispositivos móviles. Bluetooth es un protocolo de comunicaciones que ofrece conexión a velocidades aceptables con un bajo consumo de energía, permitiéndole a dos o más dispositivos móviles enviar y recibir información. Y J2ME permite el desarrollo de aplicaciones para dispositivos con bajas capacidades de memoria y procesamiento y de pantallas pequeñas.

El incremento de usuarios de teléfonos móviles y asistentes digitales personales proporciona un mercado considerable para los juegos inalámbricos en línea. El hardware y las herramientas de desarrollo están ya disponibles para empezar a explorar este mercado potencial.

¹ Java 2 Micro Edition

1. PRESENTACION.

1.1. DESCRIPCIÓN DEL DOCUMENTO.

Esta tesis ofrece una introducción a la tecnología Bluetooth, al desarrollo de aplicaciones con Java 2 Micro Edition enfatizando a los APIs Java para la tecnología Bluetooth, y al desarrollo de juegos para dispositivos móviles.

El Capítulo 1 hace la presentación formal del proyecto, haciendo mención de los objetivos general y específicos, justificación, impacto y viabilidad, y soporte del mismo.

El Capítulo 2 contiene el fundamento teórico del proyecto y se divide en cinco partes. La primera parte da una visión global de lo que es la Computación Móvil. En la segunda parte se presenta un estudio detallado de la tecnología Bluetooth, sus características, arquitectura, topología, perfiles, seguridad, etc. La tercera parte proporciona un vistazo general a la tecnología J2ME y los APIs que ofrece para el desarrollo de aplicaciones para dispositivos móviles. La cuarta parte describe los APIs Java para Bluetooth. Y la parte final del capítulo contiene algunas nociones de cómo desarrollar juegos para dispositivos móviles.

El Capítulo 3 describe la metodología de desarrollo de software utilizada en el desarrollo de este proyecto.

El Capítulo 4 presenta el diseño e implementación de la aplicación, mostrando su evolución prototipo por prototipo.

El Capítulo 5 contiene una evaluación hecha al proyecto a medida que avanza el desarrollo del mismo.

El Capítulo 6 expone las conclusiones obtenidas en el proceso de desarrollo del proyecto.

El Capítulo 7 presenta algunas recomendaciones para trabajos futuros en esta línea de investigación que puedan involucrar el uso de las tecnologías Bluetooth y J2ME.

El Capítulo 8 enumera las referencias bibliográficas que contienen la información relevante acerca de las fuentes utilizadas en el proyecto.

El documento finaliza con la Bibliografía y algunos anexos de interés.

1.2. OBJETIVOS

Los objetivos presentados en el plan de proyecto de grado fueron los siguientes [9]:

1.2.1 OBJETIVO GENERAL

Desarrollar un juego multijugador en tiempo real vía Bluetooth para dispositivos móviles basado en J2ME (Java 2 Micro Edition).

1.2.2 OBJETIVOS ESPECÍFICOS

1. Establecer una base teórica que recopile las características, capacidades, aplicaciones, ventajas y desventajas de la tecnología Bluetooth.

2. Diseñar un juego multijugador en el cual los jugadores compartan un escenario y tengan la posibilidad de visualizarse el uno al otro, especificando la lógica y las reglas del mismo, tal que pueda ser adaptado para jugarse desde dispositivos móviles.
 - 2..1 Seleccionar el perfil Bluetooth, el protocolo de comunicación y las APIs Java 2 Micro Edition más apropiados para el desarrollo del proyecto.
 - 2..2 Diseñar e implementar un servicio Bluetooth de Juego, basado en el juego previamente mencionado, para el caso particular de modo multijugador de dos jugadores.

3. Diseñar e implementar una aplicación que permita al usuario:
 - Iniciar el servicio Bluetooth de Juego.
 - Descubrir dispositivos bluetooth cercanos que ofrezcan el servicio de juego.
 - Conectarse a uno de los dispositivos que ofrecen el servicio,
 - Jugar en línea utilizando el servicio de Juego.

Dicha aplicación debe poseer capacidad de almacenamiento persistente de datos en el dispositivo y funcionalidad multimedia.

1.3 JUSTIFICACIÓN

Actualmente, el mundo ha llegado a un punto en el que la tecnología esta muy ligada al estilo de vida de las personas. Cada vez son más comunes dispositivos electrónicos como teléfonos móviles, PDAs², buscapersonas, entre otros. Aunque estos dispositivos no aportan gran beneficio por sí solos, lo verdaderamente importante es poder establecer comunicación entre ellos y otros dispositivos.

Estos dispositivos vienen apareciendo cada vez con más capacidades, ofreciendo una mayor cantidad de servicios que no están únicamente orientados a la vida laboral, si no que también se enfocan hacia el entretenimiento y el ocio. El avance a pasos agigantados de la tecnología ha provocado que las aplicaciones desarrolladas para dispositivos móviles ocupen hoy un lugar privilegiado en el negocio de las telecomunicaciones y los juegos para móviles no son ajenos a este fenómeno. Este nuevo mercado en crecimiento se ha convertido en uno de los principales puntos de interés tanto para los fabricantes de dispositivos como para empresas que no pertenecen al medio.

Se podría pensar que el bajo nivel de procesamiento y tamaño de las pantallas restringirían las posibilidades de los juegos para dispositivos móviles, no obstante, el número de usuarios es cada vez mayor y crece continuamente. La tecnología actual permite tanto el desarrollo de juegos monojugador así como de aplicaciones multijugador. El desarrollo de juegos para dispositivos móviles esta en plena revolución y es un mercado con un enorme potencial de crecimiento.

Tecnologías como Bluetooth y soluciones como J2ME, facilitan el desarrollo de juegos en línea vía redes inalámbricas para dispositivos móviles. Bluetooth provee

² Personal Digital Assistant. Asistente Personal Digital.

comunicación entre los dispositivos móviles sin costo y, además, viene embebido en la mayoría de teléfonos celulares de última generación. Por otro lado, J2ME proporciona una plataforma para el desarrollo de este tipo de aplicaciones, ya que incluye librerías específicas para el desarrollo de juegos y la mayoría de teléfonos celulares soportan Java, entre ellos, todos los teléfonos celulares de última generación Nokia, Sony Ericsson y Motorola, entre otros.

Es importante tener bases de conocimiento de estas nuevas tecnologías, porque la comprensión de su estructura y funcionamiento, es el punto de partida para encontrar nuevas formas de aplicación, que den solución a problemas de la vida cotidiana ó, que sencillamente faciliten el estilo de vida de las personas. El propósito de este proyecto es precisamente explorar un nuevo ámbito tecnológico, como es el caso de la computación móvil, y más específicamente, llevar a cabo un estudio acerca de Bluetooth, como tecnología para la comunicación entre dispositivos móviles, y de J2ME, como entorno de desarrollo de aplicaciones para dispositivos móviles.

Basado en el estudio de Bluetooth y J2ME, se desarrolló una aplicación, que permitió analizar el funcionamiento de la tecnología Bluetooth y la utilidad que ofrece J2ME como entorno para el desarrollo de aplicaciones para dispositivos móviles. La aplicación consiste en un juego que permite a dos usuarios, jugar en tiempo real, utilizando como terminales cada uno, un dispositivo móvil (teléfonos celulares). El juego permite al usuario bien sea, ofrecer el servicio de juego, ó encontrar dispositivos bluetooth cercanos, buscar en ellos el servicio de juego, y conectarse a uno de ellos para iniciar el juego. Para su desarrollo se utilizó la Edición de Java para dispositivos móviles J2ME, buscando hacer el mayor uso de los APIs que esta ofrece para el desarrollo de aplicaciones, destacando entre éstos el API Java para Bluetooth (JSR-82).

1.4 IMPACTO Y VIABILIDAD

1.4.1 IMPACTO

El proyecto permite:

- Despertar el espíritu investigativo en estudiantes que estén en búsqueda de innovaciones y fortalezas en el área de computación móvil. Fortalezas que permitirán una consolidación y potenciación de la línea investigativa “Computación Móvil”, proporcionando continuidad y fortalecimiento al Grupo de Investigación en Ingeniería Telemática y Sistemas Inteligentes (GITSI).
- Sentar una base de conocimiento para el desarrollo de futuros proyectos del Grupo de Investigación en Ingeniería Telemática y Sistemas Inteligentes en el área de la Computación Móvil.
- Concientizar a la comunidad en general de las posibilidades y ventajas que nos ofrecen las nuevas tecnologías para el mejoramiento de nuestro estilo de vida.
- Dar primeros aportes para la adaptación de una metodología de desarrollo de aplicaciones para dispositivos móviles.

1.4.2 VIABILIDAD

Se encuentran disponibles recursos necesarios tales como software, hardware, bibliografía, personal, que permiten un entorno más que propicio para el desarrollo del proyecto.

En el desarrollo de este proyecto se utilizan herramientas que ofrece Java para el desarrollo de aplicaciones móviles bajo el concepto de *GPL (General Public License)* tales como Editores de código, Emuladores de dispositivos y un Simulador Bluetooth.

Existe una gran variedad de dispositivos móviles en el mercado que soportan Java. Nokia, Motorola y Ericsson son empresas fabricantes de teléfonos celulares que le han apostado a Java como lenguaje de programación para el desarrollo sus de aplicaciones. Además, la gran mayoría de dispositivos que se han lanzado al mercado en el último año tienen capacidad Bluetooth para comunicarse entre sí y para realizar sincronizaciones con PCs.

1.5 SOPORTE

A continuación se enuncian cada uno de los requerimientos, tanto de software como de hardware, que fueron necesarios para el desarrollo del proyecto.

1.5.1 SOPORTE HARDWARE

La Tabla 1 contiene el equipo hardware que fue requerido para el desarrollo de este proyecto.

Tabla 1. Soporte Hardware.

EQUIPO	CANTIDAD	CARACTERÍSTICAS	DESTINACIÓN
Computador Personal.	1	Procesador de 2.8 GHz. Memoria RAM 512 MB. Disco Duro 80 GB. Puerto USB 2.0 Tarjeta de Red.	Alojamiento y utilización de las distintas herramientas de desarrollo utilizadas en el proyecto (editores de texto, editores de código y emuladores)

Teléfono Celular.	2	Memoria interna 7Mb Tecnología Java MIDP 2.0 con CLDC 1.1 y Conectividad Bluetooth. Capacidad para descarga de juegos Java. Pantalla multicolor de alta definición.	Pruebas de funcionalidad de la aplicación en circunstancias reales.
-------------------	---	--	---

1.5.2 SOPORTE SOFTWARE

1.5.2.1 Java 2 JDK 1.5

El *JDK*³ es un ambiente de desarrollo para construir aplicaciones y componentes usando el lenguaje de programación Java. Incluye herramientas útiles para desarrollar y probar programas escritos en el lenguaje de programación Java y ejecutarlos sobre la Plataforma Java.

1.5.2.2 Java 2 JRE 1.5

El *JRE*⁴ es un ambiente de ejecución. Provee un soporte completo para la ejecución de aplicaciones Java.

1.5.2.3 NetBeans IDE 5.0

NetBeans IDE es un entorno integrado de desarrollo, una plataforma de aplicaciones que se puede utilizar como marco genérico para construir cualquier

³ Java 2 Platform Standard Edition Development Kit. Kit de desarrollo de la plataforma Java 2 Standard Edition (J2SE)

⁴ Java Runtime Environment. Entorno de Ejecución Java.

tipo de aplicación basada en tecnología Java; incluye las plataformas Java 2 Standard Edition (J2SE) y Java 2 Enterprise Edition (J2EE).

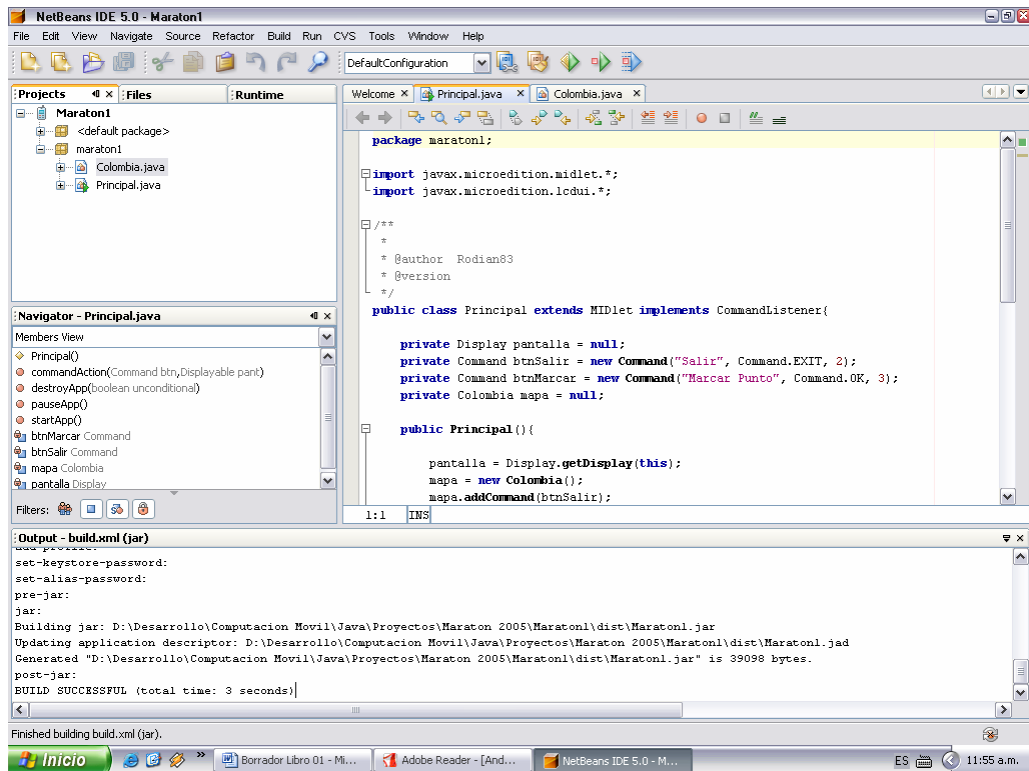


Figura 1. Vista del NetBeans IDE 5.0

1.5.2.4 NetBeans Mobility Pack 5.0

Parque para el NetBeans IDE 5.0 con el cual este obtiene la capacidad de crear, construir, probar y depurar aplicaciones MIDP⁵ 2.0 de Java. Se puede integrar fácilmente los emuladores para tener un ambiente de testeo robusto.

⁵ Mobile Information Device Profile. Perfil de dispositivo móvil de información.

1.5.2.5 Nokia Developer's Suite 3.0 for J2ME

Conjunto de herramientas que ayuda a crear contenido Java para dispositivos Nokia. Con él los desarrolladores pueden crear, probar y desplegar software Java para dispositivos Nokia [6]. Puede ejecutarse sólo o integrado con IDEs (*Integrated Development Environment*) ó Ambientes Integrados de Desarrollo, entre ellos NetBeans, Sun Java Studio, Eclipse y JBuilder.

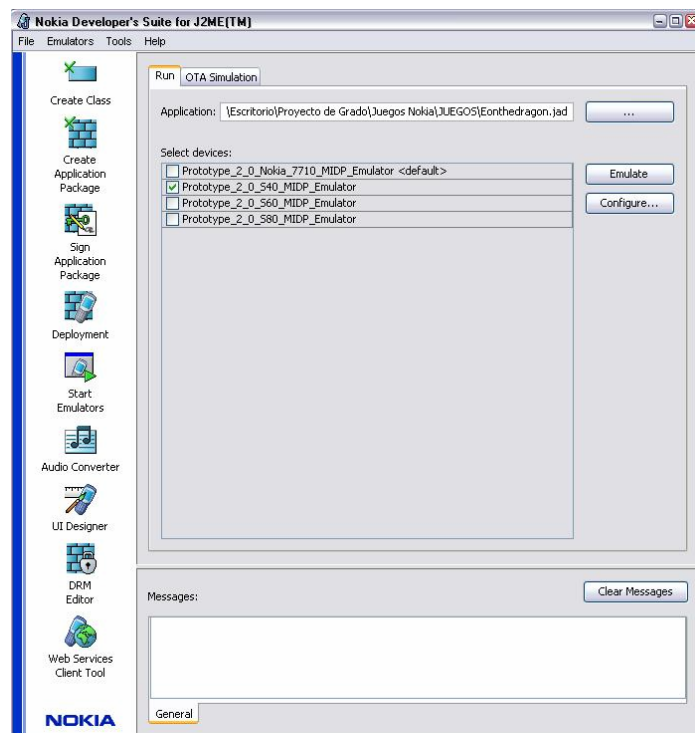


Figura 2. Vista del Nokia Developer's Suite 3.0 for J2ME™

1.5.2.6 Nokia Connectivity Framework

El Nokia Connectivity Framework (NFC lite) es una herramienta para habilitar la comunicación entre varios productos software Nokia dentro de una máquina local

[6]. Estos productos pueden soportar tecnologías, formatos y versiones diferentes. El NFC lite está construido de tecnologías y componentes de comunicación diferentes.

1.5.2.7 *Nokia PC Suite*

Software Nokia que permite editar, almacenar y sincronizar los datos del teléfono móvil Nokia con un sistema de Computador Personal basado en Microsoft Windows [6].

2. MARCO TEÓRICO

2.1. COMPUTACIÓN MÓVIL

2.1.1 INTRODUCCIÓN

En las últimas décadas, las industrias de la computación, la microelectrónica y las comunicaciones han exhibido grandes avances. Estos avances se ven materializados en sistemas operativos más eficientes en el uso de los recursos disponibles, sofisticados y ergonómicos dispositivos portátiles con mayores capacidades de procesamiento y almacenamiento, y redes omnipresentes proveídas por operadores celulares, instaladas en las empresas e incluso en los hogares.

Como resultado a todos estos adelantos, ha surgido un nuevo paradigma computacional conocido como *Computación Móvil*. Es de resaltar que el término utilizado ha sido “computación móvil” y no “computación inalámbrica”. Existe una generalizada y equivocada tendencia a establecer relaciones de implicación entre los términos inalámbrico y móvil.

Si bien es cierto que la utilización de redes inalámbricas fortalece el potencial de las aplicaciones móviles, es totalmente errado afirmar que el hecho de que una aplicación haga uso de tecnologías inalámbricas la clasifique como una aplicación móvil [3]. Por ejemplo, una aplicación ejecutada desde un computador de escritorio, que hace uso de una red WLAN⁶, no es una aplicación móvil, pues un computador de escritorio no provee movilidad alguna.

⁶ Wireless Local Area Network. Red Inalámbrica de Área Local.

También es común la afirmación de que sin el uso de las tecnologías inalámbricas no es posible hacer aplicaciones móviles, afirmación también equivocada. Prueba de ello, puede ser una aplicación de automatización de fuerza de ventas, en la que los empleados recolectan todo el día la información y la sincronizan a un servidor central al final de la jornada a través de una interfaz serial alámbrica.

2.1.2 CONCEPTUALIZACIÓN

El concepto de *Computación Móvil* surgió por primera vez en el centro de investigación de *Xerox* en el año de 1987 como un subproducto de lo que se considera como el próximo salto en la evolución informática. Una visión mucho más ambiciosa de lo que debe ser esta tecnología en el nuevo siglo: la computación centrada en el individuo [3]. La computación móvil constituye un nuevo paradigma cibernético, cuya conceptualización está en continua evolución.

Existen varias definiciones de *Computación Móvil*, y todas ellas hacen énfasis en lo mismo: movilidad y acceso a recursos en cualquier momento y desde cualquier lugar. Estas dos simples ideas son lo suficientemente discriminatorias para determinar cuando una solución informática puede ser considerada como una solución móvil [3].

Definiremos la *Computación Móvil* como el conjunto de técnicas que permiten el acceso a recursos digitales en cualquier momento y desde cualquier lugar y están asociadas a la movilidad de datos, hardware y software.

Las plataformas de la computación móvil están centradas en tres pilares tecnológicos, entorno a los cuales se han establecido frentes de investigación bien definidos. El primero de ellos es la *Movilidad*, relacionada con la evolución de

los sistemas operativos de los dispositivos móviles, y encaminada a resolver los desafíos asociados con la transparencia, la dependencia y sensibilidad de las aplicaciones a la localización. Por ejemplo, para un empleado de una compañía que viaja entre las distintas sucursales de una compañía, sería deseable que las variables del sistema como la impresora predeterminada o el proveedor de acceso se ajusten automáticamente a medida que se traslada entre una sucursal y otra.

El segundo pilar tecnológico es la *Portabilidad*, y se centra en los dispositivos móviles. Los desafíos que aquí se plantean se centran en disminuir las limitaciones de procesamiento, almacenamiento, comunicación y visualización, y minimizar el impacto asociado con el peso, autonomía y consumo de energía.

Finalmente, el tercer pilar tecnológico lo constituyen las *Comunicaciones Inalámbricas*, aquí los retos están relacionados a la calidad de la conectividad, el ancho de banda, latencia y cobertura.

2.1.3 VENTAJAS Y DESVENTAJAS

Como es de esperarse, la computación móvil ofrece una gran cantidad de ventajas. A continuación se enumeran algunas de las más relevantes:

- Portabilidad de hardware, software y datos, y acceso a información en cualquier momento desde cualquier lugar.
- Optimización de procesos de captura de datos críticos – procesos que involucran trabajo de campo.
- Aprovechamiento en ámbitos empresariales de dispositivos móviles de uso personal, con los cuales un gran porcentaje de personas en el mundo se

encuentran familiarizadas, como lo son teléfonos celulares, PDAs y buscapersonas.

- Explotación de los distintos sistemas de redes de comunicaciones y tecnologías de comunicación existentes, dando pie al desarrollo de aplicaciones multicanal.
- Creación de redes personales temporales que no requieran de una infraestructura definida.

Pero no todo son ventajas, aún existen algunos aspectos en los cuales la computación móvil muestra debilidades y es necesario mejorar. Si bien es cierto que cada vez aparecen dispositivos móviles más sofisticados, no se puede desconocer que a pesar de éstos avances, las capacidades de almacenamiento, procesamiento e interacción con el usuario aún son limitadas. La seguridad es otro punto delicado, ya que el concepto de seguridad es muy amplio y este aspecto suele ser el talón de Aquiles de la gran mayoría de las tecnologías inalámbricas.

2.2 BLUETOOTH

2.2.1 INTRODUCCIÓN

El cable funciona para las transmisiones de datos eficientemente, pero no deja de ser molesto; el infrarrojo eliminó esta molestia, pero su tecnología tiene limitaciones innatas. Las señales de radio resultan ser más adecuadas para estos menesteres, ya que no sólo suprimen la necesidad de cableado, si no que también eliminan las restricciones de línea de visión que posee el infrarrojo.

La tecnología inalámbrica *Bluetooth* es un sistema de comunicación de radiofrecuencia de corto alcance, cuyo propósito es eliminar los cables en las conexiones entre dispositivos electrónicos, tanto portátiles como fijos [2]. Las principales características de esta tecnología son su fiabilidad, bajo consumo de energía y bajos costos [4]. Los dispositivos Bluetooth se reconocen y pueden comunicarse entre sí, el canal permanece abierto y no requiere la intervención directa y constante del usuario en cada envío.

Hasta el momento han sido lanzadas dos versiones de la especificación: la versión 1.2, adoptada en noviembre del 2003, que tenía una velocidad de transmisión de datos de hasta 1Mbps; y la versión 2.0 y especificación EDR (Transferencia de datos Mejorada), adoptada en noviembre del 2004, con velocidad de transmisión de datos de 3Mbps.

Bluetooth opera en la banda de frecuencia Industrial, Científica y Médica (ISM⁷) entre los 2.4 y 2.485 GHz y no requiere de licencia en la mayoría de países del mundo. Además utiliza un Espectro Ensanchado de Salto de Frecuencia ó FHSS (Frequency Hopping Spread Spectrum), la banda de radio es dividida en 79 subcanales, el radio Bluetooth usa uno de estos canales de frecuencia y salta de canal a canal demorándose 625 microsegundos en cada canal, alcanzando una velocidad de 1600 saltos por segundo [2].

La Función de Salto Adaptable de Frecuencia ó *AFH (Adaptive Frequency Hopping)* de la tecnología Bluetooth fue diseñada para reducir las interferencias generadas por las otras tecnologías inalámbricas con la que comparte el espectro de los 2.4 GHz. Esta función utiliza la frecuencia disponible en el espectro. Primero detecta los dispositivos conectados y descarta las frecuencias que éstos

⁷ Industrial, Scientific and Medical band.

están utilizando, de esta manera logra tener un alto grado de tolerancia a las interferencias.

El alcance nominal de esta tecnología depende de la clase de dispositivo. Para alcanzar una mayor distancia de cobertura se requiere un mayor consumo de energía. Existen tres clases de dispositivos. Los radios de clase 1 logran un alcance de hasta 100 metros y se utilizan principalmente en el sector industrial. Los radios de clase 2 logran un alcance de hasta 10 metros y son habituales de los dispositivos móviles. Los radios de clase 3 suelen tener un alcance de entre 1 y 3 metros. Los radios más utilizados son los de la clase 2, con una potencia de 2.5mW [2].

2.2.2 HISTORIA

El nombre Bluetooth hace referencia a Harald Blaatand II, un rey vikingo danés del siglo X, quien logró la unificación danesa. Blaatand se traduce al inglés como Bluetooth y al español como “diente azul”. El grupo de ingeniería de Ericsson resolvió nombrar así a esta tecnología en su honor, pues dentro los objetivos de la tecnología están la unificación y la armonía, es decir, permitir a diferentes dispositivos comunicarse entre sí usando un estándar ampliamente aceptado para la conectividad inalámbrica [4].

Su origen se remonta al año 1994, cuando Ericsson inició un estudio para investigar la factibilidad de una interfaz vía radio, barata y de bajo consumo de energía, con el propósito de interconectar sus teléfonos celulares y otros accesorios, y así eliminar el uso de cables. Fue así que se inició un proyecto conocido como *MC link*, basado en un enlace de radio de corto alcance, que podía ser usado en distintos teléfonos celulares, e incluso en otro tipo de dispositivos

debido a que el chip utilizado era de tamaño pequeño (9x9mm) y de muy bajo costo [2].

Tres años después, a comienzos de 1997, Ericsson hizo despertar el interés de otras compañías fabricantes de dispositivos móviles en el proyecto. De esta manera fue que a comienzos de 1998 se creó el Grupo de Interés Especial conocido como *SIG (Special Interest Group)*, constituido por cinco compañías: Ericsson, Toshiba, Intel, Nokia e IBM. El grupo estaba conformado por dos líderes fabricantes de teléfonos celulares, dos líderes en la fabricación de computadores y un líder en la fabricación de chips, logrando un conjunto adecuado de áreas de negocio, de tal manera que se asegurara la interoperabilidad de los equipos entre los distintos fabricantes.

Poco después se unieron al SIG otras empresas dentro de las que se destacan Motorola, Lucent, Microsoft y 3com. El SIG se hizo público en mayo de 1998, y en 1999, cerca de 650 compañías firmaron el Acuerdo de Suscriptores de Bluetooth. Actualmente son más de 2500 las compañías que han firmado este acuerdo.

2.2.3 ARQUITECTURA

La especificación Bluetooth define una estructura uniforme para que una amplia gama de dispositivos de diversos fabricantes puedan conectarse y comunicarse entre sí. Por esta razón no es suficiente especificar únicamente un sistema de radio. La especificación presenta además una colección completa de protocolos para asegurarse que los dispositivos Bluetooth puedan descubrirse entre sí, explorar sus servicios, y hacer uso de éstos.

Bluetooth utiliza un sistema de capas o niveles en su arquitectura, al igual que los sistemas OSI⁸ y TCP/IP⁹. No todas las aplicaciones utilizan todos los protocolos de la pila, debido a que hay protocolos paralelos en la pila de protocolos Bluetooth, en su lugar se ejecutan sobre una o más de las columnas de esta. Además, la especificación está abierta a implementaciones de nuevos protocolos de uso común.

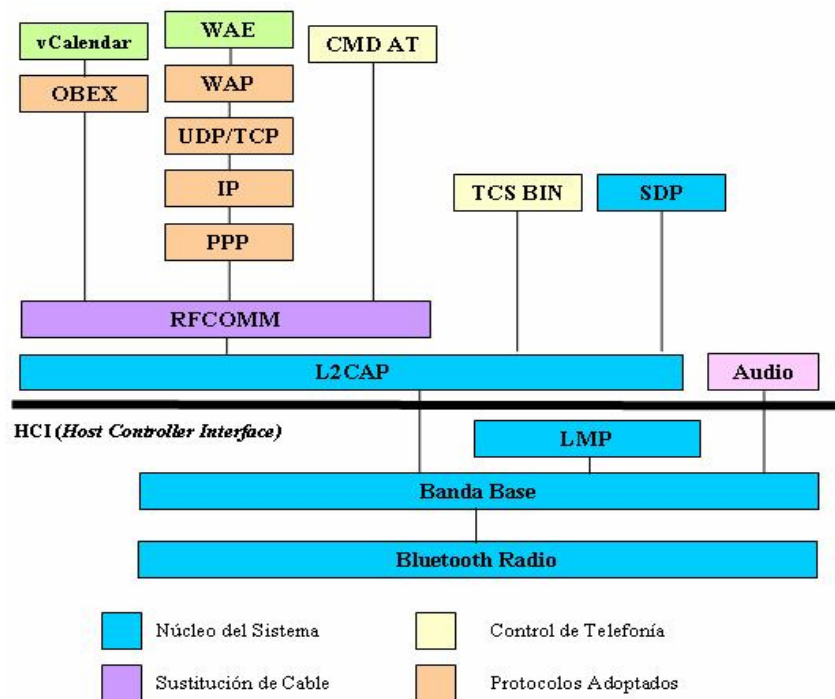


Figura 3. Pila de Protocolos Bluetooth [4]

Como puede observarse en la Figura 3, la estructura de la pila está formada por un Núcleo del Sistema Bluetooth, con protocolos propios de la tecnología Bluetooth, un protocolo para la sustitución de Cable, protocolos para el control de

⁸ Open Systems Interconnection. Modelo estándar para la interconexión de sistemas abiertos.

⁹ Conjunto básico de protocolos de comunicación de redes, popularizado por Internet, que permiten la transmisión de información en redes de computadoras. El nombre TCP/IP proviene de dos protocolos: el Transmission Control Protocol (TCP) y el Internet Protocol (IP).

telefonía, y un conjunto de Protocolos Adoptados [4]. Además de estos protocolos, la especificación define la HCI (Host Controller Interface) ó Interfaz de Controlador de Anfitrión, que proporciona una interfaz de comandos al controlador de la Banda Base, al manejador de enlace, y da acceso al estado del hardware y a los registros de control.

Los protocolos del Núcleo del Sistema Bluetooth deben ser implementados por la mayoría de dispositivos Bluetooth, situación que no ocurre con el resto de protocolos, ya que su uso está sujeto a los casos en que sean necesarios.

La pila de protocolos que conforma el Núcleo del Sistema Bluetooth está compuesta por el Protocolo de Control de Enlace ó *LCP (Link Controller Protocol)*, el Protocolo de Gestión de Enlace ó *LMP (Link Manager Protocol)* y el Protocolo de Adaptación y Control de Enlace Lógico ó *L2CAP (Logical Link Control and Adaption Protocol)*. Además, hay que agregar a éstos el Protocolo de Descubrimiento de Servicios ó *SDP (Service Discovery Protocol)* requerido por todas las aplicaciones Bluetooth.

Adicionalmente, cuenta con un protocolo de Sustitución de Cable, protocolos para el control de telefonía y algunos protocolos adoptados. De esta manera las capas más altas de la pila pueden reutilizar una gran cantidad de protocolos. A continuación se describirán cada una de las capas de la pila de protocolos.

2.2.3.1 Radio Bluetooth

Incorpora los circuitos de radio. Se encarga de modular los datos para su transmisión y de demodularlos para su recepción. Opera en la banda de los 2.4 GHz y admite una velocidad de transmisión aérea de 2 a 3 Mbps [4].

2.2.3.2 Banda Base.

Controla los enlaces físicos, los saltos de frecuencia, construcción de los paquetes de datos y sincronización de los dispositivos Bluetooth. Además, se encarga del reconocimiento y retransmisión de paquetes, y del control de flujo. El transmisor de Banda Base emplea un esquema de dúplex por división en el tiempo (TDD¹⁰). Maneja dos clases de enlaces: Síncrono Orientado a Conexión ó SCO (*Synchronous Connection Oriented*) para voz y datos, y Asíncrono no Orientado a la Conexión ó ACL (*Asynchronous Connectionless*), para datos [2].

2.2.3.3 Link Manager Protocol (LMP)

El LMP ó Protocolo de Gestión de Enlace, se encarga de la autenticación, cifrado, control y configuración de los enlaces; además se ocupa del control del tamaño de los paquetes, control de la potencia y los ciclos de trabajo en la transmisión de radio y control de la calidad de servicio (QoS).

2.2.3.4 Host Controller Interface (HCI)

La HCI ó Interfaz de Controlador de anfitrión se ocupa de manejar la comunicación entre el Anfitrión y el módulo Bluetooth. Brinda una interfaz para acceder al hardware Bluetooth. Es independiente del hardware, permitiendo la transparencia del software, y haciendo fácil la programación.

¹⁰ Time Division Duplex, una tecnología de radio.

2.2.3.5 Audio Bluetooth

Esta es una capa especial usada para enviar audio sobre Bluetooth sobre enlaces síncronos orientados a conexión. Los paquetes de audio no pasan por la capa L2CAP, sino que son enviados directamente a través del enlace. Bluetooth no especifica una presión sonora máxima para un dispositivo de audio [4].

2.2.3.6 Logical Link Control and Adaption Protocol (L2CAP)

El L2CAP ó Protocolo de Control y Adaptación de Enlace Lógico, se encarga de adaptar los paquetes de los protocolos de capas superiores al protocolo Banda Base. Provee servicios orientados y no orientados a conexión, con capacidades de multiplexación, segmentación y reensamblaje de paquetes de hasta 64 KB de longitud.

2.2.3.7 Service Discovery Protocol (SDP)

El SDP ó Protocolo de Descubrimiento de Servicio, como su nombre lo sugiere, se ocupa de la búsqueda y descubrimiento de servicios en el dispositivo remoto. Es usado para consultar los servicios ofrecidos y sus características, pero no ofrece acceso a los mismos.

2.2.3.8 RFCOMM

Es un protocolo que proporciona emulación de un puerto Serial RS-232 sobre el protocolo L2CAP [2]. Ofrece capacidades de transporte a servicios de niveles

superiores que usan líneas seriales como mecanismo de transporte. Soporta hasta 60 conexiones simultáneas entre dos dispositivos Bluetooth.

2.2.3.9 Protocolos de Control de Telefonía

Encontramos el TCS Binary (TCS BIN) y los Comandos AT (AT Commands). El TCS BIN es un protocolo que define la señalización del control de llamadas, y la señalización para facilitar el manejo de grupos Bluetooth. Además ofrece la funcionalidad de intercambio de información de señalización que no esté relacionada con llamadas entrantes. Bluetooth soporta un número de comandos AT para el control de telefonía a través de emulación de puerto serial (RFCOMM) [4].

2.2.3.10 Protocolos Adoptados

La especificación define una significativa cantidad de protocolos adoptados, e incluso esta abierta a implementaciones de protocolos de uso común. Dentro de éstos protocolos encontramos PPP (Point to Point Protocol), OBEX (Object Exchange), WAP (Wireless Application Protocol), TCP (Transmission Control Protocol), UDP (User Datagram Protocol) e IP (Internet Protocol).

2.2.4 TOPOLOGÍA

Bluetooth soporta comunicación Punto a Punto (P2P, Point-to-Point) y Punto a Multipunto. Cuando dos o más dispositivos Bluetooth comparten el mismo canal de conexión, forman una red conocida como *piconet* [5]. Hasta ocho dispositivos

pueden estar activos en una piconet, en la cual uno de ellos actuará como dispositivo maestro, y el resto lo harán como dispositivos esclavos.



Figura 4. Conexión Punto a Punto [5]

Los dispositivos esclavos solamente pueden transmitir datos durante el tiempo de transmisión otorgado por el maestro, y no se pueden comunicar directamente con otros dispositivos esclavos, toda comunicación debe ser dirigida por el dispositivo maestro. Los dispositivos esclavos sincronizan su salto de frecuencia con el maestro usando el reloj del maestro y su dirección Bluetooth. Las piconets toman la forma de estrella, con el dispositivo maestro en el nodo central, como se muestra en la Figura 5.

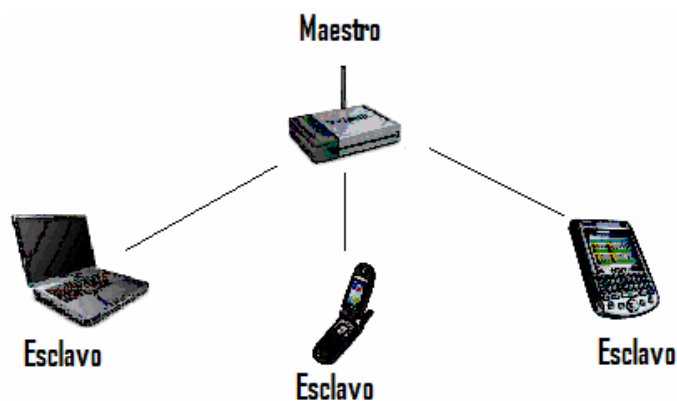


Figura 5. Piconet [5]

Dos o más piconets pueden coexistir en el mismo rango, pero el salto de frecuencia no está sincronizado entre piconets, por esta razón el salto de

frecuencia puede llegar a coincidir para ambas en un momento dado. El resultado de conectar dos o más piconets se conoce como *scatternet* [5]. Como se puede observar en la Figura 6, un dispositivo esclavo puede tener varios maestros, y un dispositivo puede actuar como maestro en una piconet, y como esclavo en otra.



Figura 6. Scatternet

2.2.5 DESCUBRIMIENTO DE DISPOSITIVOS BLUETOOTH

El comportamiento típico de un dispositivo Bluetooth es conectarse a otros dispositivos Bluetooth en una piconet, e intercambiar datos con dichos dispositivos. Dado que Bluetooth es una tecnología de comunicaciones inalámbrica *ad-hoc*¹¹, hay un cierto número de procedimientos y modos operacionales que permiten la formación de una piconet para que se pueda dar lugar a una comunicación.

¹¹ Término en la Ciencia de la Computación para definir al tipo de red formado por un grupo de nodos o host móviles que forman una red temporal sin ayuda de infraestructura externa alguna.

Los dispositivos Bluetooth usan el procedimiento de *inquiry* ó inspección para descubrir otros dispositivos Bluetooth cercanos. Un dispositivo Bluetooth que trata de encontrar otros dispositivos cercanos es conocido como dispositivo en inspección. Un dispositivo en inspección envía peticiones de inspección. Los dispositivos Bluetooth cercanos listos para ser descubiertos, escuchan estas peticiones de inspección y envían respuestas. Ocurrido esto, los dispositivos están en capacidad de empezar a transmitir. En seguida, se procede a establecer una conexión con cualquiera de los dispositivos encontrados [5].

Los dispositivos Bluetooth pueden ofrecer diferentes servicios. De aquí que un dispositivo Bluetooth necesita iniciar una búsqueda de los servicios que ofrece el dispositivo remoto para obtener información de cuales están disponibles. Las búsquedas de servicios pueden tratarse de una búsqueda general de servicios en el dispositivo, o limitarse a la búsqueda de un servicio específico.

Los dispositivos Bluetooth guardan información acerca de sus servicios Bluetooth en una Base de Datos de Descubrimiento de Servicio ó *SDDB* (*Service Discovery DataBase*), como puede observarse en la Figura 7. La SDDB contiene entradas de registro de Servicios, donde cada registro contiene atributos que describen a un servicio en particular.

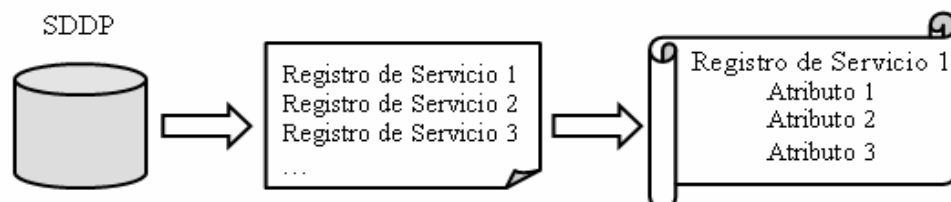


Figura 7. Service Discovery DataBase (SDDP) [5]

2.2.6 PERFILES BLUETOOTH

Para que un dispositivo pueda utilizar la tecnología Bluetooth, debe saber interpretar los perfiles Bluetooth que describen las distintas aplicaciones posibles. Los perfiles son guías que indican los procedimientos por los que los dispositivos Bluetooth se comunican entre sí [5]. Existe una amplia gama de perfiles que detallan los diferentes tipos de uso y aplicaciones de la tecnología Bluetooth. Al ceñirse a las pautas suministradas en la especificación, los desarrolladores pueden crear aplicaciones compatibles con otros dispositivos que se ajusten al estándar. Los perfiles lanzados con la versión 2.0 se presentan en la Tabla 2.

Tabla 2. Perfiles Bluetooth¹²

PERFILES	DESCRIPCIÓN
Advanced Audio Distribution Profile (A2DP) ó El Perfil de Distribución de Audio Avanzado	Describe cómo transferir sonido estéreo de alta calidad de una fuente de sonido a un dispositivo receptor. El perfil hace la distinción entre fuente de sonido y receptor.
Audio/Video Remote Control Profile (AVRCP) ó El Perfil de Control Remoto de Audio y Vídeo	Proporciona una interfaz estándar para manejar televisores, equipos de alta fidelidad o cualquier otro equipo electrónico, permitiendo que un único control remoto controle todo el equipo de audio y vídeo al que el usuario tiene acceso.
Basic Imaging Profile (BIP) ó Perfil Básico de Imagen	Establece cómo puede controlarse un dispositivo de imagen remotamente, así como la manera de enviarle órdenes de impresión y de transferencia de imágenes a un dispositivo de almacenamiento. Incluye también funciones para cambiar el tamaño y modificar las imágenes de forma que se ajusten al dispositivo receptor.
Extended Service Discovery Profile (ESDP)	Describe cómo se lleva a cabo la función Plug and Play a través de una conexión inalámbrica Bluetooth
Basic Printing Profile (BPP) ó Perfil Básico de Impresión	Permite enviar mensajes de texto, correo electrónico, imágenes, entre otras cosas, a las impresoras disponibles. No requiere controladores de impresión específicos. Es más indicado para dispositivos que no pueden actualizarse fácilmente con controladores específicos de cada fabricante, como teléfonos celulares o cámaras digitales.
Headset Profile (HSP) ó Perfil de Auricular	Describe cómo un auricular equipado con tecnología Bluetooth se comunica con un ordenador u otro dispositivo compatible, como un teléfono móvil.

¹² Basado en información encontrada en <http://www.bluetooth.com/>.

Common ISDN Access Profile (CIP) ó Perfil de Acceso ISDN Común.	Establece cómo se deben transferir las señales ISDN ¹³ a través de una conexión inalámbrica Bluetooth. Proporciona acceso sin restricciones a los servicios, datos y señales de la ISDN.
Fax Profile (FAX) ó Perfil de Fax.	Describe cómo un dispositivo terminal puede utilizar a otro como puerta de enlace para la transmisión de faxes. Ofrece una interfaz entre un teléfono celular o teléfono fijo y un ordenador con un programa de fax instalado.
Cordless Telephony Profile (CTP) ó Perfil de Telefonía Inalámbrica	Describe la implementación de un teléfono inalámbrico a través de un enlace inalámbrico Bluetooth. Puede utilizarse tanto para teléfonos celulares como para teléfonos inalámbricos.
Dial-up Networking Profile (DUN) o Perfil de Red de Marcado.	Proporciona un acceso telefónico estándar a Internet y a otros servicios de marcado a través de una conexión Bluetooth. Se utiliza, por ejemplo, al acceder a Internet desde un computador portátil utilizando un teléfono celular como módem, de forma que este realiza la función de marcado de forma inalámbrica. Se basa en el perfil SPP.
File Transfer Profile (FTP) ó Perfil de Transferencia de Archivos	Establece los procedimientos de exploración de carpetas y archivos de un servidor a través de un dispositivo cliente. Una vez que el cliente localiza el archivo en el servidor, puede copiarlo. Permite además, obtener el árbol de carpetas y pasar de una a otra, así como copiar, pegar y eliminar archivos. Utiliza el protocolo OBEX para la transmisión de datos y el perfil GOEP.
Generic Access Profile (GAP) ó Perfil de Acceso Genérico.	Proporciona las bases para los demás perfiles y establece los pasos que deben seguirse para crear un enlace de banda base entre los dispositivos con tecnología Bluetooth. Además, incluye funciones que deben implementarse en todos los dispositivos Bluetooth y procedimientos generales para detectar y conectar dispositivos.
General Audio/vides Distribution Profile (GAVDP) ó Perfil de Distribución Genérica de Audio y Vídeo	Establece las bases de los perfiles A2DP y VDP, pilar de los sistemas diseñados para la transmisión de sonido e imagen mediante la tecnología Bluetooth.
Generic Object Exchange Profile (GOEP) ó Perfil Genérico de Intercambio de Objetos.	Se utiliza para transferir objetos, tales como imágenes, documentos, etc., de un dispositivo a otro. Este perfil diferencia entre el servidor, dispositivo del que se extrae el objeto, y cliente, dispositivo que inicia la acción. Depende del perfil SPP. Proporciona un esquema general de los perfiles que utilizan el protocolo OBEX y describe la función de servidor y cliente de los dispositivos.
Synchronization Profile (SYNC) ó Perfil de Sincronización	Se utiliza junto al GOEP para sincronizar los elementos del Administrador de Información Personal (PIM), como agendas y datos de contacto, entre dispositivos con tecnología Bluetooth.

¹³ Integrated Services Digital Network, siglas inglesas de la RDSI, Red Digital de Servicios Integrados. Sistema para transmisión telefónica digital.

Hard Copy Cable Replacement Profile (HCRP) ó Perfil de Sustitución de Cable de Copia Impresa	Describe cómo imprimir archivos mediante un enlace inalámbrico Bluetooth utilizando controladores en el proceso. Distingue entre dispositivo cliente y servidor. El perfil no establece un estándar de comunicación con la impresora, por lo que se requieren los controladores específicos del modelo utilizado.
WAP Over Bluetooth Profile (WAP) ó Perfil WAP sobre Bluetooth	Describe cómo aplicar el conjunto de protocolos en la tecnología inalámbrica Bluetooth. Se utiliza, por ejemplo, al conectar un teléfono celular a un punto de acceso público para explorar contenidos a través de portales WAP.
Human Interface Device Profile (HID) o Perfil de Dispositivo de Interfaz Humana	Recoge los protocolos, procedimientos y características empleados por las interfaces de usuario Bluetooth tales como teclados, dispositivos punteros, consolas o aparatos de control remoto. Este perfil utiliza la definición de dispositivo HID del Bus de Serie Universal (USB) para aprovechar los controladores de clase para dispositivos USB.
Intercom Profile (ICP) ó Perfil de Intercomunicador	Establece cómo conectar dos teléfonos móviles con tecnología Bluetooth dentro la misma red sin utilizar la red telefónica pública. Esta función permite, por ejemplo, conectar dos teléfonos dentro de una misma oficina. También se le conoce como perfil Walkie-Talkie.
Object Push Profile (OPP) ó Perfil de Objeto Push	Distingue entre servidor y cliente de introducción de objetos push. Ambos roles son análogos y deben interactuar con los dispositivos cliente y servidor definidos en el perfil GOEP. Este perfil también puede emplearse para enviar objetos como imágenes o entradas de una agenda.
Personal Area Networking Profile (PAN) ó Perfil de Red De Área Personal	Describe cómo dos o más dispositivos con tecnología Bluetooth pueden formar una red ad-hoc y cómo ese mismo mecanismo permite acceder a la red de forma remota a través de un punto de acceso. El perfil diferencia entre el punto de acceso a la red, la red ad-hoc y el usuario de la red de área personal.
Service Discovery Protocol (SDP) ó Perfil de Descubrimiento de Servicios	Describe el proceso que seguirá una aplicación cliente Bluetooth para averiguar las características y servicios de un servidor compatible. Proporciona los medios para descubrir los nuevos servicios disponibles cuando el cliente accede a un área donde hay un servidor Bluetooth operativo. También permite detectar si un servicio ya no está disponible.
Service Discovery Application Profile (SDAP) ó	Detalla cómo una aplicación debe utilizar el perfil SDP para identificar los servicios de un dispositivo remoto. Se ocupa de que cualquier aplicación pueda localizar los servicios disponibles en el dispositivo conectado. Controla la búsqueda tanto de servicios específicos ya conocidos como de cualquier otro servicio nuevo
Hands-Free Profile (HFP) ó Perfil Manos Libres.	Describe cómo un dispositivo que actúa como puerta de enlace puede utilizarse para realizar y recibir llamadas a través de un dispositivo manos libres.

SIM Access Profile (SAP) ó Perfil de Acceso SIM	Permite a dispositivos con transmisores GSM integrados, como los sistemas de teléfono de los automóviles, conectarse a la tarjeta SIM de un teléfono celular equipado con tecnología Bluetooth. De esta forma, el teléfono del automóvil no requiere ninguna otra tarjeta SIM.
Serial Port Profile (SPP) ó Perfil de Puerto Serial.	Describe cómo configurar puertos de serie y conectar dos dispositivos con tecnología Bluetooth. Reemplaza a los cables por un sustituto inalámbrico en las aplicaciones de comunicación y señales de control basadas en conexiones de serie RS-232.
Video Distribution Profile (VDP) o Perfil de Distribución de videos	Establece los pasos que deben seguir los dispositivos con tecnología Bluetooth para la transferencia continua de vídeos. Esto permite, por ejemplo, visualizar en reproductores portátiles los vídeos almacenados en aplicaciones multimedia de cualquier ordenador o transferir las imágenes desde una videocámara digital a la televisión.

2.2.7 SEGURIDAD

La seguridad es un aspecto muy importante de las comunicaciones inalámbricas. Si un dispositivo Bluetooth se encuentra “descubrible”, cualquier otro dispositivo en las cercanías puede encontrarlo, determinar que servicios ofrece, y tratar de conectarse a ellos. Para manejar estas amenazas, la especificación define un modelo de seguridad basado en tres componentes: Autenticación, Encriptación y Autorización. Adicionalmente, son definidos tres modos de Seguridad estableciendo diferentes niveles de seguridad.

2.2.7.1 Modos de Seguridad

Los modos de Seguridad hacen parte del Perfil de Acceso Genérico (GAP¹⁴). Todos los dispositivos Bluetooth deben tener una implementación del Perfil GAP,

¹⁴ Generic Access Profile. Perfil de Acceso Genérico.

de aquí que todos los dispositivos Bluetooth tendrán implementado un modo de seguridad. El perfil GAP define tres modos de seguridad [4]:

- Modo de Seguridad 1: No Seguro.
- Modo de Seguridad 2: Seguridad impuesta a nivel del servicio.
- Modo de Seguridad 3: Seguridad impuesta a nivel del enlace.

En el modo No Seguro, los dispositivos nunca iniciarán procedimiento de seguridad alguno. En la actualidad son muy pocos los dispositivos que usan este modo de seguridad.

El modo de Seguridad 2, es el modo de seguridad más usado en dispositivos móviles. La seguridad se refuerza a nivel del servicio, el servicio decide que tipo de seguridad es la que se requiere. Esto da la posibilidad a los desarrolladores de decidir que tipo de seguridad manejarán sus servicios.

En el modo de Seguridad 3, los procedimientos de seguridad se inician durante la configuración del enlace Bluetooth. Si la seguridad falla, la configuración del enlace fallará. Los procedimientos de seguridad se ejecutan en las capas más bajas de la pila de protocolos Bluetooth, por lo cual los desarrolladores no tienen ingerencia en la seguridad.

2.2.7.2 Autenticación

La Autenticación usa un esquema de desafío-repuesta en que se verifica el conocimiento del demandante de una llave secreta a través de un protocolo que usa llaves secretas simétricas. Se compone de dos procedimientos: *Bonding* (vinculación) y *Pairing* (apareamiento) [5].

Bonding es el procedimiento en el que un dispositivo Bluetooth autentica a otro dispositivo Bluetooth. Depende de una llave de autenticación compartida. Si los dispositivos no comparten una llave de autenticación, una nueva llave debe ser creada antes que el proceso de vinculación sea completado. El proceso con el cual se genera una llave de autenticación se conoce como Pairing.

El proceso de Pairing involucra la generación de una llave de inicialización y una llave de autenticación, seguido por una autenticación mutua. La llave de inicialización se basa en una entrada del usuario, un número randómico y la dirección Bluetooth de uno de los dispositivos. La llave de autenticación se basa en números randómicos y direcciones Bluetooth de ambos dispositivos. La llave de la inicialización se usa para la encriptación al intercambiar los datos para crear la llave de la autenticación, y se desecha después de esto. Cuando el proceso de Pairing se completa, los dispositivos se han autenticado. Ambos dispositivos comparten la misma llave de autenticación, a menudo llamada llave de combinación puesto que ambos dispositivos han contribuido a la creación de la llave.

Cuando dos dispositivos han completado el proceso de Pairing, pueden guardar la llave de autenticación para uso futuro. Los dispositivos se aparean entonces y pueden autenticarse a través del proceso de Bonding sin el uso de una llave maestra. Los dispositivos se quedarán apareados hasta que un dispositivo pida un nuevo proceso de apareamiento, o la llave de autenticación se anule en alguno de los dispositivos. Guardar la llave de autenticación es útil para dispositivos que se conectan frecuentemente.

2.2.7.3 Encipción

La información del usuario puede ser protegida con la encipción de la carga útil de los paquetes; el código de acceso y el título del paquete nunca serán encipitados. Puede haber tres estados para la Banda Base con respecto al encipción [4]:

- No Encipción.
- Únicamente Punto a Punto.
- Encipción de paquetes de transmisión y Punto a Punto.

Cuando se conectan sólo dos dispositivos, el modo de encipción únicamente punto a punto es una opción natural. En el modo de no encipción sólo se seleccionará si ambos dispositivos no soportan encipción. Cuando se ha solicitado y ambos dispositivos soportan encipción, el tamaño de la llave de encipción se negocia. El dispositivo maestro sugerirá entonces el tamaño de llave de encipción más grande que soporte. El dispositivo esclavo puede entonces aceptar o rechazar este tamaño de la llave. Si el esclavo acepta, puede empezarse la encipción. Si el esclavo lo rechaza, el maestro puede sugerir un tamaño de llave más pequeño o puede decidir terminar la conexión. Este procedimiento se repite hasta que los dispositivos se ponen de acuerdo en un tamaño de la llave o el maestro decida terminar el enlace.

2.2.7.4 Autorización

La Autorización es el proceso por el cual se da permiso a un dispositivo Bluetooth remoto para acceder a un servicio particular. Para ser autorizado el dispositivo remoto debe autenticarse primero a través del proceso de Bonding. El acceso

puede entonces, concederse temporal o permanentemente. El atributo *trust* se relaciona a la autorización, vinculando los permisos de autorización a un dispositivo particular. Un dispositivo que posee el atributo *trust*, puede conectarse a un servicio Bluetooth, y el proceso de autorización se completará con éxito sin la intervención del usuario. De lo contrario, se requerirá la autorización del usuario.

2.2.8 VENTAJAS Y DESVENTAJAS

Al instante de comunicar dispositivos de corto alcance, de una manera fácil y sin la presencia de cables, Bluetooth es la opción más natural. Se trata de un estándar inalámbrico disponible en todo el mundo, con el que se pueden conectar entre sí teléfonos celulares, agendas electrónicas, computadores portátiles, y sus accesorios, entre muchos otros dispositivos. Gracias al exclusivo concepto de “perfiles”, no es necesario instalar controladores en los dispositivos provistos de la tecnología Bluetooth. La especificación Bluetooth se encuentra disponible de forma gratuita para compañías afiliadas de todo el mundo.

Uno de los puntos más fuertes de la tecnología Bluetooth es el alto grado de interoperabilidad que permite entre aplicaciones alojadas en diferentes dispositivos equipados con un módulo de comunicación Bluetooth [2].

Otra de sus ventajas es la capacidad para administrar simultáneamente tanto transmisiones de voz como de datos. Esto permite a los usuarios disfrutar de una gran variedad de soluciones innovadoras, tales como el uso de manos libres para atender llamadas, funciones de impresión y fax, o la sincronización de aplicaciones entre PDA, ordenadores y móviles, entre otras muchas.

Además, Bluetooth fue concebida como una tecnología inalámbrica de muy bajo consumo de energía, poco peso y bajo costo, lo cual le permite ser utilizada en una gran variedad de dispositivos móviles existentes en el mercado.

Obviamente, existen aún algunos desafíos que hay que superar. La velocidad de transmisión en uno de los aspectos a mejorar, ya que es considerablemente baja (hasta 3 Mbps para la versión 2.0) comparada con otras tecnologías inalámbricas existentes. Adicionalmente, existe otro aspecto crítico de la tecnología, como lo es la seguridad, algo inherente a cualquier tecnología de comunicación vía radio. Aún así, son más los beneficios que ofrece la tecnología Bluetooth comparados con estas desventajas.

2.2.9 APLICACIONES

La tecnología Bluetooth no sólo se usa para conectar un ratón, un teclado ó una impresora a un computador personal sin necesidad de cable. También permite la sincronización de dispositivos móviles automáticamente, conectar toda clase de dispositivos a una LAN y a Internet a través de puntos de acceso Bluetooth, conectar electrodomésticos y cámaras de vigilancia, además puede servir como control remoto para una gran variedad de dispositivos electrónicos, como televisores, puertas de garaje, equipos de sonido, etc.

Pero no solo encontramos aplicaciones para el hogar. También es usado en medicina para el monitoreo de los pacientes sin la necesidad de conectar cables al cuerpo, por empresas de servicios públicos para la lectura de contadores, por industrias para el control de sus inventarios, por la NASA para el control de los robots que envían al espacio, etc.

2.3 JAVA 2 MICRO EDITION (J2ME)

2.3.1 INTRODUCCIÓN

A mediados de década de los 90, la compañía Sun Microsystems lanzó el lenguaje de programación Java, diseñado en un principio para crear aplicaciones que controlaran electrodomésticos, pero gracias a su robustez e independencia de plataforma se empezó a utilizar para la generación de componentes interactivos integrados a páginas Web y programación de aplicaciones independientes. Con el correr del tiempo Java progresó considerablemente en diferentes contornos como servicios HTTP¹⁵, servidores de aplicaciones y acceso a bases de datos.

La explosión tecnológica de los últimos forzó a Sun a desarrollar soluciones Java personalizadas para cada ámbito tecnológico, agrupando cada uno de éstos ámbitos en una edición distinta de Java. Ellas son *J2SE* (Java 2 Standard Edition), orientada al desarrollo de aplicaciones independientes, *J2EE* (Java 2 Enterprise Edition), orientada al desarrollo de aplicaciones empresariales, y *J2ME* (Java 2 Micro Edition), orientada al desarrollo de aplicaciones para dispositivos con baja capacidad de procesamiento y almacenamiento de información, y limitaciones gráficas.

Al igual que las otras ediciones Java, J2ME busca mantener las cualidades por las cuales se ha conocido la tecnología Java, como robustez, portabilidad del código, seguridad en el trabajo de red, y escalabilidad. Esta Edición de Java está enfocada a la aplicación de la tecnología Java en dispositivos electrónicos con capacidades computacionales y gráficas muy reducidas, tales como teléfonos móviles, agendas digitales, buscapersonas, o electrodomésticos inteligentes.

¹⁵ Hypertext Transfer Protocol. Protocolo de Transferencia de Hipertexto.

2.3.2 ARQUITECTURA

La arquitectura J2ME es modular y escalable; además, está formada por ciertos componentes básicos. En la Figura 8 se muestra el esquema general del entorno de ejecución J2ME. Un entorno de ejecución determinado de J2ME se compone de una selección de: sistema operativo, máquina virtual Java, configuración, perfil y paquetes opcionales. A continuación se detalla cada uno de éstos componentes.

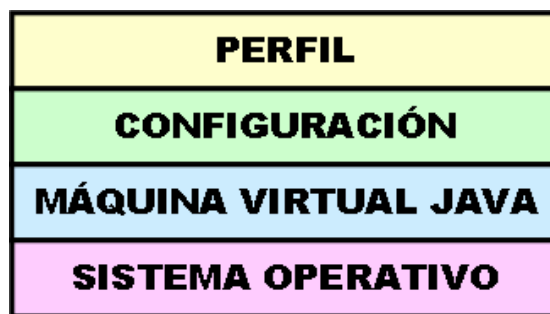


Figura 8. Entorno de Ejecución J2ME [8]

2.3.2.1 Sistema Operativo: *Middleware*

Conocido también como *Middleware*, es el sistema operativo del dispositivo móvil en el que se va a ejecutar la aplicación implementada con J2ME. El sistema operativo de los dispositivos móviles provee varios servicios: una abstracción software de las características hardware del dispositivo y servicios para la gestión de aplicaciones. Los sistemas operativos para dispositivos móviles más conocidos son BlackBerryOS, PalmOS, WindowsCE y SymbianOS.

2.3.2.2 Máquina Virtual Java

Una máquina Virtual Java (JVM, Java Virtual Machine) es un programa que se ocupa de interpretar el código neutro (bytecodes) de los programas Java precompilados y convertirlo a código máquina ejecutable por la plataforma. Así se evita tener que realizar un programa diferente para cada plataforma, proporcionando al programa Java independencia frente al hardware y al sistema operativo. J2ME cuenta con dos implementaciones de máquina virtual. Estas son la *KVM (Kylo Virtual Machine)* y la *CVM (Compact Virtual Machine)*.

La KVM es la máquina virtual Java más reducida desarrollada por Sun y está orientada a dispositivos con bajas capacidades de procesamiento (procesadores de 16 bits) y de memoria (máximo de 1MB). La CVM es también una máquina virtual Java reducida y está orientada a dispositivos electrónicos con procesadores de 32 bits de gama alta y memoria total de 2MB o mayor.

2.3.2.3 Configuración

Una configuración es el conjunto básico de APIs Java que definen un entorno generalizado de ejecución soportado por un conjunto horizontal de dispositivos [7]. Dichas APIs describen las características mínimas comunes a una gama de dispositivos con características similares en capacidad de almacenamiento y poder de procesamiento. Una configuración define las características del lenguaje Java, las características de la máquina virtual y las librerías básicas Java soportadas.

Existen dos configuraciones definidas en J2ME: *CLDC (Connected Limited Device Configuration)* enfocada a dispositivos con limitaciones de memoria y de

procesamiento; y *CDC (Connected Device Configuration)* enfocada a dispositivos con cierta capacidad computacional y de memoria.

La configuración CDC usa la máquina virtual CVM y está orientada a dispositivos con cierta capacidad de procesamiento y de memoria, y conectividad de algún tipo. Ejemplo de este tipo de dispositivos son los decodificadores de TV y sistemas de navegación para automóviles.

La configuración CLDC utiliza la máquina virtual KVM y está orientada a dispositivos con limitadas capacidades de procesamiento y memoria, bajo consumo de energía y con algún tipo de conectividad de bajo ancho de banda. Dentro de los dispositivos que usan esta configuración encontramos los teléfonos celulares, agendas digitales y buscapersonas.

2.3.2.4 Perfil

Un perfil es un conjunto de APIs orientadas a un ámbito de aplicación determinado, para una configuración específica soportado por un conjunto vertical de dispositivos [7]. Los perfiles identifican un grupo de dispositivos por la funcionalidad que proporcionan y el tipo de aplicaciones que se ejecutarán en ellos. Hay que tener en cuenta que un perfil siempre se construye sobre una configuración determinada. De allí, podemos afirmar que un perfil es un conjunto de APIs que dotan a una configuración de funcionalidad específica [8].

Para la configuración CDC existen los perfiles: Foundation Profile (FP), Personal Profile (PP) y Remote Method Invocation Profile (RMIP). El Foundation Profile está orientado a dispositivos que carecen de interfaz gráfica, como decodificadores de televisión digital. El Personal Profile proporciona por su parte, un entorno con un

completo soporte gráfico. El perfil RMIP permite trabajar con llamadas a métodos remotos. Tanto el Personal Profile como el RMI Profile trabajan sobre una implementación del Foundation Profile.

Para la configuración CLDC existen los perfiles Mobile Information Device Profile (MIDP) y Personal Digital Assistant Profile (PDAP). El MIDP está orientado a dispositivos con reducida capacidad computacional y de memoria, conectividad limitada, capacidad gráfica reducida y entrada de datos alfanuméricos reducida. El PDAP esta orientado a asistentes digitales personales de gama baja, con puntero y pantalla sensible, tipo Palm.

Ya han sido expuestos los conceptos básicos para entender el entorno de ejecución de J2ME. En la Figura 9 se muestra la arquitectura completa de J2ME.

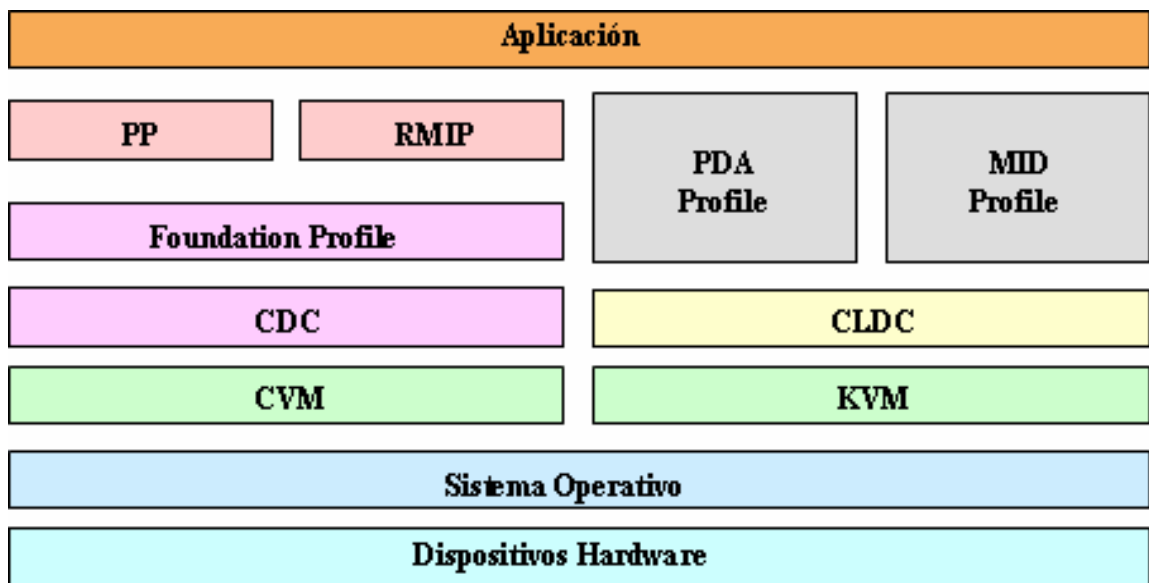


Figura 9. Arquitectura J2ME [7]

2.3.3 CONFIGURACIÓN CLDC

La meta de la especificación de CLDC es estandarizar la plataforma de desarrollo de aplicación Java para dispositivos portátiles con limitación de recursos y con capacidad para conectarse a algún tipo de red. Fue desarrollada a través del Java Community Process (JCP¹⁶), compuesto por más de 500 miembros, incluyendo operadores móviles y proveedores de servicio, fabricantes de dispositivos de mano inalámbricos, y vendedores de software móvil. CLDC se ha diseñado con los siguientes objetivos:

- Reducir los requerimientos de marca a niveles convenientes para el despliegue de un mercado en masa.
- Facilitar la portabilidad de las aplicaciones compendiando las operaciones del sistema nativo en los API estandarizados.
- Extender la funcionalidad del dispositivo, permitiendo la descarga dinámica de aplicaciones al dispositivo.

CLDC esta diseñada para llevar las ventajas de la plataforma Java a dispositivos con conexión a red que tienen limitaciones en su capacidad de procesamiento, memoria, y capacidad gráfica, como teléfonos celulares, buscapersonas y asistentes digitales personales de gama baja. Además, CLDC también puede desplegarse en los dispositivos para el hogar, consolas de juego para televisión, y terminales de punto de venta. Los dispositivos designados deben poseer al menos las siguientes capacidades:

- Procesador de 16 ó 32 bits con velocidad de reloj de al menos 16MHz.

¹⁶ Organización Internacional abierta de desarrolladores Java que desarrollan y revisan las especificaciones de la tecnología Java, implementaciones de referencia y kits de compatibilidad de tecnología a través de un proceso formal.

- Al menos 160 KB de memoria no volátil destinada para las librerías CLDC y su máquina virtual.
- Al menos 192 KB de memoria total disponible para la plataforma Java.
- Bajo consumo de energía, a menudo recurriendo al uso de baterías.
- Algún tipo de conectividad a red, a menudo redes inalámbricas, con conexión intermitente y ancho de banda limitado (9600 bps).

El JCP incluye el desarrollo de cada innovación o mejora de tecnología de Java en una Java Specification Request (JSR¹⁷). Los JSR para CLDC incluidos son los siguientes:

- CLDC Versión 1.0 (JSR 30) fue el primer lanzamiento de la especificación CDLD, la cual provee una máquina virtual compacta y librerías básicas para dispositivos de recursos limitados.
- CLDC Versión 1.1 (JSR 139) es una versión mejorada compatible de la especificación CLDC 1.0, con el mismo objetivo, dispositivos pequeños con recursos limitados. Sus muchas adiciones y mejoras incluyen soporte para matemáticas en punto flotante y referencias débiles.

Con respecto a la máquina virtual de CLDC, KVM, necesita entre 40 y 80 KB dependiendo de las opciones de compilación y el tipo de dispositivo para el que se compile. Esto significa que se podrán ejecutar aplicaciones con un total de 128 KB. Adicionalmente, se necesitan otros 32 KB para memoria dinámica de la aplicación. La KVM está implementada en lenguaje de programación C y está diseñada para ser tan completa y rápida como sea posible.

¹⁷ Descripción actual de propuestas y especificaciones finales para la plataforma Java. Los JSR son revisados por el Java Community Process antes de que sea hecho un lanzamiento final de la especificación.

2.3.4 PERFIL MIDP

MIDP es el acrónimo de Perfil para Dispositivos de Información Móvil (Mobile Information Device Profile), elemento clave de la plataforma J2ME. MIDP provee un entorno de ejecución Java para los dispositivos de información móviles más populares de hoy, como los teléfonos celulares y asistentes digitales personales. La especificación MIDP se definió también a través del Java Community Process (JCP) por un grupo especialista de más de 50 compañías, que incluyen a los principales fabricantes de dispositivos y vendedores de software móvil.

CLDC y MIDP proporcionan el núcleo funcional requerido por las aplicaciones móviles, en la forma de un estandarizado entorno de ejecución Java y un rico juego de APIs Java. Los desarrolladores que usan MIDP pueden codificar y desplegar rápidamente sus aplicaciones para una amplia variedad de dispositivos móviles de información. MIDP ha sido adoptado ampliamente como la plataforma de preferencia para las aplicaciones móviles. Está implementado en millones de teléfonos celulares y PDAs alrededor del mundo, y se soporta por los principales ambientes integrados de desarrollo (IDEs). Compañías alrededor del mundo ya han aprovechado MIDP para desarrollar una amplia gama de aplicaciones móviles tanto para consumidores como para empresas.

Hasta el momento han sido lanzadas dos versiones de MIDP. MIDP versión 1.0 (JSR 37) es la versión original y provee un núcleo funcional básico requerido para la implementación de aplicaciones móviles, incluyendo una interfaz de usuario básica y seguridad de red. La versión 2.0 (JSR 118), es la versión mejorada de MIDP, con nuevas características incluyendo interfaz de usuario mejorada, funcionalidad multimedia, soporte para juegos, mayor conectividad y seguridad extremo a extremo.

El API de MIDP 2.0 (JSR 118) se compone de los siguientes paquetes:

Tabla 3. Librerías del perfil MIDP¹⁸

PAQUETE	DESCRIPCIÓN
javax.microedition.midlet	Es el paquete del ciclo de vida de de las aplicaciones. Permite a los midlets interactuar con el entorno en el cual se ejecutan
javax.microedition.lcdui	Paquete de interfaz de usuario. Proporciona un conjunto de características para la implementación de interfaces de usuario en MIDP.
javax.microedition.rms	Paquete de persistencia de datos. Permite a los midlets guardar persistentemente datos en el dispositivo y recuperarlos posteriormente.
javax.microedition.io	Paquete de conexión Genérica. Proporciona soporte de red basándose en CLDC
javax.microedition.pki	Paquete de clave pública. Provee de certificados usados para autenticar información proveniente de conexiones seguras

Existen varias herramientas para el desarrollo de aplicaciones basadas en CLDC y MIDP, entre estas el Sun Java Wireless Toolkit, Sun ONE Studio Mobile Edition, NetBeans Mobility Pack, Nokia Developer's Suite 3.0 for J2ME(TM), Java ME SDK for CLDC de Sony Ericsson, Samsung Wireless Toolkit y el SDK Motorola for J2ME.

2.3.5 MIDLETS

Un MIDlet es una aplicación Java realizada con el perfil MIDP sobre la configuración CLDC [8]. Los MIDlets están diseñados para ser ejecutados en dispositivos con limitada capacidad gráfica, de procesamiento y de memoria. En dichos dispositivos no se dispone de líneas de comandos donde se puedan ejecutar las aplicaciones, si no que poseen un software que es el encargado de

¹⁸ Basado en JavaDocs de Sun Wireless Toolkit Beta 2.3.

ejecutar los MIDlets y administrar los recursos disponibles, dicho software es conocido como gestor de aplicaciones ó AMS (Application Management System).

El AMS reside en el dispositivo y permite ejecutar, pausar o destruir las aplicaciones J2ME. Sus principales funciones son: gestionar el ciclo de vida de los MIDlets y controlar los estados por lo que pasa mientras esta en ejecución. El ciclo de vida del MIDlet pasa por cinco fases, como puede observarse en la Figura 10, estas son: descubrimiento, instalación, ejecución, actualización y borrado.

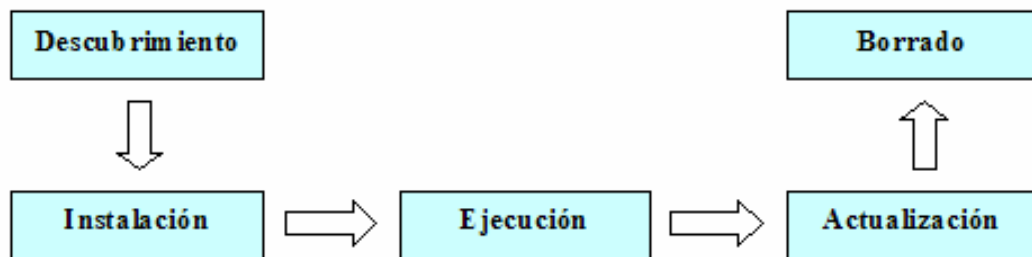


Figura 10. Ciclo de Vida del MIDlet [8]

Un MIDlet puede permanecer en el dispositivo permanentemente, debido a que después de ser instalado es almacenado en una zona de memoria persistente del dispositivo. El usuario decide en qué momento eliminar la aplicación a través del AMS.

Durante la fase de ejecución el MIDlet puede pasar por tres estados, estos son: Activo, Pausado y Destruído. En el estado activo el MIDlet está ejecutándose en el dispositivo. En el estado Pausado el MIDlet no está en ejecución y puede volver a ejecución cambiando al estado Activo. En el estado Destruído el MIDlet no esta ejecutándose y no puede transitar a otro estado, además se liberan los recursos que estaba utilizando. En la Figura 11 se puede apreciar el diagrama de estados de un MIDlet.

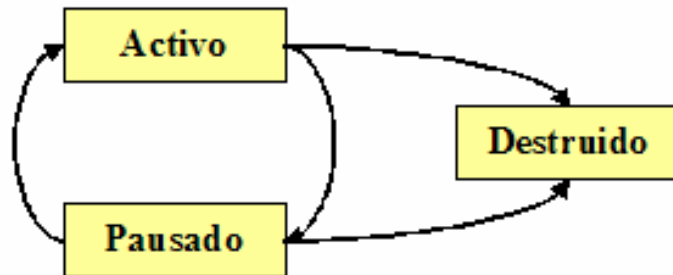


Figura 11. Diagrama de Estados del MIDlet [8]

2.3.6 PAQUETES OPCIONALES

Adicionales a los perfiles, J2ME tiene unos paquetes opcionales que agregan funcionalidades específicas a uno o más perfiles. En la Tabla 4 se presentan algunos de los más conocidos.

Tabla 4. Tabla 4. Paquetes Opcionales para J2ME¹⁹

PAQUETE	JSR	DESCRIPCIÓN
Information Module Profile (IMP).	JSR 195	Proporciona un entorno de aplicación para dispositivos embebidos que no tiene grandes capacidades gráficas o con recursos limitados de alguna otra manera: paneles de emergencia, parquímetros, sistemas de alarma domésticos y similares
Multimedia API (MMAPI)	JSR 135	Este paquete extiende la funcionalidad de la plataforma J2ME incorporando soporte de audio, video y otros tipos de datos multimedia basados en tiempo a dispositivos de recursos limitados.
Wireless Messaging API (WMA)	JSR 120 JSR 205	El API para mensajería sin cables proporciona acceso independiente de plataforma a recursos de comunicación sin cable como la mensajería SMS (Short Message Service).

¹⁹ Basado en información encontrada en <http://jcp.org/>

Bluetooth API	JSR 082	Proporciona un estándar para la creación de aplicaciones Bluetooth, de forma que las aplicaciones desarrolladas con el paquete opcional puedan ejecutarse utilizando esta tecnología.
Location API for J2ME	JSR 179	Permite la localización de dispositivos móviles para dispositivos con recursos limitados. El API se ha diseñado para generar información sobre la localización geográfica actual del terminal para las aplicaciones Java.
Security and Trust Services API	JSR 177	Este paquete amplía las características de seguridad para la plataforma J2ME añadiendo APIs de cifrado, servicio de firma digital y gestión de credenciales de usuario.
Mobile 3D Graphics API	JSR 184	Permite generar gráficos tridimensionales a frecuencias de imagen interactivas en dispositivos móviles de recursos restringidos.
SIP API for J2ME	JSR 180	El protocolo Session Initiation Protocol (SIP) se utiliza para establecer y gestionar sesiones IP multimedia. El API se ha diseñado para permitir que las aplicaciones Java envíen y reciban mensajes SIP.
J2ME Web Services APIs (WSA),	JSR 172	Amplía la plataforma de servicios Web para incluir J2ME. Estas APIs permiten que los dispositivos J2ME puedan ser clientes de servicios Web mediante un modelo de programación consistente con la plataforma estándar de servicios Web.
J2ME RMI Optional Package, (RMI OP)	JSR 066	Es un paquete que se puede utilizar sobre CDC. EL paquete RMI Optional (RMI OP) permite a dispositivos de consumo y aplicaciones embebidas interactuar como y con aplicaciones distribuidas.

Después de este vistazo hecho a la tecnología se presenta la Figura 12 en donde se puede apreciar el mapa actual de la tecnología J2ME.

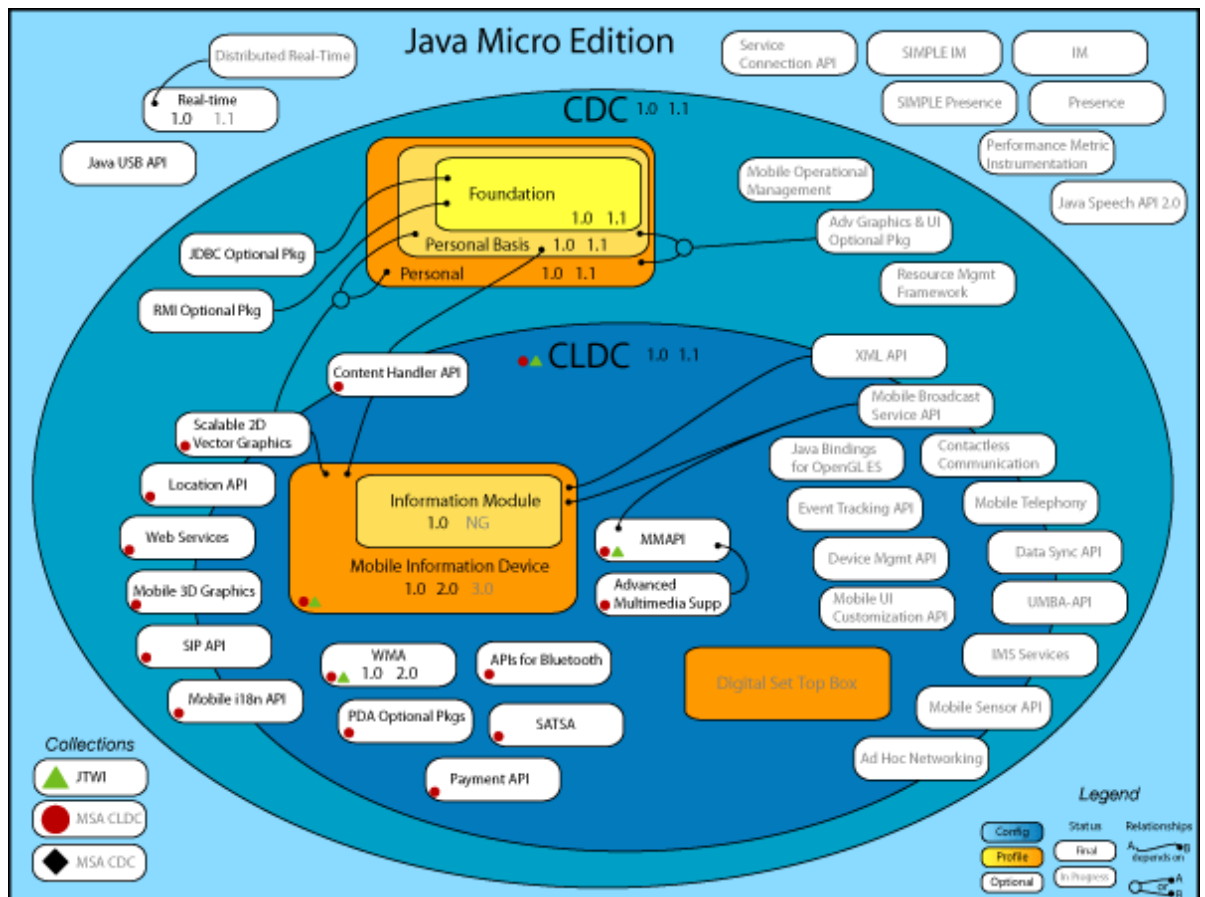


Figura 12. Mapa Actual de la Tecnología J2ME²⁰

2.4 APIS JAVA PARA LA TECNOLOGÍA BLUETOOTH

Aún cuando el hardware Bluetooth ha presentado grandes avances, no existía una manera estandarizada para desarrollar aplicaciones Bluetooth. Fue así, como se constituyó el grupo experto JSR-82²¹, que desarrolló un API Java de alto nivel para la programación de dispositivos Bluetooth. Dicho grupo experto está compuesto

²⁰ Obtenido de <http://java.sun.com>

²¹ Toda la información sobre la especificación JSR-82 se encuentra en el portal <http://jcp.org/en/jsr/detail?id=82>

por compuesto por compañías como IBM, Mitsubishi Electric, Motorola (Líder de la especificación), Newbury Networks, Nokia, Parthus Technologies, Research in Motion, Rococo Software, Sharp Laboratories of America, Sony Ericsson Mobile Communications, Smart Fusion, Smart Network Devices, Sun Microsystems, Symbian, Telecordia, Vaultus y Zucotto.

Esta especificación esconde la complejidad de dicha tecnología detrás de unos APIs que permiten concentrarse en el desarrollo en lugar de prestar atención a los detalles de bajo nivel de Bluetooth. El objetivo de la especificación es definir un API estándar abierto, no propietario que pueda ser usado en todos los dispositivos que implementen J2ME. Por tal razón fue diseñado usando los APIs J2ME y el entorno de trabajo CLDC/MIDP. Los APIs JSR-82 son muy flexibles, ya que permiten trabajar tanto con aplicaciones nativas Bluetooth como con aplicaciones Java Bluetooth. En la Figura 13 se muestra la arquitectura de la especificación JSR 082.

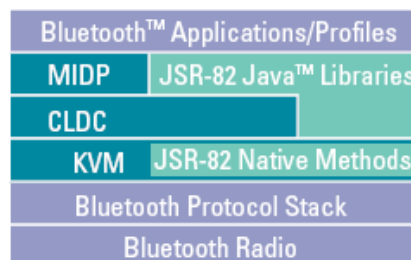


Figura 13. Arquitectura JSR-082

La especificación define dos paquetes: javax.bluetooth y javax.obex. Hay que resaltar que el API obex está definido independientemente de la capa de transporte Bluetooth. El paquete javax.bluetooth hace referencia al API central Bluetooth. El paquete javax.obex contiene los APIs para el protocolo de intercambio de objetos (OBject EXchange).

El paquete `javax.bluetooth` ofrece un conjunto de clases e interfaces Java que permiten la búsqueda de dispositivos Bluetooth cercanos, y la búsqueda y utilización de servicios Bluetooth.

2.5 DESARROLLO DE JUEGOS PARA MÓVILES

El desarrollo de juegos requiere atención al detalle y un entendimiento de varios aspectos técnicos, legales, sociales, psicológicos, culturales y económicos, cada uno de los cuales afecta la viabilidad del juego [6]. Hoy en día, juegos para PC y consolas de juego son grandes negocios, casi imposible de desarrollar sin una gran inversión de tiempo, dinero y otros recursos. Esto hace al nuevo sector de los juegos móviles e inalámbricos tan excitante, ya que hay un espacio considerable para que diseñadores y programadores innovadores para dejar su marca y edificar negocios con una cantidad limitada de recursos.

La codificación de juegos para teléfonos móviles requiere ingenio, pensamiento lateral, y una comprensión de los usuarios y de los factores sociales. Esto demanda una determinación para desarrollar lo mejor dentro de ciertas limitaciones. Para diseñadores, esto requiere la visión para crear una experiencia completamente nueva para usuarios que no tienen comprendido, hasta este punto, que desean jugar los juegos de su teléfono.

Miembros del panel de discusión de la Conferencia Global de Juegos Nokia 2002 en Londres convinieron que las descargas de juegos móviles más populares fueron las de emulaciones de deportes, carreras y de juegos de rol, con algunos mercados también atractivos para los juegos de máquinas tragamonedas [6]. Sin embargo, debe notarse que algunos de los juegos existentes son multijugador, y el

hecho de que dispositivos móviles estén conectados a una red y puedan soportar múltiples jugadores es uno de sus fuertes.

Los juegos deben ser rápidos y fáciles de iniciar y navegar. Deben ser fáciles de entender, no solamente en términos de permitir a los jugadores aprender rápidamente los procedimientos y procesos usados en el juego, contemplando también los conceptos que envuelven los juegos – los objetivos, riesgos, desafíos y recompensas.

Los juegos para móviles son típicamente más rápidos y baratos para desarrollar que los juegos digitales convencionales: un título de PC o consola ofrece un costo de \$2.5 millones de dólares o más para llegar al mercado, y hay un ciclo de desarrollo de dos o tres años, mientras que los juegos J2ME son a menudo desarrollados en tres o cuatro meses a un presupuesto de menos de \$100.000 dólares [6]. Los desarrolladores pueden completar su trabajo y ver a la gente jugando su juego en mucho menos tiempo, y producir muchos más títulos en una cantidad dada de tiempo que en la industria de juegos convencional.

3. MARCO METODOLÓGICO

Las metodologías de desarrollo de software revelan como construir técnicamente software. Estas abarcan tareas que incluyen planeación de proyectos, recolección y análisis de requerimientos, diseño, implementación de la aplicación, pruebas de funcionamiento, documentación y mantenimiento. La metodología de desarrollo de software varía según el tipo de proyecto, necesitando diferentes clases de tareas y distintos órdenes de las mismas. Una elección desacertada en la selección de una metodología de desarrollo de software puede desembocar en la omisión de tareas o en la secuenciación inapropiada de las mismas, lo cual puede afectar la calidad del producto final.

Existen muchos modelos de ciclo de vida de un proyecto con diferentes enfoques, cada uno con sus ventajas e inconvenientes. Proceso Unificado, Programación Extrema, Desarrollo en Cascada, Desarrollo en Espiral, Prototipado Evolutivo, Desarrollo Rápido de Aplicaciones (DRA), y Modelo Incremental, son algunas de las metodologías más conocidas y usadas para el desarrollo de proyectos de software.

La metodología usada para el desarrollo de este proyecto es el Prototipado Evolutivo, pues se considero que es la más apropiada para el tipo de aplicación que se deseaba desarrollar, debido a que no presenta el inconveniente de requerir la definición completa de los requerimientos antes de iniciar el diseño global.

3.1 **PROTOTIPADO EVOLUTIVO**

El Prototipado Evolutivo es un modelo que toma sus bases en el Prototipado desechable, pero con mayores controles sobre la calidad. Es un modelo en el que se desarrolla el concepto de sistema conforme el proyecto avanza; el sistema se desarrolla mediante incrementos, de tal manera que pueden realizarse cambios inmediatos de acuerdo a los requerimientos del cliente o usuario final, que colabora realimentando información al proyecto [1]. En el momento en el que el cliente o usuario final está satisfecho se le hace entrega de ese prototipo final. Esta metodología se compone de cuatro fases bien diferenciadas: Concepto Inicial, Diseño e Implementación del Prototipo Inicial, Refinamiento del prototipo y entrega del Prototipo Final. En la Figura 14 se presenta gráficamente el modelo de Prototipado evolutivo.

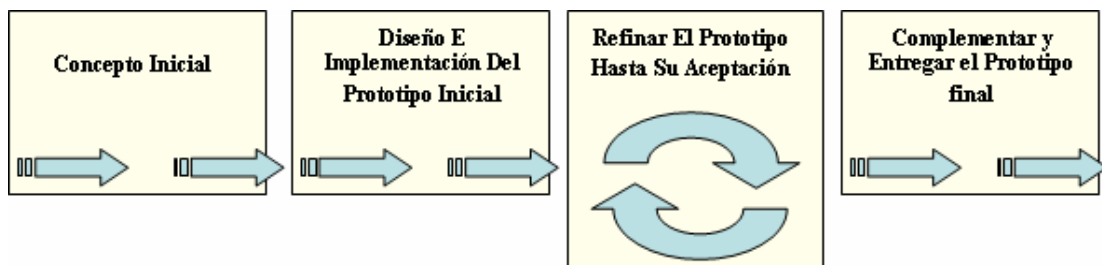


Figura 14. Modelo de Prototipado Evolutivo [1]

3.1.1 **CONCEPTO INICIAL**

El objetivo de esta fase es delimitar los aspectos generales del software. El desarrollador y el cliente definen y especifican los objetivos globales para el software, identifican los requerimientos conocidos y las áreas del esquema en donde es necesario profundizar la definición.

3.1.2 DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO INICIAL

En esta fase, se diseña y construye un prototipo inicial a partir de los requerimientos globales definidos en la fase de concepto inicial, para que el desarrollador en conjunto con el cliente, puedan evaluar el prototipo para modificarlo en la siguiente fase de ser necesario. El diseño que se lleva a cabo en esta fase es rápido y centrado en los aspectos del software que son visibles para el cliente o usuario.

3.1.3 REFINAMIENTO DEL PROTOTIPO HASTA SU ACEPTACIÓN

El objetivo de esta fase es que desarrollador y cliente evalúen el prototipo en evolución, corrijan fallas de diseño y refinen los requerimientos del software en desarrollo basándose en la realimentación, hasta que el cliente quede conforme con el producto.

3.1.4 ENTREGA DEL PROTOTIPO FINAL

En esta etapa se completa cualquier tarea pendiente realizando las revisiones pertinentes alcanzando la satisfacción total del cliente, pues se han obtenido las características, capacidades y desempeño necesarios. Finalmente, se hace entrega del producto final.

4. DESARROLLO DEL PROYECTO

4.1. BANDERCOL. JUEGO MULTIJUGADOR BLUETOOTH.

BANDERCOL es un juego multijugador Bluetooth para dispositivos móviles, en el cual el usuario puede crear su propio personaje de tres modelos de personaje diferentes. Una vez creado el personaje, el usuario puede iniciar una partida contra otro jugador que disponga de un dispositivo móvil con conectividad Bluetooth, para tal fin el juego permite crear y buscar partidas.

4.1.1 OBJETIVO

El fin del juego es recoger tres banderas ubicadas aleatoriamente en el escenario, cada una con uno de los colores respectivos de la Bandera de Colombia, y llevarlas una a la vez, hasta el asta ubicada en el punto de partida antes que el adversario. En cada escenario se encontrarán dos banderas de cada color.

4.1.2 PERSONAJES

El usuario puede crear su propio personaje, darle nombre y escoger su apariencia de cualquiera de los tres modelos que proporciona el juego. En la Figura 15 se muestran los tres modelos de personaje del juego: un soldado, un explorador y un indígena.
















Figura 15. Personajes de Juego

4.1.3 ESCENARIOS.

El juego cuenta con tres escenarios alusivos a sitios turísticos de Colombia. Son ellos: El Parque Tayrona, El Cabo de la Vela y El Parque Arqueológico San Agustín. Cada escenario esta compuesto por diversos objetos, algunos propios de cada escenario. Hay tres tipos de objetos: suelo firme, obstáculos, y terreno hostil. En la Tabla 5 se presentan cada uno de los objetos detallando el tipo de objeto, su tamaño en píxeles y los escenarios en los cuales aparece.

Tabla 5. Objetos de los Escenarios

OBJETO	IMAGEN	TIPO	TAMAÑO (Píxeles)	ESCENARIOS
Agua		Terreno Hostil	32x32	Todos los Escenarios
Árbol 1		Obstáculo	64x96	Parque Tayrona y Parque Arqueológico San Agustín
Árbol 2		Obstáculo	64x128	Parque Tayrona y Parque Arqueológico San Agustín.
Arbusto1		Obstáculo	32x32	Parque Tayrona y Parque Arqueológico San Agustín.
Arbusto 2		Obstáculo	64x64	Cabo de la Vela
Arena		Suelo Firme	32x32	Cabo de la Vela.

Arena Movediza		Terreno Hostil	32x32	Cabo de la Vela
Cactus		Obstáculo	32x32	Cabo de la Vela
Camino 1		Suelo Firme	32x32	Todos los Escenarios
Camino 2		Suelo Firme	32x32	Todos los Escenarios
Camino 3		Suelo Firme	32x32	Todos los Escenarios
Camino 4		Suelo Firme	32x32	Parque Tayrona y Cabo de la Vela
Camino 5		Suelo Firme	32x32	Parque Tayrona y Cabo de la Vela
Choza Tayrona 1		Obstáculo	64x64	Parque Tayrona
Choza Tayrona 2		Obstáculo	96x64	Parque Tayrona
Escultura 1		Obstáculo	96x64	Parque Arqueológico San Agustín
Escultura 2		Obstáculo	32x64	Parque Arqueológico San Agustín
Escultura 3		Obstáculo	32x64	Parque Arqueológico San Agustín
Palmera		Obstáculo	32x64	Parque Tayrona y Cabo de la Vela
Pantano		Terreno Hostil	32x32	Parque Tayrona
Playa 1		Terreno Hostil	32x32	Todos los Escenarios
Playa 2		Terreno Hostil	32x32	Todos los Escenarios
Prado 1		Suelo Firme	32x32	Parque Tayrona
Prado 2		Suelo Firme	32x32	Parque Arqueológico San Agustín
Roca		Obstáculo	64x32	Todos los Escenarios

Los objetos de Suelo firme, son aquellos sobre los cuales, el jugador puede desplazarse con toda libertad. Los objetos de Obstáculo impiden el avance del jugador, obligándolo a cambiar de rumbo. Finalmente están los objetos de Terreno Hostil, los cuales al hacer contacto con el jugador hacen su avance mucho más lento de lo normal.

Todos los escenarios tienen un tamaño de 800x800 píxeles, y están divididos en una cuadrícula de 25x25 baldosas (tiles), en donde son ubicados partes de objetos u objetos completos según se requiera. En la Figura 16 se presenta el escenario Parque Arqueológico San Agustín completamente construido.

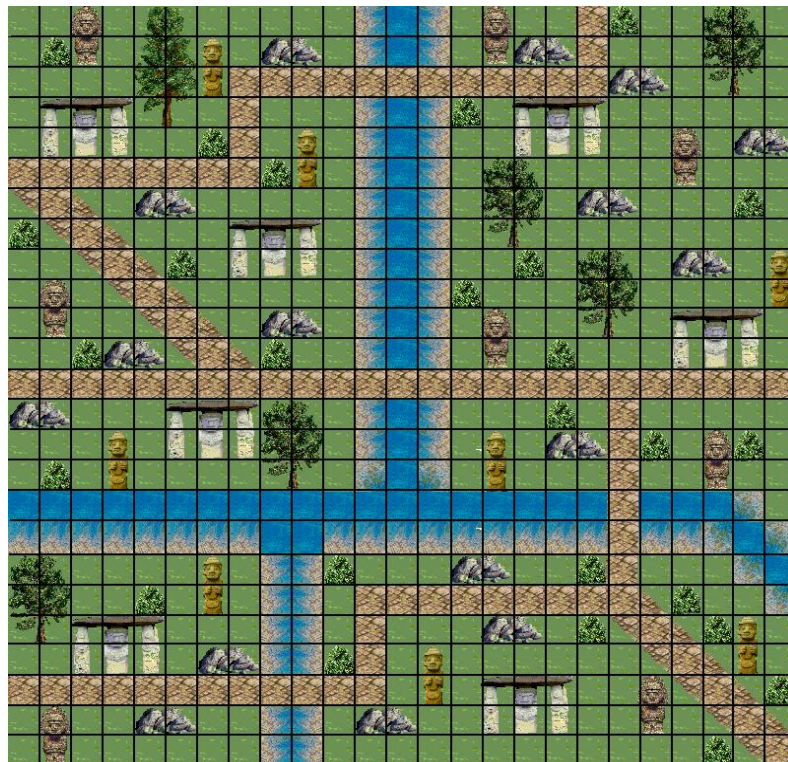


Figura 16. Escenario Completo.

4.2 ANÁLISIS DE REQUERIMIENTOS.

La aplicación desarrollada en el presente proyecto no busca ser únicamente una demostración de las tecnologías involucradas en él. Los requerimientos generales que se deben cumplir son los siguientes: Portabilidad, Jugabilidad e Interfaz de Usuario sencilla y amigable.

El primer requerimiento planteado es la *portabilidad*. Se pretende que la aplicación pueda ser ejecutada en una amplia gama de dispositivos móviles. Por dicha razón la aplicación ha sido desarrollada sobre la plataforma de desarrollo J2ME, que dentro de las varias ventajas que brinda ofrece dicha portabilidad, debido a que la gran mayoría de dispositivos móviles de tercera generación soportan juegos Java con independencia del sistema operativo.

Otro requerimiento de la aplicación es *Jugabilidad*, es decir, que sea un juego fácil de entender y navegar en él, ya que la aplicación no va dirigida a un usuario especializado, sino a cualquiera que posea un dispositivo móvil y desee instalarlo. Por tal motivo la aplicación debe manejar una Interfaz de Usuario sencilla, amigable, sobre la cual sea fácil navegar. En el caso que se produzca algún error, sea de comunicación o de configuración del dispositivo, se debe informar al usuario inmediatamente.

Adicional a estos requerimientos generales, es necesario seleccionar la configuración y el perfil J2ME sobre los cuales se va a desarrollar la aplicación, y los APIs J2ME que se utilizarán. Con respecto a la comunicación Bluetooth se debe elegir el protocolo de comunicación más adecuado para la aplicación. Igualmente es necesario diseñar un servicio Bluetooth que proporcione a la aplicación la conectividad que requiere.

4.3 DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

4.3.1 ASPECTOS GENERALES

Debido a que la aplicación va dirigida a dispositivos móviles con capacidades limitadas de procesamiento, memoria y conectividad, la aplicación se desarrolló sobre el perfil *MIDP*²² de la configuración *CLDC*²³. En la tabla 6 se muestra la lista de los APIs J2ME que fueron utilizados en el desarrollo de la aplicación.

Tabla 6. Paquetes utilizados en el Desarrollo de la Aplicación

PAQUETE	FUNCIONALIDAD
javax.microedition.midlet	Define las aplicaciones MIDP y las interacciones entre la aplicación y el entorno de ejecución.
javax.microedition.lcdui	Provee un conjunto de clases para la implementación de interfaces de usuario para aplicaciones MIDP
javax.microedition.lcdui.game	Provee una serie de clases que permiten el desarrollo de juegos para dispositivos móviles.
javax.microedition.rms	Proporciona un mecanismo para que las aplicaciones MIDP puedan almacenar datos persistentemente en los dispositivos, para recuperarlos después.
javax.microedition.io	Clases para la Estructura de Conexión Genérica de MIDP
javax.microedition.media	Define una serie de clases que permiten a las aplicaciones MIDP la reproducción y grabación audio.
java.io	Provee una serie de clases para el manejo de flujos de entrada y salida de datos.
java.lang	Proporciona un conjunto de clases fundamentales para la programación en lenguaje Java
java.util	Provee una colección de clases que facilitan la programación en el lenguaje Java
javax.bluetooth	Proporciona un mecanismo para que las aplicaciones MIDP puedan establecer conexiones Bluetooth.

²² Ver Perfil MIDP, Página 45

²³ Ver Configuración CLDC, Página 43

La elección del protocolo de comunicación Bluetooth se realizó basada en el artículo de Nokia Games Over Bluetooth [6], en el cual se hace especial énfasis en los protocolos L2CAP y RFCOMM como los más convenientes para este tipo de aplicaciones. Por esta razón, y por la facilidad a la hora de la programación, se escogió *RFCOMM* como protocolo de comunicación de la aplicación.

Después de realizar un análisis de cada uno de los perfiles Bluetooth, se concluyó que el Perfil de Puerto Serial o SPP por sus siglas en ingles (Serial Port Profile) es el perfil Bluetooth más apropiado para la aplicación, ya que provee los procedimientos básicos para buscar y conectarse a dispositivos bluetooth y es el perfil sobre el cual funciona el protocolo de comunicación RFCOMM puesto que es un protocolo que proporciona emulación de un puerto serial RS-232.

4.3.2 SERVICIO BLUETOOTH DE JUEGO

Para el desarrollo de la aplicación es necesario el diseño de un servicio Bluetooth de Juego. Dicho servicio posee las siguientes características:

Tabla 7. Servicio Bluetooth de Juego

Nombre	Juego
UUID²⁴	F0E0D0C0B0A000908070605040302010
Autorización	No
Encriptación	No
Atributos	Perfil Servidor Escenario

El servicio le permite al dispositivo maestro publicar su partida de tal modo que otros dispositivos puedan encontrarla, obteniendo así los clientes, el perfil del Servidor y el escenario en el cual se efectuará la partida. Una vez un cliente

²⁴ Identificador Universal Único del Servicio.

decide conectarse a la partida, el servicio cierra las conexiones con otros clientes, genera aleatoriamente las posiciones de las banderas en el escenario y las envía al cliente. De esta manera se da inicio al juego. El cliente le envía a su oponente, en cada ciclo de juego, un mensaje con los movimientos que él efectúa. El servidor recibe dicha información y responde enviando un mensaje con idénticas características al cliente. Adicionalmente, cliente y servidor se envían sus respectivas posiciones para corroborar que las posiciones de ambos jugadores sean las mismas en las dos terminales. En el caso que el usuario del dispositivo maestro cierre la aplicación, el servicio se encarga de informar al cliente que la partida terminó.

4.3.3 DISEÑO DE LA APLICACIÓN

Basados en las características del juego, es de entenderse que únicamente existe un tipo de actor, al cual nos referiremos como Jugador. A través de la aplicación el jugador podrá modificar las opciones de juego, Iniciar una partida del juego, buscar una partida abierta en un dispositivo móvil cercano, consultar la ayuda, o simplemente salir del juego. Ver Figura 17.

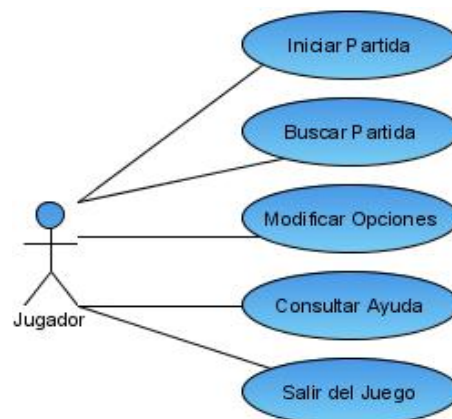


Figura 17. Diagrama de Casos de Uso

Al ejecutarse la aplicación esta debe verificar primero el acceso al modulo Bluetooth del dispositivo, en el caso de que este no sea efectivo la aplicación debe informar al usuario y cerrarse. Si el modulo Bluetooth esta disponible la aplicación debe verifica la existencia de un perfil; si este no existe la aplicación debe desplegar el formulario de Opciones, obligando al usuario a crear uno. Superando estas verificaciones de rigor la aplicación debe mostrar al usuario el menú principal del juego, el cual contiene las opciones: Iniciar Partida, Buscar Partida, Opciones, Ayuda y Salir.

De elegir la opción *Iniciar Partida*, la aplicación inicia el servicio bluetooth de juego y muestra al usuario una pantalla de espera, mientras un cliente se conecta a la partida. En la opción *Buscar Partida*, la aplicación inicia una búsqueda de dispositivos Bluetooth que tengan el servicio bluetooth de juego; al terminar la búsqueda despliega un menú con la lista de las partidas encontradas. En la opción *Opciones* el usuario tiene la oportunidad de modificar el perfil de su móvil, encender o apagar el audio del juego y elegir el escenario de juego en caso de que desee iniciar una partida. En la opción *Ayuda* se da una breve explicación del juego y la configuración de control. En la opción *Salir* se cierra la aplicación. Para un mejor entendimiento del funcionamiento inicial de la aplicación obsérvese la Figura 18 en donde se muestra el diagrama de actividades inicial de la aplicación.

Como se puede observar, en el diagrama de actividades se describe el flujo principal de cada uno de los casos de uso de la aplicación.

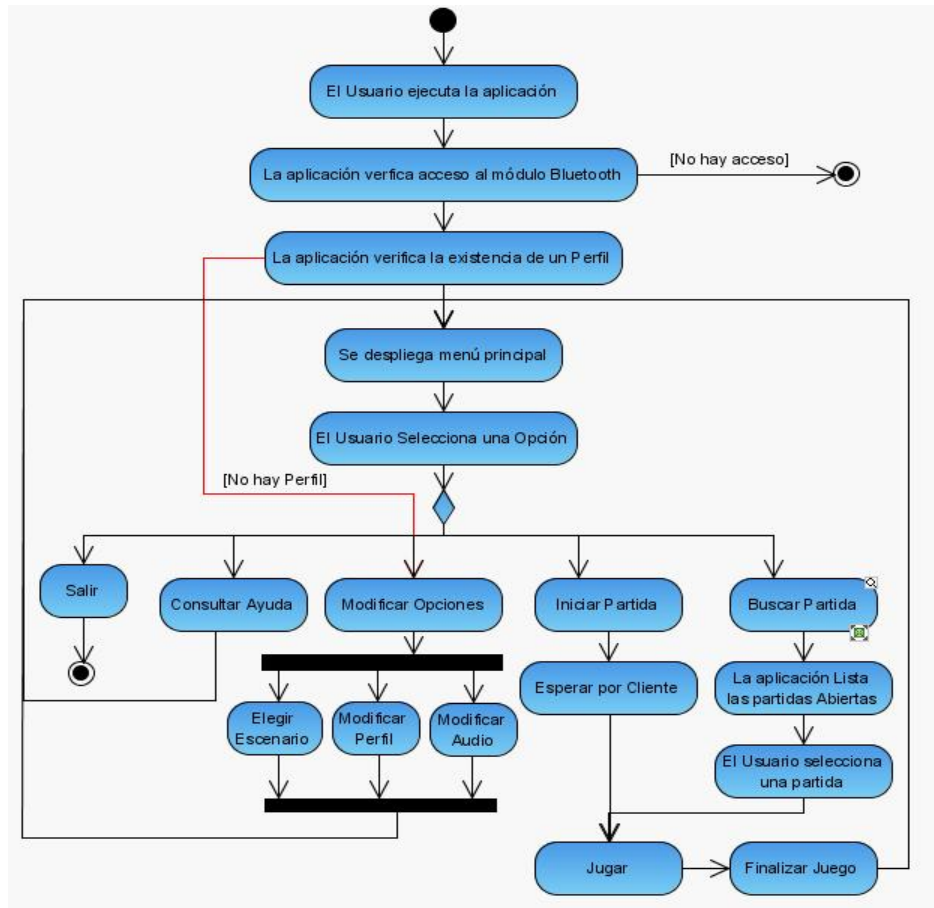


Figura 18. Diagrama de Actividades inicial de la Aplicación

En lo que respecta a las clases necesarias, estas fueron distribuidas en cuatro paquetes de acuerdo al tipo de función que desempeñan: principal, interfaz de usuario, juego y comunicación. El paquete principal está compuesto de una única clase, la clase principal que extiende de la clase midlet. En el paquete interfaz de usuario se encuentran todas las clases que van a generar la interfaz de usuario de la aplicación. En el paquete juego se encuentran todas las clases necesarias para la ejecución del juego. Y en el paquete comunicación están las clases que se encargan de la conexión Bluetooth.

Debido a que no es posible presentar todas las clases en un único diagrama de clases por razones de espacio, se presentarán agrupadas según su relación.

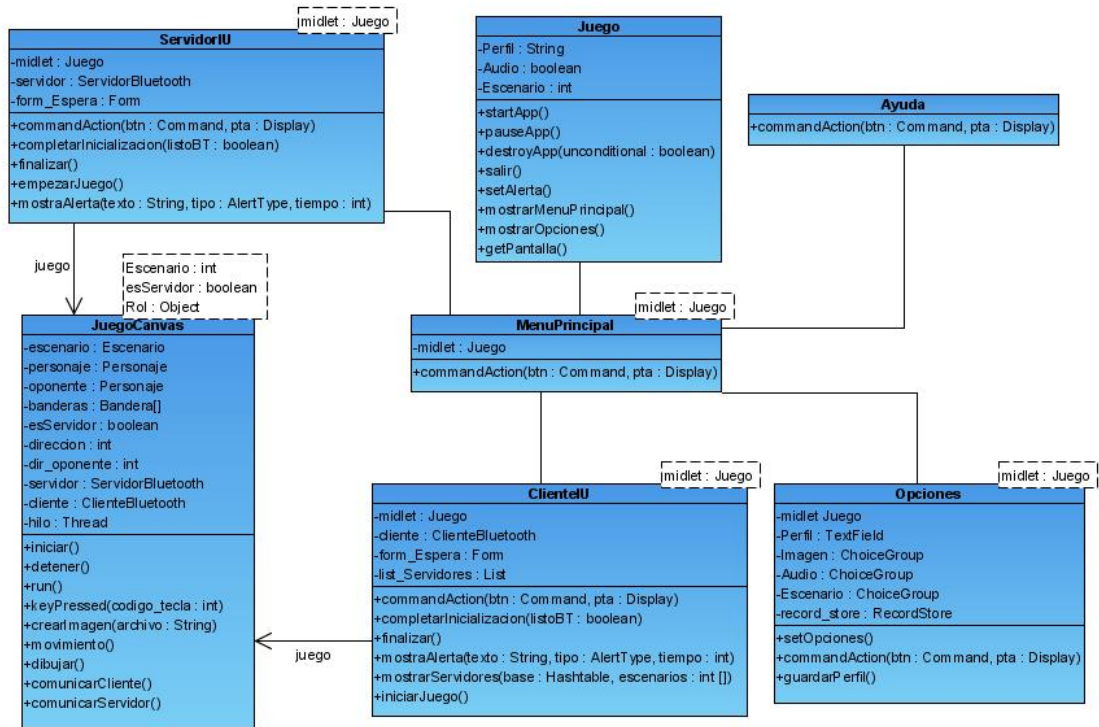


Figura 19. Diagrama de Clases de Interfaz de Usuario

En la Figura 19 se muestra un diagrama que incluye las clases que tienen que ver con la interfaz de usuario, mientras que en la Figura 20 se presenta un diagrama con las clases que se relacionan directamente con la implementación del juego.

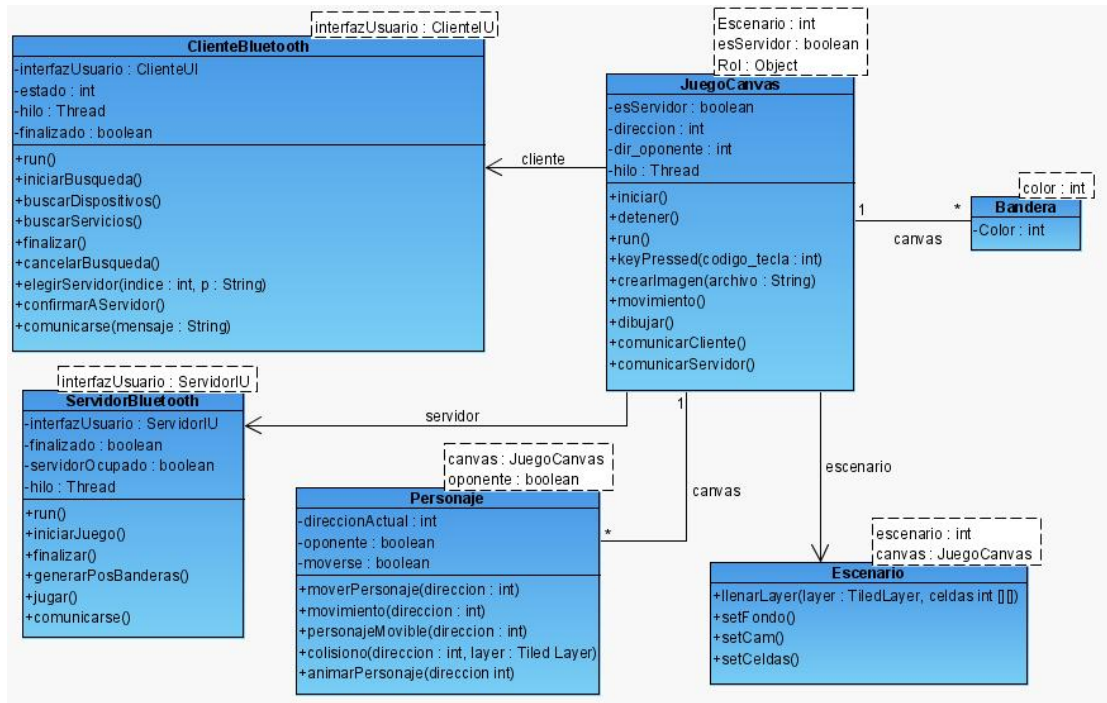


Figura 20. Diagrama de Clases del Juego

4.3.4 IMPLEMENTACIÓN DE LA APLICACIÓN

En el desarrollo de aplicaciones para dispositivos móviles es necesario tener en cuenta detalles como, poder de procesamiento y capacidad de almacenamiento restringidos de los dispositivos, además de medios limitados de interacción con el usuario (tamaño de pantallas y teclados).

Durante el desarrollo de la aplicación del presente proyecto y por la misma metodología de desarrollo de software utilizada, se construyeron tres prototipos de la aplicación, antes de llegar a un prototipo final. A continuación se detallan las características y alcances de cada uno de los prototipos.

4.3.4.1 Primer Prototipo

La implementación de este primer prototipo abarca la parte de la aplicación concerniente a la interfaz de usuario inicial y no involucra aún algún tipo de conectividad. Las clases utilizadas son todas las presentadas en el diagrama de clases de interfaz de usuario de la Figura 19²⁵, con excepción de las clases JuegoCanvas, ServidorIU y ClienteIU.

La clase Juego extiende de la clase midlet, y es la clase principal de la aplicación. La clase MenuPrincipal extiende de la clase List y representa el menú principal de la aplicación; el menú está compuesto por cinco opciones: iniciar partida, buscar partida, opciones, ayuda y salir. Las dos primeras opciones, Iniciar Partida y Buscar Partida, no tienen funcionalidad alguna.



Figura 21. Menú Principal de la Aplicación

La selección del ítem opciones crea una instancia de la clase opciones, la selección del ítem ayuda crea una instancia de la clase ayuda y la selección del ítem salir cierra la aplicación.

²⁵ Ver Página 67

La clase Opciones extiende de la clase Form y representa el formulario en donde el usuario puede modificar opciones de juego tales como su perfil, audio y escenario de juego, este último en caso de que desee iniciar una partida. La información del perfil, nombre y personaje, son almacenadas persistentemente en el dispositivo, para lo cual se requirió la utilización del paquete javax.microedition.rms de MIDP, destinada para tal fin. Las opciones de audio y escenario no tienen aún funcionalidad.



Figura 22. Formulario Opciones de Juego



Figura 23. Pantalla de Ayuda

La clase Ayuda se encarga de mostrar la pantalla de ayuda, en la cual el usuario puede consultar la lógica del juego y la configuración del teclado.

El funcionamiento de este primer prototipo se comprobó mediante la utilización de diversos emuladores de dispositivos móviles, entre ellos los emuladores del Sun Wireless Toolkit Beta 2.3 y los emuladores de los teléfonos celulares Nokia serie 40 y serie 60, en los cuales los resultados fueron satisfactorios.

4.3.4.2 Segundo Prototipo

Continuando con el refinamiento del primer prototipo, se llegó a un segundo prototipo, el cual posee en adición a al prototipo inicial, algunas funcionalidades de conexión bluetooth. Este prototipo permite al usuario iniciar el servicio Bluetooth de modo que otros usuarios puedan accederlo. Además, permite también buscar partidas abiertas en las cercanías y lista en pantalla, los perfiles de otros usuarios que hayan iniciado el servicio. Para tal efecto se implementaron cuatro clases adicionales: ServidorIU, ServidorBluetooth, ClienteIU, y ClienteBluetooth.



Figura 24. Interfaz de Usuario del Servidor.

La clase ServidorIU se encarga de presentar la interfaz al usuario que haya elegido iniciar una partida nueva. Inicialmente presenta una pantalla de espera, mientras aguarda a que un cliente se conecte a la partida; ocurrido esto, presenta una imagen del escenario escogido por el usuario en el formulario opciones (Ver Figura 24). Por su parte, la clase ServidorBluetooth se encarga de crear e iniciar el servicio bluetooth de Juego, y pone al dispositivo local en estado descubrible, de tal modo que otros dispositivos puedan acceder al servicio.

La clase ClienteIU funciona de manera análoga a la clase ServidorIU, presentando una interfaz al usuario que ha elegido la opción buscar partida del menú principal. Primero presenta una pantalla de espera, mientras se buscan partidas abiertas. Terminada la búsqueda presenta una lista con los perfiles de los usuarios que tengan iniciado el servicio bluetooth de Juego en las cercanías. En caso de no encontrar ninguna partida abierta, la aplicación lo informa al usuario inmediatamente. Si el usuario se conecta a una de las partidas, presenta en pantalla el escenario de la partida a la cual se conecto.



Figura 25. Interfaz de Usuario del Cliente

La clase ClienteBluetooth se encarga de hacer la búsqueda de dispositivos Bluetooth en las cercanías, la búsqueda del servicio Juego en los dispositivos

encontrados, y la obtención de la información de cada una de las partidas encontradas (Perfil del usuario remoto y escenario de la partida).

En caso de presentarse algún error de conexión las clases ClienteBluetooth y ServidorBluetooth se encargan de informar a las clases ClienteIU y ServidorIU respectivamente, para que lo comuniquen al usuario.

4.3.4.3 Tercer Prototipo

Comprobado el correcto funcionamiento del segundo prototipo en los emuladores, se procedió al desarrollo de un tercer prototipo. Dicho prototipo reúne las características de sus predecesores y adicionalmente implementa algunas de las características del juego.

Una vez iniciada una partida, los dos usuarios de la aplicación (cliente y servidor) pueden movilizar su personaje por todo el escenario y visualizarse el uno al otro, esto último en caso de estar dentro del rango de la pantalla del oponente. La aplicación impide el paso del usuario por sobre los objetos del escenario considerados como obstáculos y dificulta el paso por sobre aquellos considerados como terreno hostil.

Ambos dispositivos intercambian la información del movimiento hecho por cada usuario en cada ciclo del juego. Adicionalmente intercambian sus posiciones en el escenario, para asegurar que estas coincidan en uno y otro dispositivo móvil. La aplicación está además, en la capacidad de informar al usuario en caso de que se pierda la comunicación.



Figura 26. Tercer Prototipo - Visualización del Juego

4.3.4.4 *Prototipo Final*

En definitiva, se llegó a un prototipo que cumplía con todas las características especificadas en el diseño de la aplicación. Dicho prototipo permite al usuario modificar opciones de juego tales como escenario, audio y perfil; iniciar una partida del juego, o buscar una partida abierta en los dispositivos cercanos y conectarse a ella. Una vez iniciada una partida, puede desplazarse por todo el escenario, recoger banderas y llevarlas hasta el asta ubicada en su punto de partida.

La aplicación posee la característica de almacenamiento persistente de datos en el dispositivo, ya que permite almacenar el perfil del usuario (nombre y modelo del personaje). Cuenta además, con reproducción de sonidos para hacer más motivante la experiencia de jugar.



Figura 27. Prototipo Final - Visualización del Juego

5. EVALUACIÓN

La evaluación del proyecto fue un proceso constante e iterativo, y se realizó durante el desarrollo del mismo. Durante la fase de diseño e implementación se realizaron pruebas a cada uno de los prototipos con el fin de valorar el desempeño de cada uno de sus diferentes procedimientos. En base a dichas valoraciones se planeaba y diseñaba el prototipo siguiente hasta que este cumpliera con los objetivos iniciales del proyecto.

Con respecto a la aplicación, en cada una de las fases de prueba se valoraron los siguientes aspectos:

Integridad de interfaz: Se probaron las interfaces de Usuario a medida que se agregaban nuevas funcionalidades a la aplicación. Se valoró su simplicidad y facilidad de entendimiento y navegación.

Validez funcional: se realizaron pruebas desde diferentes emuladores de dispositivos móviles con el fin de descubrir errores funcionales. Además, se realizó una prueba adicional en dispositivos móviles reales para verificar su funcionamiento.

Rendimiento: Se llevaron a cabo pruebas diseñadas para verificar los límites de rendimiento establecidos durante la ejecución de la aplicación.

Finalmente, se obtuvo un prototipo lo suficientemente completo para cumplir con las perspectivas establecidas al inicio del proyecto.

CONCLUSIONES

Los objetivos planteados desde un principio fueron cumplidos durante el transcurso del desarrollo del proyecto. Además se experimentó y familiarizó con tecnologías aplicadas en la línea de la computación móvil fortaleciendo y potenciando fortalezas en dicho campo.

En el plano investigativo se realizó un estudio de la tecnología inalámbrica Bluetooth y de la plataforma de desarrollo para dispositivos móviles J2ME. Basado en dicho estudio se concluyó que Bluetooth es una tecnología inalámbrica con un gran potencial debido a su alto grado de interoperabilidad, su bajo consumo de energía, bajo peso y poco costo, pero es difícil obtener una visión suficientemente global de la tecnología, debido a que esta descrita por una compleja y extensa especificación.

Se concluyó además, que J2ME es la mejor alternativa a la hora de desarrollar aplicaciones para dispositivos móviles por su robustez, portabilidad y confianza, características distintivas del lenguaje de programación Java.

Se constató que las tecnologías existen y funcionan, y que además en nuestro país, y particularmente en nuestra región, es posible implementar aplicaciones que involucren tales tecnologías sin que sea necesario importar el conocimiento, ni hacer grandes inversiones de dinero.

En el plano de desarrollo de la aplicación, se concluyó que las metodologías típicas para el desarrollo de software, y específicamente la metodología de

Prototipado Evolutivo, son perfectamente aplicables para los procesos de desarrollo de aplicaciones móviles.

El desarrollo de aplicaciones móviles inalámbricas representa un nuevo ámbito tecnológico con un gran potencial dentro de la sociedad de la información. Más aún cuando existen tecnologías como Bluetooth y J2ME que cuentan con una tecnología hardware que les brinda soporte. Actualmente, en nuestro país es posible encontrar una amplia gama de dispositivos que soportan estas dos tecnologías, como teléfonos celulares, PDAs y BlackBerrys.

Gracias a los esfuerzos del grupo JSR-82, es posible afirmar que el desarrollo de aplicaciones Java Bluetooth no es tan complejo como podría pensarse, puesto que esta abstraer en unos APIs, fáciles de usar, toda la complejidad de la especificación Bluetooth.

Dentro de las características más importantes del juego desarrollado están el manejo de interfaces gráficas para teléfonos móviles, un uso intensivo de la tecnología inalámbrica Bluetooth y la capacidad para establecer interactividad entre dos dispositivos móviles sin incurrir en costos de conexión.

La implementación de nuevas aplicaciones que involucren tecnologías como Bluetooth y J2ME no está limitada únicamente al ocio, como lo es el caso particular del desarrollo de juegos para móviles. Existe un mundo de posibilidades que puede dar espacio a la creación de nuevas aplicaciones. Por ejemplo, el uso del teléfono celular como medio de identificación de usuarios o como control universal de una Home base-station (Casa Inteligente).

RECOMENDACIONES Y TRABAJOS A FUTURO

Una vez finalizado el desarrollo del proyecto, es posible realizar algunas recomendaciones para trabajos a futuro en la línea de la Computación Móvil.

Es de vital importancia para la Universidad Industrial de Santander que a su interior se sigan llevando a cabo estudios e investigaciones sobre las tecnologías involucradas con esta nueva línea de investigación conocida como Computación Móvil, no sólo por el simple hecho de estar a la vanguardia dentro de la sociedad de la información, si no también porque esto le permitirá ser foco de conocimiento e innovación que estimulen el desarrollo de la región.

Es recomendable el estudio de nuevas tecnologías (como Bluetooth y J2ME) e incorporar su uso en el desarrollo de aplicaciones empresariales, buscando crear una nueva clase de aplicación empresarial que incorpore la utilización dispositivos móviles de uso común como teléfonos celulares, buscapersonas y PDAs.

Se recomienda la realización de proyectos que involucren el desarrollo de aplicaciones móviles multicanal, de modo tal que sea posible evaluar varias tecnologías de comunicación inalámbricas al mismo tiempo y que se pueda ver la viabilidad de una posible integración entre distintos sistemas de comunicación.

Con respecto a trabajos a futuro basados en el presente proyecto, se podrían desarrollar aplicaciones Bluetooth basadas en otras plataformas de desarrollo, como C++ o Delphi, con el fin de comprobar su interoperabilidad con las aplicaciones J2ME.

REFERENCIAS

- [1] MCCONELL, STEVE. Desarrollo y Gestión de Proyectos Informáticos. Primera Edición. McGraw-Hill. España, 1997.
- [2] BLUETOOTH.COM, The Official Bluetooth Wireless Info Site (en línea). <http://www.bluetooth.com/>
- [3] CRUZ ORTÍZ, HAROLD. Revista ACIS. Edición N° 87. Colombia, Enero del 2004. Artículo: Computación Móvil: Retos y Oportunidades.
- [4] GARCIA DÍAZ, JORGE. Estudio Tecnológico Bluetooth. Universidad Carlos III Madrid. España. <http://www.it.uc3m.es/pervasive/documentos/Bluetooth.pdf>.
- [5] KLINGSHEIM, ANDRÉ. J2ME Bluetooth Programming. University of Bergen. Junio de 2004.
- [6] FORUM NOKIA, Resources for Mobile Application Developers (en línea). <http://www.forum.nokia.com/>.
- [7] FROUFE QUINTAS AGUSTÍN, JORGE CÁRDENAS PATRICIA. J2ME Java 2 Micro Edition, Manual de Usuario y Tutorial. Editorial Ra–ma. Diciembre de 2003.
- [8] GÁLVEZ ROJAS SERGIO, ORTEGA DÍAZ LUCAS. Java a Tope: J2ME (Java 2 Micro Edition), Edición Electrónica. Universidad de Málaga, 2003.

[9] LAMUS BRAVO RODIAN HERNANDO. Plan de Proyecto de Grado. Universidad Industrial de Santander. Colombia, 2006.

BIBLIOGRAFÍA

PRESSMAN, Roger. Ingeniería del software. Un enfoque práctico. Cuarta edición. *McGraw Hill*. España, 1998

GRECH, Pablo. Introducción a la Ingeniería. Un enfoque a través del diseño. *Pearson Educación de Colombia*, 1ª edición. Bogotá, 2001.

JACOBSON, Ivar. BOOCH, Grady. RUMBAUGH, James. El Lenguaje Unificado de Modelado. Primera edición. *Addison Wesley*. España, 1999.

FOWLER, Martin. SCOUT, Kendall. UML Gota a Gota. Primera Edición. *Addison Wesley Longman de Mexico S.A de C.V.* Mexico 1999.

HARKEY, Dan. APPAJODU, Shan. LARKIN, Mike. Wireless Java Programming for Enterprise Applications: Mobile Devices Go Corporate. Primera Edición. *John Wiley & Sons*. September 13, 2002.

FROUFE QUINTAS, Agustín; JORGE CÁRDENAS, Patricia. J2ME Java 2 Micro Edition, Manual de Usuario y Tutorial. *Editorial Ra–ma*. Diciembre del 2003.

KUMAR, Bala. KLINE, Paul. THOMPSON, Tim. Bluetooth Application Programming with the Java APIs. *Morgan Kaufmann*. September 17, 2003.

BLUETOOTH.COM, The Official Bluetooth Wireless Info Site (en línea). <http://www.bluetooth.com/>.

FORUM NOKIA, Resources for Mobile Application Developers (en línea).
<http://www.forum.nokia.com/>.

GARCIA DÍAZ, JORGE. Estudio Tecnológico Bluetooth. Universidad Carlos III Madrid. España. <http://www.it.uc3m.es/pervasive/documentos/Bluetooth.pdf>.

SOLARTE, Mario. HURTADO, Javier. CAICEDO, Oscar. Marco Conceptual, Técnico y Metodológico para el desarrollo de Aplicaciones para Dispositivos Móviles. Tecnocon 2003, Medellín - Mayo 2003.

BUTTERY, S. SAGO, A. Future Applications for Bluetooth. BT Technology Journal; 2003; 21, 3; ABI/INFORM Global, Pág. 48.

Wai Sum Sam Mok. The Electronic Library. Oxford: 2002. Tomo 20, N° 2; Pág. 113 - 119.

HURTADO, Javier. CHICANGANA, Mary Luz. BASTO, Fabian. Desarrollo de Aplicaciones Java Bluetooth utilizando el API Java JSR-82. Tecnocon 2004, Medellín - Abril 2004.

Revista ACIS, de la Asociación Colombiana de Ingenieros de Sistemas. Edición N° 87. Colombia, Enero del 2004.

KLINGSHEIM, ANDRÉ. J2ME Bluetooth Programming. University of Bergen. Junio de 2004.

XPROGRAMMING.COM, an Agile Software Development Resource (En línea)
[http:// www.xprogramming.com](http://www.xprogramming.com)

JAVA TECHNOLOGY, Sun Developer Network (En línea). <http://java.sun.com>

JAVA EN CASTELLANO, Programación de Juegos para móviles con J2ME (En línea). http://programacion.com/java/tutorial/ags_j2me.

ANEXO A: MANUAL DE USUARIO

BANDERCOL es un juego multijugador para dispositivos móviles en el cual el usuario puede gestionar su perfil (Nombre y Modelo de personaje). Una vez creado el perfil, el usuario puede jugar una partida contra otro jugador, para lo cual el la aplicación permite iniciar y buscar partidas.

INSTALACIÓN

La instalación de la aplicación depende de dos factores primordiales: el modo de descarga y del fabricante del dispositivo. Un juego Java para dispositivo móvil puede ser instalado por descarga directa o mediante una instalación externa al dispositivo. Por descarga directa, normalmente se hace mediante el acceso a una página Web. Una vez descargado se procede a instalar el juego siguiendo los pasos indicados en la pantalla del dispositivo. La instalación externa depende del dispositivo móvil utilizado. Los fabricantes de dispositivos ofrecen herramientas software para la sincronización de sus dispositivos con computadores personales mediante el uso de cables USB o vía Bluetooth.

En el caso de no disponer de un teléfono celular, puede probarse la ejecución de la aplicación mediante el uso de emuladores de móviles, como es el caso del *Java ME SDK for CLDC* de Sony Ericsson, disponible para descargar desde el sitio Web http://developer.sonyericsson.com/site/global/docstools/java/p_java.jsp.

Una vez instalado el SDK, es posible probar el juego ejecutando el archivo Run MIDP Application del submenú WTK2, el cual mostrará un cuadro donde se selecciona el archivo .jad de la aplicación a probar y se da clic en el botón Run (Ver Figura 28).

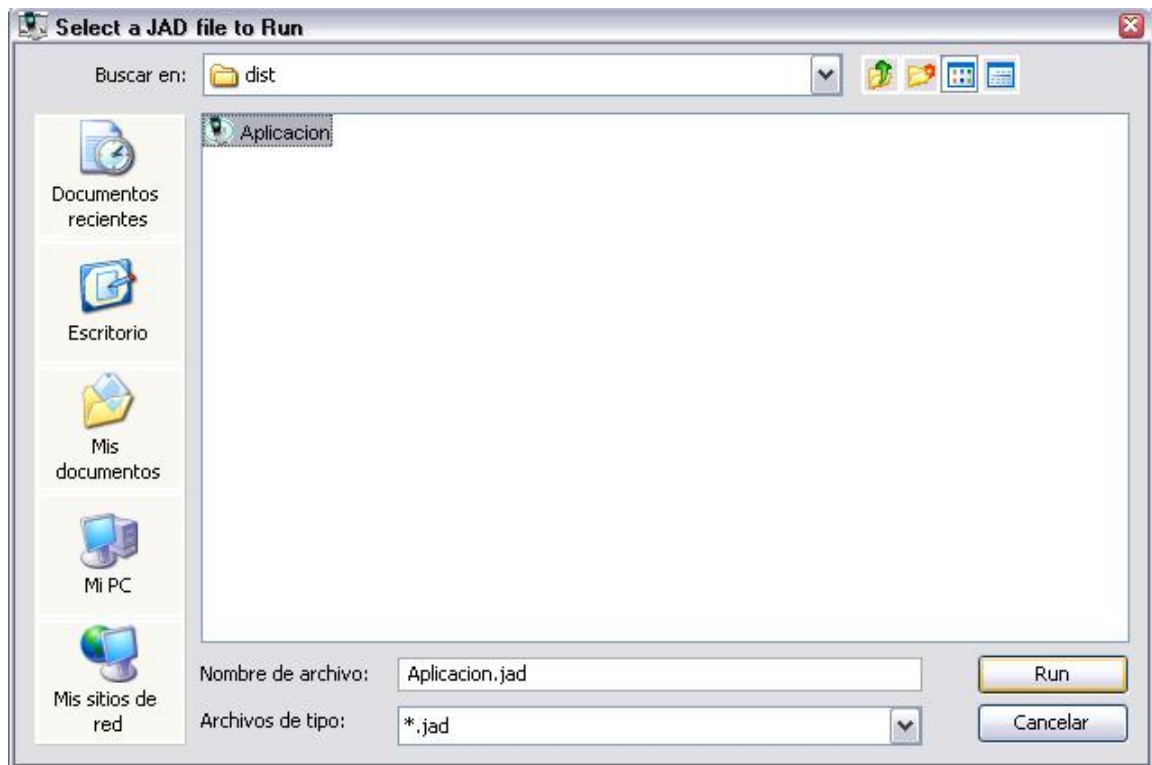


Figura 28. Run MIDP Application del Java ME SDK for CLDC de Sony Ericsson

EJECUCIÓN DE LA APLICACIÓN

Al momento de iniciar el juego por primera vez aparecerá un Formulario de Opciones, en donde el usuario puede crear su perfil, dándole un nombre y seleccionando uno de los modelos de personaje que presenta el juego: Soldado, Explorador e Indígena. Además el usuario tendrá la opción de encender o apagar el audio del juego y de seleccionar un escenario para el caso en que decida iniciar una partida (Ver Figura 29). El nombre y modelo del perfil son almacenados persistentemente en el dispositivo, de modo que el usuario no necesita realizar

esta operación cada vez que ejecute el juego. Además estas opciones pueden ser modificadas por el usuario accediendo al ítem opciones del Menú Principal.

Para navegar por el formulario, se utilizan las teclas de navegación del dispositivo (marcadas con azul en la Figura 29 para el caso del Emulador Sony Ericsson K750). Para ingresar el nombre se utiliza el teclado alfanumérico del dispositivo. Para seleccionar cualquiera de las opciones con botón de radio se debe oprimir el botón de selección (marcado con amarillo en la Figura 29). Para Aceptar las modificación de las opciones es necesario oprimir el botón Aceptar (marcado con verde) y para cancelar se debe oprimir el botón Cancelar (marcado con rojo).

Excepto por la primera vez que se ejecute el juego, aparecerá una pantalla de selección donde el usuario podrá elegir que opción desea realizar. El está compuesto por cinco opciones: Iniciar Partida, Buscar Partida, Opciones, Ayuda y Salir. En el ítem iniciar Partida, se crea una partida del juego sobre el escenario seleccionado en el formulario Opciones. En el ítem Buscar Partida, el juego busca partidas abiertas en otros dispositivos móviles y permite conectarse a una de ellas. El ítem opciones ya ha sido descrito anteriormente. El ítem Ayuda presenta una descripción breve del juego y la configuración de los controles del juego. El ítem Salir permite cerrar la aplicación. Para navegar por el menú, se utilizan las teclas de navegación del dispositivo, y para seleccionar una de las opciones se utiliza el botón de selección del dispositivo.

En caso de elegir Iniciar Partida, la aplicación esperará por otro jugador para iniciar el juego. En el caso de elegir Buscar Partida, la aplicación mostrará una pantalla con la lista de las partidas abiertas encontradas, de la cual el usuario podrá seleccionar una para conectarse a ella e iniciar así el juego.



Figura 29. Primera Pantalla de la Aplicación

Una vez iniciado el juego, cada jugador en su dispositivo visualizará su personaje ubicado al lado de un asta de bandera , logrando moverlo a través del escenario utilizando las teclas de navegación de su dispositivo. En el escenario existen ciertos objetos que presentan obstáculo al personaje y otros que dificultan su andar. Además de esto, hay 6 banderas cada una con uno de los colores emblemáticos de la bandera de Colombia, hay dos banderas por cada color (amarillo, azul y rojo). El juego consiste llevar una bandera de cada color hasta el asta ubicada en el punto de partida. El jugador que logre armar primero la bandera de Colombia gana.

Para finalizar el juego basta con pulsar cualquiera de las teclas diferentes a las teclas de navegación, de selección o del teclado alfanumérico del dispositivo. La aplicación informará inmediatamente al oponente el fin de la conexión.