

## Capítulo 8

# APLICACIONES DESARROLLADAS

## 8.1 Introducción

En este capítulo se explica el desarrollo de una aplicación que implementa un servicio de educación a distancia, en el ámbito de los teléfonos móviles. Este servicio permite al usuario recibir contenido didáctico y hacer exámenes virtuales.

Los componentes que intervienen en este servicio pueden dividirse en dos:

- **Servidor.** Dispositivos y conjunto de programas que constituyen el contenido de las clases virtuales y los exámenes. Al estar usándose Internet para transmitir estos documentos, deben estar especificados en lenguajes de marcas.
- **Cliente.** Incluye los dispositivos y los programas usados, los microbrowsers, por el usuario del servicio.

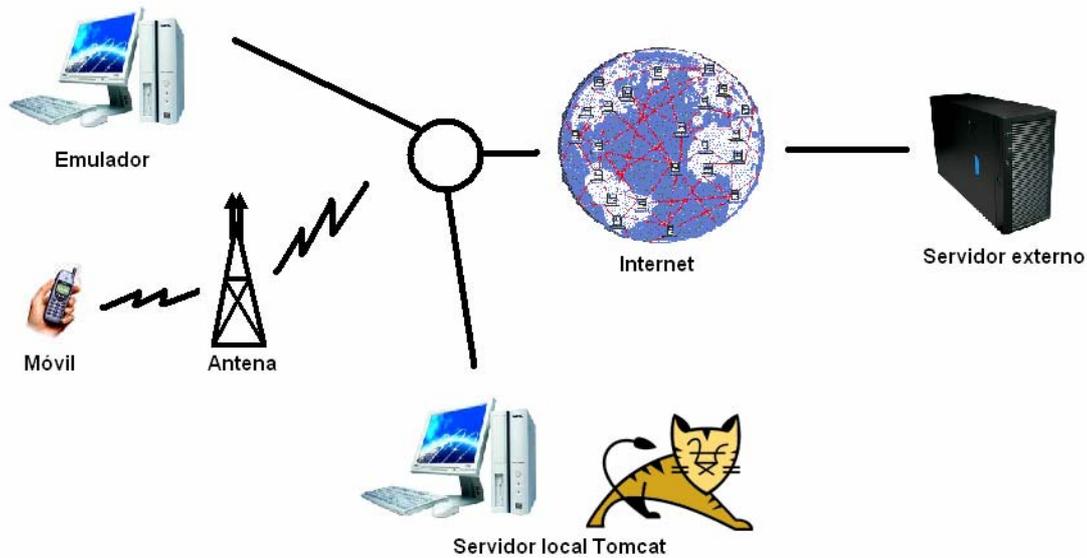
También se explica en este capítulo la creación de contenido web para realizar las pruebas de funcionamiento de los microbrowsers, tanto en HTML, como en los propios de dispositivos móviles como el XHTML MP y el WML. En los siguientes apartados se detallan las aplicaciones, servidores y clientes usados.

## 8.2 Aplicaciones desarrolladas

El lenguaje de marcas WML (*Wireless Marked Language*) sirve para crear contenido web específico para dispositivos móviles. Se crearon algunas páginas para probar el funcionamiento de los microbrowsers que interpretan este lenguaje.

También se han creado páginas tanto en el lenguaje de marcas habitual de Internet, el HTML (*Hipertext Marked Language*) como en el citado WML para implementar una aplicación de educación a distancia.

En el siguiente esquema puede verse el ámbito de funcionamiento de las aplicaciones. Por un lado está el cliente (móvil o emulador de móvil) y por el otro los servidores (local o externo). Las aplicaciones deben almacenarse en el árbol de directorios de uno de los servidores.



**Figura 8.1** Esquema de ejecución de las aplicaciones.

En primer lugar se va a explicar la creación de contenido web para realizar pruebas de funcionamiento de los microbrowsers.

### 8.2.1 Creación de contenido web

Para estudiar el funcionamiento de los microbrowsers se han creado varias páginas web en distintos tipos de lenguajes de marcas, su código puede encontrarse en el apéndice de esta memoria:

- **HTML** (*Hipertext Marked Language*). El contenido web escrito en este lenguaje es muy abundante dentro de Internet y fue utilizado para realizar las pruebas de funcionamiento de los microbrowsers. Además de usar páginas existentes se han creado algunas páginas de ejemplo para explicar el funcionamiento de los lenguajes de marcas.
- **WML** (*Wireless Marked Language*). Este es el lenguaje de marcas más extendido en el entorno inalámbrico. Al tratarse de un lenguaje pensado para dispositivos de pequeña pantalla, no existen muchas páginas de esta especificación que contengan imágenes. Por esta razón fue necesario diseñar varias páginas para probar los distintos formatos de imágenes soportados por los microbrowsers que usan este lenguaje, como son WBMP, PNG y GIF. La etiqueta para poner imágenes en las páginas WML es `<img>` y se usó con los atributos que indican el archivo fuente (`src`), un texto alternativo por si no pudiera verse (`alt`), la altura (`height`) y el ancho (`width`). Por ejemplo:

```



```

**Figura 8.2** Código para mostrar una imagen en WML.

- **XHTML MP** (*XHTML Mobile Profile*). Se trata de un lenguaje para dispositivos móviles creado recientemente, subconjunto de HTML, por lo que el número de páginas existentes es muy reducido. Se ha creado alguna página de ejemplo para explicar el funcionamiento de los lenguajes de marcas bien formados para teléfonos móviles.

### 8.2.2 Servicio de educación a distancia

La educación a distancia (en inglés *e-learning*) es un servicio que incluye la posibilidad de descargarse contenidos didácticos, comunicarse con el responsable de una asignatura y también responder cuestionarios a modo de exámenes a través de un navegador de Internet. Se va a diseñar una aplicación especificada en lenguajes de marcas para implementar este servicio en teléfonos móviles.

Estos dispositivos tienen una serie de limitaciones como una pantalla pequeña y un modo de entrada de datos restringido. Son dos aspectos a tener en cuenta para el diseño de la aplicación. Los cuestionarios serán creados mediante una serie de formularios que incluyen cajas de texto, botones, menús desplegados, etc; según las posibilidades que el lenguaje de marcas ofrezca. Se ha implementado en WML y en HTML.

#### 8.2.2.1 Servicio de educación a distancia en WML

Las etiquetas empleadas en WML son:

- La etiqueta `<input>` sirve para crear una caja de texto.
- La etiqueta `<select>` muestra una lista de opciones entre las que el usuario puede elegir una o más. Cada opción se define con la etiqueta `<option>` ó `<optgroup>` (muestra un desplegable, pero no es soportado por todos los dispositivos) y se muestra en el orden en el que lo especifiquemos.
- El formulario se envía mediante `<go>`, indicando el método “post” y la dirección completa de la dirección destino. Los campos que se van a enviar se definen mediante la etiqueta `<postfield>`. Estos valores pueden ser ocultos, enviar información no solicitada por `<input>`. Un campo tiene un nombre y un valor. Para que se obtengan los valores de las cajas de texto se especifican como `$(NOMBRE_DEL_CAMPO)`.

El formulario creado en WML para el servicio de educación a distancia, también disponible en el apéndice, es:

```

<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card title="Formulario">
<p>
<fieldset title="login">
Nombre de usuario:<br/>
<input type="text" title="Nombre de usuario" name="Ident"/><br/>
Email<br/>
<input type="text" maxlength="100" title="Email" name="Email" />
</fieldset>
1- Un motor de 8 cilindros en V, ¿ cuantos cigüñales tiene.?
<p>
<select iname="option1" ivalue="1">
<option>a) Uno</option>
<option>b) Dos</option>
<option>c) Cuatro</option>
</select>
</p>
<p>
2- Los segmentos de compresión impiden la fuga de gases.
<select iname="option2" ivalue="2">
<option>a) Al cárter.</option>
<option>b) Al colector de admisión.</option>
<option>c) A la culata.</option>
</select>
</p>
<do type="accept" label="Enviar formulario">
<go method="post" href="http://forms.melodysoft.com/index.html">
<postfield name="id" value="CÓDIGO_IDENTIFICATIVO_DEL_FORMULARIO"/>
<postfield name="Ident" value="$(nombre)"/>
<postfield name="Email" value="$(mensaje)"/>
<postfield name="option1" value="$(uno)"/>
<postfield name="option2" value="$(dos)"/>
</go>
</do>
</p>
</card>
</wml>

```

**Figura 8.3** Código WML con formulario.

### 8.2.2.2 Servicio de educación a distancia en HTML

En HTML los formularios se definen por medio de las etiquetas `<form>` y `</form>`. Entre estas dos etiquetas colocaremos todos los campos y botones que componen el formulario, de una forma muy similar a como se hace en WML. Dentro de esta etiqueta `<form>` debemos especificar algunos atributos:

- **action.** Define el tipo de acción a llevar a cabo con el formulario. Existen dos posibilidades:
  - El formulario es enviado a una dirección de correo electrónico.
  - El formulario es enviado a un programa o script que procesa su contenido.

- **method.** Este atributo se encarga de especificar la forma en la que el formulario es enviado. Los dos valores posibles que puede tomar este atributo son “post” y “get”. Se suele usar “post” porque es más seguro.
- **enctype.** Se utiliza para indicar la forma en la que viajará la información que se mande por el formulario. En el caso más corriente, enviar el formulario por correo electrónico, el valor de este atributo debe de ser "text/plain". Así conseguimos que se envíe el contenido del formulario como texto plano dentro del email.

Para crear las cajas de texto y los botones se usan las mismas etiquetas <input> y <select> de WML.

El formulario que se creó para el servicio de educación a distancia en HTML es el siguiente:

```
<HTML>
<HEAD>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
<TITLE></TITLE>
</HEAD>
<BODY>
<form action="mailto:test@mecanico.com" method="post"
enctype="text/plain">
Nombre <input type="text" name="nombre" size="30" maxlength="100">
<br>
Email <input type="text" name="email" size="25" maxlength="100"
value="@">
<br> <font color="#0066FF">
1- Un motor de 8 cilindros en V, ¿ cuantos cigüeñales tiene.?
<br> <font color="#000000">
<input type="radio" name="p2" value="a2">
<b>a)</b>Uno.
<br>
<input type="radio" name="p2" value="b2">
<b>b)</b>Dos.
<br>
<input type="radio" name="p2" value="c2">
<b>c)</b>Cuatro.
<br> <font color="#0066FF">
2- Los segmentos de compresión impiden la fuga de gases.
<br> <font color="#000000">
<input type="radio" name="p3" value="a3">
<b>a)</b>Al cárter.
<br>
<input type="radio" name="p3" value="b3">
<b>b)</b>Al colector de admisión.
<br>
<input type="radio" name="p3" value="c3">
<b>c)</b>A la culata.
<br>
<input type="submit" value="Enviar formulario">
<br>
</form>
</BODY>
</HTML>
```

Figura 8.4 Código HTML con formulario.

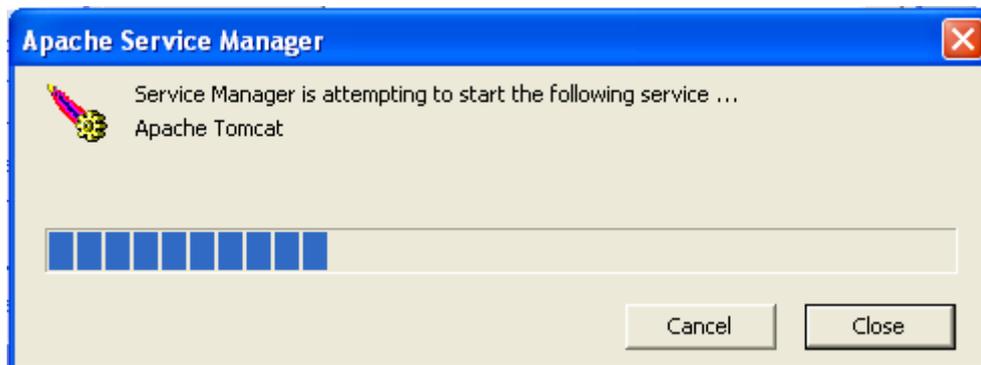
A continuación se explican los servidores que se usaron para almacenar las aplicaciones explicadas anteriormente.

### 8.3 Servidor Web

Un servidor web es un programa que implementa el protocolo HTTP (*hypertext transfer protocol*). Este protocolo está diseñado para transferir las páginas web, que pueden estar escritas en diferentes lenguajes

- Lenguajes de marcas propios de la tecnología inalámbrica (*Wireless*). Como es el caso de WML.
- Lenguaje HTML, es el usado habitualmente en las páginas web destinadas a los ordenadores personales.

Se han utilizado servidores remotos accesibles a través de Internet y el servidor local Apache Tomcat, versión 5.5, ejecutado en un ordenador personal. La versión utilizada se puede descargar de la página web de Apache [49]. Este servidor es de fácil manejo gracias a su interfaz gráfica muy intuitiva. Tras completar la instalación y arrancar el programa, sin necesidad de configuración adicional, es capaz de servir páginas. En la siguiente figura se muestra el servidor iniciándose.



**Figura 8.5** Arrancando el servidor Apache Tomcat.

El ordenador personal en el que se instaló el servidor Tomcat tenía las siguientes características:

- Procesador Intel Pentium IV a 3 GHz Prescott.
- 1024 Megas de RAM.
- 80 Gigas de disco duro.

Los servidores almacenan las páginas web que sirven en un árbol de directorios específico desconocido por el cliente. En el caso de Tomcat, por defecto está en C:\Archivos de programa\Apache Software Foundation\Tomcat 5.5\webapps\ROOT. Para acceder a sus páginas hay que usar la dirección URL, que incluye el nombre del servidor y de la página que queramos ver. Por ejemplo, en el caso de Tomcat la URL tiene una estructura como la siguiente: <http://localhost:8080/-nombredeweb>, donde

“localhost” es el nombre del servidor, “8080” es el puerto y “nombredeweb” es el nombre de la página.

El último componente necesario para ejecutar las aplicaciones desarrolladas que queda por explicar es el cliente.

## 8.4 Cliente Web

El cliente está formado por los dispositivos y programas empleados por el usuario del servicio web, en principio son teléfonos móviles y microbrowsers, pero en vez de usar un móvil se ha utilizado un emulador, *Sun Java Wireless Toolkit 2.5 Beta*, porque permite monitorizar más aspectos del funcionamiento y almacenar los resultados de forma más sencilla. Se trata de un conjunto de herramientas que permiten el desarrollo de aplicaciones inalámbricas basadas en CLDC y MIDP, las propias de los teléfonos móviles, PDA's y otros dispositivos de este estilo. Este emulador está disponible en la página de Sun Microsystems [48]:

Los siguientes son los requisitos software para su instalación:

- Microsoft Windows XP.
- *Java™ 2 Standard Edition Runtime Environment (JRE)*, version 1.5.0.

Los requisitos hardware son:

- 50 MB de disco duro.
- 128 MB de RAM.
- Procesador 800 MHz Pentium III.

El emulador sirve para ejecutar MIDlets, que son las aplicaciones de MIDP. Los microbrowsers son un tipo de MIDlet que sirve para acceder a Internet y para mostrar contenido web. Se han empleado cinco modelos distintos:

- jBrowser 1.0.4.
- WebViewer 3.2.
- WebViewer 4.0.
- Opera Mini Basic.
- Opera Mini Advanced.

Cada MIDlet consta de dos archivos, un JAR (*Java Archive*) y un JAD (*Java Archive Descriptor*). El primero contiene el ejecutable y el otro presenta información asociada. La información incluida en los archivos JAD de los microbrowsers empleados es la siguiente:

De jBrowser 1.0.4:

```
MIDlet-1: jBrowser,/icons/jataayu.png, com.jataayu.Browser.jBrowser
MIDlet-Jar-Size: 61117
MIDlet-Jar-URL: http://www.phonesoft.cz/java/469/jBrowser.jar
MIDlet-Name: jBrowser
MIDlet-Vendor: Jataayu Software (P) Ltd.
MIDlet-Version: 1.0.4
```

**Figura 8.6** Descriptor de jBrowser 1.0.4.

De WebViewer versión 3.2:

```
MIDlet-1: WebViewer, /w.png, WebViewer
MIDlet-Jar-Size: 47916
MIDlet-Jar-URL: WebViewer.jar
MIDlet-Name: WebViewer
MIDlet-Vendor: Reqwireless Inc.
MIDlet-Version: 3.2
MIDlet-Data-Size: 6144
```

**Figura 8.7** Descriptor de WebViewer 3.2.

De WebViewer versión 4.0:

```
MIDlet-1: WebViewer, /w.png, WebViewer
MIDlet-Jar-Size: 49782
MIDlet-Jar-URL: http://www.reqwireless.com/WebViewer-4.0/WebViewer.jar
MIDlet-Name: WebViewer
MIDlet-Vendor: Reqwireless Inc.
MIDlet-Version: 4.0
MIDlet-Data-Size: 6144
```

**Figura 8.8** Descriptor de WebViewer 4.0.

De Opera Mini Basic:

```
MIDlet-Version: 2.0
MIDlet-1: Opera Mini, /icon1616.png, Browser
MIDlet-Data-Size: 10240
MIDlet-Description: Opera Mini
MIDlet-Info-URL: http://mini.opera.com/
MIDlet-Install-Notify: http://mini.opera.com/notify/2.0.4509/lofi_woodland_es
MIDlet-Jar-Size: 63174
MIDlet-Jar-URL: opera-mini-2.0.4509-basic-es.jar
MIDlet-Name: Opera Mini
MIDlet-Permissions: javax.microedition.io.Connector.http
MIDlet-Permissions-Opt: javax.microedition.io.Connector.sms,
javax.wireless.messaging.sms.send
MIDlet-Vendor: Opera Software ASA
Content-Folder: Applications
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-1.0
```

**Figura 8.9** Descriptor de Opera Mini Basic.

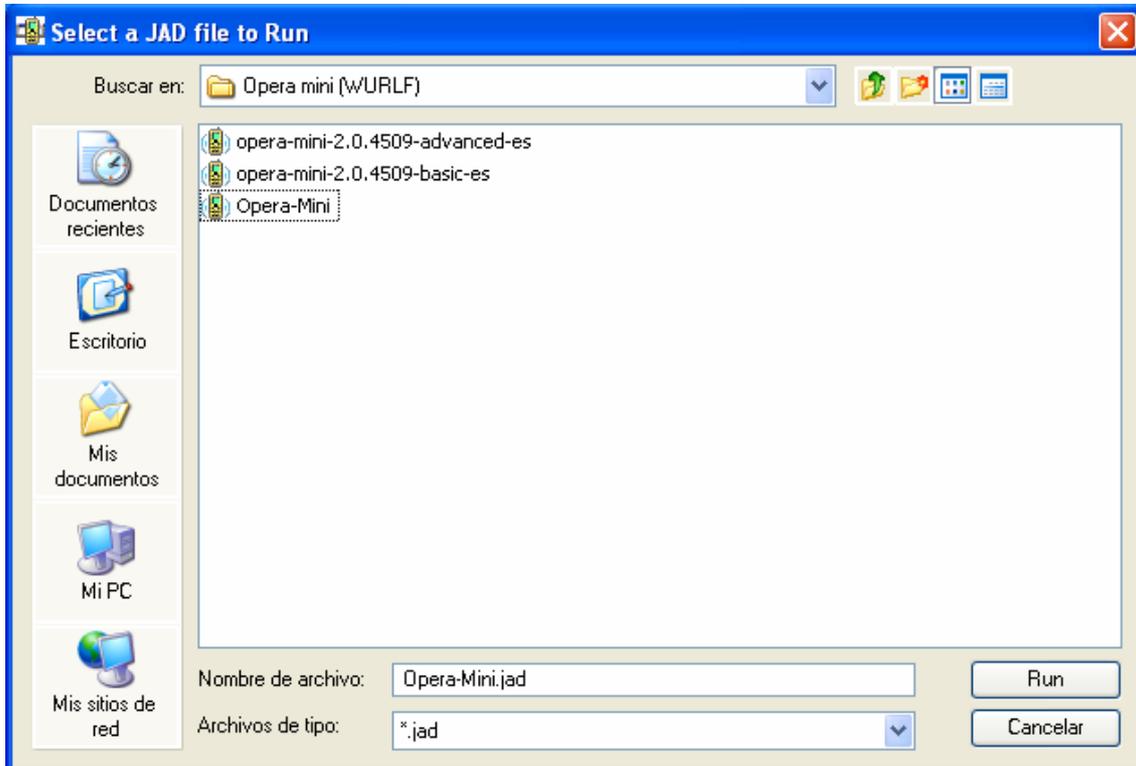
De Opera Mini Advanced:

```

MIDlet-Version: 2.0
MIDlet-1: Opera Mini, /icon1616.png, Browser
MIDlet-Description: Opera Mini
MIDlet-Info-URL: http://mini.opera.com/
MIDlet-Install-Notify: http://mini.opera.com/notify/2.0.4509/hifi_woodland_es
MIDlet-Jar-Size: 98171
MIDlet-Jar-URL: opera-mini-2.0.4509-advanced-es.jar
MIDlet-Name: Opera Mini
MIDlet-Permissions: javax.microedition.io.Connector.http
MIDlet-Permissions-Opt: javax.microedition.io.Connector.sms,
javax.wireless.messaging.sms.send
MIDlet-Vendor: Opera Software ASA
RIM-COD-SHA1-1: 99 27 0e af 86 2f 2e 76 69 c7 62 1c 18 4c 6c a1 18 02 bb 2d
RIM-COD-Size-1: 64700
RIM-COD-URL-1: http://mini.opera.com/builds/releases/operette-2.0.4509/operette-hifi-es-1.cod
RIM-COD-Creation-Time: 1152007420
RIM-COD-Module-Dependencies: net_rim_cldc
RIM-COD-Module-Name: operette-hifi-es
RIM-COD-SHA1: 65 35 91 3f f5 9c 39 26 e4 b9 4e 20 a8 e8 4a 6c bc 5f 0f 93
RIM-COD-Size: 75056
RIM-COD-URL: http://mini.opera.com/builds/releases/operette-2.0.4509/operette-hifi-es.cod
Content-Folder: Applications
Created-By: 1.5.0_05-b05 (Sun Microsystems Inc.)
MicroEdition-Configuration: CLDC-1.0
MicroEdition-Profile: MIDP-2.0
    
```

**Figura 8.10** Descriptor de Opera Mini Advanced.

*Sun Java Wireless Toolkit 2.5 Beta*, mediante la opción “Run MIDP Application”, permite la ejecución de MIDlets en ordenadores personales. En primer lugar se selecciona el fichero JAD, que invocará al JAR. En la siguiente figura se muestra la solicitud del fichero JAD.



**Figura 8.11** Ejecutar un MIDlet.

A continuación aparece un display con forma de teléfono móvil como se ve en la siguiente figura con la lista de las MIDlets instaladas.



**Figura 8.12** Display de Sun Java Wireless Toolkit 2.5 Beta.

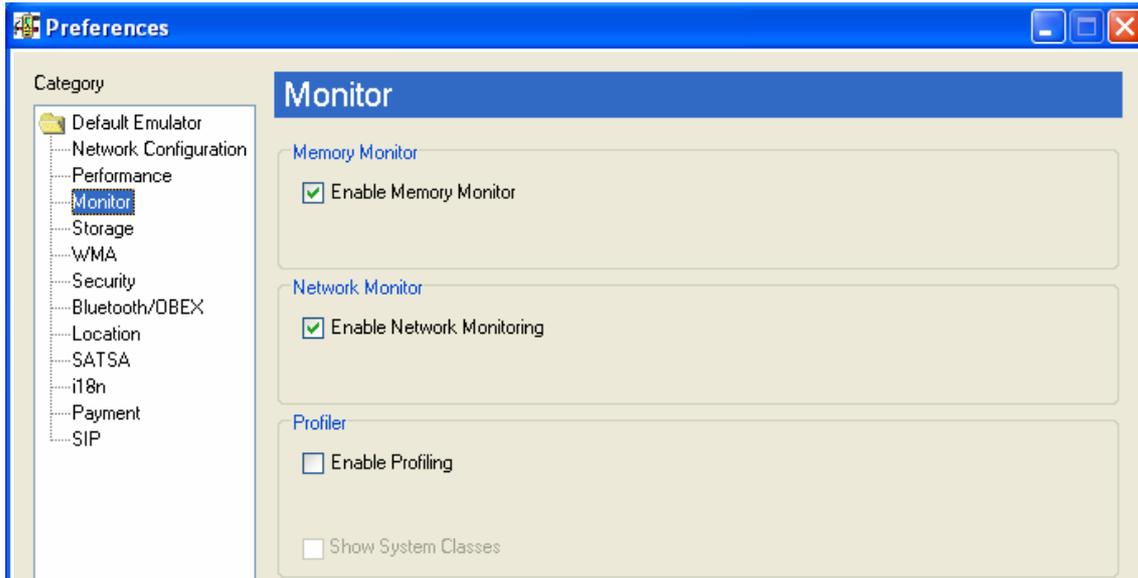
Otras opciones que tiene el emulador *Sun Java Wireless Toolkit 2.5 Beta* es la edición de MIDlets. Para ello está la herramienta “KToolbar”.

Como se está usando un ordenador personal para ejecutar los microbrowsers, se ha utilizado una conexión ADSL para descargarlos. En los teléfonos móviles habría que realizar un proceso de descarga llamado OTA (*Over the Air*), tal y como se explica en el

capítulo 6 dedicado a J2ME. El emulador puede simular este proceso mediante la opción “OTA Provisioning” comportándose como un Gestor de Aplicaciones.

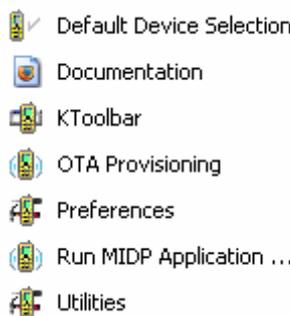
En “Preferences” podemos variar algunos aspectos de la simulación, como el tipo de protocolo HTTP, configurar tamaños de buffer, conexiones Bluetooth, etc.

También existe la posibilidad de monitorizar el uso de la memoria y de la red en tiempo real mediante la opción *Memory Monitor Extension* y *Network Monitor Extension*. Para que activarlas hay que seleccionar las casillas “Enable Memory Monitor” y “Enable Network Monitor”, tal y como se muestra en la siguiente figura:



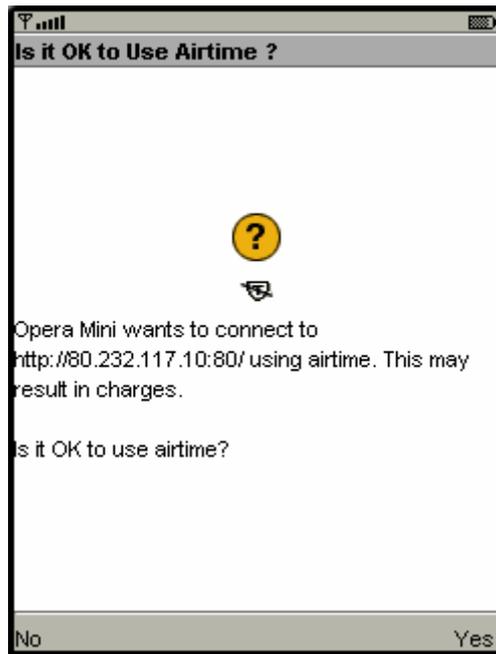
**Figura 8.13** Monitorización en tiempo real de la memoria y del uso de la red.

En la siguiente figura se muestran todas las opciones disponibles en *Sun Java Wireless Toolkit 2.5 Beta*.



**Figura 8.14** Opciones de *Sun Java Wireless Toolkit 2.5 Beta*.

Los microbrowsers necesitan acceder al medio inalámbrico para transmitir y recibir información, cada vez que esto se produce aparece una advertencia, como la que se ve en la siguiente figura, sobre una posible tarificación por parte de la operadora de telefonía.



**Figura 8.15** Aviso de tarificación

Otra de las ventajas de usar un emulador de dispositivos móviles es no tener que solicitar acceso al medio inalámbrico, evitando pagar a la operadora de telefonía por cada transmisión de datos.

## 8.5 Conclusiones

En este capítulo se han explicado las aplicaciones desarrolladas:

- Servicio de educación a distancia implementado con lenguajes de marcas.
- Diseño de páginas web en HTML, XHTML MP y WML.

El código de todas estas aplicaciones puede encontrarse en el apéndice “Aplicaciones desarrolladas” de esta memoria.

El entorno de ejecución de las aplicaciones consta de un lado servidor y otro cliente. En el lado del servidor se han usado servidores externos y el local Apache Tomcat para almacenar el código de las aplicaciones. Por el lado del cliente están los microbrowsers, programas para teléfonos móviles que permiten el acceso a Internet. Para poder monitorizar su funcionamiento se han instalado en el emulador *Sun Java Wireless Toolkit 2.5 Beta* en vez de en un dispositivo real.