



SOFTWARE EN DISPOSITIVOS MÓVILES

Celeste Campo
celeste@it.uc3m.es

Parte de este material se basa en transparencias de Natividad Martínez Madrid (nati@it.uc3m.es)



ÍNDICE

- Introducción
- Sistemas operativos
- Interfaces de usuario
- Interfaces de comunicaciones
- Nuevas funcionalidades

BIBLIOGRAFÍA

- *Mobile and Wireless Design Essentials*. Mallick, Martyn. Wiley [2003]. L/D 621.396.4 MAL. Capítulo 1, 2 y 3.
- *Pervasive computing handbook*. Hansmann, Uwe. Springer [2001]. L/D 621.39:004 PER.



INTRODUCCIÓN

- Multitud de nuevos dispositivos asociados a los usuarios: agendas electrónicas, móviles, buscas,...
- Multitud de nuevos dispositivos con capacidad de computación: electrodomésticos, electrónica de consumo,...
- Estos dispositivos tienen capacidad de comunicación gracias a los nuevos protocolos inalámbricos:
 - Bluetooth.
 - WLAN.
- Los usuarios a través de estos dispositivos pueden acceder a sus servicios tradicionales y a nuevos servicios.
- Aparece el m-commerce, el m-learning, el m-business, etc.
- Desarrollo de las aplicaciones para estos nuevos dispositivos presenta nuevos retos:
 - Limitaciones en capacidad de proceso y memoria.
 - Diferentes interfaces con el usuario.
 - Varias tipos de conectividad: intermitente, bajo ancho de banda.



SISTEMAS OPERATIVOS

- Dispositivos personales:
 - Windows CE
 - PalmOS
 - Symbian
 - Linux
- Electrónica de consumo y electrodomésticos:
 - Sistemas operativos propietarios.



SISTEMAS OPERATIVOS

- **Windows CE:**
 - Versión reducida de Windows para dispositivos limitados, fundamentalmente PDAs, en la actualidad también teléfonos móviles.
 - PDAs más potentes (133-206 Mhz 16-32 MB RAM) y más grandes y pesados (180-300 gr).
- **PalmOS:**
 - Sistema operativo pensado exclusivamente para PDAs.
 - Menos sofisticación y prestaciones.
 - 80% cuota de mercado, más de 7000 aplicaciones.
 - Necesidades asequibles de potencia (16-33 Mhz 2-8 MB RAM).



SISTEMAS OPERATIVOS

- **Symbian:**
 - Empresa fundada por Nokia, Motorola, Ericsson y Psion.
 - Su objetivo era crear un sistema operativo para dispositivos inalámbricos, especialmente teléfonos móviles.
 - Sistema operativo de 32 bits con características de tiempo real y multitarea.
- **Linux:**
 - Mismo software que en el PC o servidor pero en un dispositivo limitado.
 - Ventajas de ser software libre: disponibilidad de fuentes, modificación y adaptación del sistema operativo (a medida).
 - Se comercializan PDAs y móviles con Linux y también existen distribuciones para instalar sobre otros sistemas.



SISTEMAS OPERATIVOS

PalmOS

Symbian



Windows CE



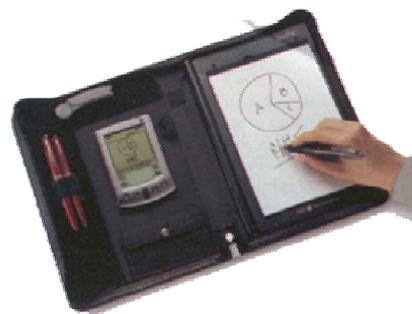
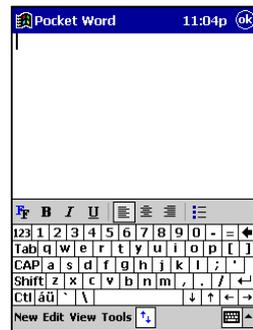
Linux



INTERFACES DE USUARIO

- Interfaces de entrada:
 - Pantallas sensibles (touch screen):
 - Lápiz especial.
 - Reconocimiento de escritura o teclado simulado.
 - Teclado:
 - Más seguro y rápido.
 - Teclado reducido o externo (plegable, de goma!).
 - Keypad:
 - Datos numéricos y caracteres asignados a números.
 - Sistema T9
 - Reconocimiento de voz.
 - Tracking balls, botones, etc.

INTERFACES DE USUARIO



INTERFACES DE USUARIO

- Interfaces de salida:
 - Pantalla
 - Pequeña, pocas líneas.
 - Poca disponibilidad de gráficos, tipos de letra, etc.
 - Leds
 - Actividad de red, estado de la batería.
 - Audio
 - Vibrador



INTERFACES DE COMUNICACIONES

- Dispositivos ofrecen varios interfaces de comunicaciones:
 - WAN:
 - GSM, GPRS, UMTS.
 - LAN:
 - WiFi.
 - PAN:
 - Bluetooth, IrDA.
- Algunos de estos interfaces vienen integrados en el propio dispositivos y otros se incluyen a través de tarjetas de expansión (CF/SD WiFi – CF/SD Bluetooth).



FUNCIONALIDADES AÑADIDAS

- Se introducen nuevas funcionalidades:
 - Varias interfaces de comunicaciones.
 - Cámara.
 - Reproductores de mp3.
 - Grabadores de voz.
 - GPS
- Convergencia entre dispositivos:
 - PDAs convergen a ser teléfonos móviles.
 - Teléfonos móviles convergen a ser PDAs.
 - Sustituyen a las cámaras digitales.
 - Sustituyen a los reproductores de audio.
 - Sustituyen a sistemas GPS.
 - ...



INTRODUCCIÓN A JAVA 2, MICRO EDITION

Celeste Campo
celeste@it.uc3m.es

Parte de este material se basa en transparencias de Natividad Martínez Madrid (nati@it.uc3m.es)



ÍNDICE

- Introducción.
- Arquitectura:
 - Máquinas Virtuales.
 - Configuraciones.
 - Perfiles.
- MIDP/CLDC/KVM:
 - CLDC/KVM.
 - MIDP.

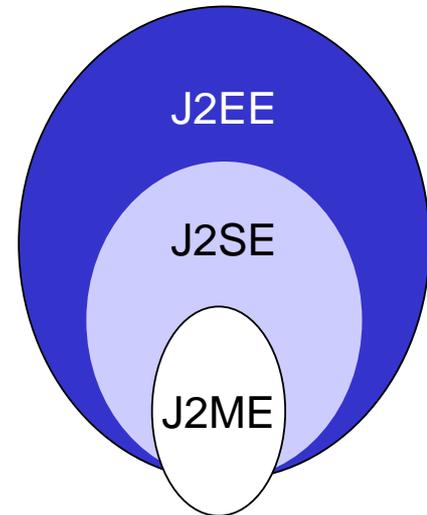
BIBLIOGRAFÍA

- *Wireless Java Programming with Java 2 Micro Edition*. Feng, Yu and Zhu, Jun. SAMS [2001] . L/D 004.438 JAVA FEN. Capítulo 2 y 3.
- <http://java.sun.com/j2me>



INTRODUCCIÓN

- Sun ha estructurado la tecnología Java 2 dirigiéndose a sectores distintos:
 - Java 2 Enterprise Edition (J2EE):
 - Soluciones de empresa: e-commerce, e-business.
 - Java 2 Standard Edition (J2SE):
 - Soluciones de PCs de sobremesa: applets, aplicaciones de usuario.
 - Java 2 Micro Edition (J2ME):
 - Dispositivos de consumo y embebidos
 - Dispositivos móviles
- También Java Card.
 - Tarjetas inteligentes.
- Diferentes API's y VM, pero siempre el lenguaje de programación es Java.



¿POR QUÉ J2ME?

- Éxito de la telefonía móvil:
 - 1 billón de teléfonos móviles en 2003.
 - Resultados de la implantación de NTT DoCoMo.
- J2ME proporciona :
 - Una plataforma estándar para el desarrollo de aplicaciones en dispositivos limitados.
 - Gran número de programadores Java: facilidad y rapidez en el desarrollo de estas nuevas aplicaciones.
 - Portabilidad de las aplicaciones entre diferentes dispositivos y distintos fabricantes.



UN POCO DE HISTORIA

- Oak (Proyecto Green) (1990):
 - Software para dispositivos electrónicos, precursor de Java.
- JavaCard (1996)
 - Plataforma Java para tarjetas inteligentes (Smart Cards).
 - CPU: 8-16 bits; 1-5Mhz.
 - Memoria: 1.2K RAM, 32K memoria no volátil.
 - Todavía activa, no integrada en J2ME.
- PersonalJava (1997)
 - Dispositivos conectados con interfaces de usuario (set-top boxes, etc).
 - Incorporado en el Personal Profile de J2ME.
- EmbeddedJava (1998)
 - Dispositivos embebidos con funcionalidad dedicada y restricciones de memoria (control automóvil)
 - Incorporado en un perfil CDC.



¿QUÉ ES Y QUÉ NO ES J2ME?

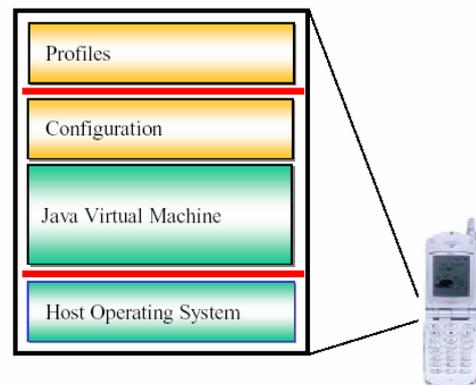
- Nueva arquitectura para diseñar aplicaciones Java en dispositivos limitados.
- J2ME abarca un gran tipo de dispositivos limitados no sólo teléfonos móviles.
- En el mundo de los sistemas móviles:
 - J2ME es complementaria, NO una alternativa a:
 - WAP.
 - i-Mode.
 - J2ME añade:
 - Mayor riqueza de contenidos.
 - Descarga de software en dispositivos móviles:
 - Personalización de servicios.
 - Nuevos servicios proporcionados por terceras partes.

ARQUITECTURA



ARQUITECTURA

- J2ME dirigido a un amplio rango de dispositivos.
- Para conseguir flexibilidad y adaptación, J2ME se estructura en tres niveles:
 - Máquina virtual
 - Mínimo conjunto de clases disponibles.
 - Engloba un segmento horizontal de mercado.
 - Configuración
 - Clases adicionales para un segmento vertical de mercado.
 - Perfil
- Un dispositivo puede soportar múltiples perfiles.





MÁQUINAS VIRTUALES

- Ligadas a una configuración.
- Existen dos VM en la actualidad:
 - CVM: C Virtual Machine.
 - KVM: “Kilo” Virtual Machine.
- CVM:
 - Orientada a dispositivos embebidos y electrónica de consumo (set-top box, TV digital, electrodomésticos,...).
 - Misma funcionalidad que JVM con:
 - Mejor uso de la memoria.
 - Procesadores de 32 bits.
 - Ligada a la configuración CDC.



MÁQUINAS VIRTUALES

- KVM:
 - Antecedentes: Spotless (VM para PalmOS).
 - 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.
 - Ligada a la configuración CLDC.



CONFIGURACIONES

- ¿Qué es una configuración?
 - Mínimo conjunto de clases disponibles en una categoría de dispositivos. Las categorías se establecen según requisitos similares de memoria y procesamiento.
 - Asociada a un máquina virtual.
- Las configuraciones se especifican vía la iniciativa *Java Community Process* (JCP) que genera los correspondientes *Java Specification Report* (JSR).
- Existen dos configuraciones actualmente:
 - Connected, Limited Device Configuration (CLDC).
 - Connected Device Configuration (CDC).



CDC

Connected Device Configuration

- Orientado a dispositivos con:
 - 512 KB de ROM.
 - 256 KB de RAM.
 - Conexión a red (fija).
 - Soporte completo a la especificación de JVM.
 - Interfaz de usuario relativamente limitado.
- Especificado en JSR 36 (definición de nueva versión JSR 218).
- Ejemplos: Internet screenphones, DTV set-top boxes y sistemas telemáticos de automóviles.
- Iniciativas anteriores: PersonalJava, JavaTV, JavaPhone.



CLDC

Connected Limited Device Configuration

- Orientado a dispositivos con:
 - 160 KB a 512 KB de memoria disponible para Java.
 - 16 MHz a 32 MHz.
 - Limitaciones de consumo (baterías).
 - Conectividad a red (inalámbrica).
 - Restricciones importantes en el interfaz de usuario.
- Especificado en el JSR 30 (CLDC 1.0) y JSR 139 (CLDC 1.1).
- Especificación CLDC 1.0/1.1 disponible:
 - Sun proporciona una implementación de referencia de CLDC sobre KVM, para Linux, Windows y Solaris.
 - Principales fabricantes de móviles la implementan en la mayoría de sus modelos (Nokia, Siemens, Samsung,...)



PERFILES

- Conjunto de clases Java que complementan una configuración para un conjunto específico de dispositivos (segmento vertical).
- Los perfiles permiten la portabilidad de aplicaciones J2ME entre diferentes dispositivos.
- Las perfiles se especifican vía la iniciativa *Java Community Process* (JCP) que genera los correspondientes *Java Specification Report* (JSR).
- A las configuraciones y perfiles se les añade también nuevas funcionalidades definiendo APIs opcionales:
 - Siguen el mismo proceso de definición que los perfiles.
 - No todos los dispositivos tienen que implementarlos.



PERFILES Y APIs CDC

- **Perfiles sobre CDC:**
 - Foundation Profile (JSR 46):
 - Perfil básico para dispositivos sin interfaz gráfico.
 - Personal Basis Specification (JSR 129):
 - Perfil gráfico para dispositivos con interfaz gráfico básico.
 - Personal Profile (JSR 62):
 - Perfil gráfico basado en AWT (dispositivos con interfaz gráfico).
 - Evolución del Personal Java.
- **Paquetes opcionales sobre CDC:**
 - RMI Optional Package (JSR 66):
 - Subconjunto de J2SE RMI.
 - JDBC Optional Package (JSR 169):
 - Soporte JDBC en dispositivos CDC.



PERFILES Y APIs CLDC

- **Perfiles sobre CLDC:**
 - Mobile Information Device Profile (JSR 37, JSR 118):
 - Perfil para dispositivos inalámbricos: móviles, PDAs,...
 - Information Module Profile (JSR 195):
 - Perfil para dispositivos con interfaz gráfica limitada: parquímetros, alarmas,...
- **Paquetes opcionales sobre CLDC:**
 - Java Technology for the Wireless Industry (JSR 185):
 - Específico para teléfonos móviles de siguiente generación.
 - Wireless Messaging API (JSR 120, JSR 205):
 - Acceso a sistemas de envío de mensajes (SMS).
 - Mobile Media API (JSR 135):
 - Acceso y reproducción de recursos multimedia (audio, video).
 - Security and Trust Services (JSR 177):
 - Mejora la seguridad añadiendo APIs criptográficas, firmas digitales, gestión de credenciales.
 - Web Services APIs (JSR 172):
 - Desarrollo de clientes Web en dispositivos móviles (web services).
 - Bluetooth API (JSR 82):
 - Desarrollo de aplicaciones que usan Bluetooth.

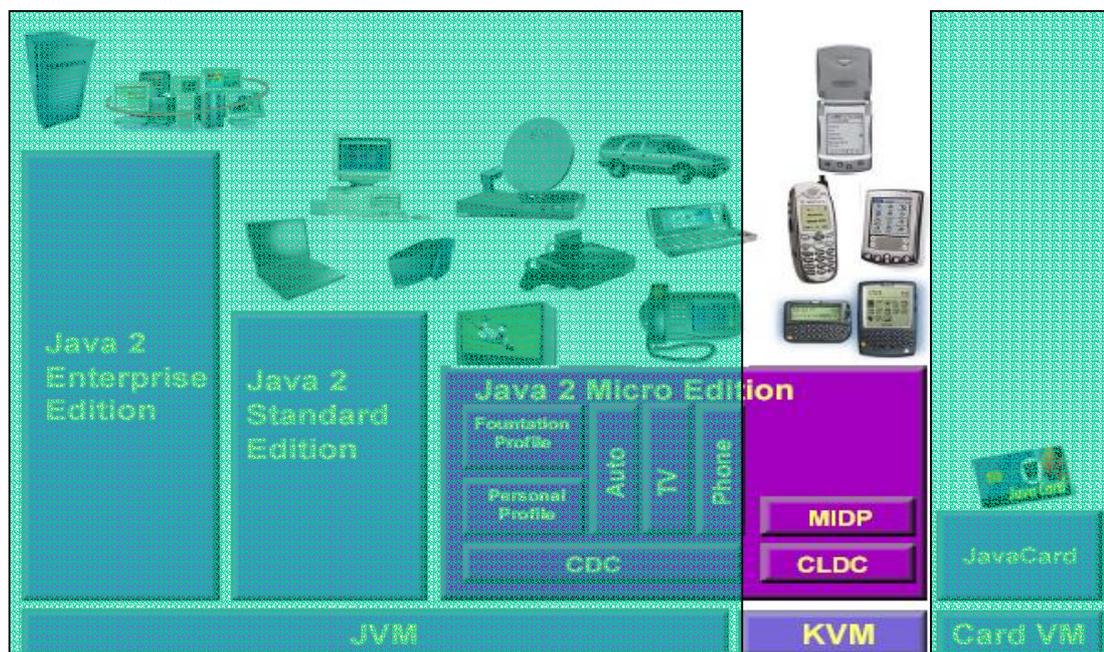


PERFILES Y APIs CLDC

- Otros paquetes opcionales sobre CLDC (en definición):
 - Location API (JSR 179):
 - Acceso a información de localización
 - SIP API (JSR 180):
 - Desarrollo de clientes SIP.
 - Mobile 3D Graphics (JSR 184):
 - Desarrollo de juegos 3D.



MIDP/CLDC/KVM

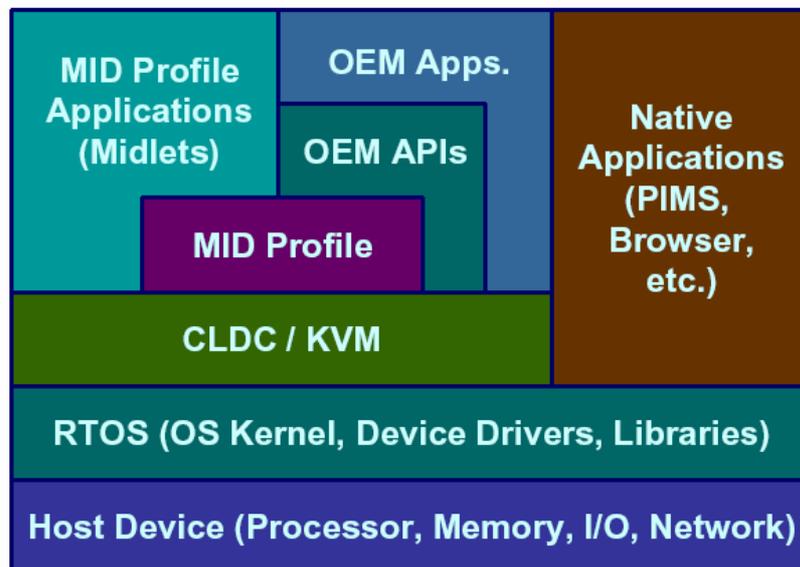




DISPOSITIVOS MIDP/CLDC/KVM



ARQUITECTURA MIDP/CLDC/KVM





CLDC/KVM - ÁMBITO

- Subconjunto de J2SE.
- CLDC/KVM cubre:
 - Máquina virtual y soporte al lenguaje Java.
 - Modelo de seguridad.
 - Entrada/Salida.
 - Soporte a conexiones de red.
 - Internacionalización.
- CLDC/KVM no cubre:
 - Instalación y gestión del ciclo de vida de las aplicaciones Java.
 - Interfaz de usuario.
 - Gestión de eventos.
 - Modelo de aplicación a alto nivel.
 - Soporte a almacenamiento persistente.



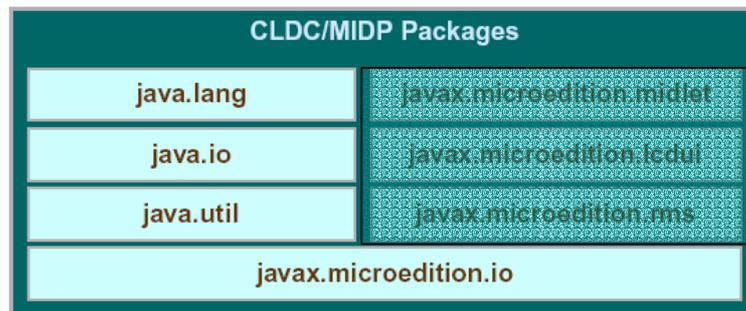
CLDC/KVM – DIFERENCIAS CON J2SE

- No soporta tipos en punto flotante (*float*).
- No soporta finalización.
- Limitaciones en el manejo de errores.
- No soporta Java Native Interface (JNI).
- No soporta reflexión (*reflection*).
- No soporta cargadores de clase definidos por el usuario.
- No soporta grupos de hilos ni demonios (*thread groups*, *daemon groups*).
- Verificación de código en dos fases: preverifier y standard Java bytecode annotations.



CLDC/KVM - LIBRERÍAS

- Clases heredadas de J2SE:
 - `java.lang.*`
 - `java.io.*`
 - `java.util.*`
- Clases específicas introducidas por CLDC:
 - `javax.microedition.io.*`



CLDC/KVM - PROPIEDADES

- CLDC permite obtener propiedades del sistema vía `java.lang.System`
- La llamada a `System.getProperty(String key)` devuelve el valor de la propiedad como un string
- CLDC deben proporcionar al menos las siguientes propiedades:
 - `microedition.platform`
 - `microedition.encoding`
 - `microedition.configuration`
 - `microedition.profiles`
- Un perfil CLDC puede requerir ciertos valores de las propiedades CLDC:
 - MIDP 1.0 requiere `microedition.profiles` contener al menos "MIDP-1.0"



CLDC/KVM - INTERNACIONALIZACIÓN

- Todos los dispositivos CLDC soportan por defecto **ISO-LATIN1** (`microedition.encoding` con valor "ISO8859_1").
- Los fabricantes pueden proporcionar códigos adicionales:
 - Por ejemplo, NTT DoCoMo requiere que los teléfonos i-mode phones soporten la codificación japonesa ShiftJIS.
- No se soportan soluciones relacionadas con el formato de fechas o tiempo.



CLDC/KVM - RED, ALMACENAMIENTO, y ENTRADA/SALIDA

- Nuevo soporte porque el de J2SE presenta los siguientes problemas:
 - Gran tamaño: más 100 clases (200 kB).
 - No estaba pensado para pequeños dispositivos:
 - Se suponía TCP/IP siempre disponible.
 - No es fácil de extender a nuevos protocolos no TCP/IP tipo Bluetooth o IrDA.
- CLDC introduce **Generic Connection Framework**:
 - Soporte a diferentes tipos de protocolos de red.
 - Permite definir y usar nuevos protocolos de forma sencilla.
 - Compatibilidad con Java standard, mediante mapeo.

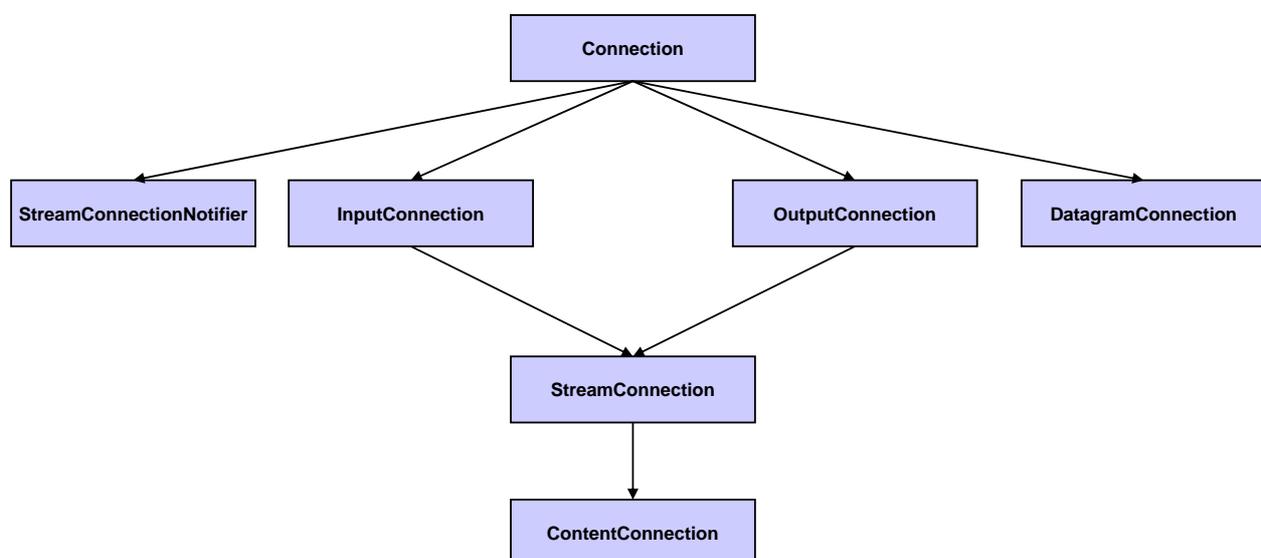


CLDC/KVM - *Generic Connection Framework*

- CLDC especifica un mecanismo general de conexión:
 - `Connector.open("<protocol>://<address>:<parameters>");`
 - Por ejemplo:
 - Archivos:
 - `Connector.open("file://midp.txt");`
 - HTTP:
 - `Connector.open("http://www.sun.com");`
 - Sockets:
 - `Connector.open("socket://129.144.111.222:9000");`
 - Puerto serie:
 - `Connector.open("comm:0;baudrate=9600");`
- CLDC no implementa ningún protocolo, son los perfiles los que deben definir qué conector(es) debe(n) implementarse.



CLDC/KVM - *Generic Connection Framework* Interfaces





CLDC/KVM - SEGURIDAD

- No soporta el modelo completo de J2SE.
- Modelo de seguridad de CLDC:
 - Seguridad a nivel máquina virtual: verificador de clases.
 - Seguridad a nivel de aplicación: modelo “sandbox”.
- Verificador de clases en dos pasos:
 - Preverifier externo.
 - Verifier en el dispositivo.
- Modelo “sandbox”:
 - No se pueden sobrescribir clases del sistema.
 - No se pueden acceder a clases nativas.
 - Restringido al API proporcionada por el CLDC y el perfil sobre el que desarrolla.



MIDP – REQUISITOS HARDWARE

- Pantalla:
 - Tamaño: 96x54.
 - Profundidad: 1 bit.
 - Aspecto pixel 1:1
- Entrada, uno o más de los siguientes mecanismos:
 - Teclado “one-handed”
 - Teclado “two-handed”
 - Pantalla táctil.
- Memoria:
 - 128 KB de memoria no volátil para los componentes de MIDP.
 - 8 KB de memoria no volátil para creación de datos persistentes de las aplicaciones.
 - 32 KB de memoria volátil para la ejecución de Java.
- Conectividad:
 - Limitada, típicamente wireless.



MIDP - ÁMBITO

- Define el conjunto de APIs disponibles para el desarrollo de aplicaciones portables entre dispositivos móviles.
- MIDP no cubre:
 - Descarga y gestión de aplicaciones (MIDlets) en los dispositivos.
 - Seguridad a bajo nivel.
 - Seguridad a nivel de aplicación.
 - Seguridad extremo-extremo.
- Se asume la existencia de Application Management System (AMS):
 - Dependiente del dispositivo.
 - Instala, interacciona con y borra MIDlets.
 - Instalación (ej. accediendo a un servidor web vía red inalámbrica).
 - Actualización de versiones de MIDlets.



MIDP - LIBRERÍAS

- Librerías específicas:
 - Ciclo de vida de la aplicación (MIDlet):
 - `javax.microedition.midlet`
 - Interfaz de usuario:
 - `javax.microedition.lcdui`
 - Memoria persistente:
 - `javax.microedition.rms`
- Complementa las de CLDC:
 - `javax.microedition.io`
 - `java.lang` y `java.util`



MIDP – APLICACIONES *MIDlets*

- Un MIDlet:
 - Tiene un ciclo de vida bien definido.
 - Da información descriptiva sobre sí mismo.
 - Extiende `javax.microedition.midlet.MIDlet`
- Existe el concepto de MIDlet permanente:
 - Reside, al menos en parte, en memoria no volátil (ROM, EEPROM).
 - Puede descargarse de la red y grabarse en memoria persistente.
 - Pueden ser ejecutados repetidas veces por el usuario sin necesidad de volver a descargarlos.
- MIDlet suite: conjunto de aplicaciones (MIDlets) que comparten recursos en el contexto de una única MV.



MIDP – INTERFAZ DE USUARIO

- Definida en el paquete `javax.microedition.lcdui`
- API de alto nivel:
 - Muy portable.
 - Orientada a “screen” y “widget”.
 - Las aplicaciones que usan este API deberían funcionar en todos los dispositivos.
 - No hay acceso a todas las funciones del dispositivo.
 - Más sencillo y menos potente que AWT.
- API de bajo nivel:
 - Primitivas de dibujo.
 - Eventos de teclado.
 - Menos portabilidad, mejor “experiencia del usuario”.



MIDP - ALMACENAMIENTO

- API independiente del dispositivo.
- Base de datos sencilla orientada a registros (RMS).
 - Registro (**record**) son array de bytes
 - Los registros se guardan en almacenes de registro (**record stores**).
 - Los almacenes de registros se comparten entre MIDlets de un mismo MIDlet suite.
- Soporta enumeración, ordenamiento y filtrado.
- Actualización atómica de registros.
- Definido en el paquete **javax.microedition.rms**.



MIDP - RED

- Implementa el *Generic Connection Framework* de CLDC definido en el paquete **javax.microedition.io**:
 - Requiere soporte de conexiones HTTP (RFC 2616) como cliente.
- Añade e implementa el interfaz **HttpConnection**, hereda directamente del interfaz **ContentConnection**.
- La implementación del interfaz **DatagramConnection**, definido en CLDC es opcional, pero recomendable.