

Tema 5: Programación de dispositivos móviles con Java (Java ME)



Tecnologías  Web

Bibliografía y enlaces

- Programación de Dispositivos Móviles con J2ME
 - Lozano Ortega, Miguel Ángel
 - Publicaciones de la Universidad de Alicante, 2004
- Web oficial de Sun
 - www.java.sun.com
- Web para desarrolladores de Nokia
 - www.forum.nokia.com



Tecnologías  Web

1. Dispositivos móviles

Características de los dispositivos



Tecnologías  Web

Dispositivos conectados

Dispositivos con pequeños ordenadores embebidos

Tienen la capacidad de conectarse a la red

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

sin interfaz



Tecnologías  Web

Limitaciones de los dispositivos

- **Escasa memoria**
 - Normalmente 128-512Kb de RAM
- **CPU lenta**
 - 1-10 MIPS (Pentium 4 3.0GHz, ~10000 MIPS)
- **Pequeña pantalla**
 - 96x65 – 178x201 píxeles, monocromo – 65536 colores
- **Dispositivos de entrada restringidos**
 - Teclado 0-9, #, *
- **Fuentes de texto limitadas**
 - Normalmente sólo una fuente



Tecnologías Web

Hay mucha variedad



96x65
Monocromo
164kb



101x64
Monocromo
150kb



178x201
4096 colores
1,4mb



128x128
4096 colores
200kb



640x200
4096 colores
8mb



240x320
65536 colores
64mb



Tecnologías Web

2. Clientes para móviles

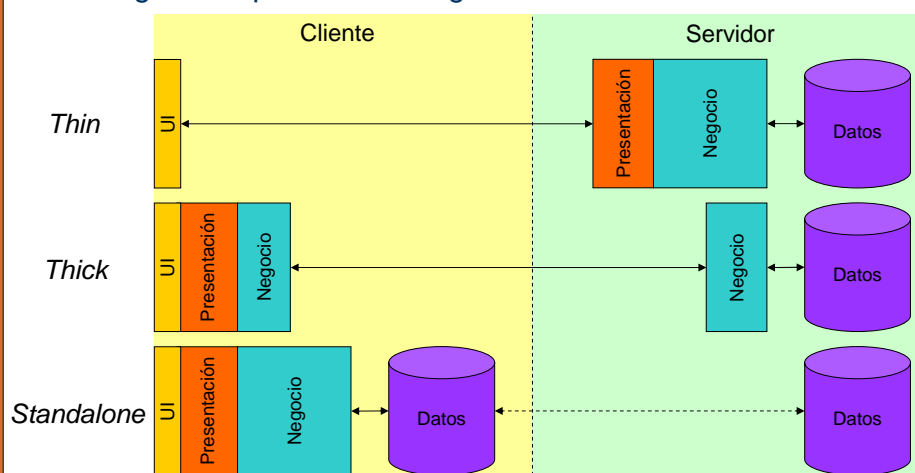
Tipos de clientes para móviles



Tecnologías Web

Tipos de cliente

Según el reparto de la carga entre cliente/servidor



Tecnologías Web

Cientes *thin*

- Todo el procesamiento se realiza en el servidor
 - El cliente sólo se ocupa de la interfaz de usuario (UI)
- El cliente normalmente se compone de:
 - Navegador
 - Documento web (p.ej. HTML)
- Descarga documentos y los muestra en un navegador
- Los documentos HTML no son adecuados para móviles
 - Se definen nuevos tipos de documentos
 - WML, XHTML-MP, cHTML, ...
 - Se muestran en un navegador que los soporte
 - Microbrowser



Tecnologías  Web

Cientes *thick*

- La aplicación se descarga e instala en el cliente
 - Se ejecuta de forma local
 - Trabaja de forma coordinada con el servidor
 - Realiza en el cliente todo el procesamiento posible
- Aplicaciones dedicadas
 - Para una tarea concreta
- Sólo necesita intercambiar información, no presentación
 - Los documentos web (p.e. HTML) no sirven
- Se puede comunicar mediante diferentes protocolos
 - Servicios Web
 - RPC
 - Otros protocolos



Tecnologías  Web

¿Thick o Thin?

■ Thin

- ☞ Requiere conectar a la red para descargar cada documento
 - Velocidad de descarga lenta en móviles
- ☞ Limitado a las posibilidades del navegador
- ☞ Fácil de mantener

■ Thick

- ☞ Mantenimiento costoso
- ☞ Interfaz de usuario (UI) más flexible
- ☞ Minimiza el tráfico en la red
 - Intercambia con el servidor sólo la información necesaria
 - Puede funcionar sin conexión
- ☞ Mayor rendimiento multimedia



Tecnológicas  Web

■ 3. Clientes thin

■ Tecnologías para clientes thin



Tecnológicas  Web

Lenguajes de marcado para clientes *thin*

- No se suele usar todo el HTML o XHTML debido a las restricciones gráficas de los dispositivos
- Algunos lenguajes de marcado para clientes *thin*
 - WML (*Wireless Markup Language*)
 - Basado en XML, totalmente distinto al HTML
 - Imágenes en formato propio: WBMP (monocromas)
 - WMLScript: lenguaje tipo Javascript
 - cHTML (*Compact HTML*)
 - Excluye tablas, hojas de estilo, fuentes,...
 - Añade enlaces a teléfonos, asociación teclas-enlaces, emojis,...
 - Imágenes GIF (no JPG)
 - XHTML Basic y XHTML-MP (*Mobile Profile*)
 - No soporta tablas complejas
 - Javascript en 1.2



Tecnológicas Web

Ejemplo de WML

```
<?xml version='1.0'?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1/EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="Carrera" title="Carrera">
    <do type="accept" label="Nombre"><go href="#Nombre"/></do>
    <p>Carrera: <select name="carrera" title="Carrera:">
      <option value="II">II</option>
      <option value="ITIS">ITIS</option>
      <option value="ITIG">ITIG</option>
    </select></p>
  </card>
  <card id="Nombre" title="Nombre:">
    <do type="accept" label="Confirma"><go href="#Confirma"/></do>
    <p>Nombre: <input type="text" name="nombre"/></p>
  </card>
  <card id="Confirma" title="Confirma:">
    <p>Tus datos son:<br/>Nombre: ${nombre}<br/>Carrera: ${carrera}<br/></p>
  </card>
</wml>
```



Tecnológicas Web

Ejemplo de cHTML

```
<!DOCTYPE "-//W3C//DTD Compact HTML 1.0 Draft//EN">
<html>
  <head>
    <META http-equiv="Content-Type" content="text/html; charset=utf-8">
    <META name="CHTML" content="yes">
    <META name="description" content="Documento cHTML">
    <title>Menu cHTML</title>
  </head>
  <body bgcolor="#ffffff" text="#000000">
    <center>Subatas</center>
    <hr>
    Menu<br>
    &#59106;<A HREF="lista.htm" accesskey="1">
      Ver subastas </A><BR>
    &#59107;<A HREF="personal.htm" accesskey="2">
      Datos personales </A><BR>
    <hr>
    &#63858;<A HREF="tel://555123456"> Incidencias </A><BR>
  </body>
</html>
```

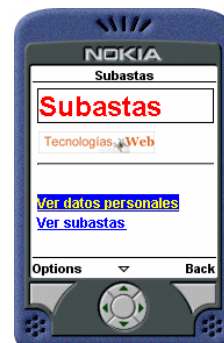


Tecnológicas Web

Ejemplo de XHTML-MP

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
  "http://www.wapforum.org/DTD/xhtml-mobile10.dtd" >

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Subastas</title>
    <link rel="stylesheet"
      href="estilo.css" type="text/css"/>
  </head>
  <body>
    <h1> Subastas </h1>
    
    <hr/>
    <p>
      <a href="personal.xhtml">Ver datos personales</a><br/>
      <a href="lista.xhtml">Ver subastas</a>
    </p>
  </body>
</html>
```



Tecnológicas Web

■ 4. Introducción a Java ME

■ Arquitectura de la plataforma



Tecnologías  Web

Java ME no es la única tecnología *thick*

■ Programar directamente sobre el sistema operativo

- Symbian OS, Palm OS, Windows Pocket PC, etc
- Poco portable
- Requiere aprender nuevas APIs
- Problemas de seguridad

■ Runtime Environments

- BREW
 - Soportado por pocos dispositivos
 - Requiere aprender una nueva API
- Java ME
 - Soportado por gran cantidad de dispositivos
 - Existe una gran comunidad de desarrolladores Java
- .NET Compact Framework
 - Soportado en Pocket PC, Windows CE y Windows Mobile



Tecnologías  Web

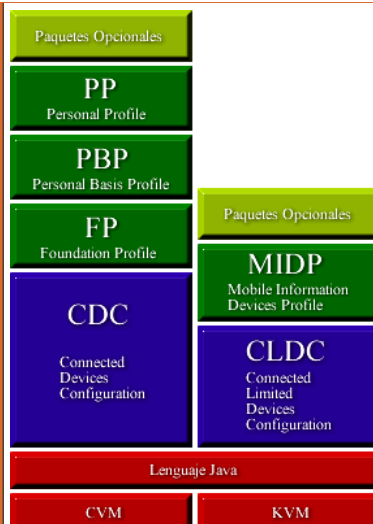
Java Micro Edition

- Edición de la plataforma Java para dispositivos móviles
- Independiente de la plataforma física
 - 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



Tecnológicas Web

Capas de Java ME



- **Paquetes opcionales:** APIs para características especiales de algunos modelos
 - Acceso a la cámara
 - 3D
- **Perfiles:** APIs para una familia de dispositivos
- **Configuración:** API para un gran conjunto de dispositivos
 - CLDC: teléfonos móviles
 - Máquina virtual limitada (KVM)
 - CDC: PDAs



Tecnológicas Web

CLDC: teléfonos y similares

- 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 (el más común, aunque hay otros)
 - Dispositivos móviles de información (MIDs)
 - Paquetes:
 - `javax.microedition.lcdui`
 - `javax.microedition.midlet`
 - `javax.microedition.rms`



Tecnología Web 

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



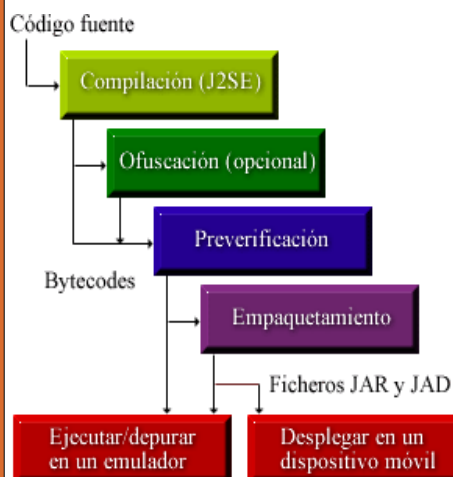
Tecnología Web 

5. El ciclo de desarrollo Java ME



Tecnología Web

Punto de vista del desarrollador



- **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



Tecnología Web

Esqueleto de un MIDlet

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

public class MiMIDlet extends MIDlet {
    protected void startApp()
        throws MIDletStateChangeException {
        // Estado activo -> comenzar
    }

    protected void pauseApp() {
        // Estado pausa -> detener hilos
    }

    protected void destroyApp(boolean incondicional)
        throws MIDletStateChangeException {
        // Estado destruido -> liberar recursos
    }
}
```



Tecnológicas Web

Herramientas

- Sun WTK
- Plugins para Eclipse
- NetBeans Mobility Pack
- ...



Tecnológicas Web

Punto de vista del usuario

■ 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



Tecnologías  Web

■ 3. Interfaz gráfica

■ Mostrar información y recibir entradas

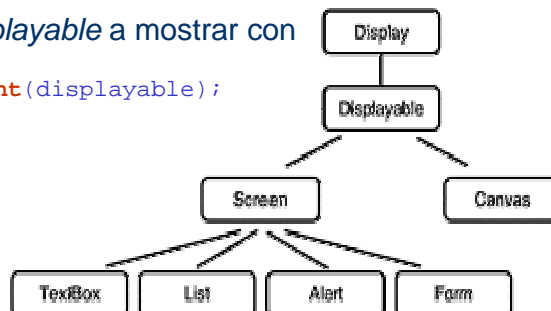


Tecnologías  Web

Display

- El *display* representa el visor del móvil: pantalla y teclado
- Cada MIDlet tiene asociado uno y sólo un *display*
`Display display = Display.getDisplay(midlet);`
- El *display* puede mostrar uno y solo un *displayable*
- Establecemos el *displayable* a mostrar con

```
display.setCurrent(displayable);
```



Tecnologías Web

Ejemplo 1: campos de texto

```
TextBox tb = new TextBox("Contraseña",  
    "", 8, TextField.ANY |  
    TextField.PASSWORD);  
  
Display d = Display.getDisplay(this);  
  
d.setCurrent(tb);
```



Tecnologías Web

Ejemplo 2 : Listas (menús, radio, checkboxes,...)

```

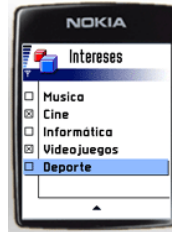
List l = new List("Menu",
                  Choice.IMPLICIT);
l.append("Nuevo juego", null);
l.append("Continuar", null);
l.append("Instrucciones", null);
l.append("Hi-score", null);
l.append("Salir", null);

Display d =
    Display.getDisplay(this);

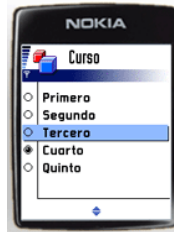
d.setCurrent(l);
    
```



Implícita



Múltiple



Exclusiva



Tecnológicas Web

Ejemplo 3: Formularios

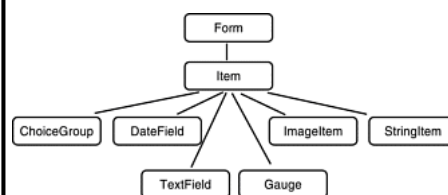
```

Form f = new Form("Formulario");

Item itemEtiqueta = new StringItem(
    "Etiqueta:",
    "Texto de la etiqueta");
Item itemTexto = new TextField(
    "Telefono:", "", 8,
    TextField.PHONENUMBER);
Item itemFecha = new DateField(
    "Fecha",
    DateField.DATE_TIME);
Item itemBarra = new Gauge("Volumen",
    true, 10, 8);

f.append(itemEtiqueta);
f.append(itemTexto);
f.append(itemFecha);
f.append(itemBarra);

Display d = Display.getDisplay(this);
d.setCurrent(f);
    
```



Tecnológicas Web

Entrada de datos

- **Comandos:** opciones que aparecen en las “esquinas” de la pantalla. Si no caben todos, el dispositivo los pone en un menú automáticamente



Tecnológicas Web

Creación de comandos

- Podemos crear comandos y añadirlos a un *displayable*

```
TextBox tb = new TextBox("Login", "", 8, TextField.ANY);
Command cmdOK = new Command("OK", Command.OK, 1);
Command cmdAyuda = new Command("Ayuda", Command.HELP, 1);
Command cmdSalir = new Command("Salir", Command.EXIT, 1);
Command cmdBorrar = new Command("Borrar", Command.SCREEN, 1);
Command cmdCancelar = new Command("Cancelar", Command.CANCEL, 1);

tb.addCommand(cmdOK);
tb.addCommand(cmdAyuda);
tb.addCommand(cmdSalir);
tb.addCommand(cmdBorrar);
tb.addCommand(cmdCancelar);

Display d = Display.getDisplay(this);
d.setCurrent(tb);
```



Tecnológicas Web

Otros APIs comunes de Java ME



Tecnologías  Web

Algunas operaciones “comunes” en Java ME

- **Conexión con el servidor**
 - Se pueden recibir/enviar datos vía HTTP
- **Envío/recepción de SMSs**
 - Se puede utilizar también para comunicar aplicaciones entre sí
- **Activación por *push***
 - Disparar una aplicación automáticamente cuando sucede un evento (hora, recepción de un SMS,...)
- **Almacenamiento de datos persistentes**
 - RMS: una especie de “ficheros de acceso aleatorio”



Tecnologías  Web

Conexión con el servidor

- Se pueden abrir conexiones HTTP
- No es habitual leer/enviar HTML (para eso usaríamos un cliente *thin*, no *thick*). Se suele leer/enviar
 - XML: + descriptivo
 - Datos binarios: +compacto (-coste)
- Para leer/enviar datos binarios, hay que usar una serie de métodos del paquete `java.io` que no hemos visto

```
HttpConnection con = (HttpConnection)Connector.open(
    "http://java.ua.es/verProducto.jsp?cod=1");
DataInputStream dis = con.openDataInputStream();
String nombre = dis.readUTF();
int precio = dis.readInt();
dis.close();
```



Tecnologías  Web

Envío/recepción de SMS

- API opcional llamado WMA (*recordad que opcional = no todos los móviles compatibles Java ME están obligados a tenerlo*)

■ Envío

```
MessageConnection mc = (MessageConnection)
    Connector.open("sms://+34555000000:6226");
String texto =
    "Este es un mensaje corto de texto";
TextMessage msg = mc.newMessage(mc.TEXT_MESSAGE);
msg.setPayloadText(texto);
mc.send(msg);
```

■ Recepción

```
MessageConnection mc = (MessageConnection)
    Connector.open("sms://:6226");
Message msg = mc.receive();
```



Tecnologías  Web

Activación por *push*

- **Pull:** el usuario pone en marcha la aplicación
- **Push:** la aplicación se pone en marcha por algún evento
 - Temporizador (alarma)
 - Conexión entrante (p.ej. SMS recibido)
- Por ejemplo, el servidor puede mandar un SMS *por un puerto determinado* para avisar de que hay una nueva entrada en el blog

