

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

DESARROLLO DE APLICACIÓN PARA DISPOSITIVOS MÓVILES

Tesis Profesional

Qué para obtener el título de:

INGENIERO EN COMPUTACIÓN

Presenta:

JUAN CARLOS VÁZQUEZ LORENZO

No. de cuenta 09712146-8

DIRECTOR DE TRABAJO:

Ing. En Comp. y M. A. O. Angélica M^a. Ramírez Bedolla



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

"Si una persona es perseverante, aunque sea dura de entendimiento, se hará inteligente; y aunque sea débil se transformará en fuerte."

... Leonardo da Vinci

...DIOS padre mío gracias por indicarme el camino correcto y permanecer siempre a mi lado.

A mis Padres y a Rocío, mí querida Esposa.

Lo único que espero es que sientan la misma alegría que yo siento y se sientan orgullosos de mi, no tengo palabras para agradecerles todo lo que por mí han hecho, sin embargo de corazón les ofrezco y dedico mis estudios concluidos. Espero que como en esta ocasión permanezcan a mi lado celebrando todos mis triunfos y logros, así como también en los momentos difíciles.

...A mis Padres agradezco sus consejos y cuidados, no hubiera llegado a mis metas sin sus enseñanzas.

...A mi Reinita Linda su paciencia y apoyo desde que inicie mi carrera, su comprensión y amor, gracias Chio.

...A mi hermoso bebé que viene en camino gracias por ser mi mayor motivo para alcanzar esta meta.

...A mi asesora de tesis le agradezco el tiempo brindado y su orientación durante todo el proceso, gracias Angélica

Juan Carlos Vázquez Lorenzo

INDICE

| | |
|--|----|
| I.- PRESENTACIÓN..... | 1 |
| II.-INTRODUCCIÓN..... | 2 |
| III.-APLICACIONES MÓVILES..... | 5 |
| A) MARCO TEÓRICO..... | 5 |
| B) SITUACIÓN ACTUAL..... | 7 |
| C) SITUACIÓN ACTUAL EN MÉXICO..... | 13 |
| D) NECESIDADES Y TENDENCIAS..... | 16 |
| IV.-LENGUAJES DE PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES..... | 19 |
| A) EN QUE DESARROLLAR..... | 19 |
| B) HERRAMIENTAS OPENSOURCE PARA DESARROLLO DE APLICACIONES PALM..... | 21 |
| C) J2ME JAVA 2 MICRO EDITION..... | 24 |
| D) VISUAL STUDIO .NET Y MICROSOFT .NET COMPACT FRAMEWORK..... | 31 |
| V.-TECNOLOGÍAS y PLATAFORMAS..... | 33 |
| A) TECNOLOGÍAS INALÁMBRICAS RELACIONADAS CON APLICACIONES MÓVILES..... | 33 |
| 1.-BLUETOOTH..... | 33 |
| 2.-REDES INALÁMBRICAS WI-FI..... | 42 |
| 3.- INFRARROJO IrDA..... | 50 |
| B) TIPOS DE MÓVILES..... | 51 |
| 1.- PDA..... | 51 |
| 2.- TELÉFONOS INTELIGENTES..... | 53 |
| C) PLATAFORMAS..... | 55 |
| 1.- PALM OS..... | 55 |
| 2.- SYMBIAN..... | 59 |
| 3.- WINDOWS MOBILE..... | 61 |
| VI.-DESARROLLO DE APLICACIÓN PARA PDA Y DISPOSITIVOS MÓVILES COMPATIBLES..... | 63 |
| A) VISIÓN DEL PROYECTO..... | 64 |
| 1.- INTRODUCCIÓN..... | 64 |
| 2.- OBJETIVO..... | 64 |
| 3.- DESCRIPCIÓN DE LA APLICACIÓN LECTORBMU..... | 64 |
| 4.- PROBLEMAS POR NO TENERLO..... | 66 |

| | |
|---|-----|
| 5.- PROCEDIMIENTO ACTUAL..... | 66 |
| 6.- BENEFICIO..... | 66 |
| B) DISEÑO CONCEPTUAL..... | 67 |
| 1.-CLASIFICACIÓN DEL PROYECTO..... | 67 |
| 2.- TIPO DE PROYECTO..... | 67 |
| 3.- SOLUCIONES PROPUESTAS PARA IMPLEMENTAR EL PROYECTO..... | 67 |
| 4.- ARQUITECTURA..... | 68 |
| 5.- METODOLOGÍA..... | 71 |
| 6.- ESTIMACIÓN DE ESFUERZO Y RECURSOS..... | 72 |
| 7.- MANEJO DE RIESGOS Y SEGURIDAD..... | 72 |
| 8.- PLAN DE PRUEBAS..... | 72 |
| 9.- PLAN DE CAPACITACIÓN..... | 74 |
| C) ESPECIFICACIÓN FUNCIONAL..... | 75 |
| 1.- DISEÑO LÓGICO..... | 75 |
| 2.- DISEÑO FÍSICO..... | 75 |
| 3.- RESULTADO DEL PLAN DE PRUEBAS..... | 91 |
| 4.- PLAN DE LIBERACIÓN..... | 95 |
| CONCLUSIONES..... | 96 |
| ANEXOS..... | 98 |
| ANEXO A EL DIAGRAMA DEL FUNCIONAMIENTO DE LA APLICACIÓN LECTORBMU..... | 99 |
| ANEXO B CÓDIGO FUENTE DE LA APLICACIÓN LECTORBMU..... | 101 |
| ANEXO C J2ME ARQUITECTURA SELECCIONADA PARA EL DESARROLLO DE LA APLICACIÓN LECTORBMU.. | 143 |
| ANEXO D DICCIONARIO DE TÉRMINOS..... | 166 |
| ANEXO E. ENCUESTA DE CAMPO EN FES ARAGÓN..... | 174 |
| BIBLIOGRAFÍA..... | 180 |

I. PRESENTACIÓN

El objetivo del presente trabajo de tesis, es exponer la aplicación Lector de Boletín para Móviles Universitarios “LectorBMU”, desarrollada en Java 2 Micro Edition, la cual permite descargar desde un servidor, sin requerir conexión a Internet, diferentes boletines informativos tales como gaceta UNAM, calendario de cursos, calificaciones, libros y cualquier otro documento de carácter informativo de utilidad para la comunidad universitaria.

Los boletines son de texto plano para no disminuir el rendimiento del dispositivo móvil ni saturar la memoria del mismo. La actualización del boletín se da a través del cambio del archivo de texto plano, es decir, la misma aplicación pero con un boletín distinto.

La aplicación podrá ser ejecutada en dispositivos móviles, principalmente dirigida a teléfonos celulares, sin embargo puede ser ejecutada en cualquier dispositivo móvil que ejecute aplicaciones Java y que tenga el puerto para la transferencia de la aplicación desde el servidor al móvil, ejemplificado en este caso por medio de bluetooth y conexión usb al equipo servidor.

Se sustenta la tesis de que la mayoría de los estudiantes de la FES Aragón, independientemente de la carrera y semestre que cursen, poseen un dispositivo móvil como lo es un teléfono celular o PDA (Personal Digital Assistant -Asistente Personal Digitalizado-), con la infraestructura necesaria para ejecutar la aplicación LectorBMU y están interesados en recibir información de carácter académico y/o cultural por este medio.

Se busca fomentar el uso de dispositivos móviles para fines académicos y no sólo de entretenimiento como pueden ser la música o los juegos, llegando de manera directa a todo aquel universitario que desee descargar el boletín y posea un celular o un PDA que cumpla con los requerimientos ya mencionados.

Se pretende con este desarrollo la implementación de un canal de difusión diferente a los tradicionales como lo son carteles y publicaciones impresas, ahorrando de esta forma recursos de impresión.

II. INTRODUCCIÓN

Lector de Boletín para Móviles Universitarios, es la aplicación que pretende fomentar el uso de los dispositivos móviles como son celulares para difundir información de carácter académico, lo único que se requiere es que el estudiante solicite la aplicación y por medio de algún puerto de transferencia se le hace llegar a su celular.

Actualmente, algunos medios de difusión son los carteles o medios impresos colocados en áreas comunes para los estudiantes como pueden ser corredores, pizarrones, entradas de laboratorios, lo cual al ser información necesaria para todos en una fecha específica genera aglomeraciones y los medios están expuestos a ser alterados o arrancados del lugar. Se sugiere, a través de esta aplicación generar un canal alternativo de difusión de información común a los estudiantes que poseen un dispositivo móvil con dos características principales, la primera, contar con puerto bluetooth, y la segunda, que ejecuten aplicaciones java para que descarguen la misma desde un servidor el cual enviará esta bajo petición del estudiante, en el presente se tratan los posibles usos de la aplicación tales como difusión de calificaciones, horarios, gaceta UNAM etc.

Si el medio resulta aceptable a los alumnos en cuanto a rapidez y comodidad paulatinamente se podría suprimir la impresión de ciertas publicaciones y se consolidaría como medio alternativo para la difusión y obtención de las mismas.

Una aplicación para dispositivos móviles implementada en J2ME permitirá el uso de la misma por la mayoría de los estudiantes del campus ya que un celular con tecnología bluetooth y con capacidad de ejecutar aplicaciones java es una pertenencia común hoy en día, tal como se demuestra en la encuesta realizada (véase Anexo E). La marca o el sistema operativo del dispositivo no son de relevancia pues la aplicación se ejecuta sobre la Máquina Virtual Java del dispositivo.

La aplicación está dirigida especialmente para su uso en celulares y ha sido desarrollada para dos perfiles: MIDP 1.0 y MIDP 2.0; está compilada en la herramienta J2ME Wireless Toolkit 1.0.4_01 y 2.0 respectivamente. El código empleado básicamente es el mismo por lo que sólo hubo que compilar en ambas herramientas, teniendo como ventaja para la primera una utilidad que convierte la aplicación en una aplicación PRC -extensión nativa de dispositivos de mano PDA- con lo cual también sería posible implementarla sin problema alguno, ya que lo único que se requiere es la Kilo Virtual Machine (KVM) Máquina Virtual Java para dispositivos palm OS que se cargaría desde el servidor y después se implementaría el LectorBMU para su uso

A continuación se describen los apartados que conforman este trabajo los cuales como característica principal son modulares y pueden ser estudiados individualmente.

APLICACIONES MÓVILES

En este capítulo se describe cómo han evolucionando las aplicaciones móviles y su adaptación a las necesidades de los usuarios; el entorno a nivel mundial en cuanto al desarrollo de aplicaciones móviles y su situación en el segundo semestre del 2006; el cómo ha ido en incremento el desarrollo de aplicaciones para dispositivos móviles tales como agendas de mano y smartphones; se explica también el cómo los smartphones están ganando terreno frente a las PDA's y se da un vistazo al mercado mexicano considerado como el de mayor potencial en América Latina. Así mismo, se expone un ejemplo de lo que es una empresa dedicada al desarrollo de aplicaciones móviles como lo es MSP, S. A. de C. V.

LENGUAJES DE PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES

En este capítulo llevamos a cabo una revisión de diferentes lenguajes de programación disponibles para el desarrollo de aplicaciones móviles, se muestran sus características, ventajas y desventajas.

Se muestra el Proceso de desarrollo de aplicaciones en J2ME y de ejecución en emuladores o dispositivos reales. Asimismo se da una revisión del desarrollo de aplicaciones para Palm OS, Visual Studio .NET y Microsoft .NET Compact Framework como entornos de programación más comunes.

TECNOLOGÍAS y PLATAFORMAS

Se describen a fondo las diferentes tecnologías inalámbricas en las cuales se basa la transferencia de aplicaciones móviles tales como son Bluetooth, Redes inalámbricas Wi-Fi, infrarrojo IrDA.

Se presenta la definición de PDA y smartphone, características de ambos y revisión de diferentes equipos PDA al igual que Smartphones. En la descripción se incluye la revisión de los equipos que fueron empleados en la implementación de la aplicación.

Se mencionan las plataformas mejor posicionadas en el mercado mundial Palm OS en dos versiones: Palm OS 6.1 y Palm OS 5.0. También se trata la plataforma Symbian, principal plataforma de Smartphones y la plataforma Windows mobile como opción que ofrece Microsoft para dispositivos móviles.

DESARROLLO DE APLICACIÓN PARA PDA Y DISPOSITIVOS MÓVILES COMPATIBLES

Es este el capítulo medular de este trabajo, aquí se explica el entorno de programación ocupado, Java 2 Micro Edition J2ME.

Se describen también, el funcionamiento de la aplicación, el desarrollo de la misma, las herramientas ocupadas durante el proceso, la ejecución de la aplicación en emuladores y después en los dispositivos físicamente, las pruebas realizadas y el uso de la aplicación.

CONCLUSIONES

En este apartado se exponen los posibles usos de la aplicación en un entorno real y a partir de la misma la idea final de este trabajo, generar un medio de difusión alternativo para la comunidad universitaria.

ANEXOS

Anexo A El diagrama del funcionamiento de la aplicación LectorBMU.

Anexo B Código fuente de la aplicación LectorBMU.

Anexo C. J2ME arquitectura seleccionada para el desarrollo de la aplicación LectorBMU.

Anexo D Diccionario de términos.

Anexo E. Encuesta de campo en FES Aragón en la cual se confirma lo común que es el uso de dispositivos compatibles con la aplicación.

III. APLICACIONES MÓVILES

A) MARCO TEÓRICO

A lo largo de la historia, las telecomunicaciones acompañaron el desarrollo, tanto en la actividad económica como en los aspectos sociales y culturales. Contribuyeron al crecimiento de la producción y a la mejora de las condiciones de vida de la población, adecuándose a las concepciones políticas de cada época.

Una aplicación móvil permite el acceso desde un dispositivo móvil a toda la información ya sea corporativa, o de uso personal que quiera movilizar, tal como agendas de clientes, catálogos de productos y precios, lista de teléfonos de empleados, amigos o cualquier información. En el siglo XXI la fuerza de trabajo móvil requiere acceso a información corporativa tanto en la oficina como fuera de ella.

Con el creciente desarrollo de dispositivos móviles y tecnologías de redes inalámbricas como Wireless LAN, GSM (Global System for Mobile communications - Sistema Global para las Comunicaciones Móviles), GPRS (General Packet Radio Service – Servicio general de Radio por Paquetes), Bluetooth e IrDA (Infrared Data Association – Asociación de Datos por Infrarrojo), ha sido perfeccionada la movilidad de los usuarios, de manera que ya no están obligados a realizar el trabajo en un lugar fijo.

Como ejemplo de esto, un inspector de calidad podría ingresar los resultados de su inspección en su dispositivo móvil, y desde el mismo lugar donde realiza éste trabajo, transmitir la información hacia la base de datos de su empresa, vía una interfase de red inalámbrica para el almacenamiento permanente o análisis inmediato. A través de la gestión de clientes dispondría en su dispositivo móvil de toda la información sobre los clientes, datos de contacto, pedidos realizados, condiciones comerciales del cliente. Con una aplicación de catálogo de productos podría acceder a la base de datos de productos de la empresa: directorio y buscador de productos, tarifas por cliente, características de productos. Una aplicación de pedidos, realizará los pedidos a la central desde cualquier lugar y en cualquier momento.

Funcionaría sobre la base de datos de productos o bien permitiría la introducción manual de líneas de pedido. Podría consultar facturas por cliente o generar nuevos cargos y facturas directamente, sobre el dispositivo móvil enlazada en tiempo real con su software de gestión.

Una aplicación de documentos/Informes tendría acceso a los distintos documentos de la empresa asociados a productos o documentos de interés general; a noticias y mensajes enviados entre el personal de la empresa y los empleados que usan los dispositivos móviles a través de una Intranet de noticias y mensajes.

Una aplicación que desde un teléfono celular nos permita iniciar sesión en un servidor de la Universidad permitiría consultar información en línea como horarios de cursos, historia académica, calificaciones de exámenes finales, etc.

Como podemos ver son numerosas las áreas en las que una aplicación móvil puede simplificar el trabajo y optimizar los resultados.

A continuación se describen algunas de las ventajas de las aplicaciones móviles:

- Permiten estar a la vanguardia en comunicaciones y sistemas de información.
- La actualización de la información, puede ser en línea. Al tener consulta en línea, se tiene información actualizada.
- Facilidad de navegación.
- Permiten a las compañías mejorar y ahorrar en la recolección de datos basada en papel.
- Hay mejora en la calidad de servicio a clientes.
- Se aumenta la productividad de los empleados.
- Se pueden tomar decisiones con mayor rapidez.
- Minimiza los costos de comunicación para el acceso remoto a información.

B) SITUACION ACTUAL

En el siglo XXI, el mundo de TI (Tecnologías de la Información) se ha estremecido con grandes cambios. Han surgido nuevos estándares de comunicación, compras millonarias entre empresas, lanzamientos de procesadores lógico-matemáticos más potentes, baterías con distintos materiales que tienen mayor duración de vida y menor tiempo de carga así como pantallas más nítidas y de menor consumo, etcétera. Todo lo anterior está mezclándose para dar paso a la nueva era del cómputo, donde las PC, los dispositivos móviles, las redes personales, las redes de telecomunicaciones y la industria del entretenimiento se fusionan y dan paso a nuevos conceptos y mejores formas de acercar la tecnología a los usuarios.

Para el segundo semestre del año 2006 ya existe una amplia oferta de dispositivos móviles en el mercado y los fabricantes distribuyen nuevos modelos constantemente. Como resultado de esta gran variedad de dispositivos, se deben afrontar los problemas cada vez que se desarrolla una aplicación Web móvil.

Las soluciones móviles están mostrando sus beneficios para la gestión de las empresas en la mejora de la productividad, en la creación de nuevos servicios.

De la misma forma empresas de renombre están concientes de estos cambios y hoy en día ofrecen cursos sobre desarrollo de aplicaciones móviles, como es el caso de la universidad virtual SunCampus. Un número creciente de profesionales técnicos se está especializando en la programación sobre la plataforma J2ME SunCampus¹, la innovadora plataforma virtual de formación de Sun Microsystems, ha experimentado un importante crecimiento en la demanda, por parte de los profesionales, de cursos relacionados con el desarrollo para dispositivos móviles sobre la plataforma Java 2 Micro Edition (J2ME).

El auge de los dispositivos móviles con tecnología Java está propiciando un gran protagonismo de la rama de formación denominada Programador de Dispositivos Móviles con J2ME. Se trata de un conjunto de cursos que se desarrollan, a modo de una auténtica carrera universitaria y que están orientados a que el alumno obtenga una formación profesional especializada en la creación de aplicaciones para dispositivos móviles mediante el empleo de las APIs (Application Programming Interfaces – Interfaces de Aplicaciones Programables) de J2ME, así como en el aprendizaje del lenguaje WML (Wireless Markup Language- Lenguaje de Marcas Inalámbrico).

La plataforma J2ME está viviendo un auge indiscutible. No en vano, ya se han superado los 250 millones de terminales móviles con Java en todo el mundo.

En todo el mundo, ya existen unos 300 modelos distintos de teléfonos móviles con Java integrado, fabricados por firmas como LG, Motorola, Nokia, Samsung, Sharp,

¹ Universidad virtual de Sun Microsystems <http://es.sun.com/services/training/suncampus/>

Siemens o Sony Ericsson. Se calcula que el 60 por ciento de todos los teléfonos móviles que se comercializan en el mundo ya llevan la tecnología Java integrada.

Tomando en cuenta los inicios del cómputo, estudiando un poco las gráficas de ventas de dispositivos como Laptops, PCs, Handhelds, dispositivos inteligentes, teléfonos celulares, smartphones así como reproductores multimedia y de entretenimiento, al mismo tiempo, ubicando nuevos lanzamientos de productos, descubrimientos y desarrollos de nuevas tecnologías que en poco tiempo son comercializables, se puede apreciar cómo evoluciona o hacia adonde se dirige la evolución del cómputo y sus vertientes.

Procesadores más potentes

En el segundo semestre de 2006 Intel, el mayor productor de procesadores, presentó a la Industria tres nuevos procesadores, Merom, Conroe (Core 2 Duo) y Woodcrest (Xeon).² Todos estos procesadores fueron creados con la intención de ofrecer mayor rendimiento, menor tiempo de respuesta, optimizar el uso de recursos y ofrecer una plataforma que soporte los nuevos estándares de comunicación del Siglo XXI, las redes de datos de alta velocidad.

Las aplicaciones que estos procesadores pueden tener son tantas, como el número de transistores que incluyen estos procesadores (millones). Al usuario lo beneficia, al darle nuevas y mejores maneras de realizar sus labores, ofrecerle productividad y brindarle entretenimiento.

Intel esta enfocando sus esfuerzos en el diseño de micro arquitectura, tecnologías de intercomunicación e innovando en el uso del Silicio, esto significa mayor conveniencia, protección y fiabilidad en el uso para los consumidores, donde sea, cuando sea y en cualquier dispositivo sea teléfono, PDA o Portátil.

Baterías más durables

Sin embargo, estos procesadores, requieren energía, y aunque su uso, ha sido optimizado en los nuevos procesadores, éstos, en conjunto con las pantallas, displays y demás factores que demandan energía, siguen limitando el uso y aplicaciones que puedan darse a los dispositivos.

Es por eso, que hoy por hoy, empresas como Toshiba están desarrollando baterías con distintos polímeros y sustancias que brindan una notable mejoría al rendimiento que brinda la limitada cantidad de espacio y peso que pueden agregar a los dispositivos.

Actualmente, ya se ha logrado que baterías sean recargadas en menos de un minuto, baterías que funcionan con hidrógeno.

² Información obtenida del sitio de Intel <http://www.intel.com/espanol/>

Mejores y más nítidas pantallas.

Grandes corporaciones asiáticas, europeas y americanas, disputan el trono de la empresa que pueda poner en el mercado global, la primera pantalla flexible en un dispositivo portátil. Ha habido grandes avances en esta materia y, Samsung es quien ha mostrado el mejor prototipo de pantalla digital flexible. Pasarán algunos meses, incluso años, antes que estas pantallas flexibles tengan grados de calidad similares a los que hoy en día, podemos encontrar en los dispositivos de cómputo móvil que vemos en todo lugar.

Bioteología y HAN

Por otro lado, los avances en bioteología y redes de comunicaciones que usan al cuerpo humano como medio de transmisión. Por citar un ejemplo de proyectos que involucren este tema, tenemos a Red Tacton³ empresa que esta enteramente dedicada al desarrollo de sistemas basados en la comunicación a través del cuerpo humano.

Se cuenta con tecnología que se acerca a este concepto. Estamos hablando de la transmisión de sonido a través de lo que se conoce como “Bone Induction” la cual transmite las ondas de sonido a través de nuestro sistema óseo, mediante la inducción de estas ondas vía los tejidos óseos.

Se cuenta también con la tecnología necesaria para realizar implantes de pequeños aparatos electrónicos, capaces de establecer comunicación e interacción con el huésped y los sistemas de detección. Tal es el caso de la tecnología RFID que ACCSYS 3000, empresa 100% mexicana, dedicada a la innovación el campo de los dispositivos móviles y soluciones verticales, ha creado en productos como Baby Safe entre otros productos.

Telecomunicaciones y Redes

Las redes de última generación, conocidas como redes 3G son capaces de transmitir millones y millones de bytes por segundo. Esto, dicho en términos coloquiales, significa que las redes de transmisión por aire que cubren cientos o miles de Km., cada vez transportarán mayor cantidad de información y a mayores velocidades, la cantidad de aplicaciones que esto implica, es inmensa. Las redes EDGE (Enhanced Data for Global Evolution - Datos Mejorados para Evolución Global), UMTS (Universal Mobile Telecommunications System – Sistema Universal de Telecomunicaciones Móviles), EVDO (Evolution Data Optimized – Evolución de Datos Optimizados) y W-CDMA (Wideband-Code Division Multiple Access - Acceso múltiple de banda ancha por división de código) son, por excelencia, las

³ Nueva tecnología de Red de Área Humana que usa la superficie del cuerpo de forma segura, como ruta de transmisión de alta velocidad <http://www.redtacton.com/>

redes de telecomunicaciones de finales del siglo XX y principios del siglo XXI redes de telecomunicaciones 100 veces más rápidas, eficientes en el consumo de ancho de banda. El proveedor Japonés DoCoMo, ha estado probando redes de 4G, capaces de transmitir a 100gbps (gigabits por segundo).

Para establecer un punto de comparación, la red EVDO (red a la que Iusacell esta migrando su infraestructura) puede transmitir a 2.4Mbps eso es .24gbps, ni siquiera el 1% de la rapidez de transmisión que estas redes de cuarta generación traerán al mercado masivo y corporativo en el caso de los proveedores de telefonía celular.

Si bien es cierto que el Internet es el medio de comunicaciones por excelencia, los servicios de datos que ofrecen los proveedores a nivel mundial aún son caros para el usuario promedio, existen iniciativas por parte de algunos proveedores por abaratar sus costos de transmisión de datos.

Los humanos, al ser más dependientes de la red de redes que es la Web, llevarán a los proveedores a buscar la forma de acercar a los usuarios más a Internet, tal es el caso de Iusacell en nuestro país que está haciendo el mejor esfuerzo por impulsar la transmisión y uso de servicios de datos. Mucha gente se preguntará acerca del futuro de las redes de telecomunicaciones que soportan los canales de voz “VoIP”. La voz sobre IP es transmitida a través de canales de datos, no de voz. Esto lleva a la conclusión de que la empresa que sepa transmitir datos en el futuro, será la empresa que prevalecerá.

En este mismo tema, podemos ampliar al hablar de las redes personales o PAN (Personal Área Network – Red de Área Personal) que se han puesto “muy de moda” con la salida al mercado de cientos de modelos de manos libres bluetooth. Bluetooth es un protocolo y un estándar de comunicación inalámbrica entre dispositivos en rangos que oscilan entre 5 a 100 metros de distancia. Cuando establecemos contacto entre nuestro PDA, el celular y el manos libres mediante Bluetooth, estamos creando una red personal de comunicación.

Por último se menciona el tema del WiFi, conocido también como “Wireless” debido a los bajos costos que presenta en la transmisión de datos vía inalámbrica, contra los servicios de datos que ofrecen los proveedores del mundo entero.

Durante 2005 y 2006, ha habido una explosión de sitios comerciales, aeropuertos y edificios en el mundo entero que ofrecen conectividad a la Internet mediante Wifi. Sin embargo, sería prácticamente imposible cubrir ciudades enteras con esta tecnología de comunicación inalámbrica de mediano alcance. Es por eso, que ahora contamos con WiMax, el cual funciona de manera parecida al Wifi, solo que los ruteadores o puntos de acceso, cuentan con un rango de transmisión que se mide por Km., no por metros.

Por otra parte, WiMax ofrecerá mejores velocidades de conexión, mayor ancho de banda y, será mucho más accesible al usuario a diferencia de las redes de datos que

actualmente ofrecen los proveedores del mundo. Lamentablemente, las grandes corporaciones que sostienen los mercados de telefonía móvil en el mundo han puesto un alto a las Instituciones que regulan el espectro de radiofrecuencia, impidiéndole al WiMax, abrirse paso como la solución idónea para los asuntos de conectividad a Internet, los usuarios no tendrán que preocuparse por servicios de Voz, ya que contarían con VoIP, servicios como Skype o Google Talk, tan de moda en los PCs de hoy en día, suplirían las necesidades de voz de los usuarios.

Smartphones y proveedores de Telefonía móvil

La tendencia del mercado es la integración de varias tecnologías y aparatos, fusionados en uno solo que realice múltiples funciones. El problema es que al ser una fusión de varias tecnologías, tiende a presentar deficiencias e inestabilidad. Los smartphones están pasando por un proceso de adaptación, donde el usuario y el proveedor deben comprender que la robustez y complejidad del sistema operativo y del hardware, tienen cierto impacto en la calidad de servicio y funcionalidad para el usuario.

La tecnología móvil es un tema emergente en el mundo. Más allá de despliegues tecnológicos y promesas de comunicación, la gran interrogante que existe entre los usuarios corporativos es cómo la movilidad puede servir a su negocio, y preocupaciones del día a día.

Más aún cuando las necesidades del usuario son muchísimas: acceder a todo tipo de información en los sistemas back office, y a todas las áreas de la empresa cuando se está fuera; actualizar la información corporativa y del cliente o compartir importante y delicada información con otros miembros del staff móvil. Sin embargo, ya existen casos concretos de optimización de negocios gracias a la integración de movilidad, destacando las industrias del transporte y la salud como innovadoras en la materia. Como botón de muestra, tenemos a FEDEX y UPS. Ambas empresas a nivel mundial ya tienen incorporada una red wireless al corazón de sus negocios. A través de PDAs sus mensajeros se comunican con la oficina central para enviar y recibir información en línea. Así, gracias al dispositivo, y su integración a los sistemas de la empresa, ellos han pasado a ser la versión remota de la oficina.

Al igual que el ejemplo anterior, en el 2007 la complejidad y posibilidades de las soluciones móviles aumentarán considerablemente, por lo que empresas y usuarios necesitan en forma urgente entender cuán valioso será para ellos esta nueva ola tecnológica. Ahora bien, si de utilidades se trata, las soluciones móviles tienen un enorme potencial para la fuerza de ventas. Imaginemos por un momento: si los ejecutivos de cuenta de una empresa, que deben visitar en terreno a sus clientes, tuvieran como parte de su “kit de ventas” un aparato handheld, con acceso directo a sistemas ERP (Enterprise Resources Planning - planificación de recursos empresariales) y otras herramientas de automatización de ventas, se ganaría un

tiempo valioso. El dispositivo permitiría la interacción inmediata con la oficina accediendo a información clave en el momento para responder a los requerimientos de su cliente. ¿El beneficio? Más rapidez en la negociación y cierre de las ventas, ya que no sería necesario hacer nuevas visitas, y además posibilitaría hacer múltiples contactos a la vez. En resumen, se maximizarían las oportunidades de ventas y mejoraría la calidad y rapidez del servicio.

En definitiva, lo que está buscando la industria de la TI (Tecnologías de la Información) al potenciar el uso de los equipos móviles se mejoran los procesos de trabajo, y la calidad de los servicios prestados por los clientes finales.

¿Qué pueden esperar, entonces, los usuarios corporativos de la movilidad? Acceso a aplicaciones de la oficina desde cualquier lugar y a cualquier hora. Comunicación rápida y simple. Menos costos y negocios más flexibles.

Pese a esto el mercado de PDAs desciende, mientras los “smart phones” no paran de crecer. El mercado mundial de handhelds y PDAs sigue perdiendo terreno frente a los “móviles inteligentes”.

La tendencia es que los dispositivos en los que convergen varias tecnologías, como los smartphones estén en condiciones de suplantar las funciones PIM (personal information manager – administrador de información personal), una característica clásica de las handhelds y PDAs.

C) SITUACION ACTUAL EN MÉXICO

El trabajo constante y la adopción clara de los cambios es quizá el mayor impulso de los desarrolladores de aplicaciones para dispositivos móviles.

Aún con los distintos matices con que pueda verse al mercado mexicano, éste es considerado como el de mayor potencial en América Latina. Pues la evolución del cómputo móvil ha sido constante, y al comparar las primeras plataformas con las actuales, hay dos mundos completamente diferentes.

El cómputo móvil ofrece enormes posibilidades de soluciones, principalmente para el sector empresarial que requiere incrementar su productividad y se auxilia mediante el uso de este tipo de herramientas para conseguir mejores resultados.

El mercado

En México el tipo de mercado al que van dirigidos es muy atractivo, y tienen ya una estrategia para llegar al mercado de la PyME (Pequeña y Mediana Empresa), que consiste en el desarrollo de una aplicación horizontal de muy fácil manejo.

El número de desarrolladores locales que existen en México no es el suficiente para la demanda, aunque esté la iniciativa de gobierno: Prosoft para exportar servicios, pero una de las limitantes más importantes es que aunque se está haciendo algo, todavía falta mucho por hacer.

Tendencias móviles

La oportunidad en México para el desarrollo de aplicaciones móviles es muy grande, pero uno de los puntos clave es que como no son nuevas, ya es posible que el usuario tenga acceso a dispositivos estándares de diversos modelos y precios tan bajos desde \$2,000 pesos hasta \$10,000 o más.

También ya es posible hacer aplicaciones cubriendo otro punto clave que es la integración de éstas con ERP, CRM, etc.

Empresas Ejemplo

Syncrologix

Es en el año 2001 cuando la compañía detecta una buena posibilidad en el desarrollo de aplicaciones móviles. El mercado iba madurando, siempre acompañado de empresas de desarrollo, además de compañías como PalmOne y los proveedores para toda la parte de datos, sobre todo en servicios de banda ancha para transmitir (datos) en forma remota.

Inició operaciones en la ciudad de México y en Buenos Aires de forma simultánea. Dedicados 100% al desarrollo de aplicaciones móviles para mercados corporativos verticales, con diferentes tipos de aplicaciones para diferentes segmentos.

La idea de Syncrologix⁴ es seguir trabajando con el fabricante PalmOne, y brindarle su apoyo en el conocimiento de los dispositivos.

Syncrologix pretende ofrecer una solución integra en el desarrollo de una estrategia de movilidad en las empresas. Su oferta se basa en cuatro pilares:

- Conocer mucho sobre los dispositivos para dar una oferta de hardware, a través de alianzas estratégicas, con PalmOne por ejemplo, y así acercar la mejor herramienta que el cliente requiera.
- El software, es decir, la compañía cuenta con un amplio equipo de desarrollo de aplicaciones sobre diferentes plataformas.
- La tercera se refiere a conectividad, pues son desarrolladores de punta a punta de todas las soluciones que brindan, entendiéndose por conectividad toda la parte de sincronización.
- Finalmente, los servicios: consultoría de desarrollo de sistemas, soporte técnico, capacitación, además de modelos comerciales.

Algunos clientes de Syncrologix son:

- Gillete
- Liconsa
- Laboratorios Aventis
- Chrysler

MSP, S. A. de C. V.

Inicia operaciones en 1992 dedicándose desde sus principios al ensamble y venta de equipos de cómputo, en 1995 comienza a desarrollar los primeros sistemas administrativos para PyME; debido a la necesidad que existe entre los clientes de contar con sistemas diseñados a la medida.

En 1996, inicia el desarrollo de aplicaciones para dispositivos móviles y desktop utilizando Internet como medio de comunicación tanto alámbrico como inalámbrico.

Actualmente, MSP⁵ desarrolla sistemas integrales en hardware y software con las herramientas de programación más avanzadas en el mundo móvil, que hacen

⁴ **Syncrologix** es una empresa de servicios que ofreciendo Mobile Consulting, desarrollo, implementación e integración de soluciones corporativas en dispositivos handheld (PDA's y teléfonos inteligentes), ayuda a las empresas a desarrollar su Estrategia Móvil <http://www.syncrologix.com/>

posible que las organizaciones sean más eficientes y productivas, simplificando y automatizando los procesos de una manera sencilla, incrementando sus capacidades tecnológicas y de cobertura en el mercado, ello gracias a un equipo de Ingenieros certificados en las principales plataformas de desarrollo móvil.

Algunos de sus productos son:

- ***Sales Force Automation*** (Sistema Móvil de Automatización de Fuerza de Ventas)

Herramienta indispensable en el proceso de ventas en campo, ya que además de tener un sistema que ayudará en el levantamiento de un pedido, también puede establecer recordatorios de futuras llamadas, controlar sus gastos, consultar sus niveles de venta contra sus metas esperadas y comunicarse con su gente de ventas. La movilidad que ofrece SFA a su equipo de ventas, puede llevarlo un paso delante de su competencia y sobre todo ofrecer a sus clientes un servicio que supere sus expectativas.

- ***Quality Control System*** (Control de Calidad Móvil)

Sistema diseñado para responder a los requerimientos de calidad de las organizaciones, permitiendo el control de calidad en plantas de producción, puntos de ventas, y el envío de dichos errores al instante a través de una fotografía.

⁵ MSP es una empresa mexicana líder en el desarrollo de aplicaciones para dispositivos móviles PDA y Teléfonos celulares en cualquier plataforma operativa (Palm OS, Pocket PC , SmartPhone, Symbian, RIM (BlackBerry) , Windows CE .Net , etc.).
<http://www.mspsmovil.net/>

D) NECESIDADES Y TENDENCIAS

Los dispositivos móviles están avanzando a una velocidad sorprendente. El hecho es que los dispositivos que ahora están en etapa de diseño llevarán a los servicios móviles del mañana a lugares que sólo podemos soñar hoy.

Dispositivos móviles discretos que registran y transmiten sonidos e imágenes instantáneamente, controlan un amplio espectro de dispositivos desde autos hasta refrigeradoras de champagne, y pueden ser descartados y reemplazados sin dudarlos. 007 ya no es algo especial – y en un futuro predecible, el tipo de magia de Bond costará sólo unos pocos dólares en una tienda de la esquina.

Esto sucederá debido a que la evolución de los dispositivos manuales está ingresando a una nueva fase a medida que los teléfonos celulares se transforman en dispositivos multimedia de alta capacidad que procesan interfaces intuitivas de usuario. Queda por ver si estas capacidades adicionales de los dispositivos garantizarán altos ingresos o una mejor experiencia para el cliente. Estos pequeños administrículos continuarán sorprendiendo, tanto por la reducción de su tamaño como por su funcionalidad exponencialmente mayor.

En los próximos cuatro años, la agrupación de nueva tecnología en dispositivos manuales será cada vez mayor, convirtiendo a los teléfonos celulares tradicionales en dispositivos audiovisuales de avanzada, procesados en redes rápidas, inalámbricas y de alta capacidad.

Se predice que el precio minorista de dispositivos de low-end se reducirá a la mitad en 2008, y que para el 2009, existirá un porcentaje de mercado casi igual en las ventas de estos dispositivos entre los modelos de tercera generación (3G) y los de 2.5G. Todos los teléfonos 3G tendrán capacidades de doble modalidad (2.5G y 3G).

El software móvil avanzará igualmente rápido. Se verá una aceleración mayor en la adopción de aplicaciones Java, especialmente en el campo de rápido crecimiento de los juegos.

Otras plataformas tales como BREW (Binary Runtime Environment for Wireless- Ambiente de Ejecución Binario para inalámbricos) open source de Qualcomm en Estados Unidos ya construyeron una sólida posición. BREW, por ejemplo, es usada por Verizon y algunos participantes regionales de Code-Division Multiple Access (CDMA) en EE.UU. y es ofrecida comercialmente en más de 100 dispositivos por fabricantes líderes y Original Device Manufacturers (ODMs).

La demanda de los clientes también impulsará el diseño de hardware. Los usuarios quieren dispositivos bonitos que soporten e-mails y aplicaciones más complejas de información personal (Personal Information Management – PIM), web-browsing y juegos. La industria responderá creando dispositivos funcionales pero modernos,

con pantallas más grandes, permitiendo un rápido y continuo cambio entre aplicaciones.

Los dispositivos también incorporarán sensores que dispararán aplicaciones industriales y personales, desde las más obligatorias hasta las críticas para la vida. Algunos actuarán como monitores del cuidado cardíaco, otras mejorarán y expandirán la experiencia de los juegos, siendo sensibles al movimiento.

Este panorama móvil emergente abrirá una amplia gama de posibilidades comerciales a las empresas en el espacio móvil y más allá de éste. Del lado del usuario, todas las industrias (desde la de servicios financieros, distribución y ventas minoristas hasta la de salud, seguridad, incluso los departamentos de policía que ya están utilizando la transmisión de fotos de delitos por telefonía celular) necesitan adaptarse a estos dispositivos, y ver cómo potenciarlos en su propio contexto único.

El primer paso para lograrlo es modelar cómo las nuevas capacidades de video, ubicación y demás impactarán la forma de vida de sus clientes y de sus propios empleados, y cómo esto abrirá nuevas formas de hacerlo.

Del lado de la industria, los fabricantes de transmisores y dispositivos deben colaborar mutuamente para permitir que los terceros incorporen las aplicaciones adecuadas rápida y fácilmente a la plataforma subyacente. Es crucial que la industria asegure que toda el área no se convierta en una pesadilla para la atención al cliente, con frustraciones de ambos lados y gente tan confundida por la tecnología que sea incapaz de recordar cómo realizar una llamada telefónica básica.

A medida que se migran las nuevas capacidades, surgen nuevas áreas de oportunidad:

- El video está a punto de despegar – las aplicaciones handset incluyen ahora videoconferencias, video clips y difusión de TV digital. El uso de estas tecnologías despegará una vez que se superen los desafíos técnicos (el tamaño de la pantalla, la claridad, la lenta inicialización de llamadas y la corta vida útil de las baterías). Tecnologías tales como Organic LED mejorarán las pantallas drásticamente, proporcionarán una mayor velocidad de visualización, mayor brillo y nitidez con una menor fuente de alimentación. En el sector de los medios, la calidad de la experiencia visual en dispositivos móviles será clave.
- Una mejor imagen – a fines del 2004, las cámaras-teléfonos de 2 mega pixels declararon la guerra a las cámaras digitales. Esta tendencia continuará con el surgimiento de aplicaciones móviles para cámaras, tales como lectores de código de barras, permitiendo a los clientes compartir detalles personales en las tarjetas e-business, comprar entradas, comparar precios y solicitar ofertas. También habrá webcams móviles y aplicaciones de video vigilancia.

- De cara a la música – iTunes de Apple marcó una tendencia con su modelo de comercialización y entrega de música digital. Los handsets equipados con reproductores de MP3, radios FM y grabadores de voz son cada vez más comunes, y los servicios musicales online permiten a los usuarios escuchar canciones en sus teléfonos digitales. Nokia está avanzando mucho más allá con sus capacidades de “Radio Visual”, permitiendo a las estaciones de radio agregar una dimensión interactiva a las transmisiones.
- Juegos: grandes desafíos y grandes recompensas. Este mercado plantea importantes desafíos técnicos y de contenido, y la compleja cadena de valor de crear, empaquetar y vender juegos móviles requiere nuevos roles para cada participante.
- Los sistemas de pago están de moda los medios de pago móviles finalmente están ganando terreno entre los consumidores, empresas minoristas, bancos, proveedores y redes de pago. El desafío más grande será actualizar los sistemas de pagos para manejar los medios de pago móviles.
- Apertura del mercado industrial. Los dispositivos móviles componen la red física de las nuevas aplicaciones industriales utilizando telemetría de sensores, GPS y tecnologías de Identificación de Radio Frecuencia (RFID). Las aplicaciones incluirán video vigilancia, lectura de medidores, monitoreo ambiental, transporte, gestión de inventario y stock, y muchos más.

El mensaje general es claro: las continuas tendencias en el diseño y capacidades de los dispositivos móviles no ocurren en forma aislada. Por el contrario, el rápido avance de los dispositivos móviles produce oportunidades importantes para cada segmento de la industria móvil. Y los operadores estarán a la vanguardia de este cambio, a medida que buscan obtener una porción más grande de las billeteras de los clientes, agrupando y proporcionando las aplicaciones y los servicios móviles que éstos quieran.

IV. LENGUAJES DE PROGRAMACIÓN PARA DISPOSITIVOS MÓVILES.

A) EN QUE DESARROLLAR.

Para tomar la decisión correcta, tanto para elegir el dispositivo correcto, como la tecnología a usar, es necesario conocer las necesidades presentes y futuras, las capacidades del dispositivo que se elija, el conocimiento actual o la facilidad de adquirir este conocimiento sobre el software de desarrollo, entre otras.

¿Qué se va a desarrollar?

Básicamente, esto dependerá de dos grandes factores: infraestructura y costos.

Infraestructura, haciendo referencia a la capacidad que se tenga como empresa y en el medio. Para citar un ejemplo, sería relativamente sencillo desarrollar un sitio web para teléfonos móviles, con acceso a un sistema, donde el usuario podría ingresar al sistema desde cualquier punto y en cualquier momento, usando Internet.

Pero para ello, sería necesario que la empresa contara con varios proveedores, para listar algunos, una línea de internet dedicada, Ip's públicas, sistemas de seguridad, etc. Los cuales, indudablemente, involucran el segundo factor, costos.

La segunda alternativa, es la de desarrollar directamente en el celular o el PDA. Con esta opción la necesidad de infraestructura puede verse reducida, ya que la información, cambios y demás, pueden ser guardados directamente en el dispositivo, y estar sincronizados con el servidor de datos, o, como segunda alternativa, al contar con una red interna, este dispositivo puede trabajar conectado a la misma, teniendo la posibilidad de realizar transacciones en línea usando la posible infraestructura ya implementada dentro de la empresa.

En resumen, la decisión de lo que se va a desarrollar, dependerá de las necesidades de cada empresa, e, indudablemente, de otros factores además de los mencionados.

Se parte del conocimiento que se pueda tener u obtener en corto plazo sobre el dispositivo en el cual se desarrollará el software, tomando como ejemplo Palm contra Pocket PC.

Palm VS Pocket PC

Por un lado, Palm¹, nos brinda una plataforma de desarrollo basada en su propio lenguaje, el cual tiene fundamentos en C y C++, el software para desarrollar para estos dispositivos es totalmente gratis, pero para poder compilar una aplicación desarrollada para éste, es necesario desembolsar una gran suma, ya que, los compiladores se venden por separado y son vendidos por terceros, y por último, si

¹ PalmOne, Inc., fabricante de hardware para PDAs basadas en Palm OS*PalmSource, Inc., desarrollador de Palm OS *Palm OS, el sistema operativo para la Palm PDA*Palm Pilots, PDAs antiguas <http://www.palm.com/>

no se posee un equipo Palm, es necesario obtener el BIOS del equipo para el cual se desea desarrollar, para ello, se deben llevar a cabo una serie de tramites entre Palm y la empresa a desarrollar y así obtener una copia de dicho BIOS. Aunque existen emuladores que simulan ser un equipo palm con todas sus funciones ejecutándose en la PC.

En contraste, Pocket PC² con Windows CE incluido, se puede programar con dos plataformas de desarrollo: .Net, y Microsoft Embedded Visual Tools, herramientas totalmente gratuitas que nos permiten programar en lenguajes como Visual Basic o Visual C.

En el caso de que queramos utilizar tecnologías de vanguardia, también podemos desarrollar con .Net, el cual acepta, no solo los lenguajes tradicionales de Microsoft, sino también lenguajes alternativos como Java, Borland C, entre otros, y por supuesto, podemos acceder a toda la potencia de .Net, pero desde el Pocket PC.

Teléfonos celulares.

En el caso de teléfonos celulares, tenemos dos alternativas:

- 1) Crear un sitio Web y
- 2) Desarrollar aplicaciones para el mismo celular.

Una vez más, se pueden plantear las dificultades de aprender un nuevo lenguaje.

Si ponemos en contraste, .Net contra otras tecnologías, veremos que, dependiendo de la marca del celular, éste posee, para el desarrollo de aplicaciones, un lenguaje específico, basado en C y C++ que cambia de celular a celular.

En el caso del desarrollo de sitios web para celulares, es necesario aprender otro lenguaje, comúnmente llamado WML, que es una mezcla de XML, con tags específicos, y el lenguaje de servidor que queramos con el código.

Por el contrario, .Net, tiene la facilidad de que, utilizando la misma forma de trabajo que el desarrollo de una pagina web normal, se pueden crear sitios webs para teléfonos celulares.

² PocketPC es un ordenador de bolsillo, también llamado PDA (Personal Digital Assistant). Se trata de un pequeño ordenador, diseñado para ocupar el mínimo espacio y ser fácilmente transportable que ejecuta el sistema operativo Windows CE de Microsoft, el cual le proporciona capacidades similares a los PCs de escritorio. <http://www.microsoft.com/mexico/>

B) HERRAMIENTAS OPENSOURCE PARA DESARROLLO DE APLICACIONES PALM

El Proceso de Desarrollo de Aplicaciones para Palm

Desarrollar aplicaciones para PDAs, independientemente de la marca que los construya, conlleva a tomar en cuenta algunas cuestiones, empezando porque la programación puede desarrollarse en C o C++ (o varios otros lenguajes para los que existen compiladores que generan código que se puede ejecutar en PDAs).

En el caso específico de la programación para dispositivos Palm, se deben generar archivos de "recursos" que describen la interfaz con el usuario y su funcionamiento. Normalmente, estos archivos se obtienen usando un programa independiente del compilador o del ambiente de desarrollo integrado (IDE, por sus siglas en inglés: Integrated Development Environment).

Los archivos de recursos contienen la jerarquía de objetos de la interfaz gráfica, empezando por la 'forma' (similar a la 'ventana' cuando se desarrolla para Windows) a la que se le van añadiendo otros objetos como barras de menú (a las que a su vez se les añaden menús), botones, campos de texto, tablas, etc. En este archivo se deben incluir también los mapas de bits (bitmaps) que se van a usar en la aplicación.

El resto del proceso de desarrollo consiste en codificar la respuesta a los eventos que le pueden ocurrir a los diferentes objetos de la forma.

Una vez que se ha generado un programa, se procede a compilarlo y depurarlo, corrigiendo los eventuales errores y repitiendo esta última parte del proceso hasta que se obtiene un programa libre de errores.

En esta etapa, es poco conveniente probar los programas en dispositivos reales, por que existen errores que pueden incluso, provocar la pérdida total de la información almacenada en la Palm. Por esto último, se recomienda probar la aplicación en un programa, llamado Palm OS Emulator, que se ejecuta en la PC y simula totalmente la funcionalidad de una Palm.

Cuando se han eliminado todos los errores, y el programa se ejecuta sin problemas en el emulador, esta listo para instalarse en un dispositivo real.

Herramientas OpenSource

Existen herramientas OpenSource para realizar cada una de las tareas descritas anteriormente: desde generar las pantallas, crear y editar los programas hasta compilarlos y probarlos.

Ambientes Integrados de Desarrollo

Básicamente un Ambiente Integrado de Desarrollo es una colección de herramientas de desarrollo de software que distribuyen en forma conjunta y que

cooperan estrechamente entre si para facilitar este proceso. Para desarrollar aplicaciones Palm, se pueden usar varios de los IDEs OpenSource disponibles, como Xwpe, abreviatura de XWindows Program Environment, que fue uno de los primeros IDEs disponibles en Linux. La interfaz de usuario se asemeja a la de la familia de productos de Borland (Turbo C y Turbo Pascal) para MS-DOS. Dentro de Xwpe es posible editar, compilar y depurar programas.

Otra opción puede ser Anjuta, que es otro IDE para C y C++ escrito para GNOME y que ofrece una gran cantidad de facilidades para el programador. Anjuta incluye administración de proyectos, un depurador interactivo y un poderoso editor de texto.

PRC-Tools

El ambiente de desarrollo conocido como PRC-Tools incluye diferentes herramientas: Una variante de GCC, el compilador GNU de C, que compila código C/C++ y genera ejecutables para los procesadores Motorola 68000. Build-PRC, que es la herramienta que construye archivos PRC (programas de Palm) a partir de código y los recursos. GDB, que es un depurador de código y PilRC que es un compilador de recursos.

Además de estas herramientas, se requiere un Kit de desarrollo de Software proveído por Palm Inc., con archivos de encabezado (.h) que hacen referencia a las diferentes librerías.

Copilot

El Emulador de Palm fue desarrollado originalmente en Unix, dentro del esquema OpenSource bajo el nombre de CoPilot, y ahora es llamado POSE (abreviatura de Palm OS Emulator).

POSE es un programa gratuito con versiones para Windows, Macintosh y Unix/Linux; es importante señalar que, cuando el programador original vendió una versión del código a Palm, esta compañía tomo la decisión de seguir desarrollándolo, manteniéndolo (al menos en Unix) como un programa OpenSource. Pero, además, aprovechando las posibilidades que el esquema OpenSource ofrece, a partir de la última versión del código fuente disponible se ha desarrollado otra versión del emulador, llamada QPilot, que es totalmente independiente de la de Palm.

Entre las ventajas que las herramientas OpenSource ofrecen se pueden mencionar dos muy importantes: La primera de ellas es que, al menos hasta este momento, todas las herramientas son gratuitas. Y aunque no hay garantía de que esta condición permanezca así por siempre, el esquema OpenSource permitiría que (al igual que ocurrió con CoPilot) nuevos equipos de desarrolladores creen versiones de las herramientas que sigan siendo gratuitas.

La segunda ventaja, inherente también al esquema OpenSource, es que las herramientas están en constante desarrollo. Cada nueva versión es probada por cientos y hasta miles de usuarios, y los errores son depurados con relativa rapidez.

La desventaja es que en los últimos tiempos, Palm ha realizado cambios importantes en el Palm OS, que de la versión 3.5 saltó a la 4 y luego a la 5 en un lapso menor a un año, además de haber hecho cambios importantes en el hardware, cambiando los procesadores 68000 de Motorola por procesadores AMD. Y aunque aseguran que se vigilará con atención la compatibilidad hacia atrás (esto es, que los programas desarrollados para las nuevas versiones del Palm OS se puedan ejecutar también en versiones anteriores), es poco probable que esto ocurra, ya que aún ahora, hay programas que se desarrollaron para una cierta versión de hardware Palm y que no pueden ejecutarse en versiones anteriores. Todo esto dificulta la labor de los programadores, tanto de las herramientas OpenSource, como de las propias aplicaciones Palm.

Informática móvil

Se pueden crear aplicaciones para Palm usando como plataforma de desarrollo los sistemas operativos Windows, Mac OS, UNIX e, incluso, el propio Palm OS. No estamos, por tanto, forzados a utilizar Windows, como sí ocurría con las eMbedded Visual Tools. También se puede elegir el lenguaje que más se adapte a nuestros conocimientos y experiencia, desde C y C++ hasta BASIC en diferentes versiones, pasando por Java y lenguajes menos habituales como Forth.

Las aplicaciones ejecutables para Palm OS se alojan en archivos con extensión PRC. Estos archivos pueden ser completamente autónomos o bien necesitar de alguna librería o módulo de apoyo para su funcionamiento. Todo el proceso de desarrollo y depuración suele efectuarse mediante un emulador, instalando la aplicación en el dispositivo real al final. Además de aplicaciones propiamente dichas, también podemos desarrollar conductos (conduits) para facilitar la conversión de formatos y sincronización entre un Palm y otra plataforma, como puede ser Windows o Mac OS.

Aplicaciones y conductos

Los dispositivos Palm cuentan con un sistema operativo, el conocido Palm OS, que permite ejecutar aplicaciones hechas a medida para dichos dispositivos. Por lo general, además de ejecutar esas aplicaciones en el Palm también es necesaria una sincronización de datos entre el sistema de escritorio y el dispositivo móvil. Es habitual, por tanto, que en el sistema de escritorio se instale una aplicación, el HotSync Manager, precisamente para hacer posible ese intercambio de información entre ambos sistemas.

Las posibilidades de comunicación o integración entre sistemas de sobremesa y Palm pueden extenderse mediante el desarrollo de conductos. Un conducto es un componente que, conjuntamente con el mencionado HotSync Manager, es capaz de

transferir información en uno u otro sentido, o bien de realizar una conversión de formatos de los archivos móviles al equipo de sobremesa o viceversa.

Los conductos, por tanto, son programas que se ejecutan en el sistema de sobremesa, mientras que las aplicaciones estándar para Palm se transfieren y ejecutan en el dispositivo móvil. Ésa, aparte de la propia finalidad del código, es la diferencia entre una aplicación estándar Palm y un conducto Palm.

C) J2ME JAVA 2 MICRO EDITION

La plataforma J2ME es una familia de especificaciones que definen varias versiones minimizadas de la plataforma Java 2; estas versiones minimizadas pueden ser usadas para programar en dispositivos electrónicos; desde teléfonos celulares, en PDAs, hasta en tarjetas inteligentes. Estos dispositivos presentan en común que no disponen de abundante memoria ni mucha potencia en el procesamiento, ni tampoco necesitan de todo el soporte que brinda el J2SE, (la plataforma estándar de Java usada en sistemas de escritorio y servidor).

Desafortunadamente, el mercado para el tipo de dispositivos que el nuevo lenguaje fue creado no se desarrolló tanto como Sun Microsystems esperaba, y al final ningún dispositivo basado en Oak fue vendido a los clientes. Sin embargo, al mismo tiempo, el inicio del conocimiento público de Internet produjo un mercado para el software de navegación para Internet (los navegadores Web). En respuesta a esto, Sun Microsystems renombró el lenguaje de programación Oak a Java y lo usó para desarrollar un navegador multiplataforma llamado HotJava. También le dio la licencia de Java a Netscape, quienes lo incorporaron en su navegador que por ese entonces era el más popular en el mercado, luego fueron incorporados los Java applets.

Las capacidades multiplataforma del lenguaje de programación Java y su potencia como plataforma de desarrollo para aplicaciones que podían ser escritas una vez y ejecutadas en diversos sistemas Windows y Unix, ha despertado el interés de usuarios finales, porque vieron en ella una manera de reducir los costos del desarrollo de software.

Con el objetivo de conocer las necesidades de los experimentados desarrolladores en Windows y Motif/X-Windows para crear aplicaciones para usuarios finales sofisticados acostumbrados a usar interfaces ricas, Sun Microsystems rápidamente expandió el alcance y tamaño de la plataforma Java. Esta plataforma extendida incluyó un conjunto más complejo de librerías de interfaces de usuario que aquellos que usaran para construir applets, además con un conjunto de características de computación distribuida y seguridad mejorada.

Irónicamente, mientras Sun Microsystems estaba desarrollando Java para Internet y para la programación comercial, la demanda empezó a crecer en los dispositivos pequeños e incluso en tarjetas inteligentes, retornando Java a sus raíces.

Sun Microsystems respondió a esta demanda creando varias plataformas Java con funcionalidades reducidas, cada una hecha a la medida de un segmento vertical y específico del mercado.

En conclusión, J2ME es la versión de Java orientada a los dispositivos móviles. Debido a que los dispositivos móviles tienen una potencia de cálculo baja e interfaces de usuario pobres, es necesaria una versión específica de Java destinada a estos dispositivos, ya que el resto de versiones de Java, J2SE o J2EE, no encajan dentro de este esquema. J2ME es por tanto, una versión “reducida” de J2SE.

Los programadores que estén acostumbrados a utilizar Java para desarrollar en Windows, UNIX, Linux o cualquier otra plataforma, también pueden usar este lenguaje a la hora de crear aplicaciones para dispositivos con el sistema Palm OS. Como promete Java, se puede desarrollar en cualquier sistema y ejecutar en otro, por lo que se puede usar Windows, Mac OS o Linux, por ejemplo, para crear programas. Posteriormente habría que transferirlo al PDA.

El sistema operativo Palm OS, no obstante, no puede ejecutar directamente el byte-code Java. Es necesario, por tanto, instalar una JVM (Java Virtual Machine) en el Palm, como en cualquier otro ordenador. La edición estándar de la plataforma Java, sin embargo, es demasiado grande como para poder ser llevada a un dispositivo que se caracteriza, entre otros aspectos, por una cantidad limitada de recursos. Por ello Sun diseñó para esos dispositivos la J2ME, una edición en miniatura de su JVM. Algunas herramientas de desarrollo comerciales, como IBM Visual Age y Borland JBuilder, han preparado sus respectivos entornos para facilitar la creación de aplicaciones Java que se ejecuten en la J2ME.³



Fig 4.1 Máquina virtual de Java ejecutando un programa en el emulador de Palm.

³ Amplia información de la tecnología J2ME en el Anexo C de este trabajo

Paquetes de desarrollo

Con el fin de crear aplicaciones y conductos para Palm se requieren, para comenzar, los correspondientes paquetes de desarrollo. Se componen, básicamente, de archivos de cabecera, librerías con funciones y algunas utilidades básicas. Estos paquetes son los recursos a utilizar desde una herramienta de desarrollo que no viene incluida en ellos.

Si se pretenden crear programas ejecutables para PDA, se necesita el Palm OS Software Development Kit. Este paquete de desarrollo es gratuito, bastando la aceptación de un contrato de uso para poder obtenerlo.

El SDK (Software Development Kit- Kit de software de Desarrollo) está disponible para Windows, Mac OS y Linux. Es decir, se pueden usar cualquiera de estos sistemas operativos como una plataforma anfitrión para el desarrollo de aplicaciones que, posteriormente, se transferirían y ejecutarían sobre Palm OS. Existen versiones específicas del paquete de desarrollo para ciertas herramientas, como CodeWarrior y PRC-Tools.

Herramientas de desarrollo

El SDK y el CDK (Component Developer's Kit – Kit de Componentes para desarrolladores) de Palm OS no servirán de mucho por sí solos, ya que son librerías, archivos de cabecera y añadidos que deberán utilizarse en alguna herramienta de desarrollo que, realmente, será en donde se introduzca el código, se diseñe la interfaz de usuario -si es que existe- y, finalmente, se compile para obtener el correspondiente ejecutable o conducto. A diferencia de Microsoft, que ofrece un completo entorno multilinguaje para el desarrollo de aplicaciones, Palm tan sólo facilita el SDK, dejando el campo de las herramientas a terceras empresas.

Si el proyecto es desarrollar algún tipo de conducto para Palm, aparte del CDK se requiere también una herramienta como puede ser Visual C++, Visual Basic o Visual Café for Java, en el caso de Windows, o bien CodeWarrior en el Mac. Una de las versiones del CDK, concretamente la dirigida a los usuarios de Visual Basic, hace posible la creación de conductos como componentes COM por lo que, en la práctica, sería posible utilizarlo para crear este tipo de programas con otras herramientas, como Borland Delphi o Visual J++.

Seguramente la herramienta más conocida para desarrollar aplicaciones ejecutables en Palm OS sea CodeWarrior, de la empresa Metrowerks. Existen versiones de CodeWarrior para Windows 9x y NT, así como para Mac OS. Con él pueden crearse aplicaciones para todas las versiones del Palm OS y los dos procesadores que incorporan estos dispositivos. El lenguaje de programación puede ser C o C++, según preferencias del programador.

Otra alternativa, a la hora de crear aplicaciones para Palm, es utilizar el lenguaje Java. Existen herramientas como el VisualAge Micro Edition de IBM, compuesta de un IDE, que se ejecutaría sobre Windows o Linux, y una máquina virtual Java para Palm OS, concretamente la conocida como J2ME (Java 2 Micro Edition).

Finalmente, y sin tratar de hacer una enumeración exhaustiva, para aquellos que prefieren productos de código abierto existen las PRC-Tools. Se trata de un conjunto de herramientas que, conjuntamente con una versión modificada del SDK, hacen posible el desarrollo de aplicaciones mediante el conocido compilador GCC, ya sea usando C o C++.

Si hay un entorno pensado específicamente para la construcción de aplicaciones para Palm es, sin duda alguna, CodeWarrior.

CodeWarrior ofrece un IDE típico para la gestión de proyectos y edición de código, por una parte, y un entorno visual para la construcción de interfaces de usuario, por otra.



Fig. 4.2 Una aplicación creada con CodeWarrior Lite ejecutándose en POSE

Entorno de prueba y depuración

Las aplicaciones desarrolladas con cualquiera de las herramientas que se hayan elegido serán, finalmente, transferidas y ejecutadas en un dispositivo móvil. Éste, sin embargo, no siempre es el objetivo más adecuado cuando lo que se pretende es probar la aplicación y depurar su funcionamiento, el mejor sistema consiste en utilizar un emulador para esta tarea.

El emulador oficial de Palm se llama POSE, existen versiones para Windows, Mac OS y UNIX, algo que le diferencia notablemente respecto al emulador y herramientas para Pocket PC.

POSE realmente emula el hardware de un dispositivo Palm, pero para funcionar necesita una ROM con el sistema operativo Palm OS, igual que ocurre con los dispositivos físicos. Sin dicha ROM el emulador no tiene utilidad alguna: no podremos utilizarlo para probar ni depurar nuestras aplicaciones.

Hay dos alternativas para conseguir una ROM que permita poner POSE en marcha: recuperarla directamente desde un dispositivo Palm, si es que se cuenta con uno, o bien obtenerla a través del programa Alliance de Palm Computing. Éste último tiene ciertas ventajas respecto a la primera opción y es que, además de la ROM, habrá un importante número de recursos adicionales al alcance. Hay que añadir también que la ROM de un dispositivo corriente no está pensada para la depuración, mientras que la obtenida a través del citado programa sí lo está.

Junto al emulador se facilita un programa, llamado ROM Transfer.prc, que se debe transferir al dispositivo del que se vaya a tomar la ROM. Acto seguido se inicia el emulador y se elige la opción correspondiente para descargar ROM, mientras se pone en marcha en el Palm el anterior programa. En unos segundos se obtendrá en el equipo un archivo Palm.ROM con la ROM del dispositivo.

En la Figura 4.3 se puede ver a POSE en marcha, funcionando en Windows, emulando un Palm V. Utilizando el ratón se simula el puntero.



Fig 4.3 El emulador POSE cuenta con diferentes skins que permiten adaptarlo a las preferencias del usuario.

AppForge Toolkit for Visual Basic

La mayor comunidad de programadores en la actualidad es la de aquellos que utilizan Microsoft Visual Basic, una herramienta que, si bien está pensada para crear aplicaciones para Windows, gracias a su extensibilidad puede utilizarse con otros fines. Mediante AppForge, por ejemplo, se extiende Visual Basic para hacer posible la creación de programas para Palm, manteniendo la simplicidad propia de Visual Basic.

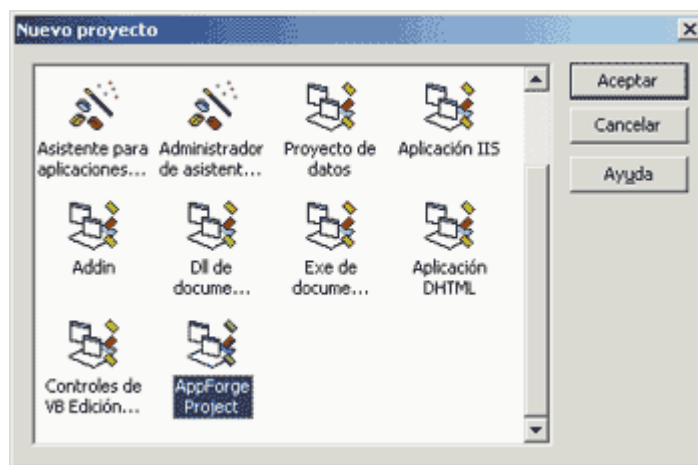


Figura 4.4 AppForge añade un nuevo tipo de proyecto a los que ya incluye por defecto Visual Basic.

Tras la instalación, al iniciar un nuevo proyecto aparece una nueva llamada AppForge Project, destacada en la figura 4.4. La aplicación generada es, aparentemente, como cualquier otra de Visual Basic, si bien el formulario muestra un tamaño y color distintos a los habituales. En la paleta de componentes, además, aparece un grupo de elementos específicos para la creación de aplicaciones para Palm. En la figura 4.5 se puede ver la citada paleta y un formulario en el cual se ha incluido una cuadrícula de datos y un botón.

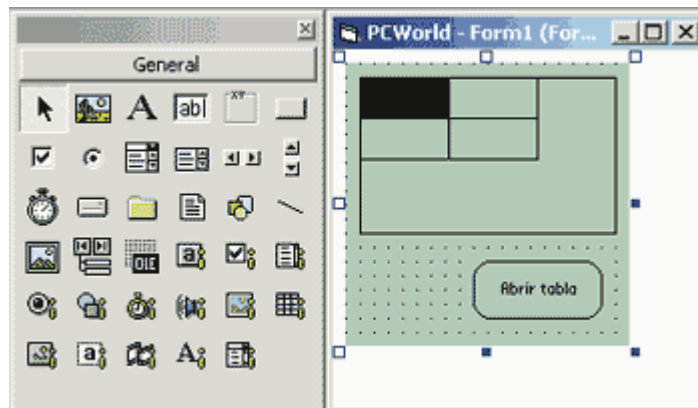


Figura 4.5 El formulario aparece con un color distinto y en la paleta aparece un nuevo grupo de componentes para aplicaciones Palm.

Creada la aplicación, siguiendo la misma metodología que se usaría para cualquier programa Visual Basic, se utilizarían las opciones del nuevo menú AppForge para compilar y transferir el ejecutable al dispositivo móvil. Las aplicaciones creadas con AppForge precisan un módulo llamado Booster, que puede instalarse desde el mismo menú añadido a Visual Basic.

Además de extender el entorno de Visual Basic para hacer posible la creación de aplicaciones para Palm, AppForge también cuenta con un conversor para bases de datos y otro para gráficos. El primero es capaz de generar bases de datos Palm a partir de bases de datos Access, mientras que el segundo se usaría para convertir cualquier gráfico de PC a Palm.

Gracias a todas estas herramientas crear aplicaciones para Palm es extremadamente fácil para aquellos que utilizan habitualmente Visual Basic.

La plataforma Palm cuenta con un importante número de herramientas y utilidades para el desarrollo de aplicaciones. La mayoría de estas herramientas pueden utilizarse en Windows, Mac OS y Linux, generando programas ejecutables para Palm OS. Es posible elegir entre múltiples lenguajes, lo cual abre las puertas a un mayor número de potenciales programadores.

Gracias a esta amplia oferta de herramientas, no es de extrañar que la plataforma Palm cuente actualmente con muchas más aplicaciones que cualquiera de sus competidoras, lo cual influye, indudablemente, en la elección por parte de la mayoría de los usuarios.

D) VISUAL STUDIO.NET Y MICROSOFT.NET COMPACT FRAMEWORK

Microsoft ha desarrollado .NET Compact Framework con un claro objetivo: la creación de aplicaciones. Aplicaciones capaces de mostrar, recopilar, procesar y enviar datos; el tipo de aplicación que justifica que los usuarios decidan llevar encima un dispositivo. Aunque normalmente estas aplicaciones tienen una interfaz, no siempre es necesario. Los datos con los que estas aplicaciones trabajan pueden ser locales, remotos o tal vez una combinación de ambos.

.NET Compact Framework simplifica el desarrollo de aplicaciones para dispositivos inteligentes. Esto incluye a los dispositivos Pocket PC, Pocket PC 2002, Pocket PC Phone Edition y otros dispositivos que ejecuten Windows CE.NET 4.1 o posterior.

Se necesita Visual Studio .NET 2003 para crear aplicaciones destinadas a .NET Compact Framework. Se pueden crear aplicaciones utilizando Visual C# .NET, Visual Basic .NET o ambos.

.NET Compact Framework tiene dos componentes principales: el tiempo de ejecución en lenguaje común y la biblioteca de clases de .NET Compact Framework.

El tiempo de ejecución es la base de .NET Compact Framework, ya que se encarga de administrar el código en el momento de la ejecución, proporcionando servicios esenciales como la administración de la memoria y de los subprocesos, al mismo tiempo que garantiza la seguridad y la precisión. Si el código está destinado al tiempo de ejecución se denomina código administrado, si no lo está, como ocurre con eMbedded Visual C++, se denomina código no administrado o nativo.

La biblioteca de clases de .NET Compact Framework es una colección de clases reutilizables que se pueden utilizar para desarrollar aplicaciones de manera fácil y rápida. Este marco se ha diseñado pensando en la portabilidad, tanto para plataformas Microsoft como de otros fabricantes. Las técnicas de codificación y las aplicaciones creadas en un Pocket PC se pueden ejecutar en otras plataformas, como un teléfono móvil o un PDA de otro fabricante, si se ha creado una versión de .NET Compact Framework para dicha plataforma.

Clases de XML y de datos

.NET Compact Framework incluye un conjunto de clases que permiten incorporar con facilidad datos (ya sea de un origen de datos relacional o no), entre los que se incluye el contenido XML, en sus aplicaciones móviles. Estas clases se definen para los espacios de nombres System.Data y System.Xml. La implementación de clases de XML y de datos en .NET Compact Framework es un subconjunto de la que se encuentra en .NET Framework.

Servicios Web

.NET Framework presta una atención especial a los servicios Web. En el espacio de nombres System.Web de .NET Compact Framework, hay una versión a escala reducida de las posibilidades y funcionalidades que ofrece el correspondiente espacio de nombres de .NET Framework. La característica más destacable es que se pueden crear clientes de servicios Web pero no se pueden alojar servicios Web en .NET Compact Framework.

Estos clientes de servicios Web XML pueden ser sincrónicos o asincrónicos. Se pueden crear fácilmente clientes de servicios Web XML destinados a .NET Compact Framework. El IDE de Visual Studio .NET hace automáticamente la mayor parte del trabajo.

Compatibilidad con GDI

.NET Compact Framework proporciona compatibilidad con los elementos básicos de dibujo de GDI incluidos mapas de bits, pinceles, fuentes, iconos y plumas mediante el espacio de nombres System.Drawing.

Compatibilidad con IrDA

Algunos dispositivos con Windows CE, por ejemplo, Pocket PC y Pocket PC, incluyen la posibilidad de comunicarse mediante infrarrojos (IR). Para poder aprovechar esta posibilidad, .NET Compact Framework incluye clases que le permiten aprovechar la comunicación mediante infrarrojos desde la aplicación. Estas clases son parte del espacio de nombres System.Net.IrDA. Puede utilizar infrarrojos para comunicarse con equipos Pocket PC, impresoras y otros dispositivos compatibles con infrarrojos.

Compatibilidad con Bluetooth

.NET Compact Framework no incorpora compatibilidad nativa con Bluetooth. Puede obtener acceso a la mayoría de implementaciones de Bluetooth en equipos Pocket PC de otros fabricantes mediante las comunicaciones del puerto serie o mediante una API del proveedor.

.NET Compact Framework ofrece a los desarrolladores una poderosa herramienta para crear aplicaciones móviles robustas. Le permite aprovechar las ventajas de los conocimientos de .NET de los que ya dispone para crear aplicaciones destinadas a los dispositivos Pocket PC y Windows CE .NET.

V. TECNOLOGÍAS Y PLATAFORMAS

A) TECNOLOGÍAS INALÁMBRICAS RELACIONADAS CON APLICACIONES MÓVILES.

1.-BLUETOOTH

Es la norma que define un Standard global de comunicación inalámbrica, que posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia. Los principales objetivos que se pretende conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre nuestros equipos personales.

La tecnología Bluetooth¹ comprende hardware, software y requerimientos de interoperabilidad, por lo que para su desarrollo ha sido necesaria la participación de los principales fabricantes de los sectores de las telecomunicaciones y la informática, tales como: Ericsson, Nokia, Toshiba, IBM, Intel y otros. Posteriormente se han ido incorporando muchas más compañías, y se prevé que lo hagan también empresas de sectores tan variados como: automatización industrial, maquinaria, ocio y entretenimiento, fabricantes de juguetes, electrodomésticos, etc., con lo que en poco tiempo se nos presentará un panorama de total conectividad de nuestros aparatos tanto en casa como en el trabajo.

¹ El nombre procede del rey danés y noruego Harald Blátand cuya traducción en inglés sería Harold Bluetooth, conocido por unificar las tribus noruegas, suecas y danesas.

De la misma manera, Bluetooth intenta unir diferentes tecnologías como los ordenadores, el teléfono móvil y el resto de periféricos. El símbolo de Bluetooth es la unión de las runas nórdicas H y B. <http://es.wikipedia.org/>

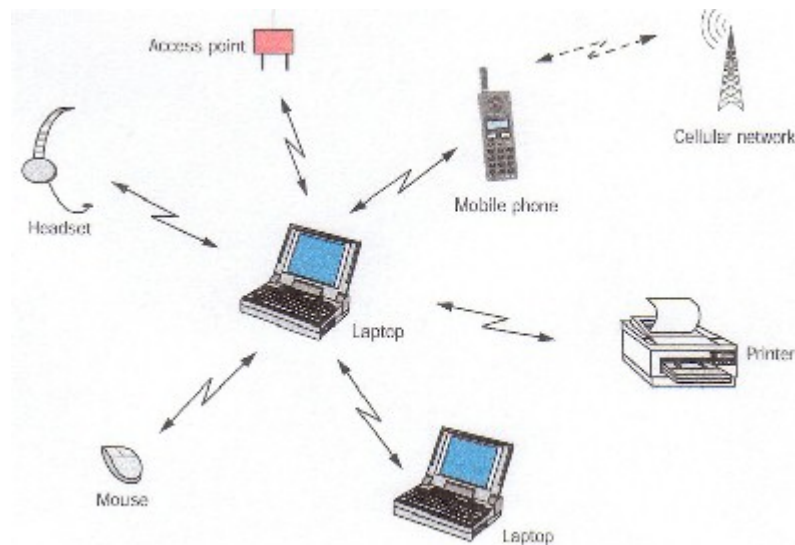


Fig. 5.1 Comunicaciones entre equipos móviles y fijos

Antecedentes

En 1994 Ericsson inició un estudio para investigar la viabilidad de una interfase vía radio, de bajo costo y bajo consumo, para la interconexión entre teléfonos móviles y otros accesorios con la intención de eliminar cables entre aparatos. El estudio partía de un largo proyecto que investigaba sobre unos multi-comunicadores conectados a una red celular, hasta que se llegó a un enlace de radio de corto alcance, llamado MC link. Conforme éste proyecto avanzaba se fue viendo claro que éste tipo de enlace podía ser utilizado ampliamente en un gran número de aplicaciones, ya que tenía como principal virtud el que se basaba en un chip de radio relativamente económico.

El SIG (Special Interest Group – Grupo de Interés Especial)

A comienzos de 1997, según avanzaba el proyecto MC link, Ericsson fue despertando el interés de otros fabricantes de equipos portátiles. Enseguida se vio claramente que para que el sistema tuviera éxito, un gran número de equipos deberían estar equipados con ésta tecnología. Esto fue lo que originó a principios de 1998, la creación de un grupo de interés especial (SIG), formado por 5 promotores que fueron: Ericsson, Nokia, IBM, Toshiba e Intel. La idea era lograr un conjunto adecuado de áreas de negocio, dos líderes del mercado de las telecomunicaciones,

dos líderes del mercado de los PCS portátiles y un líder de la fabricación de chips. El propósito principal del consorcio fue y es, el establecer un Standard para la interfase aérea junto con su software de control, con el fin de asegurar la interoperabilidad de los equipos entre los diversos fabricantes.

La interfase aérea Bluetooth

El primer objetivo para los productos Bluetooth de primera generación eran los entornos de la gente de negocios que viaja frecuentemente. Por lo que se debería pensar en integrar el chip de radio Bluetooth en equipos como: PCS portátiles, teléfonos móviles, PDA's y auriculares. Esto originaba una serie de cuestiones previas que deberían solucionarse tales como:

- El sistema debería operar en todo el mundo.
- El emisor de radio deberá consumir poca energía, ya que debe integrarse en equipos alimentados por baterías.
- La conexión deberá soportar voz y datos, y por lo tanto aplicaciones multimedia.

Banda de frecuencia libre

Para poder operar en todo el mundo es necesaria una banda de frecuencia abierta a cualquier sistema de radio independientemente del lugar del planeta donde nos encontremos. Sólo la banda ISM (médico-científica internacional) de 2,45 Ghz cumple con éste requisito, con rangos que van de los 2.400 Mhz a los 2.500 Mhz, y solo con algunas restricciones en países como Francia, España y Japón.

Salto de frecuencia

Debido a que la banda ISM está abierta a cualquiera, el sistema de radio Bluetooth deberá estar preparado para evitar las múltiples interferencias que se pudieran producir. Éstas pueden ser evitadas utilizando un sistema que busque una parte no utilizada del espectro o un sistema de salto de frecuencia. En los sistemas de radio Bluetooth se suele utilizar el método de salto de frecuencia debido a que ésta tecnología puede ser integrada en equipos de baja potencia y bajo coste. Éste sistema divide la banda de frecuencia en varios canales de salto, donde, los transceptores, durante la conexión van cambiando de uno a otro canal de salto de manera pseudo aleatoria. Con esto se consigue que el ancho de banda instantáneo sea muy pequeño y también una propagación efectiva sobre el total de ancho de banda. En conclusión, con el sistema FH (Salto de frecuencia), se pueden conseguir transceptores de banda estrecha con una gran inmunidad a las interferencias.

Definición de canal

Como se ha comentado, Bluetooth utiliza un sistema FH/TDD (salto de frecuencia/división de tiempo duplex), en el que el canal queda dividido en intervalos de 625 μ s, llamados slots, donde cada salto de frecuencia es ocupado por un slot. Esto da lugar a una frecuencia de salto de 1600 veces por segundo, en la

que un paquete de datos ocupa ocupar un slot para la emisión y otro para la recepción y que pueden ser usados alternativamente, dando lugar a un esquema de tipo TDD.

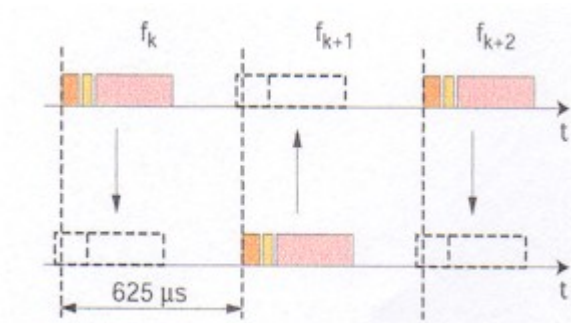


Fig. 5.2 Salto de frecuencia/división de tiempo duplex

Dos o más unidades Bluetooth pueden compartir el mismo canal dentro de una piconet, donde una unidad actúa como maestra, controlando el tráfico de datos en la piconet que se genera entre las demás unidades, donde estas actúan como esclavas, enviando y recibiendo señales hacia el maestro. El salto de frecuencia del canal está determinado por la secuencia de la señal, es decir, el orden en que llegan los saltos y por la fase de ésta secuencia. En Bluetooth, la secuencia queda fijada por la identidad de la unidad maestra de la piconet (un código único para cada equipo), y por su frecuencia de reloj. Por lo que, para que una unidad esclava pueda sincronizarse con una unidad maestra, ésta primera debe añadir un ajuste a su propio reloj nativo y así poder compartir la misma portadora de salto.

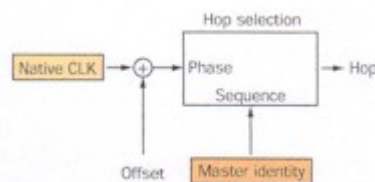


Fig. 5.3 Unidad maestra de la piconet y su frecuencia de reloj.

En países donde la banda está abierta a 80 canales o más, espaciados todos ellos a 1 Mhz se han definido 79 saltos de portadora, y en aquellos donde la banda es más estrecha se han definido 23 saltos.

Definición de paquete

La información que se intercambia entre dos unidades Bluetooth se realiza mediante un conjunto de slots que forman un paquete de datos. Cada paquete comienza con un código de acceso de 72 bits, que se deriva de la identidad maestra, seguido de un paquete de datos de cabecera de 54 bits. Éste contiene importante información de control, como tres bits de acceso de dirección, tipo de paquete, bits de control de flujo, bits para la retransmisión automática de la pregunta, y chequeo de errores de campos de cabeza. Finalmente, el paquete que contiene la información, que puede seguir al de cabeza, tiene una longitud de 0 a 2745 bits. En cualquier caso, cada paquete que se intercambia en el canal está precedido por el código de acceso.

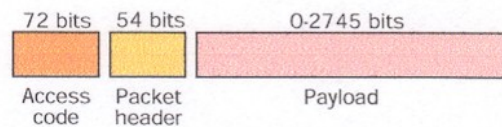


Fig. 5.4 Paquete de datos

Los receptores de la piconet comparan las señales que reciben con el código de acceso, si éstas no coinciden, el paquete recibido no es considerado como válido en el canal y el resto de su contenido es ignorado.

Definición de enlace físico

En la especificación Bluetooth se han definido dos tipos de enlace que permitan soportar incluso aplicaciones multimedia:

- Enlace de sincronización de conexión orientada (SCO)
- Enlace asíncrono de baja conexión (ACL)

Los enlaces SCO soportan conexiones asimétricas, punto a punto, usadas normalmente en conexiones de voz, estos enlaces están definidos en el canal, reservándose dos slots consecutivos (envío y retorno) en intervalos fijos.

Los enlaces ACL soportan conmutaciones punto a punto simétricas o asimétricas, típicamente usadas en la transmisión de datos.

Un conjunto de paquetes se han definido para cada tipo de enlace físico:

Para los enlaces SCO, existen tres tipos de slot simple, cada uno con una portadora a una velocidad de 64 kbit/s. La transmisión de voz se realiza sin ningún mecanismo de protección, pero si el intervalo de las señales en el enlace SCO disminuye, se puede seleccionar una velocidad de corrección de envío de 1/3 o 2/3.

Para los enlaces ACL, se han definido el slot-1, slot-3, slot-5. Cualquiera de los datos pueden ser enviados protegidos o sin proteger con una velocidad de corrección de 2/3. La máxima velocidad de envío es de 721 kbit/s en una dirección y 57.6 kbit/s en la otra.

Inmunidad a las interferencias

Como se mencionó anteriormente Bluetooth opera en una banda de frecuencia que está sujeta a considerables interferencias, por lo que el sistema ha sido optimizado para evitar éstas interferencias. En este caso La técnica de salto de frecuencia es aplicada a una alta velocidad y una corta longitud de los paquetes (1600 saltos/segundo, para slots-simples). Los paquetes de datos están protegidos por un esquema ARQ (repetición automática de consulta), en el cual los paquetes perdidos son automáticamente retransmitidos, aun así, con este sistema, si un paquete de datos no llegase a su destino, sólo una pequeña parte de la información se perdería. La voz no se retransmite nunca, sin embargo, se utiliza un esquema de codificación muy robusto. Éste esquema, que está basado en una modulación variable de declive delta (CSVD), que sigue la forma de la onda de audio y es muy resistente a los errores de bits. Estos errores son percibidos como ruido de fondo, que se intensifica si los errores aumentan.

Piconets.

Si un equipo se encuentra dentro del radio de cobertura de otro, éstos pueden establecer conexión entre ellos. En principio sólo son necesarias un par de unidades con las mismas características de hardware para establecer un enlace. Dos o más unidades Bluetooth que comparten un mismo canal forman una piconet. Para regular el tráfico en el canal, una de las unidades participantes se convertirá en maestra, pero por definición, la unidad que establece la piconet asume éste papel y todos los demás serán esclavos. Los participantes podrían intercambiar los papeles si una unidad esclava quisiera asumir el papel de maestra. Sin embargo sólo puede haber un maestro en la piconet al mismo tiempo.

Cada unidad de la piconet utiliza su identidad maestra y reloj nativo para seguir en el canal de salto. Cuando se establece la conexión, se añade un ajuste de reloj a la propia frecuencia de reloj nativa de la unidad esclava para poder sincronizarse con el reloj nativo del maestro. El reloj nativo mantiene siempre constante su frecuencia, sin embargo los ajustes producidos por las unidades esclavas para sincronizarse con el maestro, sólo son válidos mientras dura la conexión.

Las unidades maestras controlan en tráfico del canal, por lo que estas tienen la capacidad para reservar slots en los enlaces SCO. Para los enlaces ACL, se utiliza un esquema de sondeo. A una esclava sólo se le permite enviar un slot a un maestro cuando ésta se ha dirigido por su dirección MAC (medio de control de acceso) en el procedimiento de slot maestro-esclavo. Éste tipo de slot implica un sondeo por parte del esclavo, por lo que, en un tráfico normal de paquetes, este es enviado a una urna del esclavo automáticamente. Si la información del esclavo no está disponible, el maestro puede utilizar un paquete de sondeo para sondear al esclavo explícitamente. Los paquetes de sondeo consisten únicamente en uno de acceso y otro de cabecera. Éste esquema de sondeo central elimina las colisiones entre las transmisiones de los esclavos.

Estableciendo conexión.

De un conjunto total de 79 (23) portadoras del salto, un subconjunto de 32(16) portadoras activas han sido definidas. El subconjunto, que es seleccionado pseudo-aleatoriamente, se define por una única identidad.

Acerca de la secuencia de activación de las portadoras, se establece que, cada una de ellas visitará cada salto de portadora una sola vez, con una longitud de la secuencia de 32 (16) saltos. En cada uno de los 2.048 (1.028) saltos, las unidades que se encuentran en modo stand by (en espera) mueven sus saltos de portadora siguiendo la secuencia de las unidades activas. El reloj de la unidad activa siempre determina la secuencia de activación.

Durante la recepción de los intervalos, en los últimos 18 slots o 11,25 MS, las unidades escuchan una simple portadora de salto de activación y correlacionan las señales entrantes con el código de acceso derivado de su propia identidad. Si los triggers son correlativos, esto es, si la mayoría de los bits recibidos coinciden con el código de acceso, la unidad se auto-activa e invoca un procedimiento de ajuste de conexión. Sin embargo si estas señales no coinciden, la unidad vuelve al estado de reposo hasta el siguiente evento activo.

Para establecer la piconet, la unidad maestra debe conocer la identidad del resto de unidades que están en modo stand by en su radio de cobertura. El maestro o aquella unidad que inicia la piconet transmite el código de acceso continuamente en periodos de 10 MS, que son recibidas por el resto de unidades que se encuentran en stand by. El tren de 10 MS. de códigos de acceso de diferentes saltos de portadora, se transmite repetidamente hasta que el receptor responde o bien se excede el tiempo de respuesta.

Cuando una unidad emisora y una receptora seleccionan la misma portadora de salto, la receptora recibe el código de acceso y devuelve una confirmación de recibo de la señal, es entonces cuando la unidad emisora envía un paquete de datos que contiene su identidad y frecuencia de reloj actual. Después de que el receptor acepta éste paquete, ajustará su reloj para seleccionar el canal de salto correcto determinado por emisor. De éste modo se establece una piconet en la que la unidad emisora actúa como maestra y la receptora como esclava. Después de haber

recibido los paquetes de datos con los códigos de acceso, la unidad maestra debe esperar un procedimiento de requerimiento por parte de las esclavas, diferente al proceso de activación, para poder seleccionar una unidad específica con la que comunicarse.

El número máximo de unidades que pueden participar activamente en una simple piconet es de 8, un maestro y siete esclavas, por lo que la dirección MCA del paquete de cabecera que se utiliza para distinguir a cada unidad dentro de la piconet, se limita a tres bits.

Scatternet

Los equipos que comparten un mismo canal sólo pueden utilizar una parte de su capacidad de este. Aunque los canales tienen un ancho de banda de un 1Mhz, cuantos más usuarios se incorporan a la piconet, disminuye la capacidad hasta unos 10 kbit/s más o menos. Teniendo en cuenta que el ancho de banda medio disponible es de unos 80 Mhz en Europa y USA (excepto en España y Francia), éste no puede ser utilizado eficazmente, cuando cada unidad ocupa una parte del mismo canal de salto de 1Mhz. Para poder solucionar éste problema se adoptó una solución de la que nace el concepto de scatternet.

Las unidades que se encuentran en el mismo radio de cobertura pueden establecer potencialmente comunicaciones entre ellas. Sin embargo, sólo aquellas unidades que realmente quieran intercambiar información comparten un mismo canal creando la piconet. Éste hecho permite que se creen varias piconets en áreas de cobertura superpuestas. A un grupo de piconets se le llama scatternet. El rendimiento, en conjunto e individualmente de los usuarios de una scatternet es mayor que el que tiene cada usuario cuando participa en un mismo canal de 1 Mhz. Además, estadísticamente se obtienen ganancias por multiplexión y rechazo de canales salto. Debido a que individualmente cada piconet tiene un salto de frecuencia diferente, diferentes piconets pueden usar simultáneamente diferentes canales de salto.

Se debe tener en cuenta que cuantas más piconets se añaden a la scatternet el rendimiento del sistema FH disminuye poco a poco, habiendo una reducción por término medio del 10%. Sin embargo el rendimiento que finalmente se obtiene de múltiples piconets supera al de una simple piconet.

Comunicación inter-piconet

En un conjunto de varias piconets, éstas seleccionan diferentes saltos de frecuencia y están controladas por diferentes maestros, por lo que si un mismo canal de salto es compartido temporalmente por piconets independientes, los paquetes de datos

podrán ser distinguidos por el código de acceso que les precede, que es único en cada piconet.

La sincronización de varias piconets no está permitida en la banda ISM. Sin embargo, las unidades pueden participar en diferentes piconets en base a un sistema TDM (división de tiempo multiplexado). Esto es, una unidad participa secuencialmente en diferentes piconets, a condición de que ésta este sólo activa en una al mismo tiempo. Una unidad al incorporarse a una nueva piconet debe modificar el offset (ajuste interno) de su reloj para minimizar la deriva entre su reloj nativo y el del, por lo que gracias a éste sistema se puede participar en varias piconets realizando cada vez los ajustes correspondientes una vez conocidos los diferentes parámetros de la piconet. Cuando una unidad abandona una piconet, la esclava informa el maestro actual que ésta no estará disponible por un determinado periodo, que será en el que estará activa en otra piconet. Durante su ausencia, el tráfico en la piconet entre el maestro y otros esclavos continúa igualmente.

De la misma manera que una esclava puede cambiar de una piconet a otra, una maestra también lo puede hacer, con la diferencia de que el tráfico de la piconet se suspende hasta la vuelta de la unidad maestra. La maestra que entra en una nueva piconet, en principio, lo hace como esclava, a no ser que posteriormente ésta solicite actuar como maestra.

Seguridad

Para asegurar la protección de la información se ha definido un nivel básico de encriptación, que se ha incluido en el diseño del clip de radio para proveer de seguridad en equipos que carezcan de capacidad de procesamiento, las principales medidas de seguridad son:

- Una rutina de pregunta-respuesta, para autenticación
- Una corriente cifrada de datos, para encriptación
- Generación de una clave de sesión (que puede ser cambiada durante la conexión)

Tres entidades son utilizadas en los algoritmos de seguridad: la dirección de la unidad Bluetooth, que es una entidad pública; una clave de usuario privada, como una entidad secreta; y un número aleatorio, que es diferente por cada nueva transacción.

Como se ha descrito anteriormente, la dirección Bluetooth se puede obtener a través de un procedimiento de consulta. La clave privada se deriva durante la inicialización y no es revelada posteriormente. El número aleatorio se genera en un proceso pseudo-aleatorio en cada unidad Bluetooth.

2.-REDES INALÁMBRICAS WI-FI

Introducción a las WLAN

Una WLAN es un sistema de comunicaciones de datos que transmite y recibe datos utilizando ondas electromagnéticas, en lugar del par trenzado, coaxial o fibra óptica utilizado en las LAN convencionales, y que proporciona conectividad inalámbrica de igual a igual (peer to peer), dentro de un edificio, de una pequeña área residencial/urbana o de un campus universitario.

En EEUU proliferan estas redes para acceso a Internet, en donde hay más de 4.000 zonas de acceso, y en Europa es previsible que pronto se extiendan.

Las WLAN se encuadran dentro de los estándares desarrollados por el IEEE (Instituto de Ingenieros Eléctricos y Electrónicos)² para redes locales inalámbricas. Otras tecnologías como Hyper LAN apoyada por el ETSI, y el nuevo estándar Home RF para el hogar, también pretenden acercarnos a un mundo sin cables y, en algunos casos, son capaces de operar en conjunción y sin interferirse entre sí. Otro aspecto a destacar es la integración de las WLAN en entornos de redes móviles de 3G (UMTS) para cubrir las zonas de alta concentración de usuarios (los denominados hot spots), como solución de acceso público a la red de comunicaciones móviles.

Como todos los estándares 802 para redes locales del IEEE, en el caso de las WLAN, también se centran en los dos niveles inferiores del modelo OSI, el físico y el de enlace, por lo que es posible correr por encima cualquier protocolo (TCP/IP o cualquier otro) o aplicación, soportando los sistemas operativos de red habituales, lo que supone una gran ventaja para los usuarios que pueden seguir utilizando sus aplicaciones habituales, con independencia del medio empleado, sea por red de cable o por radio.

Otra tecnología de acceso inalámbrico en áreas de pequeña extensión (WPAN/WLAN Personal Área Network) es la denominada Bluetooth, que aunque pueda parecer competencia directa de las WLAN, es más bien complementaria a ella. Bluetooth pretende la eliminación de cables, como por ejemplo todos los que se utilizan para conectar el PC con sus periféricos, o proporcionar un medio de enlace entre dispositivos situados a muy pocos metros, sirviendo también como mando a distancia.

Las WLAN tienen su campo de aplicación específico, igual que Bluetooth, y ambas tecnologías pueden coexistir en un mismo entorno sin interferirse gracias a los métodos de salto de frecuencia que emplean, Sus aplicaciones van en aumento y, conforme su precio se vaya reduciendo, serán más y más los usuarios que las

² IEEE corresponde a las siglas de *The Institute of Electrical and Electronics Engineers*, el **Instituto de Ingenieros Eléctricos y Electrónicos**, una asociación técnico-profesional mundial dedicada a la estandarización, entre otras cosas. Es la mayor asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como ingenieros eléctricos, ingenieros en electrónica, ingenieros en sistemas e ingenieros en telecomunicación

utilicen, por las innegables ventajas que supone su rápida implantación y la libertad de movimientos que permiten.

Redes locales inalámbricas 802.11

El origen de las LAN inalámbricas (WLAN) se remonta a la publicación en 1979 de los resultados de un experimento realizado por ingenieros de IBM en Suiza, consistente en utilizar enlaces infrarrojos para crear una red local en una fábrica. Estos resultados, publicados por el IEEE, pueden considerarse como el punto de partida en la línea evolutiva de esta tecnología.

Las investigaciones siguieron adelante tanto con infrarrojos como con microondas, donde se utilizaba el esquema de espectro expandido (spread spectrum). En mayo de 1985, y tras cuatro años de estudios, la FCC³ (Federal Communications Comisión- Comisión Federal de Comunicaciones), la agencia federal del Gobierno de Estados Unidos encargada de regular y administrar en materia de telecomunicaciones, asignó las bandas ISM (Industrial, Scientific and Medical – Industria Medica y Científica) 902-928 MHz, 2,400-2,4835 GHz, 5,725-5,850 GHz para uso en las redes inalámbricas basadas en Spread Spectrum (SS), con las opciones DS (Direct Sequence) y FH (Frequency Hopping). La técnica de espectro ensanchado es una técnica de modulación que resulta ideal para las comunicaciones de datos, ya que es muy poco susceptible al ruido y crea muy pocas interferencias. La asignación de esta banda de frecuencias propició una mayor actividad en el seno de la industria y ese respaldo hizo que las WLAN empezaran a dejar ya el entorno del laboratorio para iniciar el camino hacia el mercado.

Desde 1985 hasta 1990 se siguió trabajando ya más en la fase de desarrollo, hasta que en mayo de 1991 se publicaron varios trabajos referentes a WLAN operativas que superaban la velocidad de 1 Mbit/s, el mínimo establecido por el IEEE 802 para que la red sea considerada realmente una LAN, con aplicación empresarial.

Las redes WLAN se componen fundamentalmente de dos tipos de elementos, los puntos de acceso y los dispositivos de cliente. Los puntos de acceso actúan como un concentrador o hub que reciben y envían información vía radio a los dispositivos de clientes, que pueden ser de cualquier tipo, habitualmente, un PC o PDA con una tarjeta de red inalámbrica, con o sin antena, que se instala en uno de los slots libres o bien se enlazan a los puertos USB de los equipos.

La principal ventaja de este tipo de redes (WLAN), que no necesitan licencia para su instalación, es la libertad de movimientos que permite a sus usuarios, ya que la

³ La Comisión Federal de las Comunicaciones (Federal Communications Commission es una agencia estatal independiente de Estados Unidos, directamente responsable del congreso. La FCC fue creada en 1934 en el acto de las comunicaciones y es la encargada de la regulación de comunicaciones interestatales e internacionales por radio, televisión, wireless, satélite y cable. La jurisdicción de la FCC cubre los 50 estados, el distrito de Colombia y las posesiones de Estados Unidos.

posibilidad de conexión sin hilos entre diferentes dispositivos elimina la necesidad de compartir un espacio físico común y soluciona las necesidades de los usuarios que requieren tener disponible la información en todos los lugares por donde puedan estar trabajando. Además, a esto se añade la ventaja de que son mucho más sencillas de instalar que las redes de cable y permiten la fácil reubicación de los terminales en caso necesario.

También, presentan alguna desventaja, o más bien inconveniente, que es el hecho de la "baja" velocidad que alcanzan, por lo que su éxito comercial es más bien escaso y, hasta que los nuevos estándares no permitan un incremento significativo, no es de prever su uso masivo, ya que por ahora no pueden competir con las LAN basadas en cable.

El uso más popular de las WLAN implica la utilización de tarjetas de red inalámbricas, cuya función es permitir al usuario conectarse a la LAN empresarial sin la necesidad de una interfaz física.

Normalización IEEE

La historia de las WLAN es bastante reciente. En 1989, en el seno de IEEE 802, se forma el comité IEEE 802.11, que empieza a trabajar para tratar de generar una norma para las WLAN, pero no es hasta 1994 cuando aparece el primer borrador, y habría que esperar hasta el año 1999 para dar por finalizada la norma.

En 1992 se crea Winforum, consorcio liderado por Apple y formado por empresas del sector de las telecomunicaciones y de la informática para conseguir bandas de frecuencia para los sistemas PCS (Personal Communications Systems). En 1993 también se constituye la IrDA (Infrared Data Association) para promover el desarrollo de las WLAN basadas en enlaces por infrarrojos. En 1996, finalmente, un grupo de empresas del sector de informática móvil (mobile computing) y de servicios forman el Wireless LAN Interoperability Forum (WLI Forum) para potenciar este mercado mediante la creación de un amplio abanico de productos y servicios inter operativos. Por otra parte, WLANA (Wireless LAN Association) es una asociación de industrias y empresas cuya misión es ayudar y fomentar el crecimiento de la industria WLAN a través de la educación y promoción.

Son cuatro los estándares reconocidos dentro de esta familia; en concreto, la especificación 802.11 original; 802.11a (evolución a 802.11 e/h), que define una conexión de alta velocidad basada en ATM; 802.11b, el que goza de una más amplia aceptación y que aumenta la tasa de transmisión de datos propia de 802.11 original, y 802.11g, compatible con él, pero que proporciona aún mayores velocidades.

WLAN 802.11

En junio del año 1997 el IEEE ratificó el estándar para WLAN IEEE 802.11, que alcanzaba una velocidad de 2 Mbit/s, con una modulación de señal de espectro expandido por secuencia directa (DSSS), aunque también contempla la opción de espectro expandido por salto de frecuencia, FHSS en la banda de 2,4 GHz, y se definió el funcionamiento y la interoperabilidad entre redes inalámbricas.

El 802.11 es una red local inalámbrica que usa la transmisión por radio en la banda de 2.4 GHz, o infrarroja, con regímenes binarios de 1 a 2 Mbit/s. El método de acceso al medio MAC (Medium Access Mechanism) es mediante escucha pero sin detección de colisión, CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance).

La dificultad en detectar la portadora en el acceso WLAN consiste básicamente en que la tecnología utilizada es Spread-Spectrum y con acceso por división de código (CDMA), lo que conlleva a que el medio radioeléctrico es compartido, ya sea por secuencia directa DSSS o por saltos de frecuencia en FHSS. El acceso por código CDMA implica que pueden coexistir dos señales en el mismo espectro utilizando códigos diferentes, y eso para un receptor de radio implicaría que detectaría la portadora inclusive con señales distintas de las de la propia red WLAN. Hay que mencionar que la banda de 2,4 GHz está reglamentada como banda de acceso pública y en ella funcionan gran cantidad de sistemas, entre los que se incluyen los teléfonos inalámbricos Bluetooth.

WLAN 802.11b (WiFi)

Un poco más tarde, en el año 1999, se aprobó el estándar 802.11b, una extensión del 802.11 para WLAN empresariales, con una velocidad de 11 Mbit/s (otras velocidades normalizadas a nivel físico son: 5,5 - 2 y 1 Mbit/s) y un alcance de 100 metros, que al igual que Bluetooth y Home RF, también emplea la banda de ISM de 2,4 GHz, pero en lugar de una simple modulación de radio digital y salto de frecuencia (FH/Frequency Hopping), utiliza una la modulación lineal compleja (DSSS). Permite mayor velocidad, pero presenta una menor seguridad, y el alcance puede llegar a los 100 metros, suficientes para un entorno de oficina o residencial.

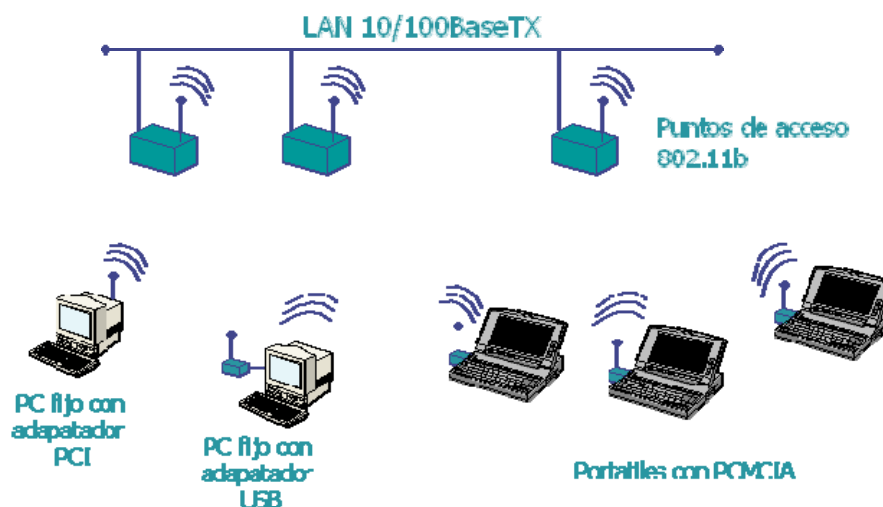


Fig. 5.5 WLAN 802.11b (WiFi)

WLAN 802.11g

El IEEE también ha aprobado en el año 2003 en el estándar 802.11g, compatible con el 802.11b, capaz de alcanzar una velocidad doble, es decir hasta 22 Mbit/s o llegar, incluso a 54 Mbit/s, para competir con los otros estándares que prometen velocidades mucho más elevadas pero que son incompatibles con los equipos 802.11b ya instalados, aunque pueden coexistir en el mismo entorno debido a que las bandas de frecuencias que emplean son distintas. Por extensión, también se le llama Wi-Fi.

WLAN 802.11a (WiFi 5)

El IEEE ratificó en julio de 1999 el estándar en 802.11a (los productos comerciales comienzan a aparecer a mediados del 2002), que con una modulación QAM-64 y la codificación OFDM (Orthogonal Frequency Division Multiplexing) alcanza una velocidad de hasta 54 Mbit/s en la banda de 5 GHz, menos congestionada y, por ahora, con menos interferencias, pero con un alcance limitado a 50 metros, lo que implica tener que montar más puntos de acceso (Access Points) que si se utilizase 802.11b para cubrir el mismo área, con el coste adicional que ello supone.

Compatibilidad y seguridad. WI-FI y WEP

A finales de la década de los 90, los líderes de la industria inalámbrica (3Com, Aironet, Lucent, Nokia, etc.) crean la WECA (Wireless Ethernet Compatibility Alliance), una alianza para la Compatibilidad Ethernet Inalámbrica, cuya misión es la de certificar la inter funcionalidad y compatibilidad de los productos de redes inalámbricas 802.11b y promover este estándar para la empresa y el hogar. Para indicar la compatibilidad entre dispositivos inalámbricos, tarjetas de red o puntos de acceso de cualquier fabricante, se les incorpora el logo "Wi-Fi" (estándar de Fidelidad Inalámbrica), y así los equipos con esta marca, soportada por más de 150 empresas, se pueden incorporar en las redes sin ningún problema, siendo incluso

posible la incorporación de terminales telefónicos Wi-Fi a estas redes para establecer llamadas de voz.

Las redes inalámbricas son inseguras aunque sólo sea porque el medio de transporte que emplean es el aire; por tanto, un elemento esencial a tener en cuenta en este tipo de redes al utilizarse la radio, es la encriptación. En general se utiliza WEP (Wired Equivalent Privacy), que es un mecanismo de encriptación y autenticación especificado en el estándar IEEE 802.11 para garantizar la seguridad de las comunicaciones entre los usuarios y los puntos de acceso. La clave de acceso estándar es de 40 bits, pero existe otra opcional de 128 bits, y se asigna de forma estática o manual (no dinámica), tanto para los clientes, que comparten todos el mismo conjunto de cuatro claves predeterminadas, como para los puntos de acceso a la red, lo que genera algunas dudas sobre su eficacia. WEP utiliza un esquema de cifrado simétrico en el que la misma clave y algoritmo se utilizan tanto para el cifrado de los datos como para su descifrado.

Con el retraso del nuevo estándar 802.11i y con el fin de resolver el tema de la seguridad, se ha lanzado la certificación WPA, aunque algunos expertos consideran que esta es sólo una solución momentánea que puede llevar a error ya que puede crear en el usuario una sensación de seguridad que este estándar no ofrece.

Otro mecanismo de seguridad definido en el estándar IEEE 802.11 es el conocido como SSID (Service Set Identifiers) o identificadores del conjunto de servicios, que es como un gestor de asignación de nombres, que proporciona un control de acceso muy rudimentario, razón por la que apenas se utiliza en las implementaciones comerciales.

Otros usuarios han preferido adquirir soluciones wireless convencionales y potenciar la seguridad con tecnología de otros fabricantes especializados en seguridad móvil en lugar de soluciones que incluyan la certificación WPA

Evolución del mercado

El mercado de las soluciones inalámbricas alcanzó en el año 2002 un volumen de negocio de unos 1.600 millones de dólares y según todas las previsiones, se espera que experimente un crecimiento anual del 20%, a pesar de algunos factores en su contra que frenan este desarrollo como los problemas de seguridad y la diversidad de estándares.

La preocupación por la seguridad es uno de los problemas que más tienen en cuenta las compañías, junto con las restricciones presupuestarias. A medida que la economía se vaya recuperando y los nuevos estándares incluyan características de seguridad mejorada, el mercado crecerá.

Por otro lado, muchas compañías se han lanzado ya a comercializar soluciones que soportan el estándar 802.11g, aunque todavía no está definido del todo. Las primeras pruebas con este tipo de equipos han demostrado que la interoperabilidad presenta lagunas, y que en redes híbridas, las prestaciones tienden a caer a los niveles del estándar anterior.

Se espera que el crecimiento venga impulsado por la tecnología Wi-Fi, así como por la mayor presencia de las tarjetas multiprotocolo, capaces de operar en estándares diversos como el 802.11b a, b y g. De hecho, en Estados Unidos esto ya se está produciendo ya que el protocolo 802.11b, convive con el 802.11a, que ofrece un mayor ancho de banda. Además, esta opción tiene la ventaja de proteger las inversiones de la obsolescencia y permite administrar el ancho de banda en función del uso o localizaciones.

La evolución del mercado de la movilidad vendrá dada sin lugar a dudas por tres "actores" fundamentales en este mercado:

- Los dispositivos móviles.
- Las redes wireless.
- Las aplicaciones móviles.

De los primeros se puede decir que cada vez son más potentes y para los próximos años se espera que los PC incorporen plataformas y tecnologías móviles y los portátiles se acerquen cada vez más al PC, hasta que compartan la misma tecnología. Se espera que esto mismo ocurra con el resto de dispositivos móviles, que converjan poco a poco hacia la compatibilidad total con el PC, a medida que su capacidad vaya incrementándose. Y por otro lado aparecerán nuevos dispositivos móviles que se adaptarán mejor a las necesidades de cada tipo de empresa.

Las redes wireless van a evolucionar de diferente manera: a través de la consolidación de redes de tercera generación, gracias a los cambios en el ancho de banda y la cobertura de las redes, etc. Sin embargo existe la amenaza de la interrelación de los diferentes estándares y tecnologías, lo que podría hacer que las

empresas tuvieran que elegir entre una tecnología concreta o tecnologías que permitan utilizar diferentes redes, a costa de una mayor complejidad y precios. Además, la posible aparición de la tecnología UWB (Ultra Wide Band), no ayudará a clarificar el mercado.

Por último se puede mencionar que tanto la mejora de las redes como una mayor capacidad permitirán montar redes con dispositivos-clientes móviles siempre conectados, de igual manera a como sucede en las redes móviles de 2.5G GSM/GPRS, al tiempo que habrá un despliegue de Web Services para aplicaciones móviles.

Una serie de facilidades y ventajas de estas tecnologías van a hacer posible la rápida expansión de estas aplicaciones móviles como por ejemplo la ubicuidad, la incorporación a los PC de tecnología Wi-Fi de serie, la mejora de los estándares de seguridad, etc.

Y una tendencia importante sería la aparición de una plataforma de conmutación centralizada que integra capacidades para gestionar la seguridad, la administración de la red y la calidad de servicio.

En cuanto a las tendencias tecnológicas se está trabajando con el fin de ofrecer soluciones inalámbricas con puntos de acceso más ligeros y económicos y con una plataforma que permita controlarlos de forma centralizada, además de incorporar otras funcionalidades.

3.- INFRARROJO IrDA

Infrared Data Association - Asociación de Datos por Infrarrojo (**IrDA**) es la organización fundada para crear las normas internacionales para el hardware y el software usados en enlaces de comunicación por rayos infrarrojos.

La tecnología de rayos infrarrojos juega un importante papel en las comunicaciones inalámbricas.

Si bien ninguna de estas tecnologías ha surgido como el líder dominante del mercado, cada una tiene una sólida base de apoyo en función de mejoras tecnológicas y de disponibilidad de productos.

Los dispositivos IrDA transmiten y reciben datos a través de pulsos de luz infrarroja. La misma tecnología que opera el control remoto de un televisor.

La mayoría de las laptops y PDA tiene un puerto infrarrojo incorporado para este tipo de comunicación de datos. Las tasas de datos típicas para las comunicaciones IrDA son de aproximadamente 4 Mbps - no son increíblemente rápidas para los parámetros actuales pero tampoco son tan malas para una comunicación fácil y rápida. No obstante, aunque pueden ser convenientes, las comunicaciones infrarrojas tienen un alcance de sólo tres pies, aproximadamente. Esto es efectivo al transmitir datos desde una notebook o PDA o desde una a una impresora personal, pero es restrictivo en una red más grande que incluya impresoras compartidas.

La tecnología IrDA es la forma más sencilla de conectar PDA y laptops con una impresora con facilidad. Este enfoque de impresión inalámbrica es una forma accesible para ofrecer impresiones rápidas y fáciles a consultores que no necesariamente tienen acceso a la red, o a un espacio de acceso público, tal como el lobby. Sin embargo, para que una solución IrDA funcione, la computadora o PDA debe tener una trayectoria sin obstáculos hacia la impresora que pueda compensar un espacio de un metro en torno a una impresora compartida.

B) TIPOS DE MÓVILES

1.- PDA.

Del inglés Personal Digital Assistant, (Asistente Personal Digital) es un computador de mano originalmente diseñado como agenda electrónica. Hoy en día se puede usar como un ordenador doméstico (ver películas, crear documentos, navegar por Internet.).

El 7 de enero de 1992, John Sculley presenta el Personal Digital Assistant, Apple Newton, en el Consumer Electronics Show (Muestra de electrónica de consumo) de Las Vegas.

Fue un sonoro fracaso de la compañía Apple, la tecnología estaba poco desarrollada y el reconocimiento de escritura era pésimo. En 1995 con la aparición de la empresa Palm comenzó una nueva etapa de un crecimiento lento pero progresivo.

La irrupción de Windows en el sector ha provocado un giro hacia una inversión mayor de capacidades multimedia o conectividad. Así, el mercado de los PDAs se divide básicamente entre los que se manejan con el sistema operativo de Palm (el aparato en sí no tiene que ser necesariamente de la marca Palm para que lleve este sistema operativo) y los que llevan el sistema de Microsoft, Pocket PC.

Los primeros son, normalmente, más económicos y sencillos de utilizar. Los equipados con el sistema de Microsoft, en cambio, tienen más opciones multimedia (permiten ver vídeos y escuchar música aunque de forma limitada por su escasa memoria) pero son más caros y gastan más baterías, aunque todo esto mejora con el tiempo y ya se empiezan a ver sistemas que incluyen Bluetooth y WiFi lo que los hace tremendamente atractivos.

Tabla comparativa de dispositivos PDA.

| DISPOSITIVO | WI-FI | BLUETOOTH | INFRARROJO | PANTALLA A COLOR | MEMORIA RAM | OS | TECLADO | CAMARA | MP3 |
|-----------------|-------|-----------|------------|------------------|-------------|-----------|---------|--------|-----|
| PALM T X | ● | ● | ● | 65000 COLORES | 128 MB | PALM OS 6 | - | - | - |
| TUNGSTEN C | ● | - | ● | 65000 COLORES | 64 MB | PALM OS 5 | ● | - | - |
| SONY CLIÉ NZ 90 | - | ● | ● | 65000 COLORES | 16 MB | PALM OS 5 | ● | ● | ● |

Tabla 5.1 Comparativo de características entre dispositivos PDA*Fig. 5.6 Palm T|X, Tungsten C y Sony Clie NZ 90*

2.- TELÉFONOS INTELIGENTES

Los últimos avances en el teléfono móvil harán que se convierta en una herramienta imprescindible en la vida moderna. El mercado de teléfonos inteligentes está creciendo tanto que está eliminando del mercado los asistentes personales digitales.

Ya no se puede describir el teléfono celular como un aparato de voz. Saldrán al mercado teléfonos capaces de escanear documentos y enviar fax. Utilizarán la cámara insertada en el móvil como un ratón y están próximos a salir dispositivos con cámara digital con resolución mayor a 6 megapíxeles.

También se abren nuevas posibilidades para el sector de video juegos porque la voz e imágenes del teléfono móvil inteligente permiten toda una nueva generación de juegos interactivos. En Japón NTT DoCoMo ha anunciado su intención de iniciar pruebas sobre una tarjeta de teléfono inteligente que podría sustituir tarjetas de identidad.

Revisión de dispositivos.

Los ordenadores de bolsillo y los teléfonos móviles están condenados a converger. Testigo de ello ha sido la reciente presentación de varios teléfonos móviles de última generación que poco tienen que envidiarles a los PDA.

El estreno de nuevos teléfonos móviles con prestaciones similares -o hasta superiores- a las de los dispositivos Pocket PC y Palm es, en este sentido, revelador. Y es que los teléfonos móviles de última generación, denominados smartphones o teléfonos inteligentes, están comenzando a invadir sin contemplaciones la cada vez más tenue línea que los separa de los ordenadores de bolsillo.

Motorola, Nokia, Panasonic, son algunas de las firmas que más fuerte están apostando por los prometedores teléfonos inteligentes.

| DISPOSITIVO | GSM | GPRS | BLUETOOTH | WI-FI | PANTALLA A COLOR | CAMARA | AUDIO/VIDEO | RANURA MEMORY CARD |
|-------------------------|-----|------|-----------|-------|------------------|--------|-------------|--------------------|
| MOTOROLA MPX | ● | ● | ● | ● | ● | ● | - | - |
| MOTOROLA MPX 100 | ● | - | ● | - | ● | - | - | ● |
| MOTOROLA A1000 | ● | ● | ● | - | ● | ● | - | - |
| MOTOROLA E 1000 | ● | - | - | - | ● | ● | ● | - |
| NOKIA 9500 COMMUNICATOR | ● | ● | ● | ● | ● | ● | - | - |
| PANASONIC X700 | ● | - | ● | - | ● | ● | - | ● |

Tabla 5.2 Comparativo de características entre dispositivos PDA



Fig. 5.7 Diferentes modelos de Smartphones

C) PLATAFORMAS

Son los 3 principales sistemas operativos o plataformas que sostienen a millones de dispositivos móviles a lo largo del mundo, estos son:

- Palm OS
- Symbian
- Windows Mobile

A continuación se hace una revisión a sus últimas versiones al igual que a las características y ventajas que cada uno nos ofrece.

1.- Palm OS

Palm OS 6.1

Es la nueva versión del sistema operativo que incorpora telefonía móvil.

Palm OS Cobalt 6.1 es una revisión completa del sistema operativo, emprendida en 2001 tras la absorción del equipo de desarrollo de la firma Be, y pretende resolver las limitaciones de versiones anteriores en cuanto a la gestión de procesos y de memoria.

Entre sus características principales destacan la integración de funciones de telefonía, el soporte nativo de redes inalámbricas WiFi y Bluetooth, un nuevo navegador Web basado en NetFront, y diversas mejoras de la interfase de usuario.

Las novedades más importantes que presenta Palm OS 6.1 son las siguientes:

- Componentes de telefonía integrados: se incluyen nuevas APIs que permiten a los fabricantes de teléfonos móviles inteligentes la creación de aplicaciones a medida de forma mucho más rápida a como se hacía hasta ahora.
Las nuevas APIs facilitan el acceso a estándares como el GSM o el GPRS.
- Soporte para Bluetooth y WiFi: Los teléfonos basados en Cobalt 6.1 podrán contar con conexiones Bluetooth 1.2 y nuevos perfiles para auriculares y dispositivos de manos libres. Asimismo, se ha previsto el soporte de redes WiFi mediante la incorporación del protocolo 802.11.
- Mejora del sistema de control con una sola mano: Ahora es más fácil acceder a las diversas aplicaciones integradas en Palm OS, además de facilitarse también el trabajo con estas mediante los dispositivos de selección y entrada que funcionan con una sola mano (joysticks).

- Soporte para SD y USB On-the-Go integrados: facilita la creación de aplicaciones residentes en una tarjeta de expansión SD (Secure Digital) y la compatibilidad con otros dispositivos móviles que cumplan el estándar USB On-the-Go.
- Interfaz gráfica renovada: El cambio más evidente introducido en Cobalt 6.1 es una sofisticada interfase gráfica que utiliza tipografías escalables TrueType, sombras y ventanas traslúcidas. Se han previsto resoluciones de pantalla VGA, QVGA y HVGA, tanto en modalidad vertical como apaisada. Ofrece, además, más posibilidades de personalización por parte del usuario, aumentándose el número de preferencias que pueden retocarse a través del panel de control.
- Multitarea real. La multitarea real ha sido una de las carencias que han afectado al sistema operativo de PalmSource desde los tiempos en que esta división formaba parte de Palm Inc., y uno de los puntos que los partidarios del Pocket PC han destacado para argumentar la superioridad de su plataforma. No obstante, y hasta hace poco tiempo, la multitarea en un dispositivo de mano no ha sido algo muy necesario, debido a su filosofía de uso.

No obstante, y con el advenimiento de las comunicaciones inalámbricas, la posibilidad de enviar un mensaje de correo electrónico en background mientras se ven unas fotografías con nuestra computadora de mano, por poner un ejemplo, ha convertido a la multitarea en una necesidad, y la nueva versión de Palm OS introducirá interesantes novedades al respecto.

Si en la versión anterior la multitarea no estaba excesivamente bien lograda según algunas críticas, para esta versión se espera una mejora del rendimiento, incluso con la introducción de la capacidad de multithreading (multihilo) para las aplicaciones. Esto, en una explicación simple, permitirá que una misma aplicación pueda realizar varias tareas a la vez.

- Nueva arquitectura de memoria. Se introduce la protección de memoria para diferentes procesos, con lo que cada uno de estos contará con su propio espacio e incluso con su propia pila. Un fallo de una aplicación ahora tendrá aún menos posibilidades de bloquear el funcionamiento de toda la máquina.
- Compatibilidad con las aplicaciones existentes. Este punto es uno de los que más preocupan a desarrolladores y usuarios. Antes de que un programa pueda aprovechar al 100% los recursos de una nueva arquitectura, debemos asegurarnos que el servicio a los usuarios no se verá interrumpido, aunque no tengamos tiempo de terminar la versión para la nueva arquitectura antes de su salida al mercado. Además, también disponemos de toda una base de aplicaciones creadas por los propios usuarios y por programadores independientes, a los cuales les costará más migrar sus desarrollos a una nueva arquitectura, tal vez por no disponer del tiempo y/o los recursos

necesarios. Es por todo esto que mantener la compatibilidad con las antiguas aplicaciones (como se ha hecho, por ejemplo, en el cambio de Mac OS 9 a Mac OS X) es algo indispensable.

Palm ya hizo una importante migración de arquitectura en la anterior versión (la 5) de Palm OS, pasando de usar chips Motorola 68k a usar los StrongARM de Intel. En el Palm OS 6 la capa de ejecución de aplicaciones antiguas, y por lo tanto preparada para funcionar sobre un chip Motorola 68k, será muy similar en su concepto a la que disponía el Palm OS 5, pero mejorando varios aspectos. Dicha capa de emulación estará preparada para hacer creer a la aplicación que se está ejecutando sobre Palm OS 4.x.

- Otro punto sobre el que poco se sabe es la compatibilidad a nivel de hardware. Palm siempre ha sido muy cuidadosa en procurar que sus dispositivos puedan instalar, al menos, una versión superior del sistema operativo, por lo que es de suponer que los usuarios de computadoras de mano con Palm OS 5.x puedan migrar al nuevo Palm OS 6, aunque la transición de la arquitectura 68k a la ARM podría hacer esta tarea algo dificultosa.
- Multimedia. La imagen y el sonido están cobrando cada día más importancia en los dispositivos móviles, ya sean PDA's o teléfonos. Y es que según muchos expertos, el destino de ambos mundos es el de fusionarse. Siguiendo con las tendencias actuales de introducir capacidades de reproducción de vídeos cortos o música en MP3 en los teléfonos, Palm OS 6 contará con un nuevo framework para multimedia con un engine de gráficos escalables.
- Los programadores también saldrán beneficiados. PalmSource pondrá a disposición de los desarrolladores una serie de herramientas para la migración de aplicaciones a código nativo para la plataforma StrongARM. Además, el nuevo sistema tiene también una serie de facilidades para este colectivo, como por ejemplo la eliminación del límite de 64 kb's en los registros, segmentos de código o direcciones de memoria, además de nuevas capacidades de la API.

Palm OS 5

Se hace revisión a esta anterior versión del sistema operativo de Palm debido a que el dispositivo en el que se llevarán a cabo las pruebas de la aplicación esta equipado con Palm OS 5.

Procesadores ARM y mayor conectividad y prestaciones multimedia.

Principales características:

El sistema operativo aporta, para empezar, más conectividad que sus versiones anteriores. Así, además del habitual soporte para WAN y Bluetooth, esta edición ofrece la posibilidad de funcionar con las llamadas redes de comunicación local inalámbricas, más conocidas por su nombre técnico, 802.11b. Esta mayor conectividad, enfocada sobre todo al mundo empresarial, aproxima las prestaciones de Palm OS en este campo a las de los dispositivos basados en el Pocket PC 2002 de Microsoft.

Las mayores prestaciones multimedia representan otro gran cambio en la filosofía de Palm OS. El sistema operativo mejora, en este sentido, tanto la calidad de la imagen como las posibilidades de audio de los PDA que lo utilicen.

La resolución de la imagen goza de una mayor definición, abandonando los ya casi históricos 160 x 160 píxeles por unos más deseables 320 x 320 píxeles. En el terreno del audio, se pueden escuchar archivos de sonido con calidad de CD.

Otra de las características es la elección de los procesadores ARM para dar vida a los PDA basados en la versión 5.0.

Los fabricantes de PDA basados en Palm OS podrán prescindir de los clásicos procesadores DragonBall de Motorola y utilizar el tipo de procesador que utiliza la mayoría de Pocket PC -basados en los sistemas operativos de Microsoft. Nos referimos a los chips de Intel, Texas Instruments o la propia Motorola.

También en la seguridad se ha esmerado la subsidiaria PalmSource. Así, las mayores posibilidades de conectividad on-line van cogidas de la mano con una mayor protección de los datos, pensada sobre todo para el envío y recepción de e-mails, la navegación por Internet y las transacciones a través de la red.

Cabe remarcar, por último, que esta versión de Palm OS permite utilizar aplicaciones creadas para la versión 4.0. Además, PalmSource garantiza que tales aplicaciones podrán ejecutarse de una forma más veloz con el Palm OS 5.0.

2.- Symbian

El sistema operativo Symbian se ha hecho un hueco en los teléfonos avanzados que están llegando al mercado.

Con 32-bits, este sistema operativo está diseñado para teléfonos móviles capaces de gestionar datos y algunas aplicaciones. La mayoría de los actuales teléfonos avanzados, o smartphones, como se denominan en inglés, utilizan las versiones 6.1 o 7.0 del sistema operativo Symbian. Pero Symbian 8.0, incluye mejoras significativas para Java y actividades multimedia. El primer teléfono 3G de Nokia, el Nokia 6630, utiliza Symbian 8.0 y cuenta con capacidades de vídeo streaming y vídeo conferencia.

Symbian OS se ejecuta aproximadamente en 30 modelos de teléfonos de fabricantes como Nokia, Sony Ericsson, Motorola y Samsung. Y, según Symbian, existen cerca de 20 millones de teléfonos en el mundo que actualmente utilizan su sistema operativo, con un crecimiento del 200 por ciento anual. Esto proporciona una enorme base, y en expansión, para los desarrolladores de software para móviles, que además deberían tener en cuenta el sistema operativo Symbian.

Ya hay vendedores de software independientes (ISVs) y organizaciones que están desarrollando aplicaciones basadas en Symbian para una base de usuarios acostumbrados a descargarse actualizaciones para sus móviles, como melodías y que están preparados y mentalizados para pagar por la descarga de nuevas aplicaciones que puedan utilizar en sus teléfonos avanzados. Juegos y aplicaciones de gestión son seguros para el mercado Symbian, principalmente por la mejor capacidad de procesamiento y mejores gráficos que pueden desarrollar los nuevos terminales.

Mejoras en el apartado multimedia, en la seguridad, y más herramientas para programadores son las claves de esta nueva versión del sistema operativo líder entre los smartphones.

Symbian es, el sistema operativo líder en el segmento de la telefonía móvil, aunque la omnipresente Microsoft y su Smartphone suponen una grave amenaza para este liderato. Palm OS queda un poco rezagado, aunque en ciertos países, y gracias a una cuidada oferta en conjunción con algunas operadoras, parece que también está ganando algo de terreno. Ambas plataformas -Windows CE/Smartphone y Palm OS- cuentan con el apoyo que les ofrece su amplia cuota de mercado en el sector de las PDA's, no siendo así en el caso de Symbian, un sistema operativo dirigido especialmente al sector de los teléfonos inteligentes.

Es por eso que para no perder cuota de mercado, Symbian tiene que innovar de forma constante, y la versión 9 es una buena muestra de ello.

El objetivo principal de la nueva versión de Symbian es asegurar el sector de los teléfonos 2,5 y 3G, permitiendo a los fabricantes reducir costes e implementar sus soluciones con más rapidez -al ser un sistema operativo ya enfocado al mundo de la telefonía-.

Los primeros aparatos funcionando bajo la versión 9.0 de Symbian OS aparecieron en el mercado durante la segunda mitad de 2005.

Las principales mejoras en esta versión son:

- Multimedia: se añade soporte para auriculares estéreo conectados por Bluetooth, dispositivos de almacenamiento masivo USB, reproducción avanzada de audio, gestión DRM (Digital Rights Management) para formato MPEG, soporte para cámaras fotográficas de varios megapíxeles, para aceleradores gráficos 3D y para diferentes orientaciones de pantalla e incluso para más de una pantalla. Como los videojuegos se están potenciando en el sector de los teléfonos móviles (y especialmente entre los usuarios más jóvenes) la nueva versión de Symbian se enfoca mucho a la optimización del hardware y el software para este uso.
- Gestión remota de dispositivos que cumplan la normativa OMA: mediante esta especificación, la operadora que proporciona el servicio de telefonía puede acceder al teléfono del usuario para actualizar determinado software que permite acceder a los servicios que presta, o bien auditar las aplicaciones que el usuario tiene instaladas.
- Seguridad mejorada: las aplicaciones instaladas en nuestro teléfono solamente tendrán acceso a recursos de comunicaciones y datos si se les da permiso específicamente, con lo que no correremos el riesgo de que nos sea robada información de nuestro móvil.
- Programa de certificación de Aplicaciones Symbian: por el cual, los productores de software pueden obtener un certificado de seguridad para las aplicaciones que desarrollen.
- Soporte para la última generación de procesadores ARM. Algunos fabricantes de chips para telefonía móvil, como Intel, están preinstalando Symbian en sus desarrollos, permitiendo de esta forma reducir el tiempo de desarrollo de un nuevo modelo de teléfono inteligente.
- Herramientas para programadores: todo sistema operativo que quiera triunfar, expandirse y ganar más usuarios, debe disponer de una gran cantidad de soluciones software para resolver cualquier problema de tipo general. Symbian no es una excepción, y para facilitar dicha tarea a los programadores, cuenta con herramientas propietarias y libres para el desarrollo de programas. Entre ellas se encuentra el conocido IDE

CodeWarrior, con la versión 3.0 para Symbian OS. La versión soportada del compilador libre GCC pasa de ser la 2.98 a la 3.4, y el propio consorcio Symbian está desarrollando una nueva herramienta basada en el IDE Eclipse.

3.- Windows Mobile

Windows Mobile aporta todo el poder del software Windows en un dispositivo móvil.

Características principales:

- Sync & Go. El software Windows Mobile para Pocket PC ayuda a gestionar comunicaciones y datos personales y profesionales sin detenerse. Con Pocket Outlook y ActiveSync, podrá sincronizar el correo electrónico, el calendario, los contactos y las tareas con el PC de escritorio. Windows Mobile puede abrir documentos importantes por correo electrónico utilizando las versiones Pocket de Word y Excel.
- Windows Mobile para Pocket PC ofrece una amplia gama de opciones de conectividad, desde la sincronización con el PC de escritorio y las redes LAN empresariales hasta las zonas activas Wi-Fi, pasando por las redes de telefonía móvil. La función Wi-Fi sin configuración del software Windows Mobile para Pocket PC detecta automáticamente las redes Wi-Fi y permite conectarse a ellas de manera rápida y sencilla y podrá navegar con Pocket Internet Explorer.
- La versión más reciente del reproductor de Windows Media para Pocket PC permite reproducir archivos MP3 y WMA. Y la compatibilidad con Plus! Digital Media Experience para PC de escritorio, puede reproducir Fotos narradas y descargar fragmentos de audio y vídeo digital desde Sync'N Go. Por su parte, Microsoft Pictures permite editar fotografías digitales, crear presentaciones con diapositivas e incluso enviar fotos digitales por correo electrónico o SMS.
- Tiene a una amplia gama de aplicaciones descargables escritas por proveedores de software independientes y por Microsoft. Desde juegos y aplicaciones multimedia hasta herramientas de productividad personal.

Aplicaciones principales:

Las aplicaciones principales para Windows Mobile para Pocket PC incluyen:

- Pocket Outlook, incluidos:
 - Correo electrónico
 - Calendario

- Contactos
- Tareas
- Notas
- Pocket Internet Explorer
- Reproductor de Windows Media
- Pocket MSN Messenger
- Pocket Word
- Pocket Excel
- Phone Application
- Microsoft Pictures
- Microsoft Reader
- ActiveSync
- Cliente de Servicios de Terminal Server
- Connection Manager
- Software de reconocimiento de escritura manuscrita
- Utilidad de transcripción
- Reconocedor de letras
- Reconocedor de bloques
- Teclado de software
- Idiomas

Los Pocket PC basados en Windows Mobile están disponibles en los siguientes idiomas: Chino (simplificado), chino (tradicional), inglés, francés, alemán, italiano, japonés, coreano, portugués (brasileño) y español.

Funciones de hardware estándar

El software Windows Mobile para Pocket PC tiene un conjunto estándar de características de hardware para garantizar que la experiencia sea uniforme en todos ellos. Además de garantizar una experiencia coherente para los usuarios, estas características de hardware estandarizadas facilitan el desarrollo de aplicaciones de otros proveedores. El resultado es que tiene a su disposición una amplia variedad de aplicaciones adicionales que amplían la potencia del Pocket PC.

Entre las características comunes se incluyen memoria Flash para permitir actualizaciones de software, infrarrojos, conectividad USB, resolución de pantalla QVGA, procesador ARM, pantalla táctil y salida estéreo.

Partners de Windows Mobile: fabricantes de equipos originales (OEM)

El software Windows Mobile para Pocket PC cuenta con el mayor número de OEM del mercado de dispositivos de mano. Nuestros partners fabricantes ofrecen un abanico de productos que satisfacen las diversas necesidades del consumidor, la empresa y los mercados verticales en todos los puntos del planeta.

**VI. DESARROLLO DE APLICACIÓN PARA PDA Y DISPOSITIVOS
MÓVILES COMPATIBLES**

LectorBMU, versión 1.0

Febrero 2007

A) VISIÓN DEL PROYECTO

1. Introducción

Es este el capítulo se explica el entorno de programación ocupado para el desarrollo de la aplicación LectorBMU (Lector de Boletín para Móviles Universitarios), Java 2 Micro Edition J2ME, también el funcionamiento de la aplicación LectorBMU el desarrollo de la misma, las herramientas ocupadas durante el proceso, la ejecución de la aplicación en emuladores y después en los dispositivos físicamente, describiendo las pruebas realizadas y el uso de la aplicación.

2. Objetivo

Generar un medio informativo alternativo para difundir información de carácter académico y/o cultural de forma electrónica por medio de un servidor, (PC con puerto bluetooth, usb o algún otro puerto de comunicación) sin necesidad de conexión a internet, generando una PAN (personal Área Network), a dispositivos móviles equipados con bluetooth o algún otro puerto de comunicación como lo son teléfonos celulares o PDA's, logrando así mayor difusión de cierta información, de forma rápida, fácil ahorrando así los medios de impresión ocupados anteriormente.

3. Descripción de la Aplicación LectorBMU

La aplicación tendrá las siguientes características:

- Ser compatible con cualquier dispositivo que independiente de su sistema operativo pueda ejecutar aplicaciones java.
- Debe ser transferida al dispositivo sin necesidad de establecer una conexión a Internet y generar costo al usuario del dispositivo móvil.
- El usuario podrá ejecutar la aplicación y hacer uso de ella sin requerir mayor conocimiento que el que posee para hacer uso de las funciones de su dispositivo móvil, sin embargo deberá poseer un apartado de ayuda para el usuario.
- Una vez instalada en el dispositivo móvil se podrá retransmitir hacia cualquier otro dispositivo móvil compatible, por los medios que el propio dispositivo móvil posee.
- La aplicación es un lector de boletines informativos que dependiendo del archivo de texto origen será la información que despliegue en pantalla, este archivo origen se podrá sustituir y solo bastará con volver a compilar la aplicación para que esta sea diferente.
- El archivo fuente que acepta la aplicación solo puede ser de texto plano *.txt

- La aplicación mostrará en pantalla el boletín informativo, el menú de la aplicación se mapeará de forma automática a las teclas principales de navegación del dispositivo móvil y nos permitirá desplazarnos hacia abajo o hacia arriba a lo largo del texto.
- La aplicación puede ser eliminada del dispositivo por medio de las opciones nativas del móvil.

Las funciones principales de la aplicación serán:

- Marcadores: Incluirá dos marcadores por default el de inicio y el final el cual nos posiciona en el inicio del texto o al final del mismo respectivamente.
Se pueden agregar mas marcadores esta opción nos permitirá agregar marcadores quitar marcadores ir a la posición de algún marcador o renombrar el marcador.
- Buscar: Nos permitirá buscar un carácter o una cadena de caracteres en el texto, una vez que lo encuentra nos posiciona ahí o nos arroja el mensaje de que no encontró el texto.
- Configuraciones: Se podrá llevar a cabo una configuración mínima como preferencias del usuario en la cual se contemplan 3 tamaños de letra, 5 niveles de interlineado, desplazamiento vertical u horizontal y fondo blanco o negro lo cual generará que la letra sea del color opuesto al seleccionado, esta configuración podrá ser guardada y respetada incluso si la aplicación es cerrada, solo puede ser modificada por medio del menú o reestablecida si se borran las bases de preferencias del dispositivo, lo cual no es necesario pues la configuración podrá ser modificada por el usuario las veces que sea necesario.
- Ayuda: Aquí a manera de texto se muestran las funciones predeterminadas que tendrán algunas teclas del dispositivo como lo es avanzar una página, o varias, acceder al menú o a la lista de marcadores.
- Quitar: Esta acción nos permite concluir con la aplicación en el dispositivo liberando los recursos del sistema de manera automática e inmediata.

4. Problemas por no tenerlo

Se generan gastos en impresión de boletines, el boletín o informativo puede ser alterado o quitado del lugar donde se pegue, si se trata de una información que en cierta fecha será consultada por muchos alumnos pueden generarse aglomeraciones, se desaprovecha la infraestructura existente en el campus, la información publicada no llega a la mayoría de los estudiantes para los cuales se ha publicado.

5. Procedimiento Actual

Actualmente algunos medios de difusión son los carteles o medios impresos colocados en áreas comunes para los estudiantes como pueden ser corredores, pizarrones, entradas de laboratorios, lo cual al ser información necesaria para todos en una fecha específica genera aglomeraciones y los medios están expuestos a ser alterados o arrancados del lugar.

6. Beneficio

Generar un canal alternativo de difusión de información común a los estudiantes que poseen un dispositivo móvil con puerto bluetooth o algún otro y que ejecuten aplicaciones java para que descarguen la misma desde un servidor el cual enviará esta bajo petición del estudiante, los posibles usos de la aplicación pueden ser difusión de calificaciones, horarios, gaceta UNAM etc.

B) DISEÑO CONCEPTUAL

1. Clasificación del proyecto

Infraestructura UNAM

2. Tipo de Proyecto

Infraestructura UNAM – Mejoras

3. Soluciones propuestas para implementar el proyecto

Se desarrollará una aplicación llamada LectorBMU la cual podrá ser descargada a dispositivos móviles compatibles y la cual nos permitirá difundir información de carácter académico o cultural a un mayor número de estudiantes mediante boletines de texto plano que se leerán en teléfonos celulares o dispositivos PDA. A continuación se mencionan las características que tendrá la aplicación y sus alcances así como su funcionamiento.

Una vez ejecutada nos presentará la pantalla de bienvenida la cual muestra información sobre la aplicación y posteriormente pasa a la pantalla que muestra el Boletín Informativo.

Con las teclas de navegación podremos desplazarnos a lo largo del texto.

La aplicación contará con un menú principal el cual incluye las siguientes funciones:

MARCADORES. Permite desplazarse en el texto a posiciones previamente establecidas por medio de marcadores permite agregar, eliminar o renombrar un marcador al igual que ir a la posición del mismo por medio del menú del marcador.

El menú del marcador contempla las siguientes opciones:

Renombrar. Permitirá renombrar un marcador ya existente mas no cambiar su posición.

Ir a. Mostrara el texto en pantalla en donde se creo el marcador.

Agregar. Permitirá agregar marcadores en la posición de texto deseada.

Eliminar. Permitirá seleccionar un marcador existente y eliminarlo de la lista de marcadores.

BUSCAR. Menú que permitirá buscar un carácter o cadena de caracteres dentro del texto, si el texto buscado no se encuentra indicará no encontrado y si lo encuentra se posicionará donde este el primer resultado de búsqueda.

CONFIGURACIONES. Menú que permitirá llevar a cabo una visualización personalizada de la aplicación como lo es el tamaño de la letra, el nivel de interlineado, el tipo de desplazamiento, vertical u horizontal y el color del fondo, al seleccionar este en negro la letra es blanca y viceversa. Esta configuración se guarda en el dispositivo móvil y se conserva para las siguientes consultas permitiendo modificarse en cualquier momento.

AYUDA. Al seleccionar el menú ayuda aparecerá en pantalla la función que tienen predeterminada algunas teclas del dispositivo permitiendo esto desplazarse mejor en el texto, búsquedas y marcadores.

QUITAR. Al seleccionar el menú quitar termina la ejecución del MIDLET y libera los recursos del sistema.

Con este menú existe opción de personalizar el Boletín, caso contrario bastará con hacer uso de las teclas de navegación para desplazarnos por el texto, dependiendo del archivo fuente será el contenido del Boletín Universitario, el cual como ya se mencionó puede ser de carácter académico o cultural.

4. Arquitectura

Se seleccionó J2ME, arquitectura ampliamente explicada en el Anexo C, para el desarrollo, debido a que la aplicación se implementará en teléfonos celulares de los cuales se desconoce sistema operativo, J2ME trabajará independiente del sistema Operativo del dispositivo solo con saber que ejecute aplicaciones java como son los juegos, que la mayoría incluye, otro motivo importante es que el tamaño de la aplicación realmente es pequeño y no ocupará mucha memoria en el dispositivo. El código fuente está segmentado en diferentes clases de acuerdo a sus funciones y coordinado por medio de una clase principal llamada LectorBMU.java.

Herramientas de desarrollo:

JCreator v2.5

Java version: 1.4.2_11

Esta aplicación originalmente es para desarrollo de aplicaciones java SE ya que no incluye los API de J2ME sin embargo se ocupó como editor de texto para las clases y funciones ya que respeta todas las reglas de la programación en general. Posteriormente se pasan los archivos a la carpeta que corresponde al código fuente en la herramienta J2ME Wireless Toolkit y ahí es donde se compila y preverifica la aplicación.

La versión que fue desarrollada para teléfonos celulares está compilada y preverificada en:

J2ME Wireless Toolkit 2.0
Java vendor: Sun Microsystems Inc.
Java version: 1.4.2_11
Operating system: Windows XP

Atributos de la aplicación:

MIDlet-Jar-Size: 42543 (42 Kb)
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-2.0

La versión que fue desarrollada para PDA está compilada y convertida a PRC (Aplicación nativa de PalmOS) en:

J2ME Wireless Toolkit 1.0.4_01
Java vendor: Sun Microsystems Inc.
Java version: 1.4.2_11
Operating system: Windows XP
Atributos de la aplicación:

MIDlet-Jar-Size: 40870 (40 Kb)
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-1.0
LectorBMU.prc (67 Kb)

Es posible trabajar independientemente con esta herramienta. A continuación se muestra como es el proceso de desarrollo de un MIDlet usando tan solo el J2ME Wireless Toolkit.

El módulo principal de esta herramienta es la llamada KToolBar.

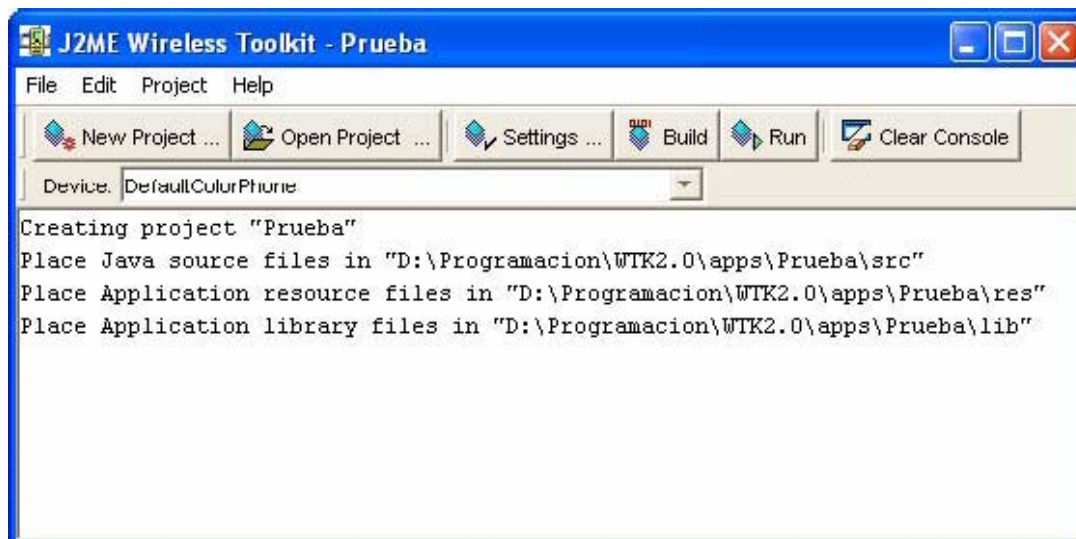


Fig. 6.1 Aspecto de la KToolBar del J2ME Wireless Toolkit.

El proceso que se realiza es el siguiente:

- En primer lugar se crea un nuevo proyecto al que se le asigna un nombre en concreto. A la hora de crear este proyecto se da la oportunidad de definir ciertos atributos en el archivo JAD.
- Una vez creado el proyecto, el entorno crea un sistema de directorios dentro de la carpeta apps. Aquí se crea una carpeta con el nombre del proyecto y dentro de ella un conjunto de subdirectorios cuya estructura puede apreciarse en la siguiente figura:

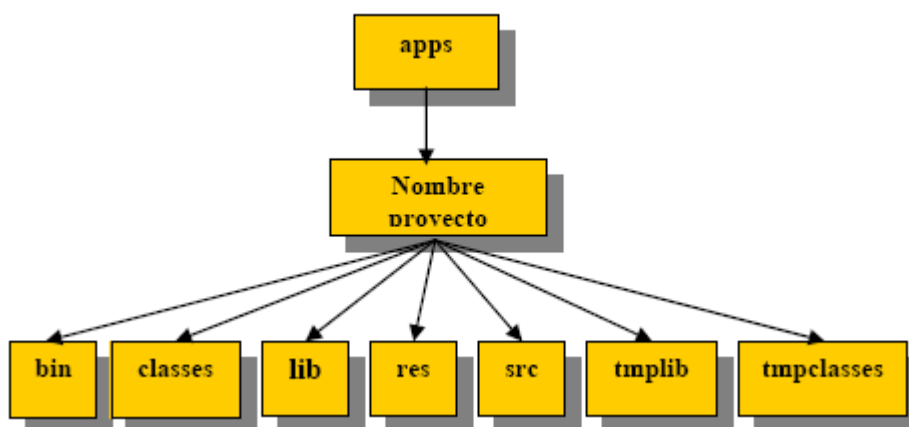


Fig. 6.2 Jerarquía de directorios.

- En el subdirectorio src es donde se deben guardar los ficheros .java con el código fuente. En el directorio res se guardarán los recursos que utilice el MIDlet. Si se usa alguna librería adicional, se debe guardar en el subdirectorio lib.
- Una vez situados los archivos en los subdirectorios correspondientes tan sólo hay que compilar el proyecto y ejecutarlo sobre el emulador.
El proceso de preverificación se realiza automáticamente después de la compilación.
- Una vez que se hayan hecho todas las pruebas necesarias con el MIDlet, esta herramienta también da la posibilidad de empaquetar el MIDlet y dejarlo de esta manera, preparado para la descarga en un dispositivo real.

5. Metodología

La base de la metodología ocupada para el desarrollo de la aplicación fue la llamada **RUP Rational Unified Process – Proceso Unificado Racional** metodología que divide en 4 fases el desarrollo del software:

- **Inicio**, El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración**, En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción**, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transmisión**, El objetivo es llegar a obtener la liberación del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala.

Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Cabe mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Traslado de las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Traslado de los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de Soporte

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.

- Distribución: Hacer todo lo necesario para la salida del proyecto

Los elementos del RUP son:

- **Actividades**, Son los procesos que se llegan a determinar en cada iteración.
- **Trabajadores**, Personas involucradas en cada proceso.
- **Artefactos**, Un artefacto puede ser un documento, un modelo, o un elemento de modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

6. Estimación de esfuerzo y recursos

Estudio del lenguaje de programación java.

Se requiere profundizar en el estudio de la tecnología J2ME al igual que la inversión de tiempo para realizar diferentes ejemplos funcionales, relacionados a la aplicación final.

Para este proyecto no es necesario realizar alguna inversión ya que la aplicación se probará en emuladores que la herramienta de desarrollo incluye y se cuentan con 2 dispositivos móviles en los que se instalará físicamente la aplicación una PDA¹ y un teléfono celular² con el perfil MIDP 1.0 y 2.0 respectivamente. Al igual que una laptop³ con conexión usb y bluetooth que servirá de servidor para transferir la aplicación a los dispositivos móviles.

7. Manejo de riesgos y seguridad

Se trata de un sistema de difusión de información alterno para la comunidad universitaria, se debe contemplar la resistencia a su uso por parte de la comunidad así como ciertas limitantes de los dispositivos móviles. No se relaciona con otros sistemas ni utiliza filtros de red especiales.

8. Plan de Pruebas

Las pruebas que se deben realizar al finalizar el proyecto son:

Perfil MIDP 1.0:

1. Compilar el proyecto en J2ME Wireless Toolkit 1.0.4_01 y ejecutarlo en un emulador propio de la herramienta.
2. Compilar el proyecto en J2ME Wireless Toolkit 1.0.4_01 Convertirlo a PRC (Aplicación nativa de PalmOS) y ejecutarla en el emulador PDA propio de la herramienta.

¹ Sony Clie modelo NZ90 MIDP 1.0 conexión Bluetooth

² Motorola c650 MIDP 2.0 conexión USB

³ Sony Vaio modelo FS815F, Celeron Mobile, conexión USB y Bluetooth

3. La versión PRC, transferirla por medio de Bluetooth a un dispositivo físico (PDA) que tenga implementado el perfil MIDP 1.0 y corroborar que la aplicación funcione.
4. Ejecutar la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 1.0.4_01, al igual que en el dispositivo físico y agregar, ir a, renombrar y eliminar un marcador verificando su funcionamiento.
5. Ejecutar la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 1.0.4_01, al igual que en el dispositivo físico y buscar una palabra existente en el texto del boletín y una palabra inexistente exponiendo los resultados.
6. Ejecutar la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 1.0.4_01, al igual que en el dispositivo físico y llevar a cabo la siguiente configuración comprobando que se respete la misma al guardar los cambios:
Fuente: grande
Interlineado: 2
Desplazamiento: Vertical
Fondo: Negro
7. Ejecutar la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 1.0.4_01, al igual que en el dispositivo físico y verificar que los comandos del texto de ayuda se ejecutan sin problema alguno.
8. Eliminar el MIDlet del dispositivo físico.

Perfil MIDP 2.0:

1. Compilar el proyecto en J2ME Wireless Toolkit 2.0 y ejecutarlo en un emulador propio de la herramienta.
2. Compilar el proyecto en J2ME Wireless Toolkit 2.0 transferirla por medio de USB a un dispositivo físico (Celular) que tenga implementado el perfil MIDP 2.0 y corroborar que la aplicación funcione.
3. Compilar el proyecto en J2ME Wireless Toolkit 2.0 transferirla por medio de Bluetooth a un dispositivo físico (Celular) que tenga implementado el perfil MIDP 2.0 y corroborar que la aplicación funcione.
4. Ejecutar la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 2.0, al igual que en el dispositivo físico y agregar, ir a, renombrar y eliminar un marcador verificando su funcionamiento.

5. Ejecutar la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 2.0, al igual que en el dispositivo físico y buscar una palabra existente en el texto del boletín y una palabra inexistente exponiendo los resultados.
6. Ejecutar la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 2.0, al igual que en el dispositivo físico y llevar a cabo la siguiente configuración comprobando que se respete la misma al guardar los cambios:
Fuente: mediana.
Interlineado: 0
Desplazamiento: Horizontal
Fondo: Blanco
7. Ejecutar la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 2.0, al igual que en el dispositivo físico y verificar que los comandos del texto de ayuda se ejecutan sin problema alguno.
8. Eliminar el MIDlet del dispositivo físico.

9. Plan de capacitación

No se requiere mayor capacitación para el usuario que el manejo de aplicaciones comunes para teléfono celular o PDA y el uso del menú de la aplicación, por lo que en sesiones cortas (10-15 minutos) se puede hablar de la aplicación desde la descarga/instalación, uso y eliminación del dispositivo móvil.

Para la persona encargada de manejar el equipo servidor se requiere:

- Capacitar para manejo y transferencia de archivos por bluetooth PC-celular.
- Manejo de herramientas de compilación de la aplicación para poder sustituir el texto fuente y volver a crear la aplicación antes de transferirla al dispositivo móvil.
- Manejo de aplicaciones comunes para teléfono celular o PDA y el uso del menú de la aplicación, por lo que se requiere de una sesión de 30-40 minutos.

C) ESPECIFICACIÓN FUNCIONAL

1. Diseño Lógico

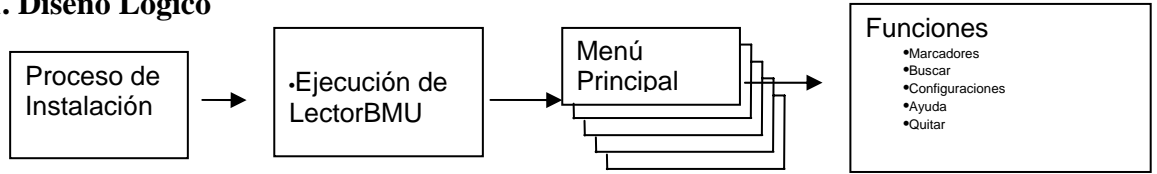


Fig. 6.3 Diagrama del diseño lógico

2. Diseño Físico

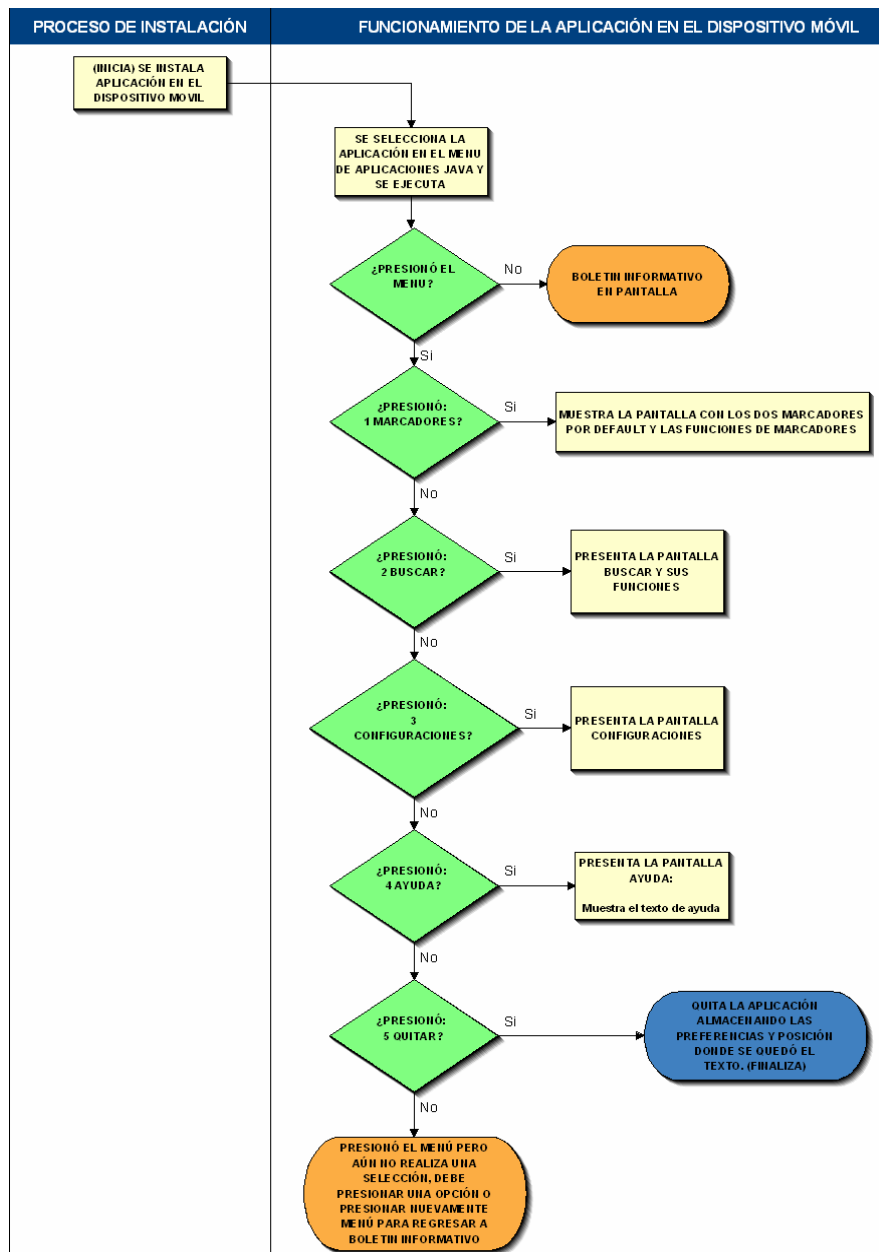


Fig. 6.4 Diagrama del diseño físico

Archivos.

A continuación se enlistan las diferentes clases que componen la aplicación mencionando la función de cada una de ellas.⁴

- LectorBMU.java
Esta es la clase que implementa el midlet que controla la ejecución de la aplicación en el dispositivo móvil.
- LectorCanvas.java
Clase que implementa la creación y presentación de las distintas capas que intervienen en el midlet. La clase también controla el desplazamiento de los elementos de las capas de forma individual.
- Menu.java
Esta es la clase que implementa el menú principal del midlet
- FormularioConfig.java
Clase que implementa el formulario del menú configuraciones del midlet.
- FormularioAyuda.java
Esta es la clase que implementa el formulario de ayuda del menú principal del midlet.
- FormularioBuscar.java
Clase que implementa las acciones del formulario buscar, permitiendo ingresar una cadena de caracteres y buscarla en todo el texto.
- FormAgregaMarcador.java
Clase que implementa la creación de marcadores y agrega los mismos en el midlet.
- InterfaceFuente.java
Clase que define la interfaz de la fuente en dimensiones.
- FormEdicionMarcador.java
Clase que implementa la edición de marcadores y actualizar los mismos en el midlet.
- Marcador.java
Clase que implementa el funcionamiento de los marcadores en el midlet.

⁴ El código fuente de la aplicación puede ser consultado en el ANEXO B de este trabajo.

- ListaMarcadores.java
Clase que implementa el menú de los marcadores en el midlet.
- SistemaDeFuentes.java
Clase que implementa la interfaz de la fuente en la pantalla.
- Convertidor.java
Clase que define la conversión de los caracteres para su posterior implementación en la pantalla del dispositivo.
- Localizacion.java
Clase que define la internacionalización de los menús para su posterior implementación en la pantalla del dispositivo.
- ImagenFuente.java
Clase que implementa la creación y presentación de las distintas capas que intervienen en el midlet. La clase también controla la imagen de la fuente cargada al dispositivo.

Funcionamiento de la aplicación.

A continuación se menciona el funcionamiento de la aplicación una vez que se ejecuta desde el emulador propio de la herramienta J2ME Wireless Toolkit 2.0.

Se ejecuta la aplicación desde el botón de ejecutar del dispositivo móvil como se muestra en la siguiente imagen.



Figura 6.5 Ejecución del LectorBMU

Una vez ejecutada presenta la pantalla de bienvenida la cual muestra información sobre la aplicación. Esta información dura 5 segundos en el dispositivo y posteriormente pasa a la pantalla que muestra el Boletín Informativo, en este momento ya se puede consultar el mismo.



Figura 6.6 Presentación del LectorBMU

Con las teclas de navegación se puede desplazar a lo largo del texto, sin embargo el menú ayuda ofrece algunas otras opciones, se revisa ahora el menú principal de la aplicación LectorBMU y cada una de sus opciones de forma individual. Para poder acceder al menú se presiona la tecla menú o tecla principal del dispositivo.



Figura 6.7 Menú principal del LectorBMU

Menú principal:

MARCADORES. Permite el desplazamiento en el texto a posiciones previamente establecidas por medio de marcadores, permite agregar, eliminar o renombrar un marcador al igual que ir a la posición del mismo para llevar a cabo estas acciones se presiona el menú principal y el número correspondiente al menú marcadores en este

caso el número 1 o simplemente seleccionándolo con el botón de selección del dispositivo móvil.

Una vez que se selecciona mostrará los 2 marcadores dados de alta por default, inicio y fin. Al seleccionar alguno de los dos se posicionará en el principio del texto o al final del mismo respectivamente, si no se selecciona ninguno por medio del menú del marcador se puede agregar renombrar o eliminar un marcador.



Figura 6.8 Menú marcadores del LectorBMU

Renombrar. Se selecciona el menú marcadores, se selecciona un marcador de la lista y posteriormente el menú renombrar, aparecerá el nombre del marcador en texto editable solo se requiere borrar y escribir el nuevo nombre y oprimir el botón con la función renombrar.

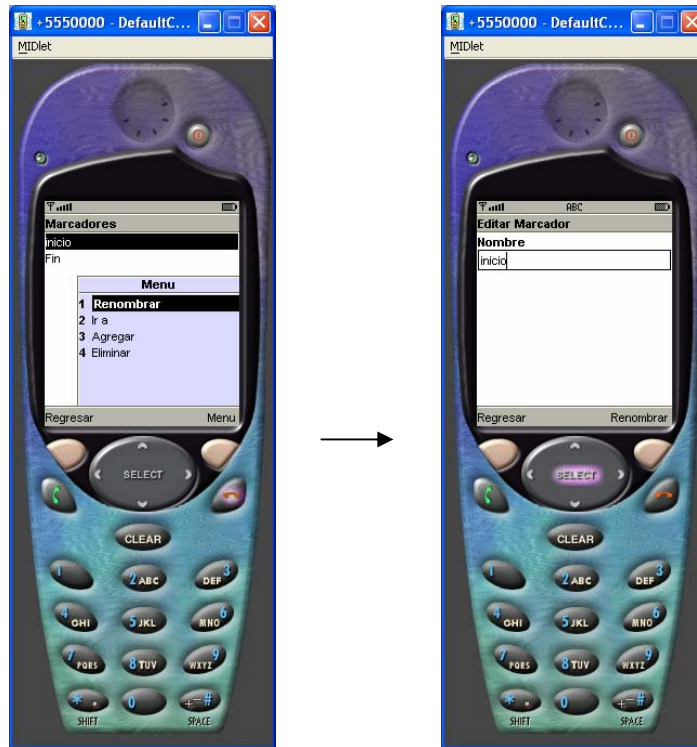


Figura 6.9 Menú renombrar del LectorBMU (1)



Figura 6.10 Menú renombrar del LectorBMU (2)

Ir a. Mostrara el texto en pantalla en donde se creo el marcador en este caso en el principio del texto. Se selecciona el menú marcadores, se selecciona un marcador de la lista y posteriormente el menú Ir a y manda a la posición del marcador.



Figura 6.11 Menú ir a del LectorBMU

Agregar. Hay que posicionarse en el apartado del texto donde se desea el nuevo marcador, se selecciona del menú principal marcadores y posteriormente el menú Agregar, se ingresa el nombre del nuevo marcador y se presiona el botón con la función Agregar. Como se muestra a continuación.

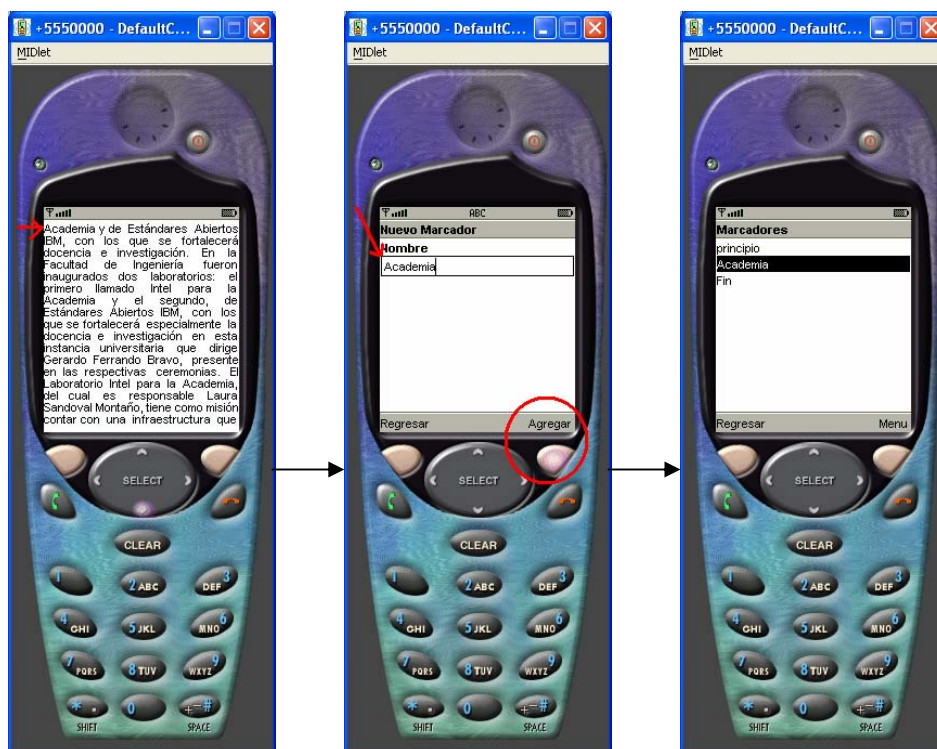


Figura 6.12 Menú Agregar del LectorBMU

Eliminar. Se selecciona del menú principal marcadores, luego se selecciona el marcador a eliminar y posteriormente el menú Eliminar.



Figura 6.13 Menú Eliminar del LectorBMU

BUSCAR. Menú que permite buscar un carácter o cadena de caracteres dentro del texto, si el texto buscado no se encuentra indicará *no encontrado* y si lo encuentra se posicionará donde este el primer resultado de búsqueda, como se mostrará más adelante en el menú ayuda una vez realizada una búsqueda la tecla * (asterisco) toma la función buscar siguiente, es decir encuentra las sucesivas coincidencias de búsqueda encontradas a lo largo del texto.

Para llevar a cabo una búsqueda se debe presionar el menú principal y seleccionar el menú Buscar, se escribe la palabra a buscar y se presiona buscar, en pantalla mostrará **NO ENCONTRADO** o la palabra encontrada según sea el caso como se muestra a continuación.

VI. DESARROLLO DE APLICACIÓN PARA PDA Y DISPOSITIVOS MÓVILES COMPATIBLES



Figura 6.14 Menú Buscar del LectorBMU

CONFIGURACIONES Permite personalizar el menú en el tamaño de la letra, el nivel de interlineado, el tipo de desplazamiento, vertical u horizontal y el color del fondo, al seleccionar este en negro la letra es blanca y viceversa. Esta configuración se guarda en el dispositivo móvil y se conserva para las siguientes consultas permitiendo modificarse en cualquier momento. Para llevar a cabo una personalización del boletín se presiona el menú configuraciones y una vez que accesa al mismo están las siguientes opciones:

Fuente: pequeña, mediana, grande.

Interlineado: 0,1,2,3,4

Desplazamiento: Horizontal, Vertical

Fondo: Blanco, Negro

Estas opciones se pueden modificar, si se presiona guardar, la configuración se muestra en pantalla modificando el boletín. Como se muestra a continuación.



Figura 6.15 Menú Configuraciones del LectorBMU (1)



Figura 6.16 Menú Configuraciones del LectorBMU (2)

AYUDA Al seleccionar el menú ayuda aparece en la pantalla la función que tienen predeterminada algunas teclas del dispositivo permitiendo esto un mejor desplazamiento en el texto, búsquedas y marcadores.

Teclas de control:

ABAJO Avanza página

DERECHA Adelanta línea

IZQUIERDA Regresa línea

ARRIBA Regresa página

(ASTERISCO) Buscar siguiente (dentro de buscar) o menú principal

(NÚMERO) Muestra el menú principal

0 (CERO) Despliega la lista de marcadores

1-9 Avance de 1-9 líneas según el número presionado.

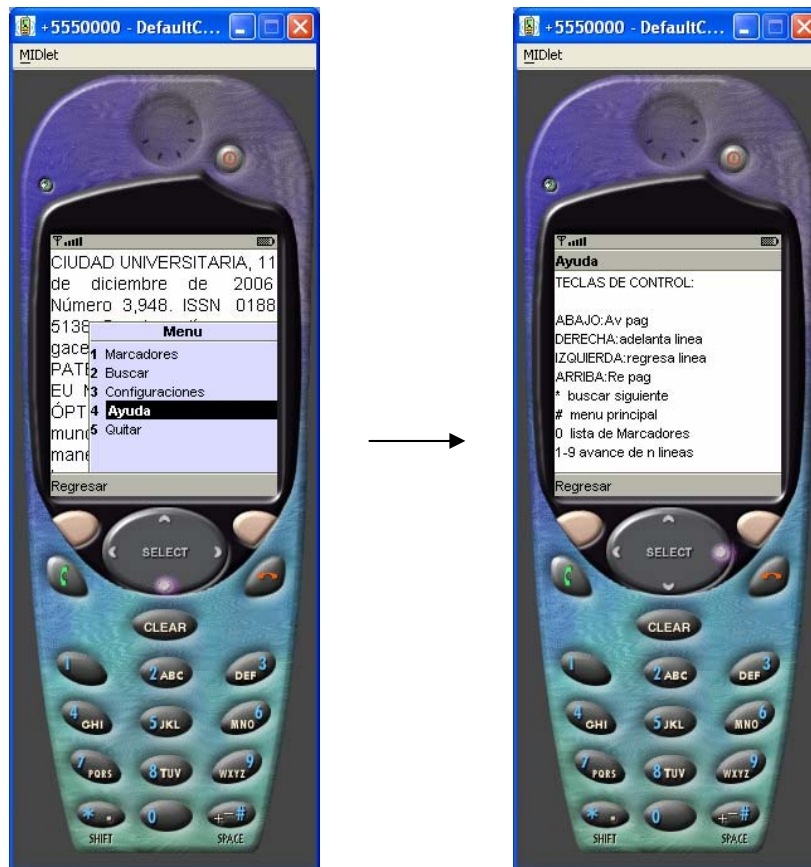


Figura 6.17 Menú Ayuda del LectorBMU

QUITAR Al seleccionar el menú quitar se termina la ejecución del MIDLET y se liberan los recursos del sistema.



Figura 6.18 Menú Quitar del LectorBMU

3. Resultados del plan de pruebas

Las pruebas realizadas para el proyecto fueron satisfactorias; a continuación se listan los procesos realizados para las pruebas.

Perfil MIDP 1.0:

- Se compiló el proyecto en J2ME Wireless Toolkit 1.0.4_01 y se ejecutó en un emulador propio de la herramienta con éxito.



Figura 6.19 Ejecución de LectorBMU en Wireless Toolkit 1.0.4_01

- Se compiló el proyecto en J2ME Wireless Toolkit 1.0.4_01 se convirtió a PRC (Aplicación nativa de PalmOS) por medio del PRC converter tool, utilidad de la misma herramienta y se ejecutó en el emulador PDA propio de la herramienta con éxito.

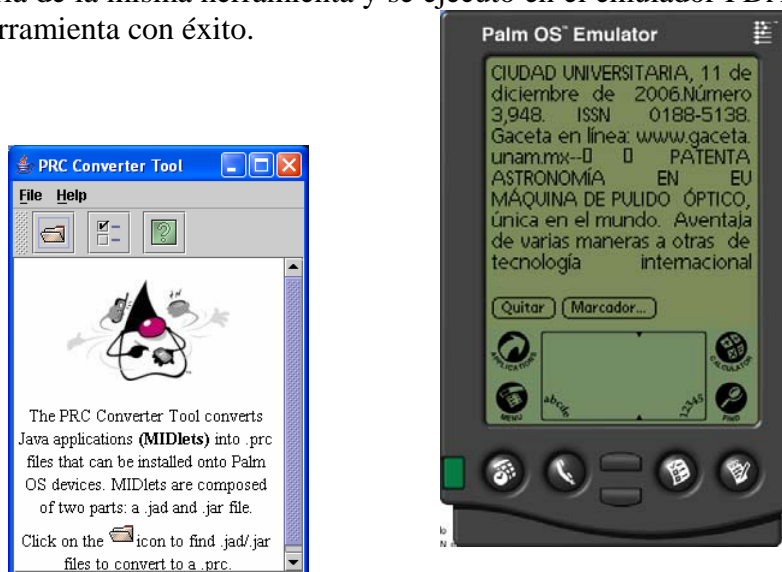


Figura 6.20 Ejecución de LectorBMU en POSE

- La versión PRC, se transfirió por medio de Bluetooth a un dispositivo físico (PDA Sony Clié NZ-90) que tiene implementado el perfil MIDP 1.0 de forma exitosa.

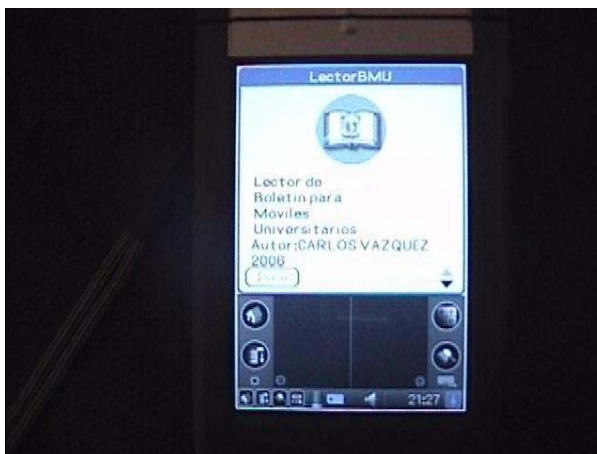


Figura 6.21 Ejecución de LectorBMU en PDA MIDP 1.0

- Se ejecutó la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 1.0.4_01, al igual que en el dispositivo físico y se agregó, se renombró, eliminó y se posicionó en un marcador verificando su funcionamiento de forma exitosa.
- Se ejecutó la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 1.0.4_01, al igual que en el dispositivo físico y se buscó una palabra existente en el texto del boletín y una palabra inexistente Encontrando la primera y posicionándose en el lugar de esta y arrojando el texto “no encontrado”, respectivamente.
- Se ejecutó la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 1.0.4_01, al igual que en el dispositivo físico y se llevó a cabo la siguiente configuración comprobando que se respetara la misma al guardar los cambios:

*Fuente: grande
Interlineado: 2
Desplazamiento: Vertical
Fondo: Negro*



Figura 6.22 Ejecución de LectorBMU con diferente configuración

- Se ejecutó la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 1.0.4_01, al igual que en el dispositivo físico y se verificó que los comandos del texto de ayuda se ejecutan sin problema alguno.
- Se eliminó el MIDlet del dispositivo físico sin problema alguno.

Perfil MIDP 2.0:

- Se compiló la aplicación en J2ME Wireless Toolkit 2.0 y ejecutó en un emulador propio de la herramienta de forma exitosa.



Figura 6.23 Ejecución de LectorBMU en J2ME Wireless Toolkit 2.0

- Se compiló el proyecto en J2ME Wireless Toolkit 2.0 y se transfirió por medio de USB a un dispositivo físico (Celular Motorola c650) que tiene implementado el perfil MIDP 2.0 corroborando que la aplicación funciona.

Nota: para poder cargar aplicaciones java por medio de USB algunos teléfonos están bloqueados de fabrica, es decir, no tienen la opción para cargar aplicaciones java, por lo que se tiene que llevar a cabo una configuración por medio de software de terceros, configuración que sale del tema de este trabajo, por lo cual no se revisará.

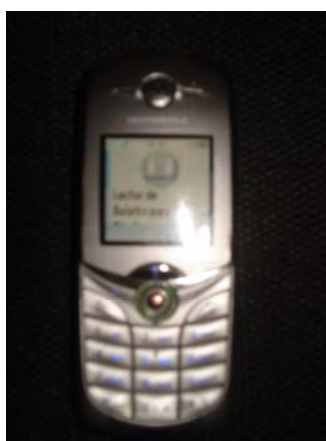


Figura 6.24 Ejecución de LectorBMU en C650 MIDP 2.0

- Se compiló el proyecto en J2ME Wireless Toolkit 2.0 y se transfirió por medio de Bluetooth a un dispositivo físico (Celular Motorola U6) que tiene implementado el perfil MIDP 2.0 corroborando que la aplicación funciona.
- Se ejecutó la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 2.0, al igual que en el dispositivo físico y se agregó, se renombró, eliminó y se posicionó en un marcador verificando su funcionamiento de forma exitosa.
- Se ejecutó la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 2.0, al igual que en el dispositivo físico y se buscó una palabra existente en el texto del boletín y una palabra inexistente Encontrando la primera y posicionándose en el lugar de esta y arrojándonos el texto “no encontrado”, respectivamente.
- Se ejecutó la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 2.0, al igual que en el dispositivo físico y se llevó a cabo la siguiente configuración comprobando que se respetara la misma al guardar los cambios:

*Fuente: mediana.
Interlineado: 0
Desplazamiento: Horizontal
Fondo: Blanco*



Figura 6.25 Ejecución de LectorBMU con otra configuración

- Se ejecutó la aplicación en el emulador propio de la herramienta J2ME Wireless Toolkit 2.0, al igual que en el dispositivo físico y se verificó que los comandos del texto de ayuda se ejecutan sin problema alguno.
- Se eliminó el MIDlet del dispositivo físico sin problema alguno.

4. Plan de liberación

Para el plan de liberación lo que se realiza es instalar las herramientas que se ocuparon para compilar la aplicación y empaquetarla, se provee al equipo servidor de conectividad USB y Bluetooth previa capacitación de la persona encargada de proveer la aplicación al usuario que la solicite.

CONCLUSIONES

Hace algún tiempo asistí a un centro de convenciones a una exposición de tecnología, en los lugares comunes había varios módulos distribuidos, estos módulos eran en forma de poste, parecido a una cabina telefónica, en los cuales solo bastaba con alinear el puerto IrDA de un dispositivo móvil PDA con el del módulo y descargaba de forma automática el mapa de ubicación del centro de convenciones al igual que información general de las exposiciones y fechas. Esta es la idea que me impulso a proponer este proyecto de tesis adecuado a las necesidades universitarias, ya que con una infraestructura básica se puede implementar como medio de difusión.

Actualmente ya existe una red inalámbrica universitaria la cual paulatinamente será extendida a todos los campus de la UNAM¹ interconectándolos a todos, esta red será la infraestructura de muchos servicios que puedan circular por aquí. Estos servicios serán capaces de ejecutarse desde dispositivos móviles tales como celulares o PDA's con interfaces gráficas amigables al usuario, gradualmente dejarán de llevarse a cabo trámites personales y de la misma manera podrán hacerse desde cualquier parte del campus ya sea en un área común, en la biblioteca o en algún centro de cómputo.

Sirva el presente proyecto como pauta de esta idea que actualmente se usa en muchos centros de convenciones o para muchos portales de internet en versiones móviles de sus sitios o envío de noticias o marketing, pero implementado para crear un medio alternativo de difusión de información que no requiera de un equipo de cómputo para procesarla, sino de un dispositivo común como lo es el teléfono celular.

Concluyo que J2ME es la mejor opción si se trata de desarrollar alguna aplicación para muy diferentes dispositivos que quizá se desconocen, con el único requisito de que pueda ejecutar aplicaciones java como lo es común en los actuales teléfonos celulares, sin importar la plataforma sobre la cual trabaja el móvil.

La aplicación LectorBMU es un ejemplo demostrativo real de lo que podría ser un gran proyecto el cual requeriría mayor desarrollo mejor presentación y podría implementarse desde internet es decir que por medio de la red inalámbrica el estudiante se conecte con su usuario y contraseña a un portal que permita descargar aplicaciones móviles las cuales ejecuten diferentes servicios escolares o difundan diferentes tipos de publicaciones, como pueden ser calificaciones oficiales, calendarios de cursos y de clases, gaceta UNAM, libros electrónicos de carácter científico o cultural, noticias, etc.

En un futuro cercano deberá ser posible entrar a un campus y encontrar algún módulo de información en el cual de forma automática podamos descargar estas

¹ La Red Inalámbrica Universitaria (RIU), que permite el acceso a Internet desde distintas áreas de la Ciudad Universitaria a través de dispositivos móviles. <https://www.riu.unam.mx/>

aplicaciones con el simple hecho de acercar nuestro teléfono celular o agenda electrónica.

Concluyo también que la actual comunidad universitaria de la FES Aragón esta lista para usar estos servicios pues la mayoría tiene un dispositivo móvil compatible con este tipo de aplicaciones y les gustaría recibir información de carácter académico por este medio.²

El desarrollo de la aplicación LectorBMU me sirvió para aprender otro uso sobre los teléfonos celulares, que el común.

Esto me permitió adentrarme en el lenguaje de programación J2ME que hasta antes del proyecto apenas tenía nociones del mismo, despertando el interés de futuros proyectos en el área de teléfonos móviles

Es para mí una satisfacción personal el haber concluido un proyecto de esta magnitud cumpliendo con el plan de pruebas.

Otro aprendizaje que obtuve durante todo el proceso fue el desarrollo de proyectos de forma metódica empleando los principios de la metodología RUP (Rational Unified Process – Proceso Unificado Racional).

El desarrollo del proyecto me demandó bastante persistencia y constancia debido a la necesidad de transferir la aplicación a diferentes dispositivos reales con características diferentes.

La aplicación LectorBMU cumple puntualmente con el proyecto sugerido, ya que todas las pruebas realizadas fueron exitosas, incluyendo dispositivos móviles reales en los cuales se ejecuto sin problema alguno cumpliendo con sus funciones de lector de Boletín para Móviles Universitarios.

² Según resultados de encuestas de campo realizadas en FES Aragón detalladas en el Anexo E de este trabajo.

ANEXOS

Anexo A El diagrama del funcionamiento de la aplicación LectorBMU.

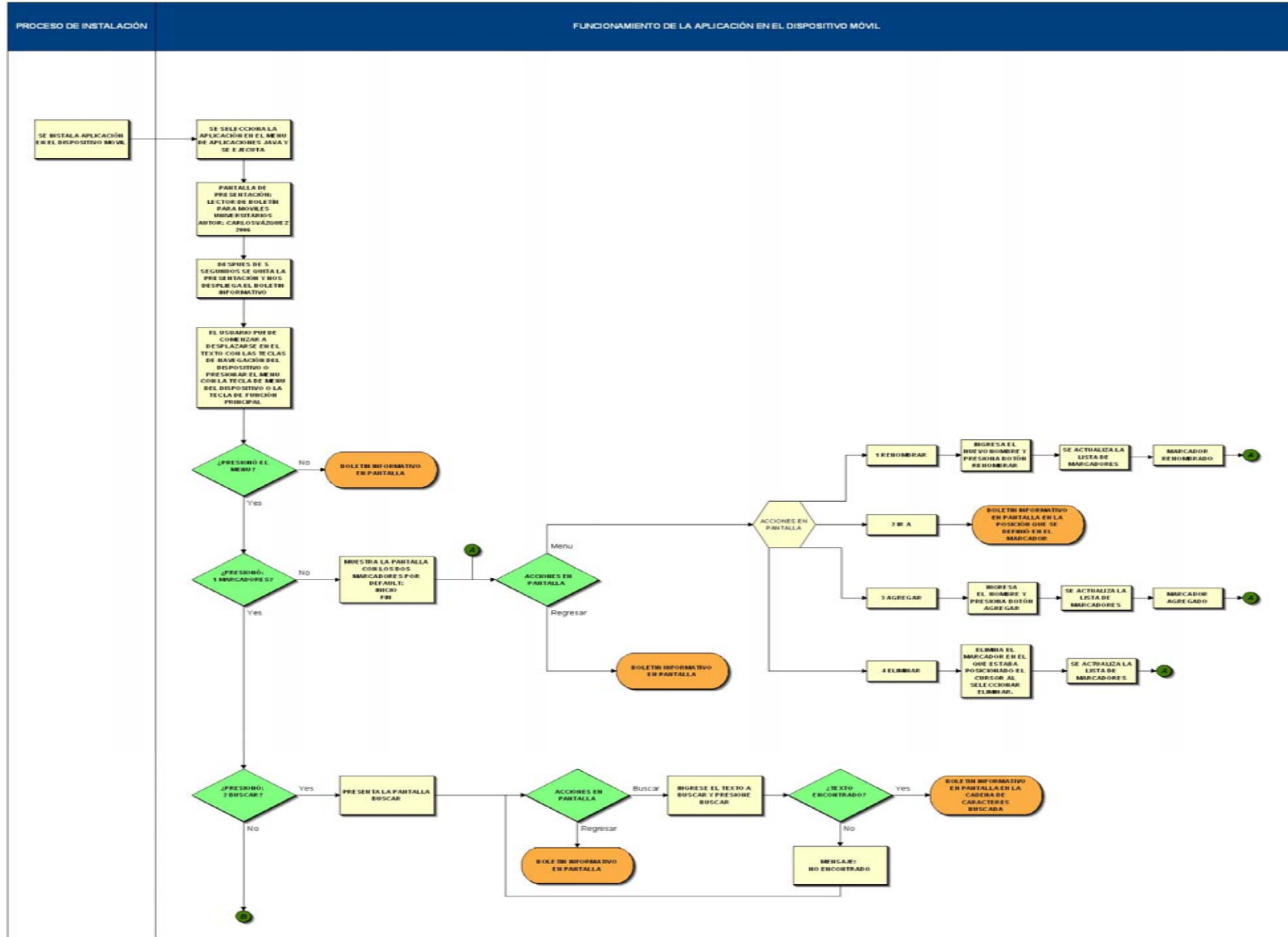
Anexo B Código fuente de la aplicación LectorBMU.

Anexo C J2ME arquitectura seleccionada para el desarrollo de la aplicación lectorBMU

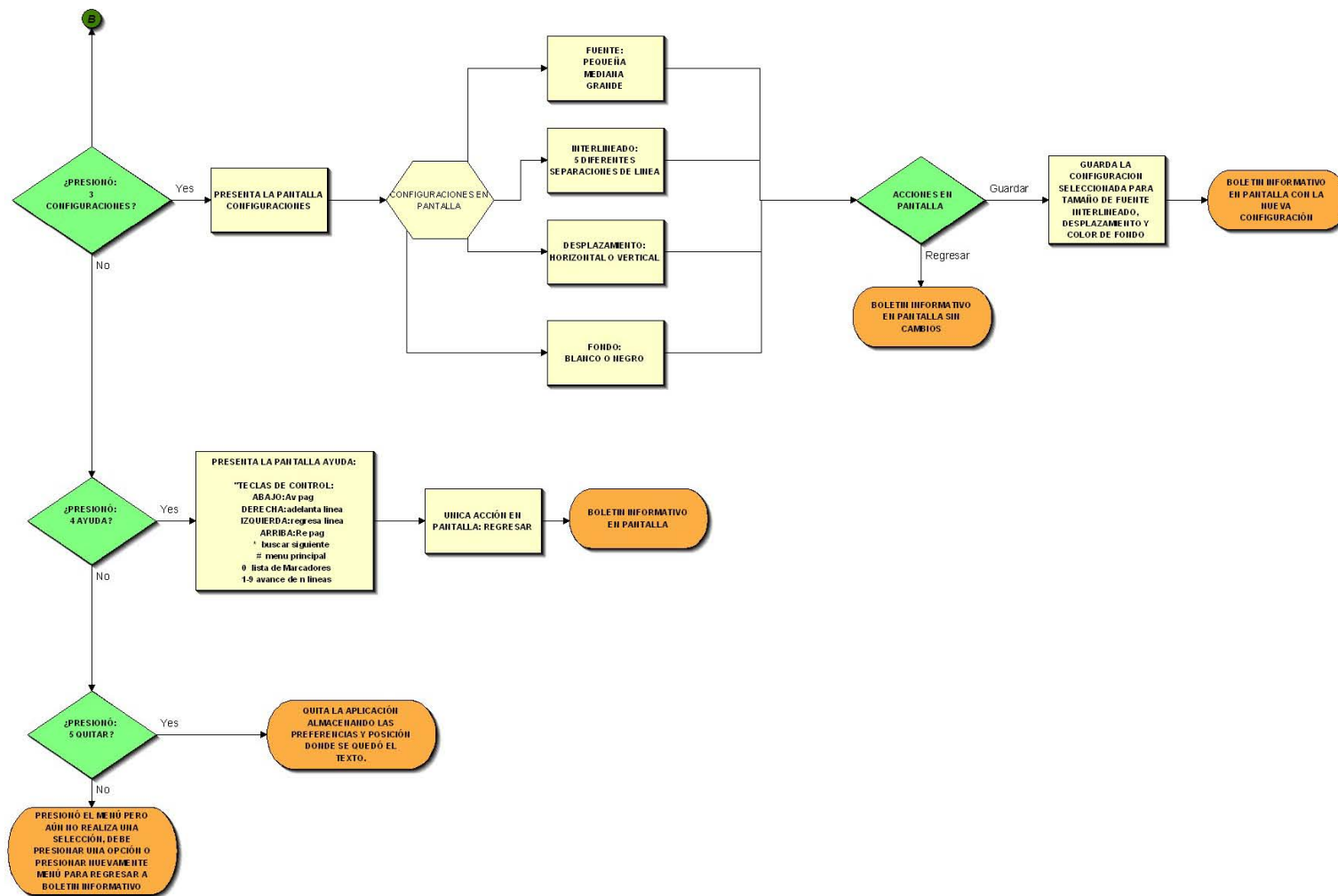
Anexo D Diccionario de términos.

Anexo E. Encuesta de campo en FES Aragón en la cual se confirma lo común que es el uso de dispositivos compatibles con la aplicación.

ANEXO A. DIAGRAMA DE FUNCIONAMIENTO DEL LECTOR BMU



REALIZADO CON IGRAF X PROCESS FOR SIX SIGMA 8.2.4.255



ANEXO B CODIGO FUENTE LECTOR BMU

```

//
// LectorBMU.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creación: 2006,2007
//-----
/*
 * Esta es la clase que implementa el midlet que controla la ejecución de
 * la aplicación en el dispositivo móvil.
 */

//Importamos los paquetes que requiere el midlet para su funcionamiento
import javax.microedition.lcdui.*;//interfaz de usuario
import javax.microedition.midlet.*;//se ocupa del ciclo de vida de la aplicación
import java.util.*;//clases de utilidades
import java.io.*;//Manejo de las I/O entradas salidas.
import javax.microedition.rms.*;//sistema de mantenimiento de registros (Record
Management System) usado para guardar información

public class LectorBMU extends MIDlet
{ // Variables correspondientes a los comandos que se utilizan en la aplicación

    static final Command REGRESA_CMD = new Command(Localizacion.BACK,
Command.BACK, 1);
    static final Command QUITAR_CMD = new Command(Localizacion.QUIT,
Command.CANCEL, 2);
    static final Command MARCADORES_CMD = new
Command(Localizacion.Marcadores, Command.SCREEN, 1);
    static final Command IR_A_CMD = new Command(Localizacion.GOTO,
Command.SCREEN, 1);
    static final Command AGREGAR_CMD = new Command(Localizacion.ADD,
Command.SCREEN, 2);
    static final Command RENOMBRAR_CMD = new
Command(Localizacion.RENAME, Command.SCREEN, 1);
    static final Command BORRAR_CMD = new
Command(Localizacion.REMOVE, Command.SCREEN, 3);
    static final Command BUSCAR_CMD = new Command(Localizacion.busca,
Command.SCREEN, 2);
    static final Command CONFIGURA_CMD = new
Command(Localizacion.SETTINGS, Command.SCREEN, 3);
    static final Command GUARDAR_CMD = new Command(Localizacion.SAVE,
Command.SCREEN, 1);

```

```
static final Command AYUDA_CMD = new Command(Localizacion.HELP,
Command.SCREEN, 4);
```

```
// Variable correspondiente a la entrada a leer como fuente de información
static final String BOLETIN = "/gaceta.txt";
```

```
//Define la clase que almacena las preferencias
static final String NOMBRE_ALMACEN =
"BOLETIN_MOVILES_UNIVERSITARIOS";
```

```
//Define el interlineado a ocupar por default
static final int DEFAULT_INTERLINEA = 1;
```

```
//Define el Tamaño inicial del buffer
static final int size_BUF = 16*1024;
```

```
// Definimos la variable de la pantalla del dispositivo
private Display display;
```

```
//Método que define la acción de insertar un nuevo marcador en la posición
actual
```

```
private int insertaMarcador(Marcador bm) {
    int pos = bm.pos;
    int l = 0, n = Marcadores.size(), r = n;
    while (l < r) {
        int m = (l + r) >> 1;
        if (pos > ((Marcador)Marcadores.elementAt(m)).pos) {
            l = m + 1;
        } else {
            r = m;
        }
    }
    Marcadores.insertElementAt(bm, r);
    return r;
}
```

```
//Método que define la acción de buscar una cadena de caracteres y posicionarnos
en el lugar donde lo encuentra
```

```
public int busca(int fromPos, String muestra) {
    char[] p = muestra.toLowerCase().toCharArray();
    int n = p.length;
    char[] q = new char[n];
    char ch, prev = '\0';
    int i = 0;
```

```

busca:
while ((ch = obtenCaracter(fromPos++)) != '\0') {
    if (ch == ' ' && prev == ' ') {
        continue;
    }
    prev = ch;
    if (i == n) {
        i = 0;
    }
    q[i++] = Character.toLowerCase(ch);
    int j = 0;
    for (int k = i; k < n; k++, j++) {
        if (p[j] != q[k]) {
            continue busca;
        }
    }
    for (int k = 0; j < n; k++, j++) {
        if (p[j] != q[k]) {
            continue busca;
        }
    }
    return fromPos - n;
}
return -1;
}

```

//Método que define la acción de almacenar un nuevo marcador que se ha insertado

```

public int agregaMarcador(Marcador bm) {
    byte[] record = bm.pack();
    try {
        bm.id = store.addRecord(record, 0, record.length);
    } catch (Exception x) {}

    return insertaMarcador(bm);
}

```

//Este método nos permite actualizar el nombre del marcador actual y volverlo a almacenar

```

public void actualizaMarcador(Marcador bm) {
    byte[] record = bm.pack();
    try {
        store.setRecord(bm.id, record, 0, record.length);
    } catch (Exception x) {}
}

```

```

//Elimina el marcador seleccionado de la clase que almacena la información
public void borraMarcador(int i) {
    try {
        store.deleteRecord(((Marcador)Marcadores.elementAt(i)).id);
    } catch (Exception x) {}
    Marcadores.removeElementAt(i);
}

//Nos permite abrir en posteriores ocasiones la aplicación en la posición donde
dejamos el texto.
public void guardaPosicionActual() {
    byte[] record = new byte[SIZE_REGISTRO];
    record[INTERLINEA_OFFS] = (byte)overlap;
    record[FONT_OFFS] = (byte)font;
    Convertidor.packInt(record, POSICION_OFFS, currPagePos);
    try {
        store.setRecord(1, record, 0, record.length);
    } catch (Exception x) {}
}

//Crea la zona de almacenamiento si no existía antes
private void abreAlmacena() {
    try {
        store = RecordStore.openRecordStore(NOMBRE_ALMACEN, true);
        int last = store.getNextRecordID();
        if (last > 1) {
            byte[] record = store.getRecord(1);
            font = record[FONT_OFFS];
            overlap = record[INTERLINEA_OFFS];
            currPagePos = Convertidor.unpackInt(record, POSICION_OFFS);
            for (int i = 2; i < last; i++) {
                try {
                    insertaMarcador(new Marcador(i, store.getRecord(i)));
                } catch (InvalidRecordIDException x) {}
            }
        } else {
            byte[] record = new byte[SIZE_REGISTRO];
            overlap = DEFAULT_INTERLINEA;
            font = FormularioConfig.SMALL_FONT;
            record[INTERLINEA_OFFS] = (byte)overlap;
            record[FONT_OFFS] = (byte)font;
            store.addRecord(record, 0, record.length);
            agregaMarcador(new Marcador(0, Localizacion.BEGIN));
        }
    }
}

```

```

        DataInputStream content = new
DataInputStream(this.getClass().getResourceAsStream("/content"));
        if (content != null) {
            while (true) {
                int pos = content.readInt();
                int len = content.readByte() & 0xFF;
                byte[] nameBytes = new byte[len];
                content.readFully(nameBytes);
                char[] nameChars = new char[len];
                for (int i = 0; i < len; i++) {
                    nameChars[i] = Localizacion.obtenCaracter(nameBytes[i]);
                }
                agregaMarcador(new Marcador(pos, new String(nameChars)));
            }
        }
    } catch (Exception x) {}

    if (longArchivo != 0) {
        agregaMarcador(new Marcador(Integer.MAX_VALUE,
Localizacion.END));
    }
}
} catch(Exception x) {
    x.printStackTrace();
}
}
// Cierra las zonas de almacenamiento
private void cierraAlmacena() {
    if (store != null) {
        try {
            store.closeRecordStore();
            store = null;
        } catch (Exception x) {}
    }
}
// Constructor del midlet, en donde se obtiene el tamaño
//y la longitud del archivo font
public LectorBMU() {
    String sizeBoletin = getAppProperty("size.Boletin");
    if (sizeBoletin != null) {
        longArchivo = Integer.parseInt(sizeBoletin);
    } else {
        try {
            DataInputStream cfg = new
DataInputStream(this.getClass().getResourceAsStream("/config"));
            longArchivo = cfg.readInt();
        } catch (IOException e) {

```



```

        e.printStackTrace();
    }
}

buf = new byte[size_BUF];
regresa ();
llenaBuffer();
Marcadores = new Vector();
abreAlmacena();
}
//Método que define la acción de llenar el buffer
private void llenaBuffer() {
    int offs = 0;
    try {
        while (offs < buf.length) {
            int rc = in.read(buf, offs, buf.length - offs);
            if (rc <= 0) {
                break;
            }
            offs += rc;
        }
    } catch (IOException x) {}

    if (offs >= 0 && offs < size_BUF) {
        buf[offs] = '\0';
    }
}
//Método que define la acción de mover a la posición de búsqueda encontrada.
private boolean mover_a(int pos) {

    int offs = pos <= size_BUF/2 ? 0 : pos - size_BUF/2;
    if (pos < bufPos) {
        regresa();
    } else {
        bufPos += size_BUF;
        offs -= bufPos;
    }
    if (offs > 0) {
        long skipped = 0;
        try {
            skipped = in.skip(offs);
        } catch (IOException x) {}

        if (skipped < offs) {

            return false;
        }
    }
}

```

```

        bufPos += offs;
    }
    llenaBuffer();
    return true;
}
// Regresa una entrada para leer el recurso especificado
private void regresa() {
    bufPos = 0;
    if (in != null) {
        try {
            in.close();
        } catch (IOException x) {}
    }
    in = getClass().getResourceAsStream(BOLETIN);
}

// Regresa la posición seleccionada obteniendo el caracter en el buffer
public char obtenCaracter(int pos) {
    if (pos < bufPos) {
        if (!mover_a(pos)) {
            return '\0';
        }
    } else if (pos >= bufPos + size_BUF) {
        if (pos == bufPos + size_BUF) {
            if (!mueveAdelante()) {
                return '\0';
            }
        } else {
            if (!mover_a(pos)) {
                return '\0';
            }
        }
    }
    return Localizacion.obtenCaracter(buf[pos - bufPos]);
}

//Modifica la posición del texto en base al buffer hacia adelante del archivo
fuente
private boolean mueveAdelante() {
    System.arraycopy(buf, size_BUF/2, buf, 0, size_BUF/2);
    bufPos += size_BUF/2;
    int offs = 0;
    try {
        while (offs < size_BUF/2) {
            int rc = in.read(buf, size_BUF/2 + offs, size_BUF/2 - offs);
            if (rc <= 0) {
                if (offs == 0) {
                    return false;
                }
            }
        }
    }
}

```

```

        }
        break;
    }
    offs += rc;
}
} catch (IOException x) {}

if (offs < size_BUF/2) {
    buf[size_BUF/2 + offs] = 0;
}
return true;
}

//Método que define el error al no localizar el texto buscado
public void error(String message, Displayable parent) {
    Alert a = new Alert(Localizacion.ERROR, message, null, AlertType.ERROR);
    a.setTimeout(Alert.FOREVER);
    Display.getDisplay(this).setCurrent(a, parent);
}

// Detiene el midlet y libera todos los recursos
protected void destroyApp(boolean unconditional) {
    // Guarda la posición actual del texto
    guardaPosicionActual();
    // Cierra la zona de almacenamiento
    cierraAlmacena();
}

// Se suspenden todas las acciones que realiza el midlet
protected void pauseApp() {
    // Guarda la posición actual del texto
    guardaPosicionActual();
}

// Arrancamos la aplicación
protected void startApp()
{
    // Recuperamos el display del objeto estático del Sistema
    display = Display.getDisplay( this);
    // Si es la primera vez que ejecutamos el midlet, presentamos
    // la pantalla del logotipo
    Displayable d = display.getCurrent();
    if( d == null ) {
        try {
            Alert intro = new Alert( null,
                " Lector de\n Boletín para\n Móviles\n Universitarios\nAutor:CARLOS
VÁZQUEZ 2007\n\n",
                Image.createImage("/lector.png"),AlertType.INFO );

```

```

        intro.setTimeout(5000);
        // La dejamos que se vea unos segundos, e indicamos que la
        // transición es a la pantalla de datos
        display.setCurrent( intro,new LectorCanvas(this));

    } catch( Exception e ) {
        // Presentamos la pantalla con el boletín
        display.setCurrent( new LectorCanvas(this) );
    }
}
else {
    // Presentamos la pantalla con el boletín
    display.setCurrent( new LectorCanvas (this) );

}

}
// Concluimos la ejecución del midlet, devolviendo todos los
// recursos acaparado al sistema
public void quit() {
    // Concluimos la ejecución del midlet
    destroyApp(true);
    // Notificamos a la plataforma su desaparición
    notifyDestroyed();
}

InputStream in;
//Se definen los parametros default en off
static final int POSICION_OFFS = 0;
static final int FONT_OFFS = 5;
static final int INTERLINEA_OFFS = 6;
static final int SIZE_REGISTRO = 8;
//Definición de variables
Vector Marcadores;
RecordStore store;
int bufPos;
int currPagePos;
int longArchivo;
int overlap;
int font;
int flags;
byte [] buf;

}
//----- Final del fichero principal LectorBMU.java

```

```

//
// LectorCanvas.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa la creación y presentación de las distintas
 * capas que intervienen en el midlet.
 * La clase también controla el desplazamiento de los elementos de las
 * capas de forma individual,*/

import javax.microedition.lcdui.*;
import java.util.*;

public class LectorCanvas extends Canvas implements CommandListener {
    //Declaración de variables
    LectorBMU lector;
    int pantallaAncho;
    int pantallaAlto;
    int alturaFuente;
    int lineasPantalla;
    int nLineas;
    int[] lineaPos;
    char[] lineaPrevChar;
    char prevPagChar;
    InterfaceFuente font;
    Image imgBuf;
    int anchoEspacio;
    int anchoGuion;
    String buscamuestra;
    int buscaPos;
    String marcadorDefault;
    boolean inicializado;
    repiteTarea repiteTarea;

    static final int WHITE = 0xffffffff;
    static final int BLACK = 0;
    static final int SLIDER_COLOR = 0x8080ff;
    static final int SLIDER_POS_COLOR = 0xff0000;
    static final int REPEAT_DELAY = 1000;
    static final int REPEAT_PERIOD = 100;
    static final int MIN_SLIDER_LENGTH = 3;
    static final int MAX_Bookmark_LENGTH = 16;
    static final int INDENTATION = 3;
    //Método que define la accion de repetir una acción al presionar una tecla
    class repiteTarea extends TimerTask {

```

```

    repiteTarea(int keyCode) {
        this.keyCode = keyCode;
    }

    public void run() {
        keyAction(keyCode);
    }

    int keyCode;
}
//Establece el modo de pantalla completa
LectorCanvas(LectorBMU lector) {
    setFullScreenMode (true);
    this.lector = lector;
    prevPagChar = '\n';
    buscaPos = -1;
    inicializado = false;

}
//Establece el menu que permite modificar las preferencias de tamaño de fuente y
propiedades
void resetFont()
{
    switch (lector.font) {
        case FormularioConfig.SMALL_FONT:
            font = new
SistemaDeFuentes(Font.getFont(Font.FACE_PROPORTIONAL,
Font.STYLE_PLAIN, Font.SIZE_SMALL));
            break;
        case FormularioConfig.MEDIUM_FONT:
            font = new
SistemaDeFuentes(Font.getFont(Font.FACE_PROPORTIONAL,
Font.STYLE_PLAIN, Font.SIZE_MEDIUM));
            break;
        case FormularioConfig.LARGE_FONT:
            font = new
SistemaDeFuentes(Font.getFont(Font.FACE_PROPORTIONAL,
Font.STYLE_PLAIN, Font.SIZE_LARGE));
            break;
        default:
            font = new ImagenFuente("/fonts/" +
FormularioConfig.nombreFuente[lector.font]);
    }
    alturaFuente = font.getHeight() - lector.overlap;
    anchoEspacio = font.charWidth(' ');
    anchoGuion = font.charWidth(Localizacion.HYPHEN);
}

```

```

pantallaAlto = getHeight();
if ((lector.flags & FormularioConfig.SLIDER_HORIZONTAL) != 0) {
    pantallaAlto -= 1;
}
pantallaAncho = getWidth();
if ((lector.flags & FormularioConfig.SLIDER_VERTICAL) != 0) {
    pantallaAncho -= 1;
}
lineasPantalla = pantallaAlto/alturaFuente;
lineaPos = new int[lineasPantalla+1];
lineaPrevChar = new char[lineasPantalla+1];
inicializado = true;

//Agrega los comandos del menú principal
addCommand(LectorBMU.MARCADORES_CMD);
addCommand(LectorBMU.BUSCAR_CMD);
addCommand(LectorBMU.CONFIGURA_CMD);
addCommand(LectorBMU.AYUDA_CMD);
addCommand(LectorBMU.QUITAR_CMD);
setCommandListener(this);
}
// Método de control de los comandos de la pantalla
public void commandAction(Command c, Displayable d)
{
    if (c == LectorBMU.MARCADORES_CMD) {
        new ListaMarcadores(this, this);
    } else if (c == LectorBMU.BUSCAR_CMD) {
        new FormularioBuscar(this, this);
    } else if (c == LectorBMU.CONFIGURA_CMD) {
        new FormularioConfig(this, this);
    } else if (c == LectorBMU.AYUDA_CMD) {
        new FormularioAyuda(this, this);
    } else if (c == LectorBMU.QUITAR_CMD) {
        lector.quit();
    }
}
//Define los espacios entre las palabras del texto de origen
static void insertSpaces(StringBuffer buf, int nSpaces) {
    if (nSpaces > 0) {
        int i = 0;
        int n = buf.length();
        while (i < n && buf.charAt(i) == ' ') {
            i += 1;
        }
        int nSeparators = 0;
        while (++i < n) {

```

```

        if (buf.charAt(i) == ' ') {
            nSeparators += 1;
        }
    }
    if (nSeparators != 0) {
        int ns1 = nSpaces / nSeparators;
        int ns2 = nSpaces % nSeparators;
        do {
            if (buf.charAt(--i) == ' ') {
                for (int j = 0; j < ns1; j++) {
                    buf.insert(i, ' ');
                }
                if (ns2 != 0) {
                    buf.insert(i, ' ');
                    ns2 -= 1;
                }
                nSeparators -= 1;
            }
        } while (nSeparators != 0);
    }
}

//Define los caracteres que generaran un espacio entre las palabras
private static boolean isSeparator(char ch) {
    return ch == ' ' || ch == '!' || ch == '.' || ch == ',' || ch == ';' || ch == '?'
        || ch == '-' || ch == '+' || ch == '=' || ch == ')' || ch == '}' || ch == ']'
        || ch == '\\' || ch == '/' || ch == ':' || ch == '>';
}

//divide las palabras e incluye un gui3n de acuerdo al ancho de pantalla del
dispositivo
int splitWord(StringBuffer buf, int n, int width) {
    char c1 = buf.charAt(n-1);
    char c2 = buf.charAt(n-2);
    if (isSeparator(c2)) {
        if (c2 == ' ') {
            buf.delete(n-2, buf.length());
            width -= anchoEspacio + font.charWidth(c1);
        } else {
            buf.delete(n-1, buf.length());
            width -= font.charWidth(c1);
        }
    }
    insertSpaces(buf, (pantallaAncho - width)/anchoEspacio);
    return 1;
}

int i = Localizacion.split(buf, n);
if (i < 0) {
    for (i = n-2; i > 0 && !isSeparator(buf.charAt(i-1)); i--);
}

```



```

    }
    if (i <= 0) {
        i = n-2;
    }
    int rest = n-i;
    boolean insertaGuion = true;
    if (i >= 1 && isSeparator(buf.charAt(i-1))) {
        i -= 1;
        insertaGuion = false;
    }
    for (int j = i; j < n; j++) {
        width -= font.charWidth(buf.charAt(j));
    }
    buf.delete(i, buf.length());
    if (insertaGuion) {
        buf.insert(i, Localizacion.HYPHEN);
        width += anchoGuion;
    }
    insertSpaces(buf, (pantallaAncho - width)/anchoEspacio);
    return rest;
}
// Método que se encarga de todo el proceso de visualización de
// los elementos de la pantalla y de presentar información incluyendo
//cuando se modifica la preferencia del fondo a blanco o negro
protected synchronized void paint(Graphics g) {
    if (!inicializado) {
        resetFont();
    }

    g.setColor((lector.flags & FormularioConfig.INVERSE) != 0 ? BLACK :
WHITE);
    g.fillRect(g.getClipX(), g.getClipY(), g.getClipWidth(), g.getClipHeight());
    g.setColor((lector.flags & FormularioConfig.INVERSE) != 0 ? WHITE :
BLACK);
    int pos = lector.currPagePos;
    char prevChar = prevPagChar;
    boolean eof = false;
    nLineas = 0;
    marcadorDefault = null;
    for (int i = 0; !eof && i < lineasPantalla; i++) {
        lineaPos[i] = pos;
        lineaPrevChar[i] = prevChar;
        nLineas += 1;
        int width = 0;
        StringBuffer buf = new StringBuffer();
        do {
            char ch = lector.obtenCaracter(pos++);

```

```

if (ch == ' ') {
    if (prevChar == '\n') {
        if (buf.length() < INDENTATION) {
            buf.append(ch);
            width += anchoEspacio;
        }
        continue;
    } else if (prevChar == ' ' || buf.length() == 0) {
        continue;
    }
} else if (ch == '\0') {
    eof = true;
    break;
} else if (ch == '\n') {
    ch = lector.obtenCaracter(pos);
    if (ch == ' ' || ch == '\n') {
        prevChar = '\n';
        break;
    }
    ch = ' ';
}
buf.append(ch);
width += font.charWidth(ch);
prevChar = ch;
} while (width <= pantallaAncho);

if (marcadorDefault == null && lineaPrevChar[i] == '\n') {
    String bm = buf.toString().trim();
    if (bm.length() > 0) {
        if (bm.length() > MAX_Bookmark_LENGTH) {
            int j = MAX_Bookmark_LENGTH;
            while (j > 0 && bm.charAt(j) != ' ') {
                j -= 1;
            }
            bm = bm.substring(0, j);
        }
        marcadorDefault = bm;
    }
}
if (width > pantallaAncho) {
    if (prevChar == ' ' || prevChar == '\n') {
        buf.deleteCharAt(buf.length()-1);
        width -= anchoEspacio;
        insertSpaces(buf, (pantallaAncho - width) / anchoEspacio);
    } else {
        int n = buf.length();
        if (!eof) {

```

```

        int p = pos;
        char ch;
        while ((ch = lector.obtenCaracter(p++)) > ' ') {
            buf.append(ch);
        }
    }
    pos -= splitWord(buf, n, width);
}
font.drawString(g, 0, i*alturaFuente, buf.toString());
}
lineaPos[nLineas] = pos;
lineaPrevChar[nLineas] = prevChar;

if (lector.longArchivo != 0) {
    if ((lector.flags & FormularioConfig.SLIDER_HORIZONTAL) != 0) {
        int sliderLength = (pos -
lector.currPagePos)*pantallaAncho/lector.longArchivo;
        if (sliderLength < MIN_SLIDER_LENGTH) {
            sliderLength = MIN_SLIDER_LENGTH;
        }
        int sliderPos = lector.currPagePos*pantallaAncho/lector.longArchivo;
        g.setColor(SLIDER_COLOR);
        g.drawLine (0, pantallaAlto, pantallaAncho, pantallaAlto);
        g.setColor(SLIDER_POS_COLOR);
        g.drawLine(sliderPos, pantallaAlto, sliderPos+sliderLength,
pantallaAlto);
    }
    if ((lector.flags & FormularioConfig.SLIDER_VERTICAL) != 0) {
        int sliderLength = (pos -
lector.currPagePos)*pantallaAlto/lector.longArchivo;
        if (sliderLength < MIN_SLIDER_LENGTH) {
            sliderLength = MIN_SLIDER_LENGTH;
        }
        int sliderPos = lector.currPagePos*pantallaAlto/lector.longArchivo;
        g.setColor(SLIDER_COLOR);
        g.drawLine(pantallaAncho, 0, pantallaAncho, pantallaAlto);
        g.setColor(SLIDER_POS_COLOR);
        g.drawLine(pantallaAncho, sliderPos, pantallaAncho,
sliderPos+sliderLength);
    }
}
}
//Define la navegación en el documento fuente hacia atras
void mueveAtras(int shift) {
    int pos = lector.currPagePos;
    int width = 0;

```

```

prevPagChar = '\n';
while (--pos >= 0 && shift > 0) {
    char ch = lector.obtenCaracter(pos);
    if (ch == '\n') {
        width = 0;
        shift -= 1;
        prevPagChar = '\n';
    } else {
        width += font.charWidth(ch);
        if (width > pantallaAncho) {
            width = 0;
            shift -= 1;
            prevPagChar = ' ';
        }
    }
}
lector.currPagePos = pos+1;
}
//Define la navegación en el documento fuente hacia adelante
void mueveAdelante(int shift) {
    if (nLineas == lineasPantalla) {
        if (shift > lineasPantalla) {
            shift = lineasPantalla;
        }
        lector.currPagePos = lineaPos[shift];
        prevPagChar = lineaPrevChar[shift];
    }
}

public void setPosition(int pos) {
    if (pos == Integer.MAX_VALUE) {
        lector.currPagePos = lector.longArchivo;
        mueveAtras(lineasPantalla/2);
    } else {
        lector.currPagePos = pos;
    }
}

//Repite la tarea que una tecla realiza mientras la tecla este presionada
public void keyPressed(int keyCode) {
    if (repiteTarea != null) {
        repiteTarea.cancel();
    }
    repiteTarea = new repiteTarea(keyCode);
}
//Libera la tecla de función al soltarla

```

```

public void keyReleased(int keyCode) {
    if (repiteTarea != null) {
        repiteTarea.cancel();
        repiteTarea = null;
    }
    keyAction(keyCode);
}
//Avance de lineas de 1 a 9 lineas segun la tecla seleccionada
private synchronized void keyAction(int keyCode)
{
    switch (keyCode)
    {
    case KEY_NUM1:
    case KEY_NUM2:
    case KEY_NUM3:
    case KEY_NUM4:
    case KEY_NUM5:
    case KEY_NUM6:
    case KEY_NUM7:
    case KEY_NUM8:
    case KEY_NUM9:
        mueveAdelante(keyCode - KEY_NUM0);
        break;
    case KEY_NUM0:
        new ListaMarcadores(this, this);
        return;
    case KEY_STAR:
        if (buscaPos >= 0) {
            buscaPos = lector.busca(buscaPos+buscamuestra.length(), buscamuestra);
            if (buscaPos >= 0) {
                lector.currPagePos = buscaPos;
                break;
            } else {
                lector.error(Localizacion.NOT_FOUND, this);
                return;
            }
        }
    }
}
//Define el movimiento del documento hacia atras, adelante
//por linea o por pagina.
case KEY_POUND:
    new Menu(this);
    return;
default:
    switch (getGameAction(keyCode)) {
    case LEFT:
        mueveAtras(1);
        break;

```

```
        case RIGHT:
            mueveAdelante(1);
            break;
        case UP:
            mueveAtras(lineasPantalla);
            break;
        case DOWN:
            mueveAdelante(lineasPantalla);
            break;
        default:
            return;
    }
}
repaint();
lector.guardaPosicionActual();
}
}

//----- Final del fichero LectorCanvas.java
```

```

//
// FormularioConfig.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa el formulario del menu configuraciones del midlet.
*/

import javax.microedition.lcdui.*;

public class FormularioConfig extends Form implements CommandListener {
    //Definición de variables
    Displayable parent;
    LectorCanvas canvas;
    ChoiceGroup font;
    ChoiceGroup slider;
    ChoiceGroup background;
    Gauge overlap;

    static final int MAX_OVERLAP = 4;

    static final int SMALL_FONT = 0;
    static final int MEDIUM_FONT = 1;
    static final int LARGE_FONT = 2;

    static final int SLIDER_HORIZONTAL = 2;
    static final int SLIDER_VERTICAL = 1;

    static final int INVERSE = 8;

    //Variable que define el tamaño de letra seleccionado
    static String[] nombreFuente = {
        "pequeña",
        "mediana",
        "grande",
    };

    //Variable que define el desplazamiento seleccionado
    static String[] tipoDesplazamiento = {
        Localizacion.HORIZONTAL,
        Localizacion.VERTICAL
    };

```

```

};
//Variable que define el fondo seleccionado
static String[] backgroundType = {
    Localizacion.WHITE,
    Localizacion.BLACK
};

// Constructor de la clase que se encarga de mostrar el menu para seleccionar el
tamaño de la letra
//el interlineado, el desplazamiento y el fondo
FormularioConfig(LectorCanvas canvas, Displayable parent) {
    super(Localizacion.SETTINGS);
    this.canvas = canvas;
    this.parent = parent;
    font = new ChoiceGroup(Localizacion.font,
        Choice.EXCLUSIVE,
        nombreFuente,
        null);
    font.setSelectedIndex(canvas.lector.font, true);

    slider = new ChoiceGroup(Localizacion.SLIDER,
        Choice.EXCLUSIVE,
        tipoDesplazamiento,
        null);
    slider.setSelectedFlags(new boolean[] {
        (canvas.lector.flags & SLIDER_HORIZONTAL) != 0,
        (canvas.lector.flags & SLIDER_VERTICAL) != 0
    });

    background = new ChoiceGroup(Localizacion.BACKGROUND,
        Choice.EXCLUSIVE,
        backgroundType,
        null);
    background.setSelectedIndex((canvas.lector.flags & INVERSE) == 0 ? 0 : 1,
true);

    overlap = new Gauge(Localizacion.OVERLAP, true, MAX_OVERLAP,
canvas.lector.overlap);
    append(font);
    append(overlap);
    append(slider);
    append(background);
    addCommand(LectorBMU.GUARDAR_CMD);
    addCommand(LectorBMU.REGRESA_CMD);
    setCommandListener(this);
    Display.getDisplay(canvas.lector).setCurrent(this);
}

```



```

// Método de control de los comandos de la pantalla
public void commandAction(Command c, Displayable d)
{
    if (c == LectorBMU.GUARDAR_CMD) {
        canvas.lector.font = font.getSelectedIndex();
        canvas.lector.overlap = overlap.getValue();
        canvas.lector.flags = 0;
        boolean[] flags = new boolean[tipoDesplazamiento.length];
        slider.getSelectedFlags(flags);
        if (flags[0]) {
            canvas.lector.flags |= SLIDER_HORIZONTAL;
        }
        if (flags[1]) {
            canvas.lector.flags |= SLIDER_VERTICAL;
        }
        flags = new boolean[backgroundType.length];
        background.getSelectedFlags(flags);
        if (flags[1] && canvas.lector.font <= LARGE_FONT) {
            canvas.lector.flags |= INVERSE;
        }
        canvas.inicializado = false;
    }
    Display.getDisplay(canvas.lector).setCurrent(parent);
}
}

//----- Final del fichero FormularioConfig.java

```

```

//
// Menu.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/*
 * Esta es la clase que implementa el menu principal del midlet
 */

import javax.microedition.lcdui.*;

public class Menu extends List implements CommandListener {
    LectorCanvas canvas;

    //Definición de las diferentes opciones del menú principal
    static String[] items = {Localizacion.Marcadores, Localizacion.busca,
Localizacion.SETTINGS, Localizacion.HELP, Localizacion.QUIT};

    //invoca a la clase LectorCanvas para mostrar el menu en pantalla
    Menu(LectorCanvas canvas) {
        super(Localizacion.MENU, IMPLICIT, items, null);
        this.canvas = canvas;
        addCommand(LectorBMU.REGRESA_CMD);
        setCommandListener(this);
        Display.getDisplay(canvas.lector).setCurrent(this);
    }

    public void commandAction(Command c, Displayable d)
    {
        if (c == LectorBMU.REGRESA_CMD) {
            Display.getDisplay(canvas.lector).setCurrent(canvas);
        } else {
            //Selección de las diferentes opciones del menu principal
            switch (getSelectedIndex()) {
                case 0:
                    new ListaMarcadores(canvas, this);
                    break;
                case 1:
                    new FormularioBuscar(canvas, this);
                    break;
                case 2:
                    new FormularioConfig(canvas, this);

```

```
        break;
    case 3:
        new FormularioAyuda(canvas, this);
        break;
    case 4:
        canvas.lector.quit();
        break;
    }
}
}
}

//----- Final del fichero Menu.java
```

```

//
// FormularioAyuda.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/*
 * Esta es la clase que implementa el formulario de ayuda del menu principal del
 midlet
 */

import javax.microedition.lcdui.*;

public class FormularioAyuda extends Form implements CommandListener {
    LectorCanvas canvas;
    Displayable parent;

    //Invoca a la clase LectorCanvas para mostrar el formulario de ayuda en pantalla
    FormularioAyuda(LectorCanvas canvas, Displayable parent) {
        super(Localizacion.HELP);
        this.canvas = canvas;
        this.parent = parent;
        addCommand(LectorBMU.REGRESA_CMD);
        append(Localizacion.HELP_TEXT);
        setCommandListener(this);
        Display.getDisplay(canvas.lector).setCurrent(this);
    }
    // Método de control de los comandos de la pantalla
    public void commandAction(Command c, Displayable d) {
        Display.getDisplay(canvas.lector).setCurrent(parent);
    }
}

//----- Final del fichero FormularioAyuda.java

```

```

//
// Marcador.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa el funcionamiento de los marcadores
 * en el midlet.*/

public class Marcador {
    int id;
    int pos;
    String name;

    byte[] pack() {
        byte[] nameBytes = Convertidor.packString(name);
        byte[] record = new byte[nameBytes.length + 4];
        Convertidor.packInt(record, 0, pos);
        System.arraycopy(nameBytes, 0, record, 4, nameBytes.length);
        return record;
    }

    void unpack(byte[] record) {
        pos = Convertidor.unpackInt(record, 0);
        name = Convertidor.unpackString(record, 4, record.length-4);
    }

    Marcador(int id, byte[] record) {
        this.id = id;
        unpack(record);
    }

    Marcador(int pos, String name) {
        this.pos = pos;
        this.name = name;
    }
}
//----- Final del fichero Marcador.java

```

```

//
// ListaMarcadores.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa el menu de los marcadores
 * en el midlet.*/

import javax.microedition.lcdui.*;
import java.util.Vector;

public class ListaMarcadores extends List implements CommandListener {
    LectorCanvas canvas;
    Displayable parent;

    //Invoca a la clase LectorCanvas para mostrar el formulario del menu de los
    marcadores en pantalla
    ListaMarcadores(LectorCanvas canvas, Displayable parent) {
        super(Localizacion.Marcadores, IMPLICIT);
        this.canvas = canvas;
        this.parent = parent;
        addCommand(LectorBMU.AGREGAR_CMD);
        addCommand(LectorBMU.RENOMBRAR_CMD);
        addCommand(LectorBMU.BORRAR_CMD);
        addCommand(LectorBMU.IR_A_CMD);
        addCommand(LectorBMU.REGRESA_CMD);
        Vector Marcadores = canvas.lector.Marcadores;
        for (int i = 0, n = Marcadores.size(); i < n; i++) {
            append(((Marcador)Marcadores.elementAt(i)).name, null);
        }
        setCommandListener(this);
        // Recuperamos el display del objeto estático del Sistema
        Display.getDisplay(canvas.lector).setCurrent(this);
    }
    // Método de control de los comandos de la pantalla
    public void commandAction(Command c, Displayable d)
    {
        //Lleva a cabo la acción seleccionada en el menú de los marcadores invocando
        a la clase correspondiente
        int i = getSelectedIndex();
        if (c == LectorBMU.REGRESA_CMD) {
            Display.getDisplay(canvas.lector).setCurrent(parent);
        } else if (c == LectorBMU.AGREGAR_CMD) {
            new FormAgregaMarcador(canvas, this);
        }
    }
}

```

```
    } else if (c == LectorBMU.RENOMBRAR_CMD) {
        new FormEdicionMarcador(canvas, this, i);
    } else if (c == LectorBMU.BORRAR_CMD) {
        canvas.lector.borraMarcador(i);
        delete(i);
    } else {
        canvas.setPosition(((Marcador)canvas.lector.Marcadores.elementAt(i)).pos);
        // Recuperamos el display del objeto estático del Sistema
        Display.getDisplay(canvas.lector).setCurrent(canvas);
    }
}
}

//----- Final del fichero ListaMarcadores.java
```

```

//
// FormAgregaMarcador.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa la creación de marcadores y agregar los mismos
 * en el midlet.*/

import javax.microedition.lcdui.*;

public class FormAgregaMarcador extends Form implements CommandListener {
    LectorCanvas canvas;
    ListaMarcadores list;
    TextField name;

    //Invoca a la clase LectorCanvas para mostrar el formulario agregar marcador en
    pantalla
    FormAgregaMarcador(LectorCanvas canvas, ListaMarcadores list) {
        super(Localizacion.NEW_BOOKMARK);
        this.canvas = canvas;
        this.list = list;
        addCommand(LectorBMU.AGREGAR_CMD);
        addCommand(LectorBMU.REGRESA_CMD);
        name = new TextField(Localizacion.NAME, canvas.marcadorDefault,
LectorCanvas.MAX_Bookmark_LENGTH, TextField.ANY);
        append(name);
        setCommandListener(this);
        Display.getDisplay(canvas.lector).setCurrent(this);
    }

    // Método de control de los comandos de la pantalla
    public void commandAction(Command c, Displayable d)
    {
        if (c == LectorBMU.AGREGAR_CMD) {
            String BookmarkName = name.getString().trim();
            if (BookmarkName.length() == 0) {
                canvas.lector.error(Localizacion.EMPTY_NAME, this);
                return;//No permite que el nombre del marcador este vacío
            }
            int pos = canvas.lector.agregaMarcador(new
Marcador(canvas.lector.currPagePos, BookmarkName));
            list.insert(pos, BookmarkName, null);//agrega en la lista de marcadores el
nuevo marcador
        }
        // Recuperamos el display del objeto estático del Sistema
        Display.getDisplay(canvas.lector).setCurrent(list);
    }
}

```



```
}  
}
```

```
//----- Final del fichero FormAgregaMarcador.java
```

```

//
// FormEdicionMarcador.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa la edición de marcadores y actualizar los mismos
 * en el midlet.*/

import javax.microedition.lcdui.*;

public class FormEdicionMarcador extends Form implements CommandListener {
    LectorCanvas canvas;
    ListaMarcadores list;
    int    index;
    TextField  name;
    Marcador  bm;

    //Invoca a la clase LectorCanvas para mostrar el formulario editar  marcador en
    pantalla y sus opciones
    //Regresa o renombrar
    FormEdicionMarcador(LectorCanvas canvas, ListaMarcadores list, int index) {
        super(Localizacion.EDIT_BOOKMARK);
        this.canvas = canvas;
        this.list = list;
        this.index = index;
        addCommand(LectorBMU.RENOMBRAR_CMD);
        addCommand(LectorBMU.REGRESA_CMD);
        bm = (Marcador)canvas.lector.Marcadores.elementAt(index);
        name = new TextField(Localizacion.NAME, bm.name,
LectorCanvas.MAX_Bookmark_LENGTH, TextField.ANY);
        append(name);
        setCommandListener(this);
        // Recuperamos el display del objeto estático del Sistema
        Display.getDisplay(canvas.lector).setCurrent(this);
    }
    // Método de control de los comandos de la pantalla
    public void commandAction(Command c, Displayable d)
    {
        if (c == LectorBMU.RENOMBRAR_CMD) {
            String BookmarkName = name.getString().trim();
            if (BookmarkName.length() == 0) {
                canvas.lector.error(Localizacion.EMPTY_NAME, this);
                return;
            }
            bm.name = BookmarkName;
            canvas.lector.actualizaMarcador(bm);
        }
    }
}

```

```
        list.set(index, BookmarkName, null);
    }
    // Recuperamos el display del objeto estático del Sistema
    Display.getDisplay(canvas.lector).setCurrent(list);
}
}
//----- Final del fichero FormEdicionMarcador.java
```

```

//
// Convertidor.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que define la conversión de los caracteres para su posterior
implementación en
*la pantalla del dispositivo.
*/

public class Convertidor {
    static void packInt(byte[] arr, int offs, int x) {
        arr[offs] = (byte)(x >> 24);
        arr[offs+1] = (byte)(x >> 16);
        arr[offs+2] = (byte)(x >> 8);
        arr[offs+3] = (byte)x;
    }

    static int unpackInt(byte[] arr, int offs) {
        return (arr[offs] << 24) | ((arr[offs+1] & 0xFF) << 16) | ((arr[offs+2] & 0xFF)
<< 8) | (arr[offs+3] & 0xFF);
    }

    static byte[] packString(String str) {
        char[] chars = str.toCharArray();
        byte[] bytes = new byte[chars.length*2];
        for (int i = 0; i < chars.length; i++) {
            bytes[i*2] = (byte)(chars[i] >> 8);
            bytes[i*2+1] = (byte)chars[i];
        }
        return bytes;//Devuelve los bytes o tamaño de la cadena de caracteres
    }

    static String unpackString(byte[] arr, int offs, int len) {
        len >>= 1;
        char[] chars = new char[len];
        for (int i = 0; i < len; i++) {
            chars[i] = (char)((arr[offs + i*2] << 8) | (arr[offs + i*2 + 1] & 0xFF));
        }
        return new String(chars);//Devuelve la cadena de caracteres
    }
}

//----- Final del fichero Convertidor.java

```

```
//  
// InterfaceFuente.java  
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)  
// Autor:Juan Carlos Vázquez Lorenzo  
// Creacion: 2006,2007  
//-----  
/* Clase que define la interfaz de la fuente en dimensiones.  
*/  
  
import javax.microedition.lcdui.*;  
  
public interface InterfaceFuente {  
    public void drawString(Graphics g, int x, int y, String s);  
    public int  charWidth(char ch);  
    public int  getHeight();  
}  
  
//----- Final del fichero InterfaceFuente.java
```

```

//
// SistemaDeFuentes.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor: Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa la interfaz de la fuente en
   la pantalla*/

import javax.microedition.lcdui.*;

public class SistemaDeFuentes implements InterfaceFuente {
    SistemaDeFuentes(Font font) {
        this.font = font;
    }

    public void drawString(Graphics g, int x, int y, String s) {
        g.setFont(font);
        g.drawString(s, x, y, Graphics.LEFT | Graphics.TOP);
    }

    public int charWidth(char ch) {
        return font.charWidth(ch);
    }

    public int getHeight() {
        return font.getHeight();
    }

    Font font;
}
//----- Final del fichero SistemaDeFuentes.java

```

```

//
// ImagenFuente.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa la creación y presentación de las distintas
 * capas que intervienen en el midlet.
 * La clase también controla la imagen de la fuente cargada al dispositivo,*/
import javax.microedition.lcdui.*;
import java.io.*;

public class ImagenFuente implements InterfaceFuente
{
    //definición de variables
    private int [] _props;
    private byte [] _ancho;
    private String _nombreFuente;
    private boolean cargada;
    private Image _buf;
    private int _contador, _altura;

    public ImagenFuente (String nombreFuente)
    {
        _nombreFuente = nombreFuente;
        cargaFuente ();
    }

    private int obtenPalabra (InputStream stream) throws IOException
    {
        int lo = stream.read ();
        int hi = stream.read ();
        if (lo < 0 || hi < 0) return -1;
        return hi * 256 + lo;
    }
    //Método que carga la fuente
    public void cargaFuente ()
    {
        if (cargada)
            clear ();
        try
        {
            InputStream in = this.getClass ().getResourceAsStream (_nombreFuente +
".props");//Fuente origen desde la cual busca el recurso

            _contador = obtenPalabra (in);
            _props = new int [_contador];

```

```

        _ancho = new byte [_contador];

        for (int i = 0; i < _contador; i++)
        {
            _props [i] = obtenPalabra (in);
            if (i > 0)
                _ancho [i - 1] = (byte) (_props [i] - _props [i - 1]);
        }
        _buf = Image.createImage (_nombreFuente + ".png");
        _altura = _buf.getHeight ();
        _ancho [_contador - 1] = (byte) (_buf.getWidth () - _props [_contador - 1]);
        cargada = true;

    } catch (Exception e)
    {
        System.out.println("Error al cargar la fuente");
        clear ();
        return;
    }
}
//Método que dibuja el caracter y carga la imagen
public void drawString (Graphics g, int x, int y, String s)
{
    if (!cargada)
    {
        g.drawString ("La fuente no ha sido cargada", 0, 0, 0);
        System.out.println ("no se puede importar fuente la fuente no ha sido
cargada");
        return;
    }
    int clipx = g.getClipX ();
    int clipy = g.getClipY ();
    int clipw = g.getClipWidth ();
    int cliph = g.getClipHeight ();
    int offs = x;
    for (int i = 0; i < s.length (); i ++)
    {
        int ch = s.charAt(i) - 1;
        ch = ch == 0x400 ? 0xa7 : ch == 0x450 ? 0xb7 : ch;
        ch = ch > 0x400 ? ch - 0x350 : ch;
        if (ch < 0 || ch >= _contador)
        {
            continue;
        }
        int width = _ancho [ch];
        if (ch > 31)

```



```

        {
            int imgx = _props [ch];
            g.setClip (offs, y, width, _altura);
            g.drawImage (_buf, offs - imgx, y, Graphics.LEFT | Graphics.TOP);
        }
        offs += width;
    }
    g.setClip (clipx, clipy, clipw, cliph);
}
//Método que determina la anchura del caracter y carga la imagen
public int charWidth (char ch)
{
    if (!cargada)
    {
        System.out.println ("Anchura de caracter sin determinar, no se puede cargar
la fuente");
        return 0;
    }
    ch = (char) (ch == 0x401 ? 0xa8 : ch == 0x451 ? 0xb8 : ch);
    ch = (char) (ch > 0x400 ? ch == 0x451 ? 0xb8 : ch == 0x401 ? 0xa8 : ch -
0x351 : ch - 1);
    if (ch < 0 || ch >= _contador)
    {
        return 0;
    }
    return _ancho [ch];
}

public int getHeight ()
{
    return _altura;
}

private void clear ()
{
    if (!cargada) return;
    _nombreFuente = "";
    _buf = null;
    _props = null;
    _ancho = null;
    _altura = 0;
}
}
//----- Final del fichero ImagenFuente.java

```

```

//
// FormularioBuscar.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que implementa la creación de marcadores y agregar los mismos
 * en el midlet.*/

import javax.microedition.lcdui.*;

public class FormularioBuscar extends Form implements CommandListener {
    LectorCanvas canvas;
    Displayable parent;
    TextField muestra;
    //Define la cadena máxima de caracteres a buscar
    static final int MAX_busca_muestra_LENGTH = 64;

    //Invoca a la clase LectorCanvas para mostrar el formulario buscar en pantalla
    FormularioBuscar(LectorCanvas canvas, Displayable parent) {
        super(Localizacion.busca);
        this.canvas = canvas;
        this.parent = parent;
        addCommand(LectorBMU.BUSCAR_CMD);
        addCommand(LectorBMU.REGRESA_CMD);
        muestra = new TextField(Localizacion.busca, canvas.buscamuestra,
MAX_busca_muestra_LENGTH, TextField.ANY);
        append(muestra);
        setCommandListener(this);
        Display.getDisplay(canvas.lector).setCurrent(this);
    }
    // Método de control de los comandos de la pantalla buscar y regresa
    public void commandAction(Command c, Displayable d)
    {
        if (c == LectorBMU.BUSCAR_CMD) {
            canvas.buscamuestra = muestra.getString().trim();
            if (canvas.buscamuestra.length() == 0) {
                return;
            }
            canvas.buscaPos = canvas.lector.busca(0, canvas.buscamuestra);
            if (canvas.buscaPos < 0) {
                canvas.lector.error(Localizacion.NOT_FOUND, this);
                return;//Regresa el mensaje de error, de que no fue encontrada la cadena
de caracteres
            }
            canvas.setPosition(canvas.buscaPos);
            Display.getDisplay(canvas.lector).setCurrent(canvas);

```

```
        return;//Nos posiciona en la cadena de caracteres encontrada
    }
    // Recuperamos el display del objeto estático del Sistema
    Display.getDisplay(canvas.lector).setCurrent(parent);
}
}
//----- Final del fichero FormularioBuscar.java
```

```

//
// Localizacion.java
// Compilador: J2ME Wireless Toolkit 2.0 (Java version: 1.4.2_11)
// Autor:Juan Carlos Vázquez Lorenzo
// Creacion: 2006,2007
//-----
/* Clase que define la internacionalización de los menus para su posterior
implementación en
*la pantalla del dispositivo.
*/
public class Localizacion {
    public static final String NEW_BOOKMARK = "Nuevo Marcador";
    public static final String EDIT_BOOKMARK = "Editar Marcador";
    public static final String Marcadores = "Marcadores";
    public static final String NAME = "Nombre";
    public static final String EMPTY_NAME = "El nombre del Marcador no debe
estar vacio";
    public static final String BACK = "Regresar";
    public static final String QUIT = "Quitar";
    public static final String GOTO = "Ir a";
    public static final String ADD = "Agregar";
    public static final String RENAME = "Renombrar";
    public static final String REMOVE= "Eliminar";
    public static final String busca = "Buscar";
    public static final String HELP = "Ayuda";
    public static final String BEGIN = "Inicio";
    public static final String END = "Fin";
    public static final String ERROR = "Error";
    public static final String MENU = "Menu principal";
    public static final String SETTINGS = "Configuraciones";
    public static final String SAVE = "Guardar";
    public static final String font = "Fuente";
    public static final String SLIDER = "Desplazamiento";
    public static final String OVERLAP = "Interlineado";
    public static final String HORIZONTAL = "Horizontal";
    public static final String VERTICAL = "Vertical";
    public static final String NOT_FOUND = "No encontrado";
    public static final String BACKGROUND = "Fondo";
    public static final String WHITE = "Blanco";
    public static final String BLACK = "Negro";

    public static final char HYPHEN = '-';

//Definición del menú ayuda
static final String HELP_TEXT =
    "TECLAS DE CONTROL:\n\n" +

```

```
"ABAJO:Av pag\n" +
"DERECHA:adelanta linea\n" +
"IZQUIERDA:regresa linea\n" +
"ARRIBA:Re pag\n" +
"* buscar siguiente\n" +
"# menu principal\n" +
"0 lista de Marcadores\n" +
"1-9 avance de n lineas\n";

public static char obtenCaracter(byte b) {
    return (char)(b & 0xFF);
}

public static int split(StringBuffer buf, int i) {
    return -1;
}
}

//----- Final del fichero Localizacion.java
```

ANEXO C. J2ME ARQUITECTURA SELECCIONADA PARA EL DESARROLLO DE LA APLICACIÓN LectorBMU

Sun, dispuesto a proporcionar las herramientas necesarias para cubrir las necesidades de todos los usuarios, creó distintas versiones de Java de acuerdo a las necesidades de cada uno. Según esto nos encontramos con que el paquete Java 2 lo podemos dividir en 3 ediciones distintas:

- J2SE (Java Standard Edition) orientada al desarrollo de aplicaciones independientes de la plataforma
- J2EE (Java Enterprise Edition) orientada al entorno empresarial
- J2ME (Java Micro Edition) orientada a dispositivos con capacidades restringidas.

Características de cada una de las versiones

Java 2 Platform, Standard Edition (J2SE): Esta edición de Java es la que en cierta forma recoge la iniciativa original del lenguaje Java. Tiene las siguientes características:

- Inspirado inicialmente en C++, pero con componentes de alto nivel, como soporte nativo de strings y recolector de basura.
- Código independiente de la plataforma, precompilado a bytecodes intermedio y ejecutado en el cliente por una JVM (Java Virtual Machine).
- Modelo de seguridad tipo sandbox proporcionado por la JVM.
- Abstracción del sistema operativo subyacente mediante un juego completo de APIs de programación.

Esta versión de Java contiene el conjunto básico de herramientas usadas para desarrollar Java Applets, así como las APIs orientadas a la programación de aplicaciones de usuario final: Interfaz gráfica de usuario, multimedia, redes de comunicación, etc.

Java 2 Platform, Enterprise Edition (J2EE): Esta versión está orientada al entorno empresarial. El software empresarial tiene unas características propias marcadas: está pensado no para ser ejecutado en un equipo, sino para ejecutarse sobre una red de ordenadores de manera distribuida y remota mediante Ejes (Enterprise Java Beans).

De hecho, el sistema se monta sobre varias unidades o aplicaciones. En muchos casos, además, el software empresarial requiere que se sea capaz de integrar datos provenientes de entornos heterogéneos. Esta edición está orientada especialmente al desarrollo de servicios web, servicios de nombres, persistencia de objetos, XML, autenticación, APIs para la gestión de transacciones, etc. El cometido de esta especificación es ampliar la J2SE para dar soporte a los requisitos de las aplicaciones de empresa.

Java 2 Platform, Micro Edition (J2ME): Esta versión 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, PDAs o electrodomésticos inteligentes. Esta edición tiene unos componentes básicos que la diferencian de las otras versiones, como el uso de una máquina virtual denominada KVM (Kilo Virtual Machine, debido a que requiere sólo unos pocos Kilobytes de memoria para funcionar) en vez del uso de la JVM clásica, inclusión de un pequeño y rápido recolector de basura y otras diferencias.¹

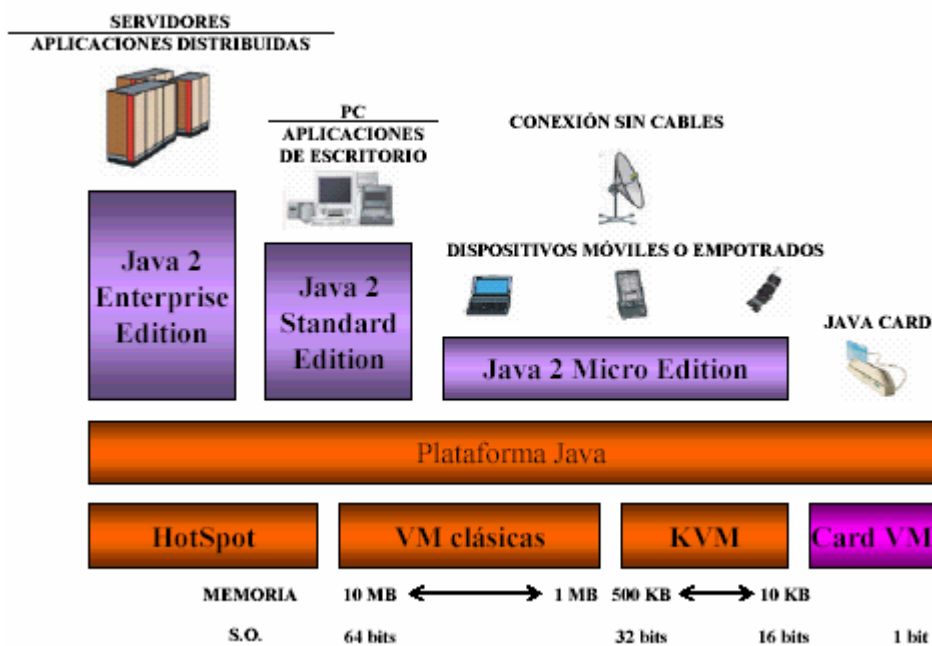


Fig. 1.1 Diversas tecnologías de la Plataforma Java

No es realista ver Java como un simple lenguaje de programación, sino como un conjunto de tecnologías que abarca a todos los ámbitos de la computación con dos elementos en común:

- El código fuente en lenguaje Java es compilado a código intermedio interpretado por una Java Virtual Machine (JVM), por lo que el código ya compilado es independiente de la plataforma.
- Todas las tecnologías comparten un conjunto más o menos amplio de APIs básicas del lenguaje, agrupadas principalmente en los paquetes `java.lang` y `java.io`.

Un claro ejemplo de éste último punto es que J2ME contiene una mínima parte de las APIs de Java. Esto es debido a que la edición estándar de APIs de Java ocupa 20 Mb, y los dispositivos pequeños disponen de una cantidad de memoria mucho más reducida. En concreto, J2ME usa 37 clases de la plataforma J2SE provenientes de los paquetes

¹ GÁLVEZ, Rojas Sergio, ORTEGA, Díaz Lucas (2003), *Java a Tope: J2ME (Java 2Micro Edition)* Madrid, Universidad de Malaga pp 3

java.lang, java.io, java.util. Esta parte de la API que se mantiene fija forma parte de lo que se denomina “configuración”. Otras diferencias con la plataforma J2SE vienen dadas por el uso de una máquina virtual distinta de la clásica JVM denominada KVM. Esta KVM tiene unas restricciones que hacen que no posea todas las capacidades incluidas en la JVM.

J2ME representa una versión simplificada de J2SE. Se separó estas dos versiones ya que J2ME estaba pensada para dispositivos con limitaciones de proceso y capacidad gráfica. También separó J2SE de J2EE porque este último exigía unas características muy pesadas o especializadas de E/S, trabajo en red, etc. Por tanto, se separó ambos productos por razones de eficiencia. J2EE es un súper conjunto de J2SE pues contiene toda la funcionalidad de éste y más características, así como J2ME es un subconjunto de J2SE (excepto por el paquete javax.microedition) ya que, como se ha mencionado, contiene varias limitaciones con respecto a J2SE.

El conjunto de clases provenientes de los paquetes java.lang, java.util y java.io, relacionan las tres tecnologías. J2ME nace al agregar a esas clases, el paquete javax.microedition.*

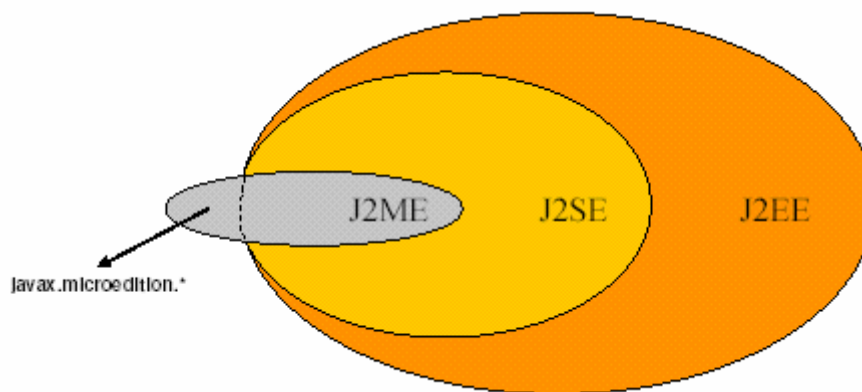


Figura 1.2 Arquitectura J2ME

La arquitectura está compuesta por capas y estas son la configuración, el perfil y los paquetes opcionales. Todos ellos tienen como base el sistema operativo del hardware donde se instalará la aplicación J2ME y será la máquina virtual la encargada de la ejecución.

Componentes que forman parte de esta tecnología:

- Por un lado una serie de máquinas virtuales Java con diferentes requisitos, cada una para diferentes tipos de pequeños dispositivos.

- Configuraciones, que son un conjunto de clases básicas orientadas a conformar el corazón de las implementaciones para dispositivos de características específicas.

Existen 2 configuraciones definidas en J2ME:

1. *Connected Limited Device Configuration (CLDC)* enfocada a dispositivos con restricciones de procesamiento y memoria.
2. *Connected Device Configuration (CDC)* enfocada a dispositivos con más recursos.²

- Perfiles, que son unas bibliotecas Java de clases específicas orientadas a implementar funcionalidades de más alto nivel para familias específicas de dispositivos.

Un entorno de ejecución determinado de J2ME se compone entonces de una selección de:

- a) *Máquina virtual.*
- b) *Configuración.*
- c) *Perfil.*
- d) *Paquetes Opcionales.*

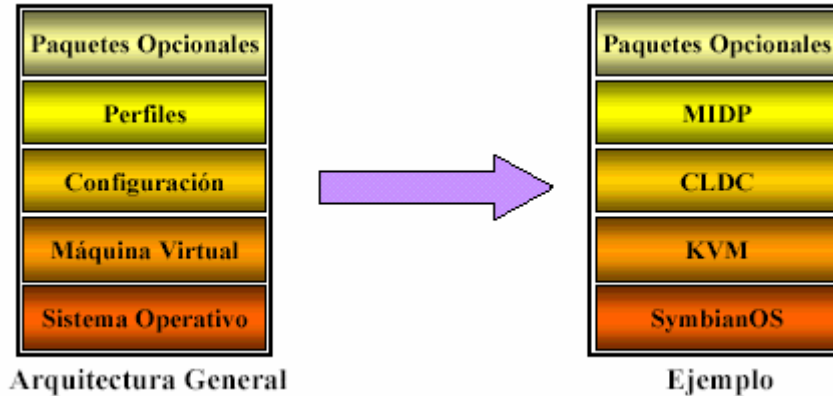


Fig. 1.3 Un entorno de ejecución de J2ME

Esta arquitectura a su vez se basa en familias y categorías de dispositivos. Una categoría define un tipo de dispositivo particular.

² FROUFE, Quintas Agustín, JORGE, Cárdenas Patricia (2004), *J2ME Java 2 Micro Edition: Manual de usuario y tutorial*, Madrid, Alfaomega. Pp 23

Ejemplo:



Fig. 1.4 Nokia 7700 (CDC-RMI Profile)

Y una familia de dispositivos está compuesta por un grupo de categorías. Para constituir una familia, los dispositivos tendrán características similares, tales como memoria, procesamiento, representación por pantalla, conexión a redes, etc.

Ejemplo:



Fig. 1.5 PDA (CLDC-PDA Profile)



Celular (CLDC-MIDP)

Estos dos dispositivos pertenecen a la misma familia.

Máquinas Virtuales J2ME

Una máquina virtual de Java (JVM) es un programa encargado de interpretar código intermedio (bytecode) de los programas Java precompilados a código máquina ejecutable por la plataforma, efectuar las llamadas pertinentes al sistema operativo subyacente y observar las reglas de seguridad y corrección de código definidas para el lenguaje Java. De esta forma, la JVM proporciona al programa Java independencia de la plataforma con respecto al hardware y al sistema operativo subyacente.³

Las implementaciones tradicionales de JVM son, en general, muy pesadas en cuanto a memoria ocupada y requerimientos computacionales. J2ME define varias JVMs de referencia adecuadas al ámbito de los dispositivos electrónicos que, en algunos casos, suprimen algunas características con el fin de obtener una implementación menos exigente.

Como ya se mencionó existen 2 configuraciones CLDC y CDC, cada una con unas características propias. Como consecuencia, cada una requiere su propia máquina virtual. La VM (Virtual Machine) de la configuración CLDC se denomina KVM y la de la configuración CDC se denomina CVM. A continuación las características principales de cada una de ellas:

³ GÁLVEZ, Rojas Sergio, ORTEGA, Díaz Lucas Op. Cit.. pp. 5.

- *KVM*

Se corresponde con la Máquina Virtual más pequeña desarrollada por Sun. Su nombre KVM proviene de Kilobyte (haciendo referencia a la baja ocupación de memoria, entre 40Kb y 80Kb). Se trata de una implementación de Máquina Virtual reducida y especialmente orientada a dispositivos con bajas capacidades computacionales y de memoria. La KVM está escrita en lenguaje C, aproximadamente unas 24000 líneas de código, y fue diseñada para ser:

- Pequeña, con una carga de memoria entre los 40Kb y los 80 Kb, dependiendo de la plataforma y las opciones de compilación.
- Alta portabilidad.
- Modulable.
- Lo más completa y rápida posible y sin sacrificar características para las que fue diseñada.

Sin embargo, esta baja ocupación de memoria hace que posea algunas limitaciones con respecto a la clásica Java Virtual Machine (JVM) entre las cuales están las siguientes:

- No hay soporte para tipos en coma flotante. No existen por tanto los tipos double ni float. Esta limitación está presente porque los dispositivos carecen del hardware necesario para estas operaciones.
- No existe soporte para JNI (Java Native Interface) debido a los recursos limitados de memoria.
- No existen cargadores de clases (class loaders) definidos por el usuario. Sólo existen los predefinidos.
- Limitada capacidad para el manejo de excepciones debido a que el manejo de éstas depende en gran parte de las APIs de cada dispositivo por lo que son éstos los que controlan la mayoría de las excepciones.

El verificador de clases estándar de Java es demasiado grande para la KVM. De hecho es más grande que la propia KVM y el consumo de memoria es excesivo, más de 100Kb para las aplicaciones típicas. Este verificador de clases es el encargado de rechazar las clases no válidas en tiempo de ejecución. Este mecanismo verifica los bytcodes de las clases Java realizando las siguientes comprobaciones:

- Ver que el código no sobrepase los límites de la pila de la VM.
- Comprobar que no se utilizan las variables locales antes de ser inicializadas.
- Comprobar que se respetan los campos, métodos y los modificadores de control de acceso a clases.

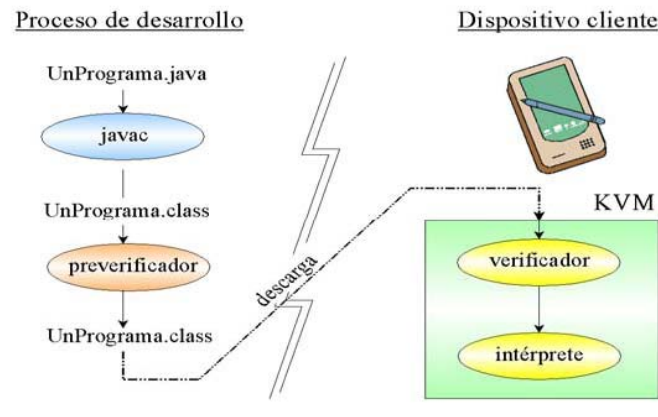


Fig. 1.6 Preverificación de clases en CDLC/KVM.

La KVM puede ser compilada y probada en 3 plataformas distintas:

1. Solaris Operating Environment.
2. Windows
3. PalmOs

- CVM

La CVM (Compact Virtual Machine) ha sido tomada como Máquina Virtual Java de referencia para la configuración CDC y soporta las mismas características que la Máquina Virtual de J2SE. Está orientada a dispositivos electrónicos con procesadores de 32 bits de gama alta y en torno a 2Mb o más de memoria RAM. Las características que presenta esta Máquina Virtual son:

- Sistema de memoria avanzado.
- Tiempo de espera bajo para el recolector de basura.
- Separación completa de la VM del sistema de memoria.
- Recolector de basura modularizado.
- Portabilidad.
- Rápida sincronización.
- Ejecución de las clases Java fuera de la memoria de sólo lectura (ROM).
- Soporte nativo de hilos.
- Baja ocupación en memoria de las clases.
- Proporciona soporte e interfaces para servicios en Sistemas Operativos de Tiempo Real.
- Conversión de hilos Java a hilos nativos.
- Soporte para todas las características de Java2 v1.3 y librerías de seguridad, referencias débiles, Interfaz Nativa de Java (JNI), invocación remota de métodos (RMI), Interfaz de depuración de la Máquina Virtual (JVMDI).

- Configuraciones

Una configuración es el conjunto mínimo de APIs Java que permiten desarrollar aplicaciones para un grupo de dispositivos. Éstas APIs describen las características básicas, comunes a todos los dispositivos:

- Características soportadas del lenguaje de programación Java.
- Características soportadas por la Máquina Virtual Java.
- Bibliotecas básicas de Java y APIs soportadas.

Existen dos configuraciones en J2ME:

- Configuración de dispositivos con conexión, CDC (Connected Limited Configuration)

La CDC está orientada a dispositivos con cierta capacidad computacional y de memoria. Por ejemplo, decodificadores de televisión digital, televisores con internet, algunos electrodomésticos y sistemas de navegación en automóviles. CDC usa una Máquina Virtual Java similar en sus características a una de J2SE, pero con limitaciones en el apartado gráfico y de memoria del dispositivo. Ésta Máquina Virtual es la que hemos visto como CVM (Compact Virtual Machine). La CDC está enfocada a dispositivos con las siguientes capacidades:

- Procesador de 32 bits.
- Disponer de 2 Mb o más de memoria total, incluyendo memoria RAM y ROM.
- Poseer la funcionalidad completa de la Máquina Virtual Java2.
- Conectividad a algún tipo de red.

La CDC está basada en J2SE v1.3 e incluye varios paquetes Java de la edición estándar. Las peculiaridades de la CDC están contenidas principalmente en el paquete `javax.microedition.io`, que incluye soporte para comunicaciones http y basadas en datagramas. La Tabla 1.1 nos muestra las librerías incluidas en la CDC.

| Nombre de Paquete CDC | Descripción |
|------------------------------------|--|
| <code>java.io</code> | Clases e interfaces estándar de E/S. |
| <code>java.lang</code> | Clases básicas del lenguaje. |
| <code>java.lang.ref</code> | Clases de referencia. |
| <code>java.lang.reflect</code> | Clases e interfaces de reflection. |
| <code>java.math</code> | Paquete de matemáticas. |
| <code>java.net</code> | Clases e interfaces de red. |
| <code>java.security</code> | Clases e interfaces de seguridad |
| <code>java.security.cert</code> | Clases de certificados de seguridad. |
| <code>java.text</code> | Paquete de texto. |
| <code>java.util</code> | Clases de utilidades estándar. |
| <code>java.util.jar</code> | Clases y utilidades para archivos JAR. |
| <code>java.util.zip</code> | Clases y utilidades para archivos ZIP y comprimidos. |
| <code>javax.microedition.io</code> | Clases e interfaces para conexión genérica CDC. |

Tabla 1.1 Librerías de configuración CDC.

- Configuración de dispositivos limitados con conexión, CLDC (Connected Limited Device Configuration).

La CLDC está orientada a dispositivos dotados de conexión y con limitaciones en cuanto a capacidad gráfica, cómputo y memoria. Un ejemplo de estos dispositivos son: teléfonos móviles, buscapersonas (pagers), PDAs, organizadores personales, etc. Algunas restricciones vienen dadas por el uso de la KVM, necesaria al trabajar con la CLDC debido a su pequeño tamaño. Los dispositivos que usan CLDC deben cumplir los siguientes requisitos:

- Disponer entre 160 Kb y 512 Kb de memoria total disponible. Como mínimo se debe disponer de 128 Kb de memoria no volátil para la Máquina Virtual Java y las bibliotecas CLDC, y 32 Kb de memoria volátil para la Máquina Virtual en tiempo de ejecución.
- Procesador de 16 o 32 bits con al menos 25 Mhz de velocidad.
- Ofrecer bajo consumo, debido a que éstos dispositivos trabajan con suministro de energía limitado, normalmente baterías.
- Tener conexión a algún tipo de red, normalmente sin cable, con conexión intermitente y ancho de banda limitado (unos 9600 bps).

La CLDC aporta las siguientes funcionalidades a los dispositivos:

- Un subconjunto del lenguaje Java y todas las restricciones de su Máquina Virtual (KVM).
- Un subconjunto de las bibliotecas Java del núcleo.
- Soporte para E/S básica.
- Soporte para acceso a redes.
- Seguridad.

La siguiente tabla nos muestra las librerías incluidas en la CLDC.

| Nombre de paquete CLDC | Descripción |
|------------------------|--|
| java.io | Clases y paquetes estándar de E/S. Subconjunto de J2SE. |
| java.lang | Clases e interfaces de la Máquina Virtual. Subconj. de J2SE. |
| java.util | Clases, interfaces y utilidades estándar. Subconj. de J2SE. |
| javax.microedition.io | Clases e interfaces de conexión genérica CLDC |

Tabla 1.2 Librerías de configuración CLDC.

Esta configuración posee un modelo de seguridad sandbox al igual que ocurre con los applets.

En cualquier caso, una determinada Configuración no se encarga del mantenimiento del ciclo de vida de la aplicación, interfaces de usuario o manejo de eventos, sino que estas responsabilidades caen en manos de los perfiles.

Perfiles.

El perfil es el que define las APIs que controlan el ciclo de vida de la aplicación, interfaz de usuario, etc. Un perfil es un conjunto de APIs orientado a un ámbito de aplicación determinado.

Los perfiles identifican un grupo de dispositivos por la funcionalidad que proporcionan (electrodomésticos, teléfonos móviles, etc.) y el tipo de aplicaciones que se ejecutarán en ellos. Las librerías de la interfaz gráfica son un componente muy importante en la definición de un perfil. Aquí nos podemos encontrar grandes diferencias entre interfaces, desde el menú textual de los teléfonos móviles hasta los táctiles de los PDAs. El perfil establece unas APIs que definen las características de un dispositivo, mientras que la configuración hace lo propio con una familia de ellos. Esto hace que a la hora de construir una aplicación se cuente tanto con las APIs del perfil como de la configuración. Se debe tener en cuenta que un perfil siempre se construye sobre una configuración determinada. De este modo, podemos pensar en un perfil como un conjunto de APIs que dotan a una configuración de funcionalidad específica.

Este entorno de ejecución se estructura en capas, una construida sobre la otra.

Existen unos perfiles que se construyen sobre la configuración CDC y otros que se construyen sobre la CLDC. Para la configuración CDC tenemos los siguientes perfiles:

- Foundation Profile.
- Personal Profile.
- RMI Profile.

Y para la configuración CLDC tenemos los siguientes:

- PDA Profile.
- Mobile Information Device Profile (MIDP).

Un perfil puede ser construido sobre cualquier otro. Sin embargo, una plataforma J2ME sólo puede contener una configuración.

Foundation Profile: Este perfil define una serie de APIs sobre la CDC orientadas a dispositivos que carecen de interfaz gráfica como, por ejemplo, decodificadores de televisión digital. Este perfil incluye gran parte de los paquetes de la J2SE, pero excluye totalmente los paquetes “java.awt” Abstract Windows Toolkit (AWT) y “java.swing” que conforman la interfaz gráfica de usuario (GUI) de J2SE. Si una aplicación requiriera una GUI, entonces sería necesario un perfil adicional. Los paquetes que forman parte del Foundation Profile se muestran en la siguiente tabla.

| Paq. del Foundation Profile | Descripción |
|-----------------------------|---|
| java.lang | Soporte del lenguaje Java |
| java.util | Añade soporte completo para zip y otras funcionalidades (java.util.Timer) |
| java.net | Incluye sockets TCP/IP y conexiones HTTP |
| java.io | Clases Reader y Writer de J2SE |
| java.text | Incluye soporte para internacionalización |
| java.security | Incluye códigos y certificados |

Tabla 1.3 Librerías del Foundation Profile.

Personal Profile: El Personal Profile es un subconjunto de la plataforma J2SE v1.3, y proporciona un entorno con un completo soporte gráfico AWT. El objetivo es el de

dotar a la configuración CDC de una interfaz gráfica completa, con capacidades web y soporte de applets Java. Este perfil requiere una implementación del Foundation Profile. La Tabla 1.4 muestra los paquetes que conforman el Personal Profile v1.0.

| Paq. del Personal Profile | Descripción |
|---------------------------|---|
| java.applet | Clases necesitadas para crear applets o que son usadas por ellos |
| java.awt | Clases para crear GUIs con AWT |
| java.awt.datatransfer | Clases e interfaces para transmitir datos entre aplicaciones |
| java.awt.event | Clases e interfaces para manejar eventos AWT |
| java.awt.font | Clases e interfaces para la manipulación de fuentes |
| java.awt.im | Clases e interfaces para definir métodos editores de entrada |
| java.awt.im.spi | Interfaces que añaden el desarrollo de métodos editores de entrada para cualquier entorno de ejecución Java |
| java.awt.image | Clases para crear y modificar imágenes |
| java.beans | Clases que soportan JavaBeans |
| javax.microedition.xlet | Interfaces que usa el Personal Profile para la comunicación. |

Tabla 1.4 Librerías del Personal Profile

RMI Profile: Este perfil requiere una implementación del Foundation Profile se construye encima de él. El perfil RMI soporta un subconjunto de las APIs J2SE v1.3 RMI. Algunas características de estas APIs se han eliminado del perfil RMI debido a las limitaciones de cómputo y memoria de los dispositivos.

PDA Profile: El PDA Profile está construido sobre CLDC. Pretende abarcar PDAs de gama baja, tipo Palm, con una pantalla y algún tipo de puntero (ratón o lápiz) y una resolución de al menos 20000 pixels (al menos 200x100 pixels) con un factor 2:1.

Mobile Information Device Profile (MIDP): Este perfil está construido sobre la configuración CLDC. Al igual que CLDC fue la primera configuración definida para J2ME, MIDP fue el primer perfil definido para esta plataforma. Este perfil está orientado para dispositivos con las siguientes características:

- Reducida capacidad computacional y de memoria.
- Conectividad limitada (en torno a 9600 bps).
- Capacidad gráfica muy reducida (mínimo un display de 96x54 pixels monocromo).
- Entrada de datos alfanumérica reducida.
- 128 Kb de memoria no volátil para componentes MIDP.
- 8 Kb de memoria no volátil para datos persistentes de aplicaciones.

- 32 Kb de memoria volátil en tiempo de ejecución para la pila Java.
- Los tipos de dispositivos que se adaptan a estas características son: teléfonos móviles, buscapersonas (pagers) o PDAs de gama baja con conectividad.

El perfil MIDP establece las capacidades del dispositivo, por lo tanto, especifica las APIs relacionadas con:

- La aplicación (semántica y control de la aplicación MIDP).
- Interfaz de usuario.
- Almacenamiento persistente.
- Trabajo en red.
- Temporizadores.

En la Tabla 1.5 podemos ver cuáles son los paquetes que están incluidos en el perfil MIDP.

| Paquetes del MIDP | Descripción |
|---------------------------|---|
| javax.microedition.lcdui | Clases e interfaces para GUIs |
| javax.microedition.rms | Record Management Storage. Soporte para el almacenamiento persistente del dispositivo |
| javax.microedition.midlet | Clases de definición de la aplicación |
| javax.microedition.io | Clases e interfaces de conexión genérica |
| java.io | Clases e interfaces de E/S básica |
| java.lang | Clases e interfaces de la Máquina Virtual |
| java.util | Clases e interfaces de utilidades estándar |

Tabla 1.5 Librerías del perfil MIDP.

Las aplicaciones que se realizan utilizando MIDP reciben el nombre de MIDlets (por simpatía con Applets). Se dice así que un MIDlet es una aplicación Java realizada con el perfil MIDP sobre la configuración CLDC.

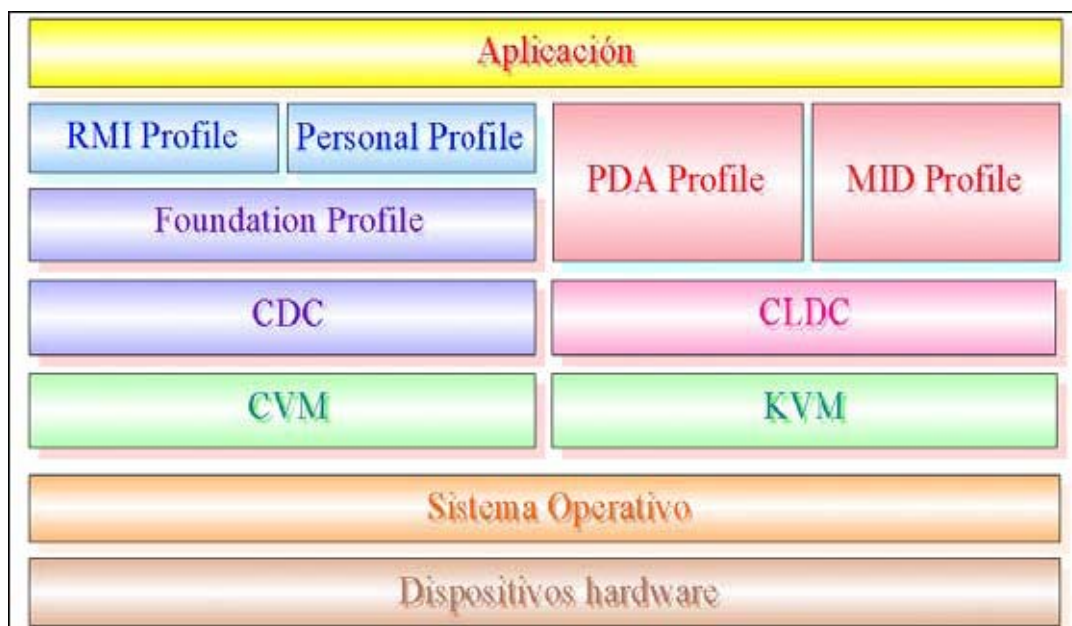


Fig. 1.7 Arquitectura del entorno de ejecución de J2ME.

Paquetes Opcionales.

Son APIs adicionales que pueden agregarse para desarrollar aplicaciones con más funciones. Por ejemplo podemos agregar APIs para desarrollar interfaces en 3D, para establecer comunicaciones Bluetooth, para conexión a Web Services, Reproducción MP3, Mensajes MMS, Correo, etc.



Fig. 1.8 Paquetes Opcionales.

Los MIDlets residen en el Application Management System, conocido también como AMS, este software se encuentra en el dispositivo y se encarga de la gestión del ciclo de vida y el control de los estados del Midlet.⁴

⁴ FROUFE, Quintas Agustín, JORGE, Cárdenas Patricia Op. Cit.. pp. 53.

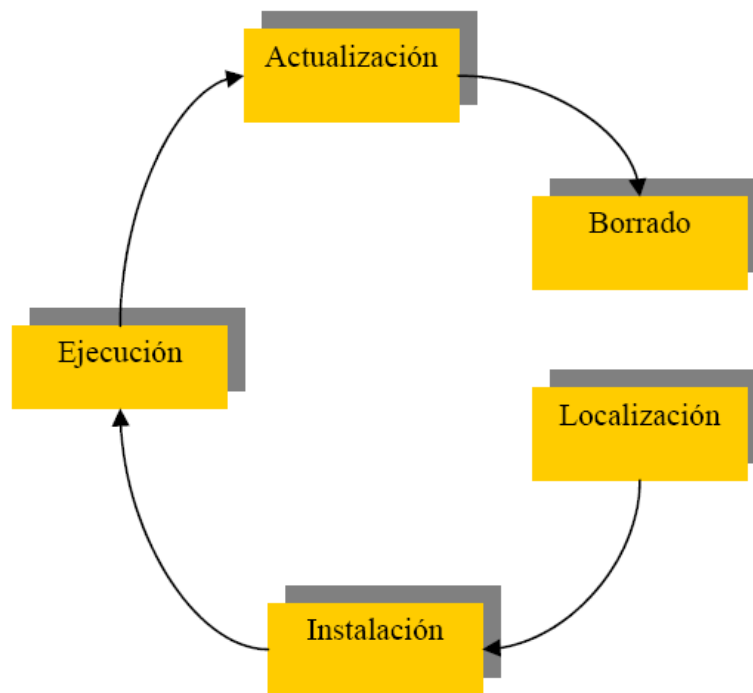


Fig.1.9 Ciclo de Vida del MIDlet.

Localización: Esta fase es la etapa previa a la instalación del MIDlet y es dónde seleccionamos a través del gestor de aplicaciones la aplicación a descargar. Por tanto, el gestor de aplicaciones nos tiene que proporcionar los mecanismos necesarios para realizar la elección del MIDlet a descargar. El AMS puede ser capaz de realizar la descarga de aplicaciones de diferentes maneras, dependiendo de las capacidades del dispositivo. Por ejemplo, esta descarga se puede realizar mediante un cable conectado a un ordenador o mediante una conexión inalámbrica.

Instalación: Una vez descargado el MIDlet en el dispositivo, comienza el proceso de instalación. En esta fase el gestor de aplicaciones controla todo el proceso informando al usuario tanto de la evolución de la instalación como de si existiese algún problema durante ésta. Cuando un MIDlet está instalado en el dispositivo, todas sus clases, archivos y almacenamiento persistente están preparados y listos para su uso.

Ejecución: Mediante el gestor de aplicaciones se inicia la ejecución de los MIDlets. En esta fase, el AMS tiene la función de gestionar los estados del MIDlet en función de los eventos que se produzcan durante esta ejecución.

Actualización: El AMS tiene que ser capaz de detectar después de una descarga si el MIDlet descargado es una actualización de un MIDlet ya presente en el dispositivo. Si es así, debe informar de ello, además de dar la oportunidad de decidir si se desea realizar la actualización pertinente o no.

Borrado: En esta fase el AMS es el encargado de borrar el MIDlet seleccionado del dispositivo. El AMS pedirá confirmación antes de proceder a su borrado e informará de cualquier circunstancia que se produzca.

Hay que indicar que el MIDlet puede permanecer en el dispositivo todo el tiempo que se desee. Después de la fase de instalación, el MIDlet queda almacenado en una zona de memoria persistente del dispositivo MID. El usuario de éste dispositivo es el encargado de decidir en qué momento quiere eliminar la aplicación y así se lo hará saber al AMS mediante alguna opción que éste nos suministre.

Un MIDlet durante su ejecución pasa por 3 estados diferentes:

- Activo: El MIDlet está actualmente en ejecución.
- Pausa: El MIDlet no está actualmente en ejecución. En este estado el MIDlet no debe usar ningún recurso compartido. Para volver a pasar a ejecución tiene que cambiar su estado a Activo.
- Destruído: El MIDlet no está en ejecución ni puede transitar a otro estado. Además se liberan todos los recursos ocupados por el MIDlet.

Estos se corresponden con los siguientes métodos:

startApp()
pauseApp()
destroyApp()

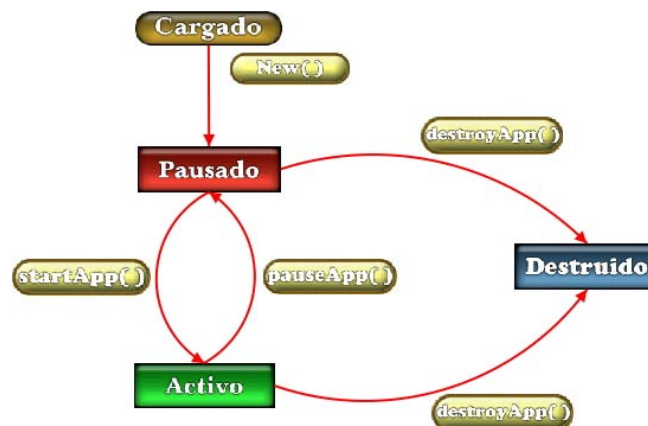


Fig. 1.10 Estados de un MIDlet.

Inicialmente las aplicaciones o MIDlets están en estado de Pausa, en el momento de ejecutarse pasará al estado activo invocando al método `startApp()`, cuando el usuario desee salir de la aplicación invocará al método `destroyApp()` y la aplicación estará en estado destruido.

Esta es la estructura básica de un MIDlet, en base a esto se construye la interfaz de usuario.

```
import javax.microedition.midlet.*;

public class Taller1 extends MIDlet {

    public Taller1( ) {
    }

    void startApp( ) {
    }

    void pauseApp( ) {
    }

    void destroyApp(boolean incondicional) {
    }

}
```

Interfaz de usuario.

Existen dos tipos de interfaces para las aplicaciones en base a CLDC-MIDP, estas son de Alto Nivel y Bajo Nivel.



Fig. 1.11 Interfaces de alto nivel.



Fig. 1.13 Interfaces de bajo nivel.

Por lo general las interfaces de alto nivel van orientada a aplicaciones empresariales, como el ingreso de datos por un formulario, listas de consultas de datos, textbox para recibir noticias vía inalámbrica. Mientras que aplicaciones multimedia, juegos inclusive interfaces de 3D son implementas por las clases de la interfaz de bajo nivel.

El paquete que brinda las clases necesarias para la construcción de interface se llama: `javax.microedition.lcdui.*`;

Algunas de sus clases son las siguientes:

Display

Es la clase que representa el manejador de la pantalla y los dispositivos de entrada. Podemos incluir todos los objetos `Displayable` que queramos. Muestra los componentes de la interfaz de usuario.

Todo `Midlet` debe tener al menos una instancia de la clase `Display`.

```
//Podemos declarar la variable a nivel global  
Display pantalla;
```

```
//y en el constructor declaramos la variable  
pantalla = Display.getDisplay(this);
```

```
//para mostrar un Displayable:  
pantalla.setCurrent(Displayable d)
```

```
// d puede ser un objeto TextBox, Form, etc.
```

Displayable

Corresponde a objetos que capacidad de ser visibles en pantalla. Esta clase es la encargada del manejo de eventos y la inserción de comandos.

Command

Esta clase mantiene información sobre un evento. Sirve para detectar o ejecutar una acción simple.

```
Command salir,
salir = new Command("Salir", Command.EXIT, 1);
```

Tipos de comandos: BACK, CANCEL, EXIT, HELP, ITEM, OK, SCREEN, STOP.

CommandListener

Se implementa en midlets que necesitan recibir los eventos de alto nivel, cualquier midlet que incluya comandos, deberá implementar esta interfaz.

```
public class Ejemplo1 extends MIDlet implements CommandListener{
```

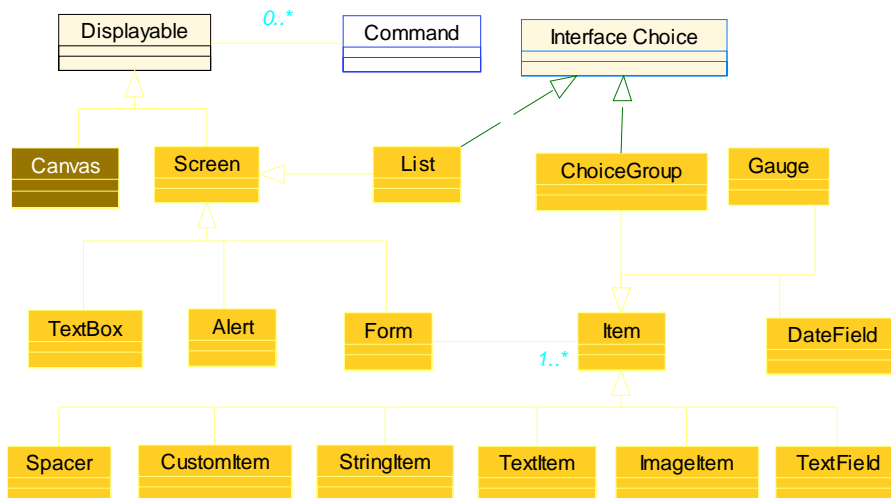


Fig. 1.14 Interfaz de alto nivel.

Las interfaces de alto nivel provienen de la clase Screen, por otro lado, La clase Canvas servirá para crear interfaces de bajo nivel.

Las principales son las List usualmente para generar menú, TextBox para recibir información, Alert para recibir información sobre alguna acción o evento y el Form, única clase a la que puede insertarse items.

Los items son componentes gráficos tales como DateField para el registro de año, mes, fechas y horas, El TextField para editar texto, ChoiceGroup para el manejo de opciones predeterminadas, ImageItem para incluir gráficos.



Fig. 1.15 Manejo de Ítems.

J2ME y las comunicaciones.

Una característica importante que tienen que deben tener los dispositivos que hagan uso de J2ME, más específicamente CLDC/MIDP es que necesitan poseer conexión a algún tipo de red, por lo que la comunicación de estos dispositivos cobra una gran importancia. Esto es así debido a la rápida evolución que han tenido los teléfonos móviles en el sector de las comunicaciones, lo que ha facilitado el desarrollo, por parte de algunas empresas, de herramientas que usaremos para crear las aplicaciones.⁵

Uno de los primeros avances de la telefonía móvil en el sector de las comunicaciones se dio con la aparición de la tecnología WAP. WAP proviene de Wireless Application Protocol o Protocolo de Aplicación Inalámbrica. Es un protocolo con el que se ha tratado de dotar a los dispositivos móviles de un pequeño y limitado navegador web. WAP exige la presencia de una puerta de enlace encargado de actuar como intermediario entre Internet y el terminal. Esta puerta de enlace o gateway es la que se encarga de convertir las peticiones WAP a peticiones web habituales y viceversa. Las páginas que se transfieren en una petición usando WAP no están escritas en HTML, sino que están escritas en WML, un subconjunto de éste. WAP ha sido un gran avance, pero no ha resultado ser la herramienta que se prometía. La navegación es muy engorrosa (la introducción de URLs largas por teclado es muy pesada, además de que cualquier error en su introducción requiere que se vuelva a escribir la dirección

⁵ GÁLVEZ, Rojas Sergio, ORTEGA, Díaz Lucas Op. Cit., pp. 14.

completa por el teclado del móvil). Además su costo es bastante elevado ya que el pago de uso de esta tecnología se realiza en base al tiempo de conexión a una velocidad, que no es digamos, muy buena.

Otra tecnología relacionada con los móviles es SMS. SMS son las siglas de Short Message System (Sistema de Mensajes Cortos). Actualmente este sistema permite comunicación de una manera rápida y sin tener que establecer una comunicación con el receptor del mensaje. Con ayuda de J2ME, sin embargo, podemos realizar aplicaciones de chat o mensajería instantánea.

Los últimos avances de telefonía móvil nos llevan a las conocidas como generación 2 y 2.5 que hacen uso de las tecnologías GSM y GPRS respectivamente. GSM es una conexión telefónica que soporta una circulación de datos, mientras que GPRS es estrictamente una red de datos que mantiene una conexión abierta en la que el usuario paga por la cantidad de información intercambiada y no por el tiempo que permanezca conectado. La aparición de la tecnología GPRS no hace más que favorecer el uso de J2ME y es, además, uno de los pilares sobre los que se asienta J2ME, ya que podemos decir que es el vehículo sobre el que circularán las futuras aplicaciones J2ME. Otras tecnologías que favorecen la comunicación son Bluetooth y las redes inalámbricas que dan conectividad a ordenadores, PDAs y teléfonos móviles. De hecho, una gran variedad de estos dispositivos disponen de soporte bluetooth. Esto nos facilita la creación de redes con un elevado ancho de banda en distancias pequeñas (hasta 100 metros). Es de esperar que todas estas tecnologías favorezcan el uso de J2ME en el mercado de la telefonía móvil.

OTA(Over the Air – a través del aire).

Las aplicaciones realizadas con J2ME están pensadas para que puedan ser descargadas a través de una conexión a internet. El medio empleado para garantizar esta descarga recibe el nombre de OTA (Over the Air), y viene totalmente reflejado en un documento denominado «Over The Air User Initiater Provisioning Recommended Practice», Sun Microsystems, 2002. Una aplicación J2ME está formada por un archivo JAR que es el que contiene a la aplicación en sí y un archivo JAD (Java Archive Descriptor) que contiene diversa información sobre la aplicación.

Los dispositivos deben proporcionar mecanismos mediante los cuales se puedan encontrar los MIDlets que se desean descargar. En algunos casos, los MIDlets estarán disponibles a través de un navegador WAP o a través de una aplicación residente escrita específicamente para identificar MIDlets. Otros mecanismos como Bluetooth, cable serie, etc., pueden ser soportados por el dispositivo.

Requerimientos: El programa encargado de manejar la descarga y ciclo de vida de los MIDlets en el dispositivo se llama Gestor de Aplicaciones o AMS (Application Management Software).

Un dispositivo que posea la especificación MIDP debe ser capaz de:

- Localizar archivos JAD vinculados a un MIDlet en la red.
- Descargar el MIDlet y el archivo JAD al dispositivo desde un servidor usando el protocolo
- HTTP 1.1 u otro que posea su funcionalidad.

- Enviar el nombre de usuario y contraseña cuando se produzca una respuesta HTTP por parte del servidor 401 (Unauthorized) o 407 (Proxy Authentication Required).
- Instalar el MIDlet en el dispositivo.
- Ejecutar MIDlets.
- Permitir al usuario borrar MIDlets instalados.

Localización de la Aplicación: El descubrimiento de una aplicación es el proceso por el cual un usuario a través de su dispositivo localiza un MIDlet. El usuario debe ser capaz de ver la descripción del MIDlet a través de un enlace que, una vez seleccionado, inicializa la instalación del MIDlet. Si éste enlace se refiere a un archivo JAR, el archivo y su URL son enviados al AMS del dispositivo para empezar el proceso de instalación. Si el enlace se refiere a un archivo JAD se realizan los siguientes pasos:

1. El descriptor de la aplicación (archivo JAD) y su URL son transferidos al AMS para empezar la instalación. Este descriptor es usado por el AMS para determinar si el MIDlet asociado puede ser instalado y ejecutado satisfactoriamente.
2. Este archivo JAD debe ser convertido al formato Unicode antes de ser usado. Los atributos del JAD deben ser comprensibles, acorde con la sintaxis de la especificación MIDP, y todos los atributos requeridos por la especificación MIDP deben estar presentes en el JAD.
3. El usuario debería de tener la oportunidad de confirmar que desea instalar el MIDlet. Asimismo debería de ser informado si se intenta instalar una versión anterior del MIDlet o si la versión es la misma que ya está instalada. Si existen problemas de memoria con la ejecución del MIDlet se intentarían solucionar liberando componentes de memoria para dejar espacio suficiente.

Instalación de MIDlets: La instalación de la aplicación es el proceso por el cual el MIDlet es descargado al dispositivo y puede ser utilizado por el usuario.

Cuando existan múltiples MIDlets en la aplicación que se desea descargar, el usuario debe ser avisado de que existen más de uno. Durante la instalación, el usuario debe ser informado del progreso de ésta y se le debe de dar la oportunidad de cancelarla. La interrupción de la instalación debe dejar al dispositivo con el mismo estado que cuando se inició ésta.

Los pasos que el AMS sigue para la instalación de un MIDlet son:

1. Si el JAD fue lo primero que descargó el AMS, el MIDlet debe tener exactamente la misma URL especificada en el descriptor.
2. Si el servidor responde a la petición del MIDlet con un código 401 (Unauthorized) o un 407 (Proxy Authentication Required), el dispositivo debe enviar al servidor las correspondientes credenciales.

3. El MIDlet y las cabeceras recibidas deben ser chequeadas para verificar que el MIDlet descargado puede ser instalado en el dispositivo. El usuario debe ser avisado de los siguientes problemas durante la instalación:

- Si no existe suficiente memoria para almacenar el MIDlet, el dispositivo debe retornar el Código de Estado (Status Code) 901.
- Si el JAR no está disponible en la URL del JAD, el dispositivo debe retornar el Código 907.
- Si el JAR recibido no coincide con el descrito por el JAD, el dispositivo debe retornar el Código 904.
- Si el archivo manifest o cualquier otro no puede ser extraído del JAR, o existe algún error al extraerlo, el dispositivo debe retornar el Código 907.
- Si los atributos “MIDlet-Name”, “MIDlet-Version” y “MIDlet Vendor” del archivo JAD, no coinciden con los extraídos del archivo manifest del JAR, el dispositivo debe retornar el Código 905.
- Si la aplicación falla en la autenticación, el dispositivo debe retornar el Código 909.
- Si falla por otro motivo distinto del anterior, debe retornar el Código 911.
- Si los servicios de conexión se pierden durante la instalación, se debe retornar el Código 903 si es posible.

La instalación se da por completa cuando el MIDlet esté a disposición del usuario en el dispositivo, o no haya ocurrido un error irre recuperable.

Actualización de MIDlets: La actualización se realiza cuando se instala un MIDlet sobre un dispositivo que ya contenía una versión anterior de éste. El dispositivo debe ser capaz de informar al usuario cual es la versión de la aplicación que tiene instalada.

Cuando comienza la actualización, el dispositivo debe informar si la versión que va a instalar es más nueva, más vieja o la misma de la ya instalada y debe obtener verificación por parte del usuario antes de continuar con el proceso.

En cualquier caso, un MIDlet que no posea firma no debe de reemplazar de ninguna manera a otro que sí la tenga.

Ejecución de MIDlets: Cuando un usuario comienza a ejecutar un MIDlet, el dispositivo debe invocar a las clases CLDC y MIDP requeridas por la especificación MIDP. Si existen varios MIDlets presentes, la interfaz de usuario debe permitir al usuario seleccionar el MIDlet que desea ejecutar.

Eliminación de MIDlets: Los dispositivos deben permitir al usuario eliminar MIDlets. Antes de eliminar una aplicación el usuario debe dar su confirmación. El dispositivo

debería avisar al usuario si ocurriese alguna circunstancia especial durante la eliminación del MIDlet. Por ejemplo, el MIDlet a borrar podría contener a otros MIDlets, y el usuario debería de ser alertado ya que todos ellos quedarían eliminados.

Básicamente cualquier aplicación en Java sigue este proceso de desarrollo excepto por las etapas de empaquetamiento y preverificación que es exclusivo de las aplicaciones desarrolladas usando la plataforma J2ME.

1. Desarrollo: En esta fase se va a escribir el código que conforma el MIDlet.
2. Compilación: Se compilará la aplicación haciendo uso de un compilador J2SE.
3. Preverificación: Antes de empaquetar el MIDlet es necesario realizar un proceso de preverificación de las clases Java. En esta fase se realiza un examen del código del MIDlet para ver que no viola ninguna restricción de seguridad de la plataforma J2ME.
4. Empaquetamiento: En esta fase se crea un archivo JAR que contiene los recursos que usa la aplicación, y se crea también un archivo descriptor JAD.
5. Ejecución: Para esta fase se hace uso de los emuladores que permitirán ejecutar el MIDlet.
6. Depuración: Esta última fase permitirá depurar los fallos detectados en la fase anterior del MIDlet.

ANEXO D. DICCIONARIO DE TÉRMINOS

3G. Es un desarrollo mundial de normas y tecnologías de comunicaciones. Este desarrollo permitirá la posibilidad de acceder a servicios multimedia desde un teléfono móvil.

AMS. Application Management System, software que se encuentra en el dispositivo y se encarga de la gestión del ciclo de vida y el control de los estados del Midlet.

BANDA. En la comunicación inalámbrica el término banda se refiere a la frecuencia o gama continua de frecuencias.

BLUETOOTH. Es el nombre común de la especificación industrial IEEE 802.15.1, que define un estándar global de comunicación inalámbrica que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia segura, globalmente y sin licencia de corto rango. Los principales objetivos que se pretende conseguir con esta norma son: Facilitar las comunicaciones entre equipos móviles y fijos, Eliminar cables y conectores entre éstos, ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre nuestros equipos personales. Los dispositivos que con mayor intensidad utilizan esta tecnología son los de los sectores de las telecomunicaciones y la informática personal, como PDAs, teléfonos celulares, ordenadores portátiles, PCs, impresoras y cámaras digitales.

BOLETÍN. Publicación periódica sobre un determinado campo del saber: boletín informativo o de noticias Espacio de radio o televisión dedicado a las noticias de actualidad.

CELULAR. En las comunicaciones inalámbricas el término “celular” se refiere fundamentalmente a la estructura de las redes de transmisión inalámbrica, formadas por células o centros de transmisión. “Celular” también es el nombre del sistema telefónico inalámbrico creado por los Laboratorios Bell, que utilizaba equipos analógicos de radio de baja potencia para la transmisión dentro de las células. El término “teléfono celular” es equivalente a teléfono inalámbrico. Dentro del sector de las comunicaciones inalámbricas, celular se refiere también a los productos y servicios no PCS.

EDGE. La tecnología EDGE (Datos Mejorados para Evolución Global) nace con el objetivo de mejorar la capacidad de transmisión de datos para telefonía móvil de las redes GSM y TDMA. Con estas mejoras, el mercado abre las puertas a la llamada 'telefonía de tercera generación'. Las mejoras se basan en dos puntos principales. Por un lado, el uso de técnicas de modulación de frecuencias diferentes a las que se aplicaban hasta el momento. El segundo punto de mejora consiste en la utilización de nuevos protocolos de comunicación por radio. La implementación de estas mejoras

permite en global un uso más eficiente de los espectros de 800, 900 y 1800 MHz, abriendo el mercado a la telefonía 3G. Un nombre alternativo para la tecnología EDGE es EGPRS (Enhanced General Packet Radio Service).

ERP. Los sistemas de planificación de recursos empresariales (ERPs) son sistemas de información gerenciales que integran y manejan muchas de las prácticas de los negocios asociados con las operaciones de producción y de los aspectos de distribución de una compañía comprometida en la producción de bienes o servicios.

EVDO. Evolution Data Optimized, EvDO por sus siglas en inglés, es un protocolo de banda ancha para la transferencia inalámbrica de datos, perteneciente al estándar CDMA2000, desarrollado por la empresa norteamericana Qualcomm, dueña de la patente CDMA.

EvDO permite alcanzar tasas de transmisión de hasta 2.4 Megabits por segundo, lo que la convierte en una de las tecnologías de mayor importancia en el futuro inmediato de las telecomunicaciones en el que los servicios de valor agregado ganan cada día más importancia. Acceso a Internet de banda ancha móvil, visualización de video y descargas de archivos MP3 y juegos Java en tres dimensiones son algunos de los servicios que permite disfrutar de manera fácil EvDO gracias a las altas velocidades de transferencia de datos que puede registrar; hasta cuarenta veces más que las alcanzadas por las redes de segunda generación.

GDI. El GDI (Graphics Device Interface), es el interfaz de dispositivo gráfico, que contiene todas las funciones y estructuras necesarias que nos permiten comunicar nuestras aplicaciones con cualquier dispositivo gráfico de salida conectado a nuestro ordenador: pantalla, impresora, plotter, etc.

GPRS. General Packet Radio Service o GPRS es una tecnología digital de telefonía móvil.

Es considerada la generación 2.5, entre la segunda generación (GSM) y la tercera (UMTS). Proporciona altas velocidades de transferencia de datos (especialmente útil para conectar a Internet) y se utiliza en las redes GSM.

GSM. Global System for Mobile communications (Sistema Global para las Comunicaciones Móviles), anteriormente conocida como "Group Special Mobile" (GSM, Grupo Especial Móvil) es un estándar mundial para teléfonos móviles digitales. El estándar fue creado por la CEPT y posteriormente desarrollado por ETSI como un estándar para los teléfonos móviles europeos, con la intención de desarrollar una normativa que fuera adoptada mundialmente. El estándar es abierto, no propietario y evolutivo (aún en desarrollo). Es el estándar predominante en Europa, así como el mayoritario en el resto del mundo (alrededor del 70% de los usuarios de teléfonos móviles del mundo en 2001 usaban GSM). GSM difiere de sus antecesores principalmente en que tanto los canales de voz como las señales son digitales. Se ha diseñado así para un moderado nivel de seguridad.

HOTSYNC. El proceso de sincronización, llamado “HotSync”, es la transferencia de la información entre una computadora de escritorio y un dispositivo Palm Powered.

IDE. Por sus siglas en inglés: Integrated Development Environment, es una colección de herramientas de desarrollo de software que distribuyen en forma conjunta y que cooperan estrechamente entre si para facilitar este proceso.

IRDA. Infrared Data Association (IrDA) define un estándar físico en la forma de transmisión y recepción de datos por rayos infrarrojo. IrDA se crea en 1993 entre HP, IBM, Sharp y otros. Esta tecnología, basada en rayos luminosos que se mueven en el espectro infrarrojo. Los estándares IrDA soportan una amplia gama de dispositivos eléctricos, informáticos y de comunicaciones, permite la comunicación bidireccional entre dos extremos a velocidades que oscilan entre los 9.600 bps y los 4 Mbps. Esta tecnología se encuentra en muchos ordenadores portátiles, y en un creciente número de teléfonos celulares, sobre todo en los de fabricantes líderes como Nokia y Ericsson.

J2ME. Es la versión de Java orientada a los dispositivos móviles. Debido a que los dispositivos móviles tienen una potencia de cálculo baja e interfaces de usuario pobres, es necesaria una versión específica de Java destinada a estos dispositivos, ya que el resto de versiones de Java, J2SE o J2EE, no encajan dentro de este esquema. J2ME es por tanto, una versión “reducida” de J2SE.

JAD. El archivo JAD contiene información necesaria para la instalación de los MIDlets contenidos en el archivo JAR.

JAR. Es un archivo comprimido que contiene las clases que ha generado la compilación de nuestro programa. Además puede contener los recursos necesarios para el MIDlet como sonidos, gráficos, etc. Finalmente, contiene un archivo con extensión .mf., es lo que se llama un archivo de manifiesto. Este archivo contiene información sobre las clases contenidas en el archivo JAR.

JAVA. Es un lenguaje de programación creado por Sun Microsystems. Es posible que se utilicen algunas versiones de Java en la creación de servicios inalámbricos. Algunos teléfonos Nokia, como el 3410, 6310i y 7210, permiten descargar aplicaciones Java a través de la conexión WAP. De este modo puedes enriquecer las funciones de tu teléfono con aplicaciones útiles e interesantes, como un reloj mundial, un convertidor de divisas o distintos juegos.

KVM. Kilobyte Virtual Machine, diseñada desde cero para dispositivos con poca memoria, capacidad de proceso limitada y con conexión a red intermitente (inalámbrica): Memoria mínima 128 KB, Procesadores de 16 ó 32 bits RISC o CISC, acepta el mismo conjunto de bytecode (con algunas excepciones) y el mismo formato de ficheros de clase que la JVM, ocupa entre 40 y 80 KB y esta ligada a la configuración CLDC.

LAN. Designa a una red de área local, conocida por sus siglas en inglés LAN (Local Area Network).

LectorBMU. MIDlet que nos permite descargar boletines informativos de diferente índole a un dispositivo móvil ya sea un teléfono celular o un PDA compatible con aplicaciones java y con conexión bluetooth o USB.

LOW END Línea de equipos de bajo costo.

MARCADORES Suele llamarse marcadores, conocidos también como favoritos o, en inglés, bookmarks, a aquellos enlaces a páginas web que hemos almacenado en nuestro navegador web o en alguna aplicación por su interés o para su posterior visualización.

MIDLET. Es el nombre adoptado por estas pequeñas aplicaciones JAVA descargada de la red. Una aplicación que puede utilizarse localmente sin conexión o estableciendo un canal de comunicación vía GSM, GPRS o UMTS para la transferencia de información con servidores remotos. Además, presenta otras características como el almacenaje de contenidos adicionales en los dispositivos donde se instalen.

MIDP. Es el acrónimo de Perfil para dispositivos de información móvil (Mobile Information Device Profile) y nos proporciona un perfil que se apoya en CLDC y que nos va a proporcionar los paquetes y clases necesarios para el desarrollo de nuestras aplicaciones. MIDP está orientado principalmente a teléfonos móviles, aunque existe una implementación para PalmOS (versión 3.5 y superiores) y PocketPC, por lo que es también utilizable en casi cualquier PDA.

MMS. (Servicio de Mensajería Multimedia) es una nueva norma que se está definiendo para su uso en terminales inalámbricos avanzados. Este servicio permite la transmisión en tiempo no real de distintos tipos de contenidos multimedia, como imágenes, audio, secuencias de vídeo, etc.

MOTIF/ X-WINDOWS. Software que se ha convertido en un estándar para Unix de Windows. Básicamente, ofrece la posibilidad de visualizar simultáneamente en pantalla varios procesos en ejecución.

OEM. (Abreviatura del inglés Original Equipment Manufacturer, en español sería Fabricante Original de Equipo). Empresas o personas que adquieren dispositivos al por mayor para ensamblar computadoras o equipos de forma personalizada que presentan con su propio nombre. Los productos OEM pueden ser tanto Hardware como Software.

OPENSOURCE. Es software de dominio público (esto significa sin licencia), cumple todos estos criterios siempre y cuando todo el código fuente esté disponible, y esté reconocido por la OSI y se le permita usar la marca de la misma.

OTA. Las aplicaciones realizadas con J2ME están pensadas para que puedan ser descargadas a través de una conexión a internet. El medio empleado para garantizar esta descarga recibe el nombre de OTA (*Over the Air*), y viene totalmente reflejado en un documento denominado «*Over The Air User Initiater Provisioning Recommended Practice*», Sun Microsystems, 2002.

PALM OS. Palm OS es un sistema operativo para ordenadores de bolsillo contruidos por palmOne. La versión actual es desarrollada por PalmSource, que junto con PalmOne formaron hasta 2003 la compañía Palm. Esta, a su vez, es una subsidiaria de 3Com, que se hizo cargo de esta tecnología al adquirir la empresa madre, US Robotics.

PAN. Una Personal Area Network (PAN / Red de Área Personal) es una red de ordenadores para la comunicación entre distintos dispositivos (tanto ordenadores, puntos de acceso a internet, teléfonos móviles, PDA, dispositivos de audio, impresoras) cercanos al punto de acceso. Estas redes normalmente son de unos pocos metros y para uso personal.

PANTALLA DE CRISTAL LÍQUIDO (LCD). Ofrecen un bajo consumo y suelen ser fáciles de leer. Estas pantallas se fabrican mediante la introducción hermética de un compuesto líquido entre dos cristales y/o un filtro. La pantalla está formada por cientos o miles de puntos que se pueden cargar o no, reflejando o no reflejando la luz y configurando de este modo letras, caracteres y números. Algunas pantallas de cristal líquido llevan un panel electroluminiscente en la parte posterior denominado "retroiluminación".

PANTALLA. Existen diversos tipos de pantallas: Pantalla iluminada - Se refiere a la luz de la pantalla de los teléfonos, que se enciende en distintas condiciones. Pantalla de cristal líquido (LCD) - Visualización fácil en todas las condiciones de iluminación. Utiliza segmentos oscuros sobre un fondo más claro. Los caracteres de este tipo de pantalla son negros sobre fondo gris. Se iluminan mediante una pequeña luz que brilla sobre la pantalla desde la parte de arriba. Pantalla de cristal líquido (LCD) en color - Invierte la imagen de la pantalla, haciendo posible la creación de segmentos luminosos en color. Ofrece unos caracteres más claros, nítidos y marcados que son más fáciles de leer. Pantalla emisora de luz (LED) - Las pantallas emisoras de luz son grandes y brillantes. Ofrecen una excelente visibilidad con luz solar directa y en la oscuridad. Pantalla fluorescente de vacío (VFD) - Esta pantalla conserva la visibilidad con luz solar directa y es muy visible en la oscuridad. Se ve sin distorsión desde un gran número de ángulos de visión y permanece totalmente operativa en una amplia gama de temperaturas.

PDA. (Personal Digital Assistant o Ayudante personal digital) es un dispositivo de pequeño tamaño que combina un ordenador, teléfono/fax, Internet y conexiones de red. A los PDAs también se les llama palmtops, hand held computers (ordenadores de mano) y pocket computers (ordenadores de bolsillo). Un PDA típico puede funcionar como teléfono móvil, fax, explorador de internet, organizador personal, GPS, etc.

POCKET PC. PocketPC es un ordenador de bolsillo, también llamado PDA (Personal Digital Assistant). Se trata de un pequeño ordenador, diseñado para ocupar el mínimo espacio y ser fácilmente transportable que ejecuta el sistema operativo Windows CE de Microsoft, el cual le proporciona capacidades similares a los PCs de escritorio.

POSE. Es un emulador de la plataforma Palm que corre bajo Windows, Mac y actualmente también sobre Unix. Con el emulador se pueden ejecutar aplicaciones específicas del sistema Palm OS desde un ordenador, así como probar las distintas versiones (OS 3.5, OS 4.1, OS 5, etc.), evaluar el funcionamiento de una determinada aplicación, etc.

RED. En las comunicaciones inalámbricas el término red se refiere a la infraestructura que permite la transmisión de las señales inalámbricas. La red sirve para unir entre sí los distintos elementos y compartir los recursos.

SMARTPHONES. O teléfono inteligente, es un teléfono móvil con capacidades extras concretas tal como la organización de la información personal. Suelen ejecutar un sistema operativo que les permite cargar aplicaciones diferentes a las que vienen de serie. Por definición, el smartphone se opone al handheld (del inglés “sostenido en la mano”, o al PDA (Personal Data Assistant, o asistente de datos personales), aunque, a fin de cuentas, realizan funciones similares. Los smartphones suelen tener menos memorias y menos posibilidades de conectividad y expansión.

SMS. El Servicio de Mensajes Cortos (SMS) es una función disponible en algunos teléfonos inalámbricos que permite al usuario enviar y/o recibir breves mensajes alfanuméricos.

SYMBIAN. Sistema operativo que fue producto de la alianza de varias empresas de telefonía celular, dentro de las que se encuentran Nokia, Sony Ericsson, Samsung, Siemens, Arima, Benq, Fujitsu, Lenovo, LG, Motorola, Mitsubishi Electric, Panasonic, Sharp, etc.

TIC. Es el acrónimo de Tecnologías de la Información y la Comunicación. Sin embargo, no existe una definición precisa y uniforme del término. Las TIC (Tecnologías de la Información y Comunicaciones) son las tecnologías que se necesitan para la gestión y transformación de la información, y muy en particular el uso de ordenadores y programas que permiten crear, modificar, almacenar, proteger y recuperar esa información.

UMTS. Es una tecnología apropiada para una gran variedad de usuarios y tipos de servicios, y no solamente para usuarios muy avanzados, UMTS ofrece: Facilidad de uso y bajos costos: UMTS proporcionará servicios de uso fácil y adaptable para abordar las necesidades y preferencias de los usuarios, amplia gama de terminales para realizar fácil acceso a los distintos servicios, bajo coste de los servicios para asegurar un mercado masivo.

USB ON THE GO. Se trata de una mejora del interfaz USB, permitiendo elegir el estado de cada puerto USB (actuar como dispositivo de datos o como servidor de datos). Así pues, es posible transferir todos los datos desde un puerto USB actuando como servidor hasta otro actuando como dispositivo sin necesidad de usar un ordenador. Además es posible cambiar en cualquier momento el perfil de servidor a dispositivo o viceversa.

USB. El Bus de Serie Universal (USB, de sus siglas en inglés Universal Serial Bus) es una interfaz que provee un estándar de bus serie para conectar dispositivos a un ordenador personal (generalmente a un PC). Un sistema USB tiene un diseño asimétrico, que consiste en un solo servidor y múltiples dispositivos conectados en serie para ampliar la gama de conexión, en una estructura de árbol utilizando concentradores especiales. Se pueden conectar hasta 127 dispositivos a un sólo servidor, pero la suma debe incluir a los concentradores también, así que el total de dispositivos realmente usables es algo menor.

WAP. Wireless Application Protocol o WAP (protocolo de aplicaciones inalámbricas) es un estándar abierto internacional para aplicaciones que utilizan las comunicaciones inalámbricas, por ejemplo acceso a servicios de Internet desde un teléfono móvil. Se trata de la especificación de un entorno de aplicación y de un conjunto de protocolos de comunicaciones para normalizar el modo en que los dispositivos inalámbricos, se pueden utilizar para acceder a correo electrónico, grupo de noticias y otros.

W-CDMA. Es el acrónimo de Wideband-Code Division Multiple Access (Acceso múltiple de banda ancha por división de código) Constituye una tecnología móvil inalámbrica de tercera generación que aumenta las tasas de transmisión de datos de los sistemas GSM utilizando la interfaz aérea CDMA en lugar de TDMA (Acceso Múltiple por División de Tiempo) y por ello ofrece velocidades de datos mucho más altas en dispositivos inalámbricos móviles y portátiles que las que se suelen ofrecer en el mercado actual. W-CDMA es la conexión 3G para GSM, mientras que EV-DO lo es para CDMA.

WEP. Acrónimo de Wired Equivalent Privacy, es el sistema de cifrado incluido en el estándar IEEE 802.11 como protocolo para redes Wireless que permite cifrar la información que se transmite. Proporciona cifrado a nivel 2. Está basado en el algoritmo de cifrado RC4, y utiliza claves de 64 bits (40 bits más 24 bits del vector de inicialización IV) o de 128 bits (104 bits más 24 bits del IV).

WI-FI. (O Wi-fi, WiFi, Wifi, wifi) es un conjunto de estándares para redes inalámbricas basados en las especificaciones IEEE 802.11. Wi-Fi se creó para ser utilizada en redes locales inalámbricas, pero es frecuente que en la actualidad también se utilice para acceder a Internet. Wi-Fi es una marca de la Wi-Fi Alliance (anteriormente la Wireless Ethernet Compatibility Alliance), la organización comercial que prueba y certifica que los equipos cumplen los estándares IEEE 802.11x.

WLAN. (Wireless Local Area Network) es un sistema de comunicación de datos inalámbrico flexible muy utilizado como alternativa a las redes LAN cableadas o como extensión de éstas. Utiliza tecnología de radiofrecuencia que permite mayor movilidad a los usuarios al minimizar las conexiones cableadas. Las WLAN van adquiriendo importancia en muchos campos, como almacenes o para manufactura, en los que se transmite la información en tiempo real a una terminal central. También son muy populares en los hogares para compartir el acceso a Internet entre varias computadoras.

ANEXO E. ENCUESTA DE CAMPO EN FES ARAGÓN



UNIVERSIDAD NACIONAL
AUTÓNOMA DE MÉXICO
FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN



FORMATO DE ENCUESTA DE CAMPO

San Juan de Aragón, Estado de México a De

de 200 7

CARRERA _____

Hola, estoy realizando una encuesta para determinar que porcentaje de estudiantes cuenta con algún dispositivo electrónico y si estaría interesado en recibir información de índole académica o cultural por ese medio.

1. CUENTAS CON UN DISPOSITIVO ELECTRÓNICO SI NO

SI LA PREGUNTA ES AFIRMATIVA ENTONCES CONTINUAR CON LA PREGUNTA DOS. EN CASO CONTRARIO, TERMINAR LA ENCUESTA

2 ¿DE QUE TIPO?

TELÉFONO CELULAR AGENDA ELECTRÓNICA (PDA)

3. ¿QUÉ MODELO? _____

4. ¿TU DISPOSITIVO MÓVIL TIENE TECNOLOGÍA BLUETOOTH?

SI NO NO SE

5. ¿TU DISPOSITIVO MÓVIL TIENE ALGÚN OTRO MEDIO DE TRANSFERENCIA (IRDA, USB)?

SI NO NO SE

6. ¿TU DISPOSITIVO MÓVIL EJECUTA APLICACIONES JAVA?

SI NO NO SE

7. ¿TE GUSTARÍA RECIBIR EN TU DISPOSITIVO MÓVIL INFORMACIÓN DE ÍNDOLE ACADÉMICA O CULTURAL SIN COSTO ALGUNO?

SI NO

GRACIAS POR TU TIEMPO.

ENCUESTO: _____

Se llevó a cabo una investigación para determinar la proporción de estudiantes que cuentan con un dispositivo electrónico (celular o PDA), siendo necesario calcular el tamaño de muestra requerido trabajando con un intervalo de confianza de 95% y con un margen de error del 5%. La investigación se llevó a cabo en una población de 15000 alumnos.¹

Según las formulas de determinación de muestras y error tenemos que:

DETERMINACIÓN DE MUESTRAS Y ERROR

N = POBLACIÓN

N = TAMAÑO DE LA MUESTRA

P = PROBABILIDAD A FAVOR DEL EVENTO

Q = PROBABILIDAD EN CONTRA DEL EVENTO

S = ERROR ESTANDAR

S² = VARIANZA

| | | MUESTRA | |
|------------------------|------|---|-------------------------|
| | | FINITA | INFINITA |
| INTERVALO DE CONFIANZA | 99 % | $n = \frac{9 P Q N}{S^2 (N-1) + 9 P Q}$ | $n = \frac{9 P Q}{S^2}$ |
| | 95 % | $n = \frac{4 P Q N}{S^2 (N-1) + 4 P Q}$ | $n = \frac{4 P Q}{S^2}$ |
| | 68 % | $n = \frac{P Q N}{S^2 (N-1) + P Q}$ | $n = \frac{P Q}{S^2}$ |

Fig. 1.1 Formulas para cálculo de muestra

| | | ERROR | |
|------------------------|------|---|------------------------------|
| INTERVALO DE CONFIANZA | 99 % | $S = \sqrt{\frac{9 P Q}{n} \left(\frac{N-n}{N-1} \right)}$ | $S = \sqrt{\frac{9 P Q}{n}}$ |
| | 95 % | $S = \sqrt{\frac{4 P Q}{n} \left(\frac{N-n}{N-1} \right)}$ | $S = \sqrt{\frac{4 P Q}{n}}$ |
| | 68 % | $S = \sqrt{\frac{P Q}{n} \left(\frac{N-n}{N-1} \right)}$ | $S = \sqrt{\frac{P Q}{n}}$ |

Fig. 1.2 Formulas para cálculo de error

¹ Estimado de alumnos sin considerar población flotante obtenidos de la unidad de planeación correspondientes al periodo 2004-2005

Considerando una población de 15 000 estudiantes:
Para una población infinita.

$$n = \frac{n = 4 p q}{s^2} = \frac{4 (50 X 50)}{25} = \mathbf{400 \text{ Entrevistas}}$$

Para una población finita.

$$n = \frac{4 p q N}{s^2 (N-1) + 4 p q} = \frac{4(50 X 50) (15000)}{25 (15000-1)+4(50 X 50)} = 389 \text{ Entrevistas}$$

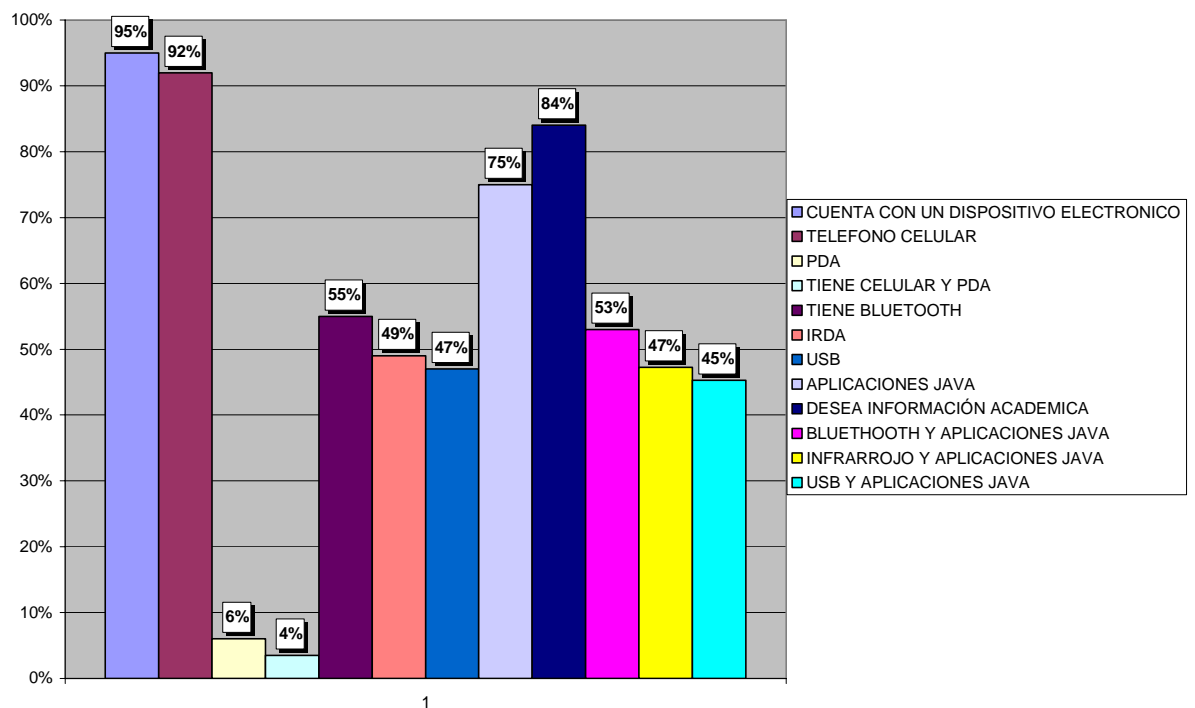
Debido al mínimo de diferencia entre las entrevistas a aplicar a una población infinita o a una finita se optó por realizar el número de encuestas para una población infinita para mejor certeza en los resultados.

Se llevaron a cabo 400 encuestas de campo en FES Aragón a los alumnos de las diferentes carreras impartidas en el plantel entre el 19 de febrero y el 22 de febrero de 2007, concluyendo que la mayoría de los alumnos cuentan con un dispositivo electrónico (teléfono celular) y que la mayoría tiene la infraestructura necesaria para transferir la aplicación LectorBMU y ejecutarla, también se concluyó que la mayoría de los alumnos si desearía recibir este tipo de aplicación de carácter académico a su celular.

A continuación se incluyen las graficas de los resultados obtenidos.

| TOPICO | % | ENCUESTAS |
|---------------------------------------|-----|-----------|
| CUENTA CON UN DISPOSITIVO ELECTRÓNICO | 95% | 380 |
| TELÉFONO CELULAR | 92% | 369 |
| PDA | 6% | 25 |
| TIENE CELULAR Y PDA | 4% | 14 |
| TIENE BLUETOOTH | 55% | 218 |
| IRDA | 49% | 197 |
| USB | 47% | 186 |
| APLICACIONES JAVA | 75% | 301 |
| DESEA INFORMACIÓN ACADÉMICA | 84% | 336 |
| BLUETHOOTH Y APLICACIONES JAVA | 53% | 212 |
| INFRARROJO Y APLICACIONES JAVA | 47% | 189 |
| USB Y APLICACIONES JAVA | 45% | 181 |

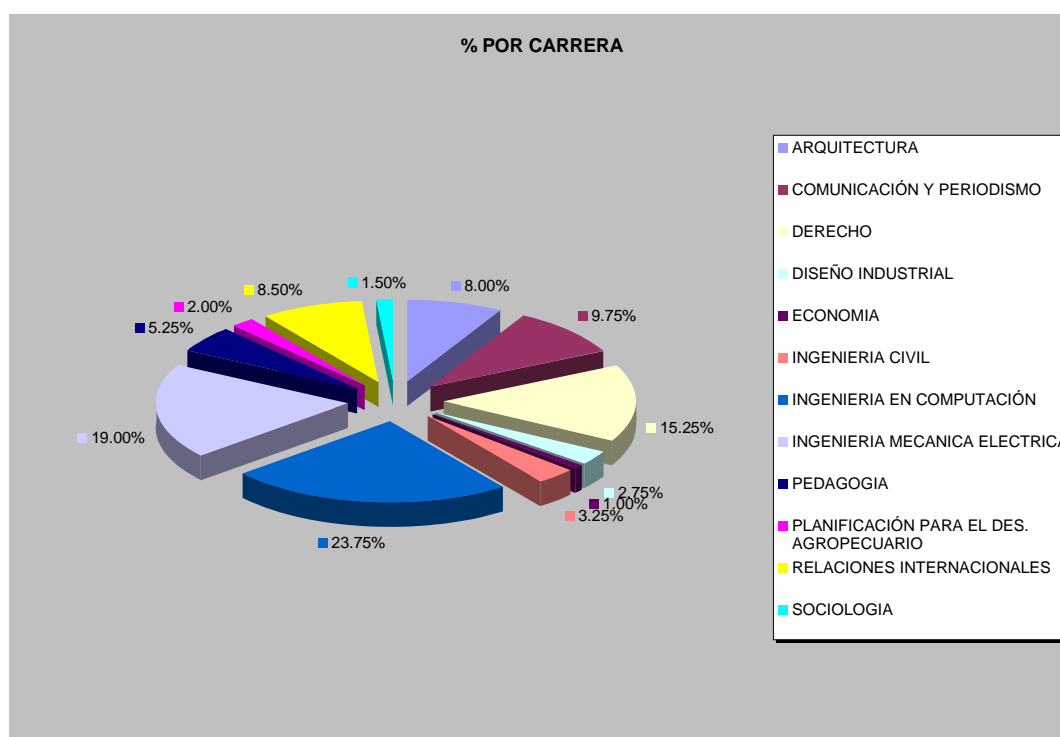
Tabla 1.1 Resultados de encuestas en FES Aragón.



Gráfica 1.1 Resultados de encuestas en FES Aragón.

| CARRERA | % | ENCUESTAS |
|---|--------|-----------|
| ARQUITECTURA | 8.00% | 32 |
| COMUNICACIÓN Y PERIODISMO | 9.75% | 39 |
| DERECHO | 15.25% | 61 |
| DISEÑO INDUSTRIAL | 2.75% | 11 |
| ECONOMIA | 1.00% | 4 |
| INGENIERIA CIVIL | 3.25% | 13 |
| INGENIERIA EN COMPUTACIÓN | 23.75% | 95 |
| INGENIERIA MECANICA ELÉCTRICA | 19.00% | 76 |
| PEDAGOGIA | 5.25% | 21 |
| PLANIFICACIÓN PARA EL DES. AGROPECUARIO | 2.00% | 8 |
| RELACIONES INTERNACIONALES | 8.50% | 34 |
| SOCIOLOGIA | 1.50% | 6 |

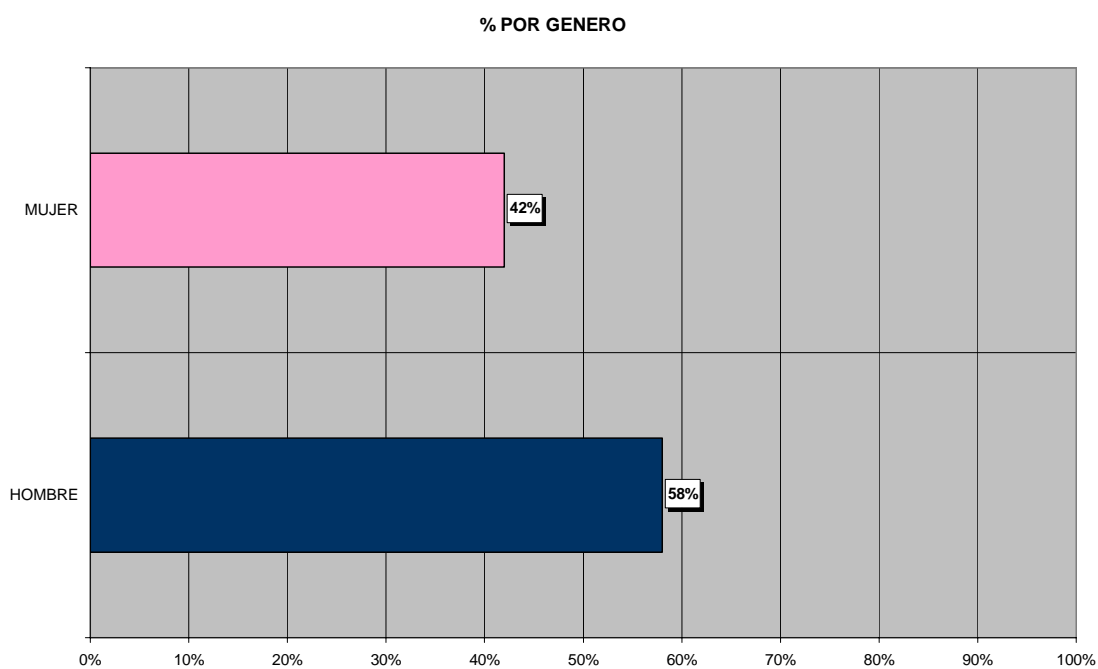
Tabla 1.2 Participación porcentual por carrera para la realización de encuestas.



Gráfica 1.2 Participación porcentual por carrera para la realización de encuestas.

| GENERO | % | ENCUESTAS |
|--------|-----|-----------|
| HOMBRE | 58% | 232 |
| MUJER | 42% | 168 |

Tabla 1.3 Participación porcentual por género para la realización de encuestas.



Gráfica 1.3 Participación porcentual por género para la realización de encuestas.

BIBLIOGRAFÍA

GARCÍA, Córdova (2005), *La tesis y el trabajo de tesis: Recomendaciones metodológicas para la elaboración de los trabajos de tesis*, México, Limusa.

FROUFE, Quintas Agustín (2003), *JAVA 2, Manual de usuario y tutorial*, Madrid, Alfaomega.

FROUFE, Quintas Agustín, JORGE, Cárdenas Patricia (2004), *J2ME Java 2 Micro Edition: Manual de usuario y tutorial*, Madrid, Alfaomega.

SOMMERVILLE, Ian (2002), *Ingeniería de Software*, México, Pearson Educación.

GÁLVEZ, Rojas Sergio, ORTEGA, Díaz Lucas (2003), *Java a Tope: J2ME (Java 2 Micro Edition)* Madrid, Universidad de Malaga

GARCÍA, Serrano Alberto (2003) *Programación de juegos para móviles con J2ME*, USA, Creative Commons.

WEBB, Chris (2002), *Wireless Programming with J2ME: Cracking the Code*, USA, Hungry Minds.

HERNANDO, Rábanos J. M. (2001) *Comunicaciones móviles de Tercera Generación. Segunda Edición*, España, Telefónica Móviles España S.A.

ENLACES:

“Sitios de fabricantes de celulares”

<http://forum.nokia.com/>

<http://www.motocoder.com/>

<http://developer.sonyericsson.com/>

“Dispositivos móviles”

<http://www.pdamexico.net/>

“Empresas de aplicaciones móviles”

<http://www.msppmovil.net/>

<http://www.accsys.com.mx/>

“Plataformas”

<http://www.todo-linux.com/>

<http://www.microsoft.com/spanish/>

“Página principal de Sun sobre J2ME”

<http://java.sun.com/j2me/>

“Diccionario”

<http://es.wikipedia.org/wiki/Portada>