

# Programación de Dispositivos Móviles



## Sesión 1: Introducción a los MIDs

# Índice



- **Características de los dispositivos**
- **Arquitectura de J2ME**
- **Aplicaciones MIDP**
- **Construcción de aplicaciones**
- **Desarrollo con Eclipse**

# Introducción a los MIDs



- **Características de los dispositivos**
- **Arquitectura de J2ME**
- **Aplicaciones MIDP**
- **Construcción de aplicaciones**
- **Desarrollo con Eclipse**

# Tipos de dispositivos



- **Dispositivos móviles de información**
    - **MIDs: Mobile Information Devices**
    - **Teléfonos móviles, PDAs, etc**
  - **Descodificadores de TV (*set top boxes*)**
  - **Electrodomésticos**
  - **Impresoras de red**
  - **Routers**
  - **etc**
- } **sin interfaz**

# Características de los MIDs



96x65  
Monocromo  
164kb



101x64  
Monocromo  
150kb



178x201  
4096 colores  
1,4mb



128x128  
4096 colores  
200kb



640x200  
4096 colores  
8mb



240x320  
65536 colores  
64mb

# Redes de telefonía celular



- **1G: Red analógica**
  - **Sólo voz**
  - **Red TACS en España**
  - **Distintos países usan distintas redes**
    - **No permite itinerancia**
- **2G: Red digital**
  - **Voz y datos**
  - **GSM (*Global System for Mobile communications*) en toda Europa**
    - **Permite itinerancia**
  - **Red no IP**
    - **Protocolos WAP (WSP)**
    - **Un gateway conecta la red móvil (WSP) a la red Internet (TCP/IP)**
  - **Commutación de circuitos (*Circuit Switched Data, CSD*)**
    - **9'6kbps**
    - **Se ocupa un canal de comunicación de forma permanente**
    - **Se cobra por tiempo de conexión**

# Redes de telefonía celular (2)



- **2,5G: GPRS (*General Packet Radio Service*)**
  - Transmisión de paquetes
    - No ocupa un canal de forma permanente
    - Hasta 144kbps teóricamente (40kbps en la práctica)
    - Cobra por volumen de información transmitida
  - Se implementa sobre la misma red GSM
- **3G: Banda ancha**
  - Red UMTS (*Universal Mobile Telephony System*)
    - Itinerancia global
  - Entre 384kbps y 2Mbps
  - Servicios multimedia
    - Videoconferencia, TV, música, etc
  - Transmisión de paquetes
  - Requiere nueva infraestructura

# Paradigmas de programación en móviles



## ■ Documentos Web

- Descarga documentos y los muestra en un navegador
- Formato adecuado para móviles (WML, XHTML, ...)
- Requiere conectar a red para descargar cada documento
- Velocidad de descarga lenta
- Documentos pobres (deben servir para todos los móviles)

## ■ Aplicaciones locales

- La aplicación se descarga en el móvil
- Se ejecuta de forma local
- Interfaz de usuario más flexible
- Puede funcionar sin conexión (minimiza el tráfico)

# Documentos Web



- **WML (Wireless Markup Language)**
  - Forma parte de los protocolos WAP (Capa de aplicación, WAE)
  - Lenguaje de marcado dirigido a móviles
  - Requiere aprender un nuevo lenguaje diferente a HTML
  - Documentos muy pobres
- **iMode**
  - Documentos escritos en cHTML (HTML compacto)
    - Subconjunto de HTML
    - Propietario de NTT DoCoMo
  - Sobre la red japonesa PDC-P (extensión de la red japonesa PDC, similar a GSM, para transmisión de paquetes)
    - En Europa se lanza sobre GPRS
- **XHTML MP**
  - Versión reducida de XHTML dirigido a móviles
  - A diferencia de cHTML, se desarrolla como estándar

# Aplicaciones locales



- **Sistema operativo**
  - **Symbian OS, Palm OS, Windows Pocket PC, etc**
  - **Poco portable**
  - **Requiere aprender nuevas APIs**
- **Runtime Environments**
  - **BREW**
    - **Soportado por pocos dispositivos**
    - **Requiere aprender una nueva API**
  - **J2ME**
    - **Soportado por gran cantidad de dispositivos**
    - **Existe una gran comunidad de desarrolladores Java**

# Conectividad de los MIDs



- **Los dispositivos deben conectarse para descargar las aplicaciones**

- **Over The Air (OTA)**

- **Conexión a Internet usando la red móvil (GSM, GPRS, UMTS)**

- **Cable serie o USB**

- **Conexión física**

- **Infrarrojos**

- **Los dispositivos deben verse entre si**

- **Bluetooth**

- **Ondas de radio (10 metros de alcance)**
- **Alta velocidad (723kbit/s)**

# Introducción a los MIDs



- **Características de los dispositivos**
- **Arquitectura de J2ME**
- **Aplicaciones MIDP**
- **Construcción de aplicaciones**
- **Desarrollo con Eclipse**

# Java 2 Micro Edition



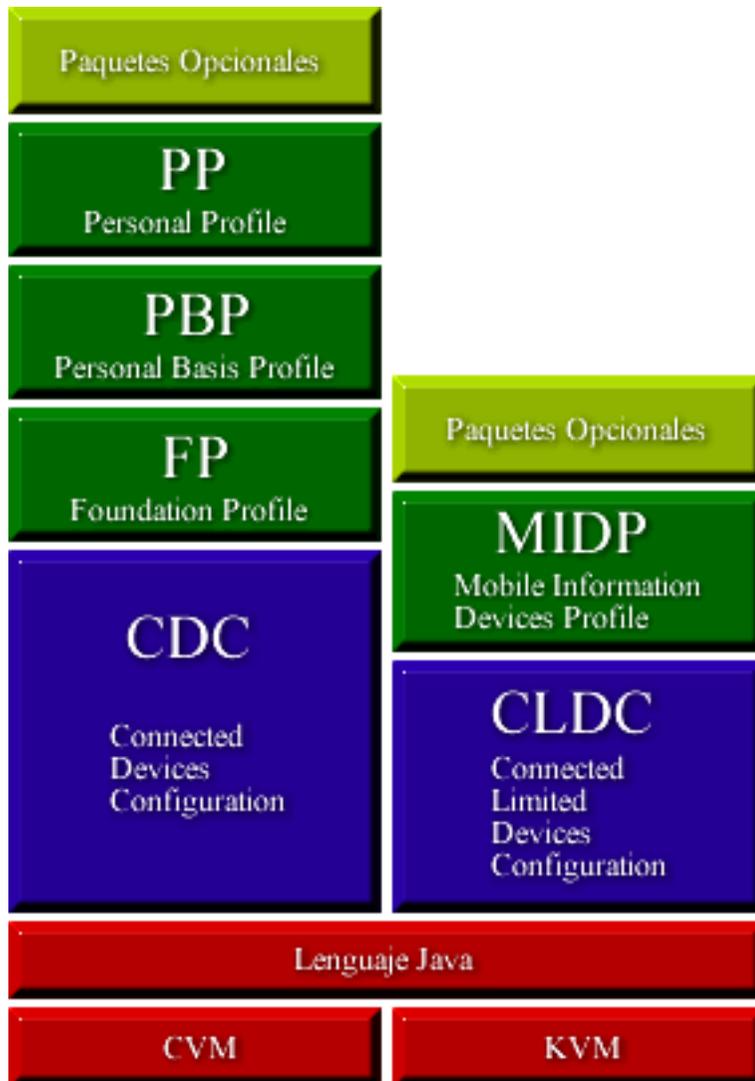
- **Edición de la plataforma Java 2 para dispositivos móviles**
- **Independiente de la plataforma**
  - **Adecuado para programar dispositivos heterogéneos**
- **Gran comunidad de desarrolladores Java**
  - **Los programadores Java podrán desarrollar aplicaciones para móviles de forma sencilla**
  - **No hace falta que aprendan un nuevo lenguaje**
- **Consiste en un conjunto de APIs**
  - **Una sola API es insuficiente para la variedad de tipos de dispositivos existente**
  - **Cada API se dedica a una distinta familia de dispositivos**

# Capas de J2ME



- **Configuraciones**
  - **API común para todo un gran conjunto de dispositivos**
  - **Elementos básicos del lenguaje**
- **Perfiles**
  - **API que cubre las características propias de una familia de dispositivos concreta**
    - **P.ej, para acceder a la pantalla de los teléfonos móviles**
- **Paquetes opcionales**
  - **APIs para características especiales de ciertos dispositivos**
    - **P.ej, para acceder a la cámara de algunos teléfonos móviles**

# APIs de J2ME



## Configuraciones

- **CDC: Dispositivos conectados**
  - Sobre JVM
- **CLDC: Dispositivos conectados limitados**
  - Sobre KVM (limitada)
  - Paquetes:
    - **java.lang**
    - **java.io**
    - **java.util**
    - **javax.microedition.io**

# Perfiles CDC



- **Dispositivos con memoria del orden de los MB**
  - Se recomienda por lo menos 2MB
  - PDAs de gama alta
- **Se ejecuta sobre CVM (equivalente a JVM)**
- **FP (Foundation Profile)**
  - Dispositivos sin interfaz: impresoras de red, routers
- **FBP (Foundation Basis Profile)**
  - Dispositivos con interfaz: descodificadores de TV
  - Sólo componentes ligeros de AWT
- **PP (Personal Profile)**
  - Incluye la especificación completa de AWT
  - Dispositivos con interfaz gráfica nativa
  - Adecuado para migrar antiguos sistemas PersonalJava



- **Dispositivos con memoria del orden de los KB**
  - Puede funcionar con sólo 128KB
  - Teléfonos móviles y PDAs de gama baja
- **Se ejecuta sobre KVM (*Kilobyte Virtual Machine*)**
- **Muy limitada, para poder funcionar con escasos recursos**
  - P.ej, no soporta reales (tipos `float` y `double`)
- **Perfil MIDP**
  - **Dispositivos móviles de información (MIDs)**
  - Paquetes:
    - `javax.microedition.lcdui`
    - `javax.microedition.midlet`
    - `javax.microedition.rms`

# Paquetes opcionales



- **Wireless Messaging API (WMA)**
  - Envío y recepción de mensajes cortos (SMS)
- **Mobile Media API (MMAPI)**
  - Multimedia, reproducción y captura de video y audio
- **Bluetooth API**
  - Permite establecer conexiones vía Bluetooth
- **J2ME Web Services**
  - Invocación de servicios web desde dispositivos móviles
- **Mobile 3D Graphics**
  - Permite incorporar gráficos 3D a las aplicaciones y juegos

# Más paquetes opcionales



- **Location API**
  - Localización física del dispositivo (GPS)
- **Security and Trust Services API**
  - Servicios de seguridad: encriptación, identificación, autenticación
- **PDA Optional Packages**
  - Consta de dos librerías:
    - *FileConnection* (FC): librería para acceso al sistema de ficheros (FC)
    - *Personal Information Management* (PIM): librería para el acceso a la información personal almacenada (agenda, contactos, etc)
- **Content Handler API**
  - Integración con el entorno de aplicaciones del dispositivo. Permite utilizar otras aplicaciones para abrir diferentes tipos de contenidos
- **SIP API**
  - Permite utilizar *Session Initiation Protocol*. Este protocolo se usa para conexiones IP multimedia (juegos, videoconferencia, etc)



- **JTWI (*Java Technologies for Wireless Industry*)**
- **Especificación que trata de definir una plataforma estándar para el desarrollo para móviles**
  - **Aumentar la compatibilidad entre los dispositivos**
- **Las tareas de esta especificación son:**
  - **Definir las APIs que deben estar presentes en los dispositivos.**
    - **CLDC 1.0, MIDP 2.0, WMA 1.1**
    - **Opcionalmente: CLDC 1.1, MMAPI**
  - **Evitar que se utilicen APIs adicionales que reducen la compatibilidad.**
  - **Aclarar aspectos confusos en las especificaciones de estas APIs.**

# Introducción a los MIDs



- **Características de los dispositivos**
- **Arquitectura de J2ME**
- **Aplicaciones MIDP**
- **Construcción de aplicaciones**
- **Desarrollo con Eclipse**



- **Las aplicaciones para dispositivos MIDP se denominan *MIDlets***
- **Estas aplicaciones se distribuyen como una *suite de MIDlets*, que se compone de:**
  - **Fichero JAD**
    - Fichero ASCII
    - Descripción de la aplicación
  - **Fichero JAR**
    - Aplicación empaquetada (clases y recursos)
    - Contiene uno o más MIDlets
    - Contiene un fichero `MANIFEST.MF` con información sobre la aplicación (algunos datos son replicados del fichero JAD).

# Fichero JAD



- **Ejemplo de fichero JAD:**

```
MIDlet-Name: SuiteEjemplos
MIDlet-Version: 1.0.0
MIDlet-Vendor: Universidad de Alicante
MIDlet-Description: Aplicaciones de ejemplo para moviles.
MIDlet-Jar-Size: 16342
MIDlet-Jar-URL: ejemplos.jar
```

- **En un dispositivo real es importante que `MIDlet-Jar-Size` contenga el tamaño real del fichero JAR**
- **Si publicamos la aplicación en Internet, `MIDlet-Jar-URL` deberá apuntar a la URL de Internet donde se encuentra publicado el fichero JAR.**

# Fichero MANIFEST.MF



- **Ejemplo de fichero MANIFEST.MF:**

```
MIDlet-Name: SuiteEjemplos
MIDlet-Version: 1.0.0
MIDlet-Vendor: Universidad de Alicante
MIDlet-Description: Aplicaciones de ejemplo para moviles.
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-1.0
MIDlet-1: Snake, /icons/snake.png, es.ua.j2ee.serpiente.SerpMIDlet
MIDlet-2: TeleSketch, /icons/ts.png, es.ua.j2ee.ts.TeleSketchMIDlet
MIDlet-3: Panj, /icons/panj.png, es.ua.j2ee.panj.PanjMIDlet
```

- **Si el dispositivo real no soporta la configuración o el perfil indicados, se producirá un error en la instalación.**

# Software gestor de aplicaciones



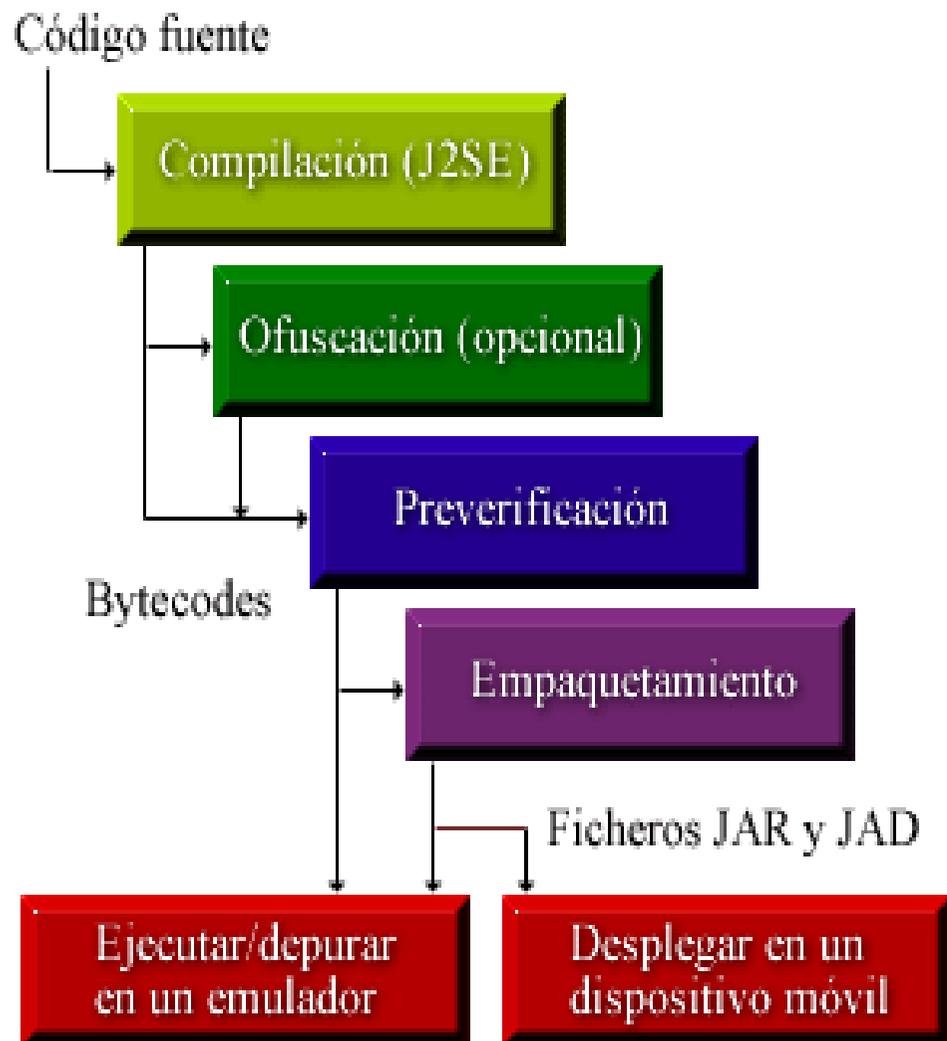
- **Los dispositivos móviles con soporte para Java tienen instalado un software gestor de aplicaciones**
  - *AMS: Application Management Software*
  
- **Gestiona las aplicaciones Java:**
  - **Descarga**
    - **Descarga primero el fichero JAD y muestra los datos de la aplicación**
    - **Si la aplicación es compatible y el usuario acepta, descarga el JAR**
  - **Instalación**
  - **Actualización**
  - **Desinstalación**
  - **Ejecución**
    - **Es el contenedor que da soporte a los MIDlets**
      - **Contiene la KVM sobre la que se ejecutarán las aplicaciones**
      - **Soporta la API de MIDP**
    - **Controla el ciclo de vida de los MIDlets que ejecuta**

# Introducción a los MIDs



- **Características de los dispositivos**
- **Arquitectura de J2ME**
- **Aplicaciones MIDP**
- **Construcción de aplicaciones**
- **Desarrollo con Eclipse**

# Pasos del proceso



- **Compilar**
  - Utilizar como clases del núcleo la API de MIDP
- **Ofuscar (optativo)**
  - Reducir tamaño de los ficheros
  - Evitar descompilación
- **Preverificar**
  - Reorganizar el código para facilitar la verificación a la KVM
  - Comprobar que no se usan características no soportadas por KVM
- **Empaquetar**
  - Crear ficheros JAR y JAD
- **Probar**
  - En emuladores o dispositivos reales

# Kits de desarrollo



- **Incluyen las APIs necesarias**
  - MIDP y APIs adicionales
- **Incluyen herramientas que no están en Java 2 SDK**
  - Preverificador
- **Incluye emuladores para probar las aplicaciones**
  - Imitan teléfonos genéricos o modelos reales
- **Facilitan el proceso de construcción de aplicaciones**
  - Entorno de creación de aplicaciones
- **Es necesario contar con Java 2 SDK para compilar y empaquetar**

# Sun Wireless Toolkit (WTK)

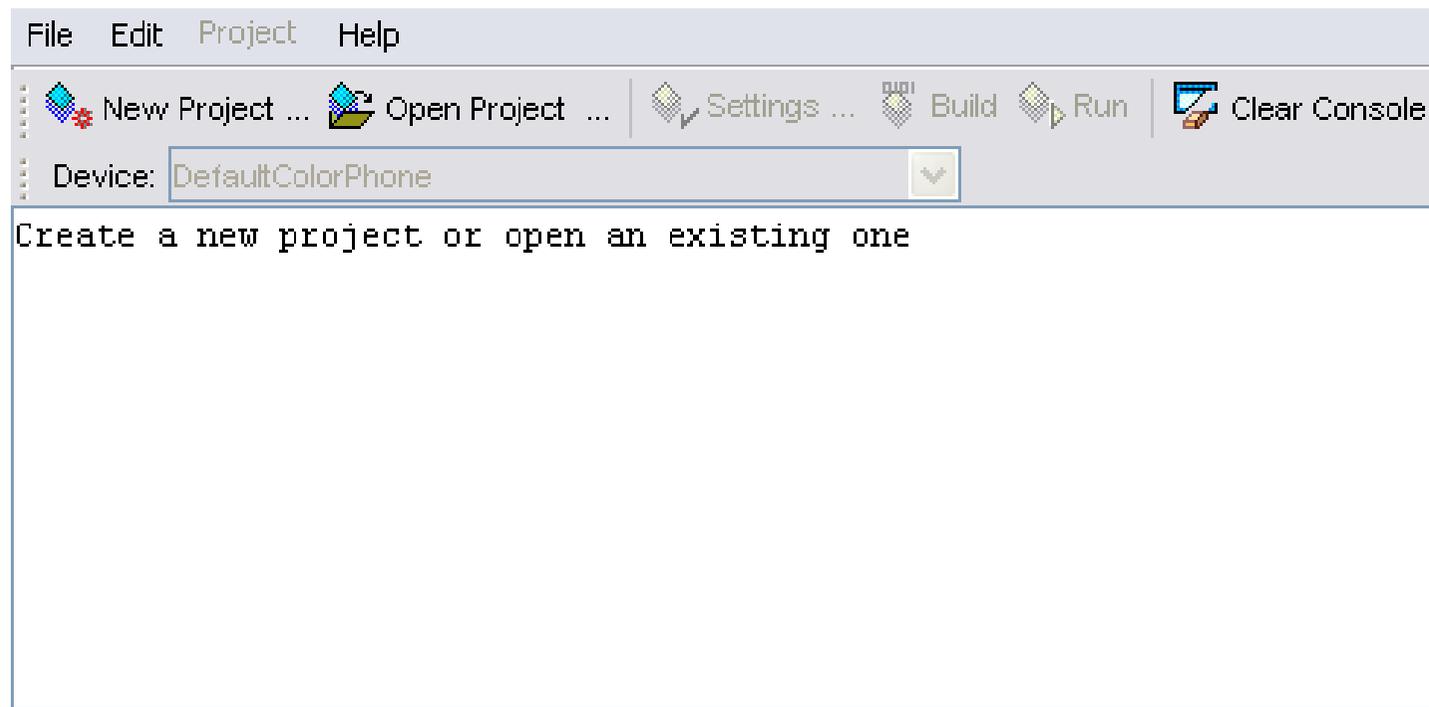


- **Kit de desarrollo genérico.**
  - **Se puede integrar con emuladores proporcionados por terceros (Nokia, Ericsson, etc).**
- **Versiones:**
  - **WTK 1.0.4: Sólo soporta MIDP 1.0**
  - **WTK 2.0: Sólo soporta MIDP 2.0**
    - **APIs opcionales: WMA, MMAPI**
  - **WTK 2.1: Soporta MIDP 1.0 y MIDP 2.0**
    - **Puede generar aplicaciones JTWI**
    - **APIs opcionales: WMA, MMAPI, WSA**
  - **WTK 2.2: Igual que WTK 2.1, añadiendo:**
    - **APIs opcionales: M3G, Bluetooth**
  - **WTK 2.3: Igual que WTK 2.2, añadiendo:**
    - **APIs opcionales: SIP, CHAPI, PDA, SATSA y Location API**

# Ktoolbar



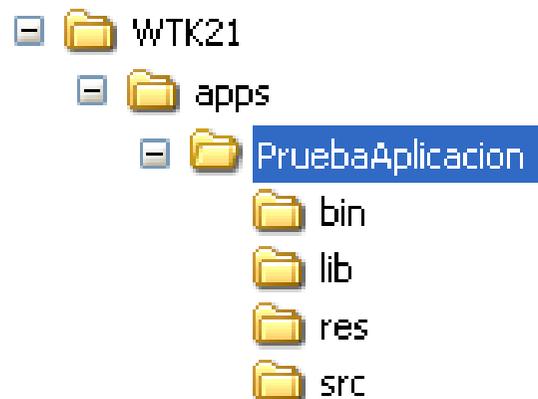
- **WTK contiene la herramienta ktoolbar para automatizar la creación de aplicaciones**



# Aplicaciones de WTK



- Se almacenan en el directorio  $\${WTK\_HOME}/apps$
- Existe un subdirectorio por aplicación
- Cada aplicación se organiza en los siguientes subdirectorios:



**src:** Código fuente

**res:** Recursos (ficheros de datos, imágenes, ...)

**lib:** Librerías (jar)

**bin:** Aquí se generan los ficheros JAD y JAR

**classes:** Clases intermedias generadas (temporal)

# Crear una aplicación



- Pulsar *New Project ...*

|                   |                                |
|-------------------|--------------------------------|
| Project Name      | PruebaAplicacion               |
| MIDlet Class Name | es.ua.j2ee.prueba.MIDletPrueba |

- Editar los datos para los ficheros JAD y JAR (MANIFEST.MF)

| API Selection              | Required             | Optional | User Defined | MIDlets | Push Registry | Permissions |
|----------------------------|----------------------|----------|--------------|---------|---------------|-------------|
| Key                        | Value                |          |              |         |               |             |
| MIDlet-Jar-Size            | 100                  |          |              |         |               |             |
| MIDlet-Jar-URL             | PruebaAplicacion.jar |          |              |         |               |             |
| MIDlet-Name                | PruebaAplicacion     |          |              |         |               |             |
| MIDlet-Vendor              | Unknown              |          |              |         |               |             |
| MIDlet-Version             | 1.0                  |          |              |         |               |             |
| MicroEdition-Configuration | CLDC-1.0             |          |              |         |               |             |
| MicroEdition-Profile       | MIDP-1.0             |          |              |         |               |             |

| Class                             |
|-----------------------------------|
| acion.png es.ua.j2ee.prueba.Ml... |

# Prueba de la aplicación



- **Construir la aplicación**
  - Pulsar sobre *Project* → *Build*
- **Ejecutar en un emulador**
  - Seleccionar un emulador del cuadro desplegable
  - Pulsar sobre *Project* → *Run*



- **Distribuir la aplicación**
  - Pulsar sobre *Project* → *Package* → *Create package*

# Provisionamiento OTA



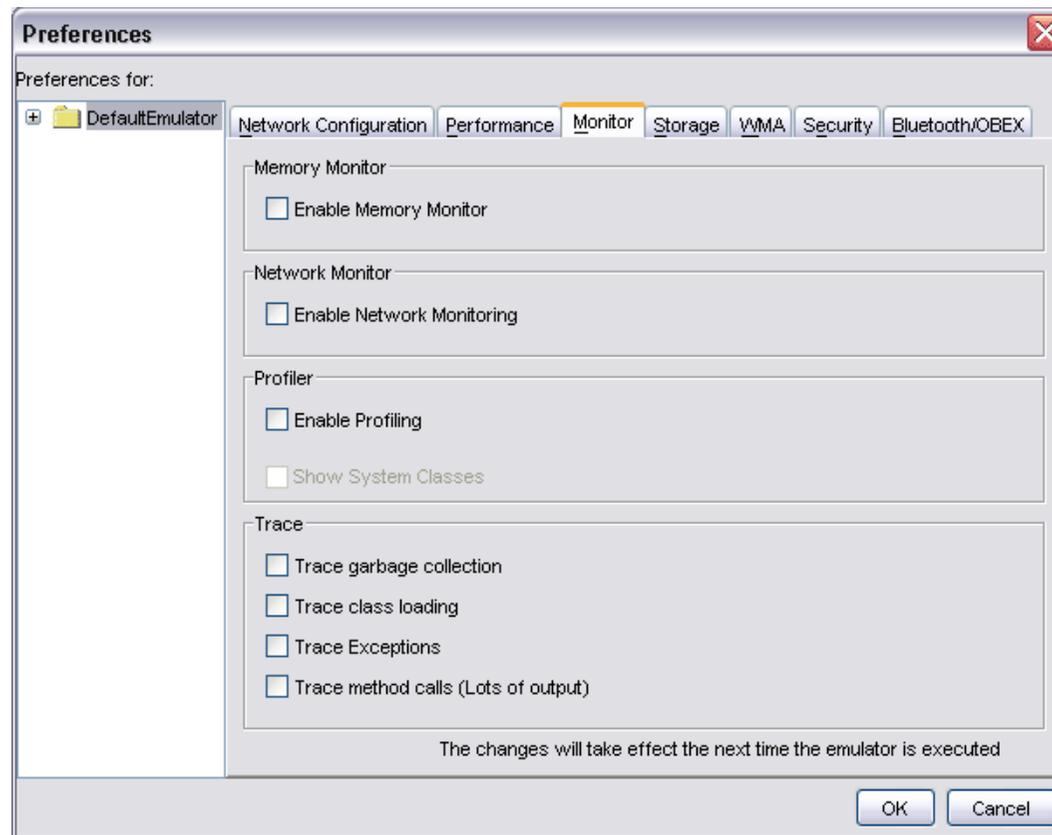
- Podemos simular la descarga real de la aplicación
- Provisionamiento OTA: *Project > Run via OTA*



# Optimización



- **Podemos activar monitores para controlar:**
  - **Trafico en la red**
  - **Ocupación de memoria**



# Introducción a los MIDs



- **Características de los dispositivos**
- **Arquitectura de J2ME**
- **Aplicaciones MIDP**
- **Construcción de aplicaciones**
- **Desarrollo con Eclipse**

# Integración de J2ME y Eclipse

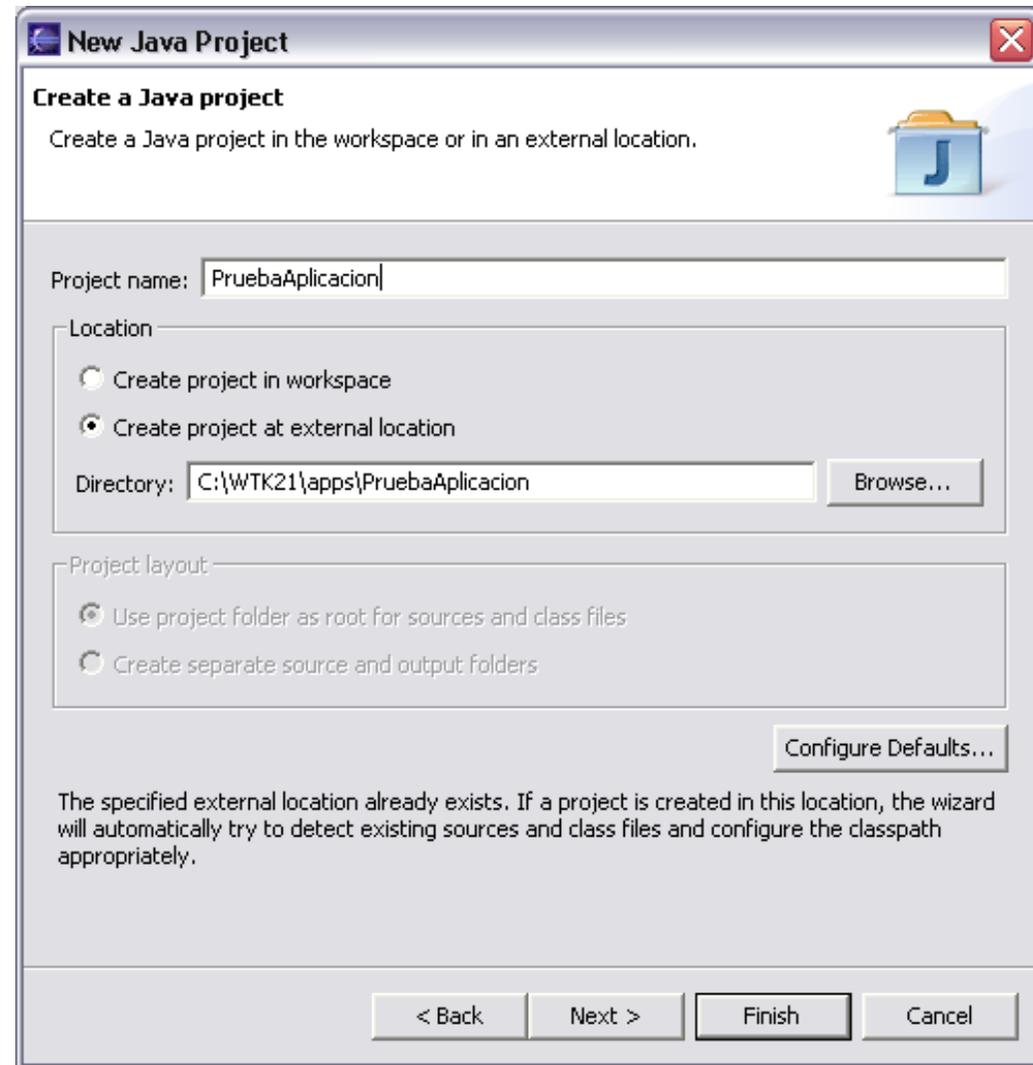


- **Eclipse no incluye soporte “de serie” para J2ME**
  
- **Tenemos varias opciones**
  - **Utilizarlo sólo como editor de código**
    - Construir las aplicaciones con WTK
  
  - **Utilizar tareas de *Ant* para el desarrollo con J2ME**
    - Utilizar librería de tareas Antenna
  
  - **Añadir *plugins* para trabajar con aplicaciones J2ME**
    - Como por ejemplo EclipseME

# Creación de un proyecto



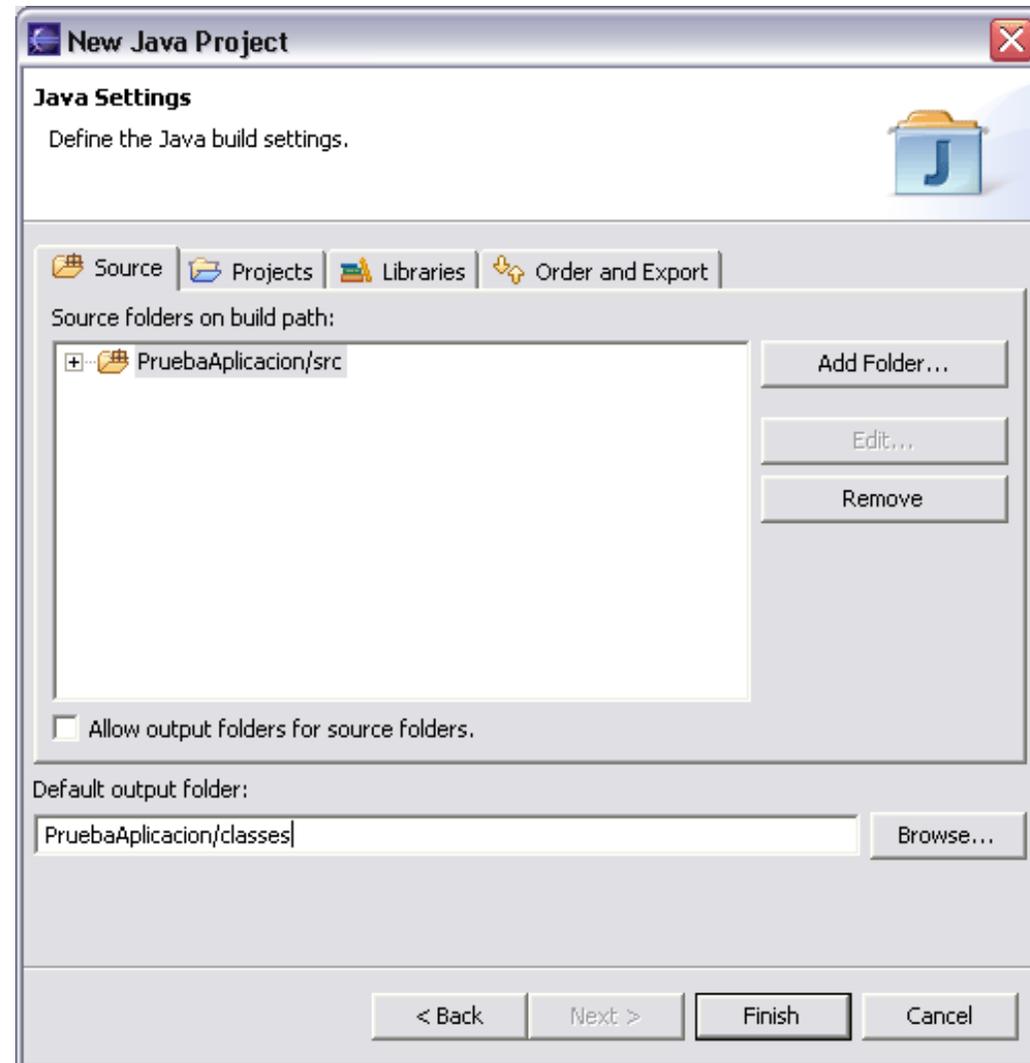
- **Asignar un nombre al proyecto**
- **Utilizar como directorio del proyecto el directorio de la aplicación creada con WTK**
- **Pulsar sobre *Next* >**



# Establecer directorios



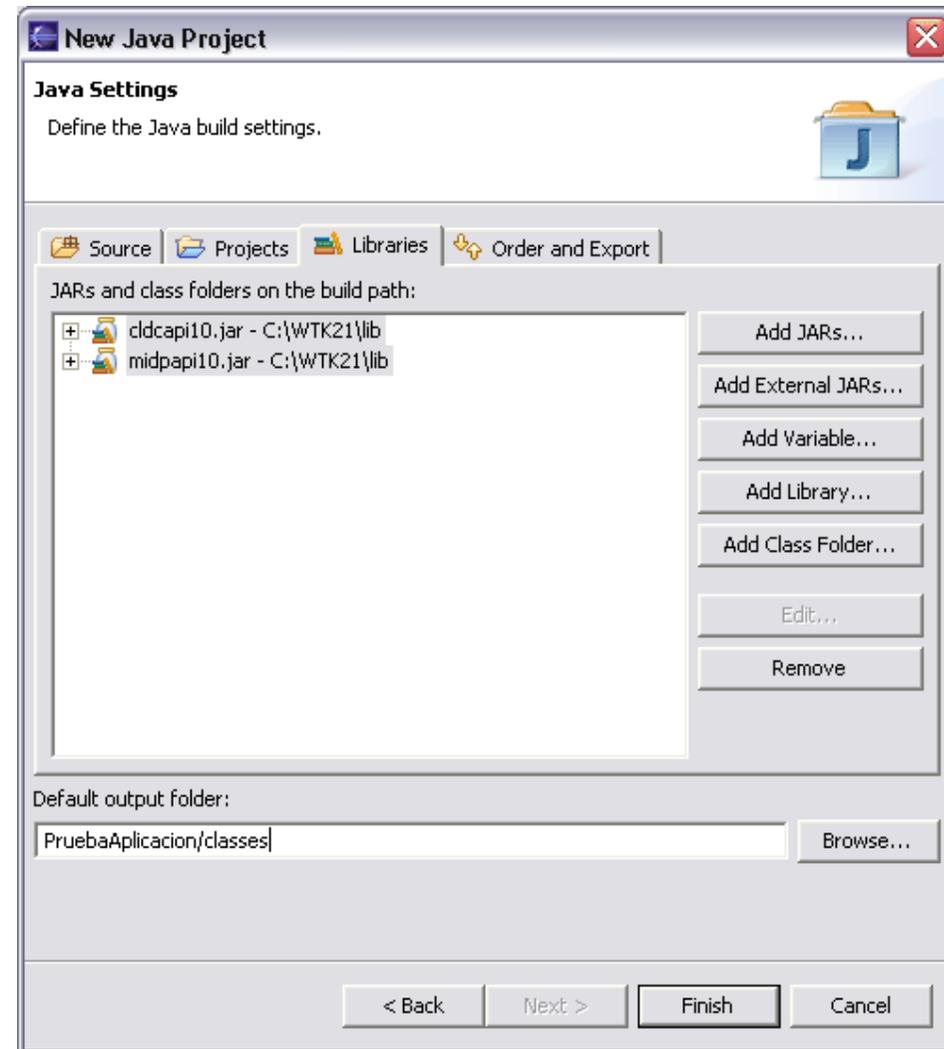
- Establecer como directorio de fuentes el directorio `src` de la aplicación
- Establecer como directorio de salida el directorio `classes` de la aplicación



# Establecer librerías



- **Eliminar la librería de clases de J2SE**
- **Añadir la librería de CLDC (cldcapi10.jar)**
- **Añadir la librería de MIDP (midpapi10.zip)**



# Crear un MIDlet



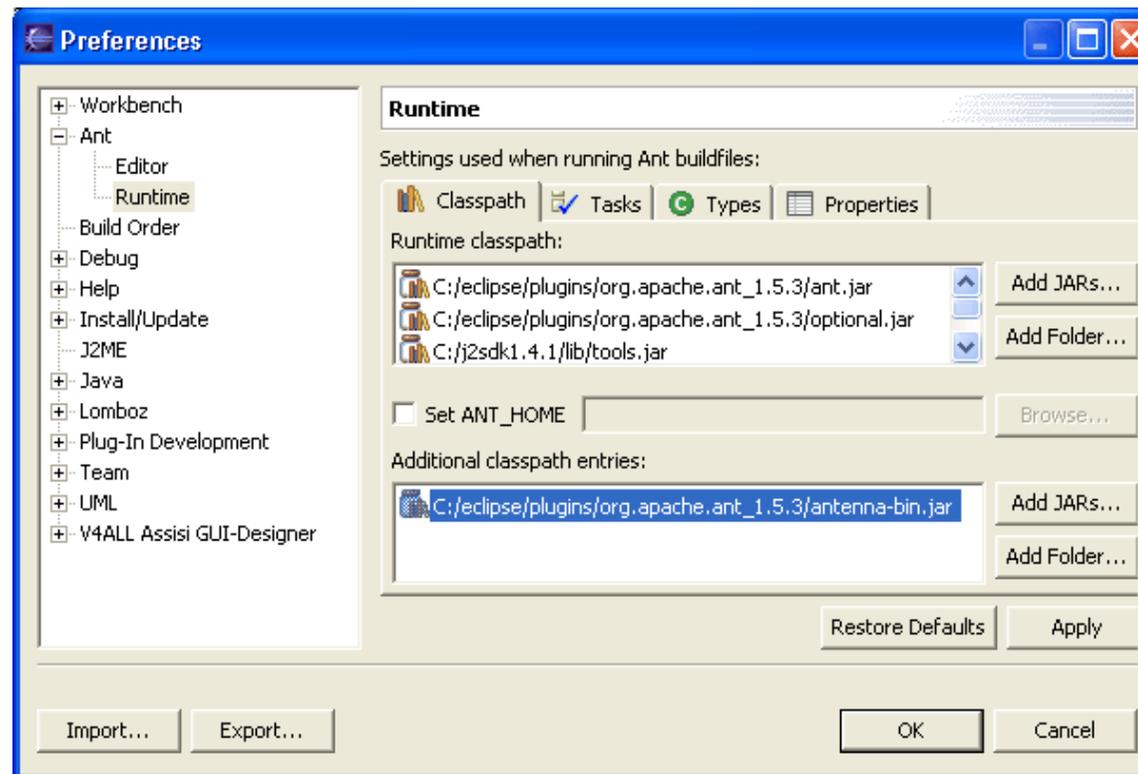
- **Crear una clase que herede de MIDlet**
- **Introducir el código necesario en la clase creada**
- **Crear todas las clases adicionales que sean necesarias para la aplicación**
- **Grabar el código editado**
- **Construir la aplicación desde WTK**

A screenshot of the 'New Java Class' dialog box in an IDE. The dialog has a title bar with a close button. The main area is titled 'Java Class' and contains the text 'Create a new Java class.' with a green circular icon. Below this are several input fields and buttons: 'Source Folder:' with 'PruebaAplicacion/src' and a 'Browse...' button; 'Package:' with 'es.ua.j2ee.prueba' and a 'Browse...' button; 'Enclosing type:' with an unchecked checkbox and a 'Browse...' button; 'Name:' with 'MIDletPrueba'; 'Modifiers:' with radio buttons for 'public' (selected), 'default', 'private', and 'protected', and checkboxes for 'abstract', 'final', and 'static'; 'Superclass:' with 'javax.microedition.midlet.MIDlet' and a 'Browse...' button; 'Interfaces:' with an empty list and 'Add...' and 'Remove' buttons; and a section 'Which method stubs would you like to create?' with checkboxes for 'public static void main(String[] args)', 'Constructors from superclass', and 'Inherited abstract methods' (checked). At the bottom are 'Finish' and 'Cancel' buttons.

# Tareas de Antenna



- *Antenna* es una librería de tareas de *Ant* para construir aplicaciones J2ME
- Podemos utilizar esta librería desde Eclipse



# EclipseME



- *Plug-in* de Eclipse
- Nos permite crear aplicaciones J2ME con este entorno de forma integrada
  - No es necesario utilizar ninguna herramienta externa
- Podemos:
  - Crear una suite de MIDlets
  - Añadir MIDlets a la suite
  - Editar el fichero JAD mediante un editor de JAD incorporado
  - Ejecutar la aplicación directamente en un emulador

