Capítulo 9

RESULTADOS

9.1 Introducción

En este capítulo se analiza el funcionamiento de varios microbrowsers. Estos son programas que permiten el acceso a Internet para ofrecer contenidos especificados en lenguajes de marcas para los teléfonos móviles. Tienen una serie de limitaciones asociadas a su ámbito de funcionamiento tales como la baja capacidad de memoria, pantallas pequeñas, dificultad para introducir datos, procesadores lentos, etc.

Ante estas restricciones, los microbrowsers usan diferentes mecanismos para buscar una navegación web satisfactoria tales como emplear servidores externos, usar páginas web adaptadas para este entorno inalámbrico, etc.

Para estudiar su funcionamiento, los microbrowsers se han ejecutado en un emulador de dispositivos móviles, *Sun Java Wireless Toolkit 2.5 Beta*, que, mediante su interfaz gráfica, permite una monitorización de los recursos empleados y almacenar los resultados obtenidos.

Se observaron los siguientes puntos:

- Tamaño del MIDlet.
- Lenguaje de marcas soportado.
- Disponibilidad del MIDlet.
- Tamaño de carga de las páginas.
- Uso de memoria.
- Velocidad de proceso.
- Modo de navegación.
- Web de fabricante.
- Compatibilidad con los dispositivos.
- Imágenes.
- Uso de scripts.
- Hojas de estilo.
- Uso del protocolo HTTP.
- Visualizar otro tipo de archivos.
- Uso de formularios.
- Otras opciones.

Se estudiaron cinco modelos de microbrowsers: J.Browser 1.0.4, WebViewer 3.2, WebViewer 4.0, Opera Mini Basic y Opera Mini Advanced. Algunos de ellos son distintas versiones de un mismo programa, en tal caso, en los apartados que tengan los mismos resultados se presentan como uno solo.

9.2 Tamaño del MIDlet

Los dispositivos móviles tienen una capacidad de almacenamiento reducida, que limita el tamaño de las MIDlets que se quiera instalar en ellos. Tales capacidades se miden en Kilobytes (Kb), Megabytes (Mb) y Gigabytes (Gb).

La capacidad tiene relación con el perfil (MIDP) al que pertenezca el teléfono:

- Los móviles de gama media, los englobados en el MIDP 1.0, suelen usar aplicaciones de reducidas dimensiones, de entre 50 y 60Kb.
- El perfil MIDP 2.0 aparece en el año 2004. A él pertenecen móviles más evolucionados tecnológicamente con mayor capacidad para almacenar aplicaciones y archivos. Existen modelos, como el Sony Ericsson W950i, con capacidad de hasta 4Gb, aunque lo más habitual es disponer de varios megas, como el Motorota V3, que posee 5.5Mb.

Las MIDlets se componen de dos archivos, de extensiones JAR (*Java Archive*) y JAD (*Java Archive Descriptor*). El primero es el que contiene el ejecutable y también el que más ocupa de los dos. Su tamaño no debe ser superior a la capacidad de memoria del teléfono para poder ser instalado. Los archivos JAR de los microbrowser analizados tienen el siguiente tamaño:

- J.Browser 1.0.4 59.6 Kb.
- WebViewer 3.2 46.8 Kb.
- WebViewer 4.0 48.6 Kb.
- Opera-Mini Basic 61.6 Kb.
- Opera-Mini Advanced 95.8 Kb.

Esos datos se han representado en la siguiente figura:

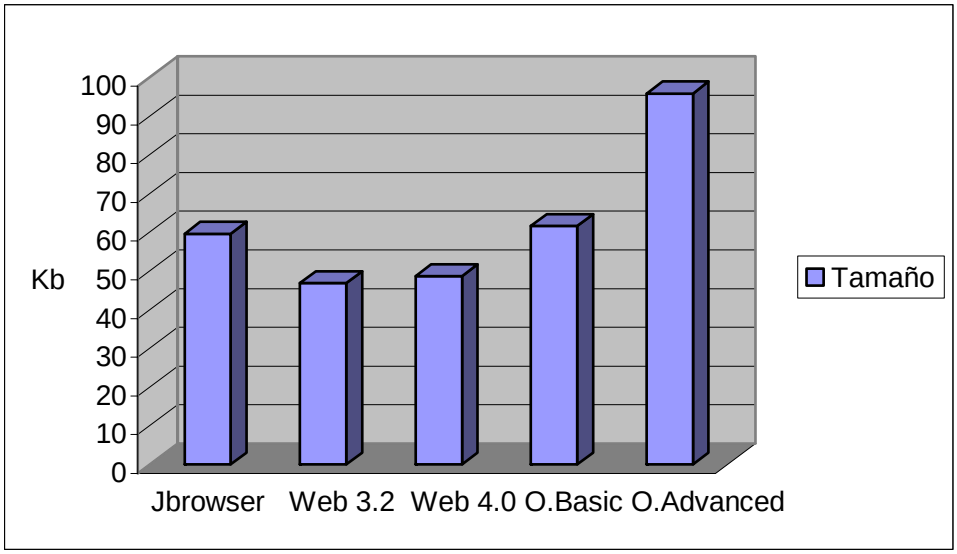


Figura 9.1 Tamaño del archivo JAR del MIDlet.

9.3 Lenguaje de marcas soportado

9.3.1 Introducción

Existen varios lenguajes de marcas para mostrar los contenidos en los dispositivos móviles. Un microbrowser concreto está especializado en leer sólo algunos de ellos. De esta manera pueden clasificarse por el tipo de lenguaje que leen:

- Lenguaje diseñado para dispositivos inalámbricos. Tales como WML o XHTML. Es el caso de jBrowser.
- Lenguaje HTML. Es el caso de WebViewer.
- Lenguaje HTML modificado. Es el caso de Opera Mini.

La página del buscador Google está escrita en todos los lenguajes de marcas existentes. Se ha utilizado para comprobar los lenguajes soportados por los microbrowsers.

9.3.2 WML

WML es un lenguaje web pensado para dispositivos inalámbricos, ámbito de trabajo de los microbrowsers.

J.Browser 1.0.4 está diseñado para leer únicamente este lenguaje. Opera Mini, en sus dos versiones, también puede ver estas páginas gracias a la traducción que le hace su servidor externo. Únicamente las dos versiones de WebViewer no son capaces de usar WML. En la siguiente figura se muestra la página http://www.google.es/wml vista por JBrowser y Opera Mini:

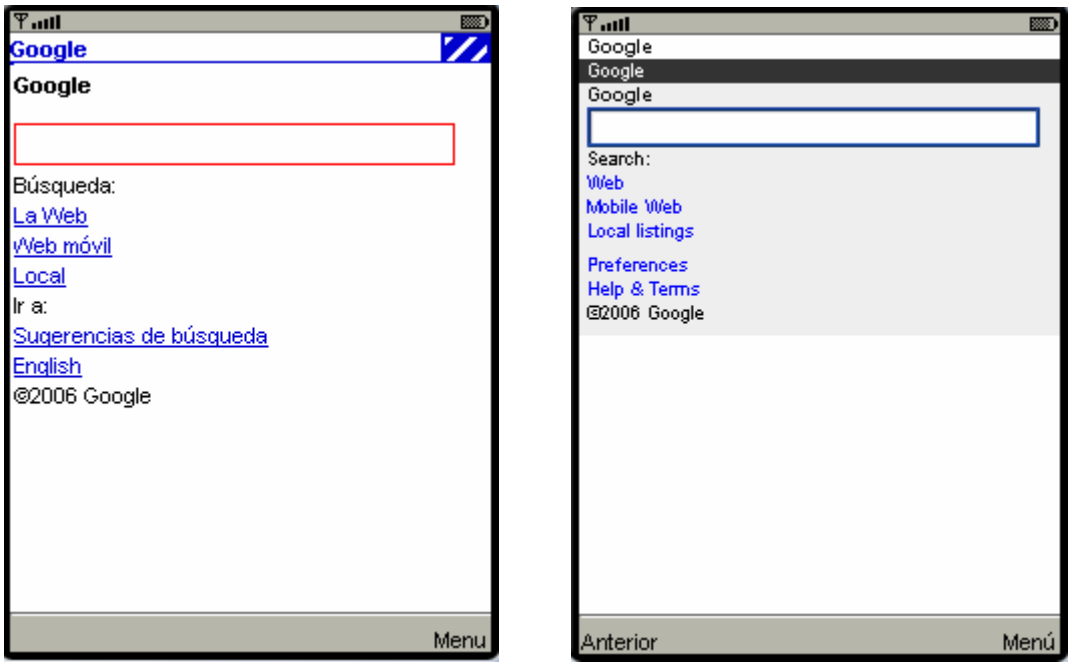


Figura 9.2 Google escrito en WML.

9.3.3 HTML

HTML es el lenguaje de marcas más extendido de Internet. Sin embargo puede verse mal en los dispositivos móviles al no estar pensado para ellos. Es trabajo del microbrowser adaptar la página a las limitaciones de los teléfonos, tales como sus pequeñas pantallas, lentitud de procesador, etc.

La página del buscador Google en HTML es http://www.google.es y se va a utilizar de ejemplo para probar cómo interpretan los microbrowsers este lenguaje.

- J.Browser 1.0.4 no soporta HTML, sólo interpreta WML.
- WebViewer está diseñado para mostrar HTML tal cual, sin adapatar a las pequeñas pantallas, por lo que las páginas suelen verse distorsionadas.
- Opera Mini modifica el código HTML en un servidor externo para adaptarlo al dispositivo y obtener una página con un aspecto parecido al original.

En la siguiente figura se muestra cómo se ve la página del buscador Google usando WebViewer y Opera Mini:

Figura 9.3 Google escrito en HTML visto por WebViewer y por Opera Mini.

En la siguiente figura se enseña la página del buscador Google mostrada correctamente por el navegador para ordenadores personales Firefox 1.5. Este es el aspecto original que los microbrowsers deben intentar mostrar. De ellos Opera Mini es el que más se acercó.

Figura 9.4 Google en HTML visualizado por el navegador Firefox 1.5.

9.3.4 XHTML

XHTML es un lenguaje similar a HTML. De él han surgido subconjuntos pensados para ser soportado en dispositivos móviles, como XHTML MP y XHTML Basic. Aquellos microbrowsers compatibles con HTML también lo son con este lenguaje.

La página del buscador Google en XHTML es http://www.google.es/xhtml. Al tratarse de un lenguaje adapatado al entorno inalámbrico WebViewer lo muestra con un aspecto mucho mejor al que ofrecía antes, como puede verse en la siguiente figura junto con la versión mostrada por Opera Mini:

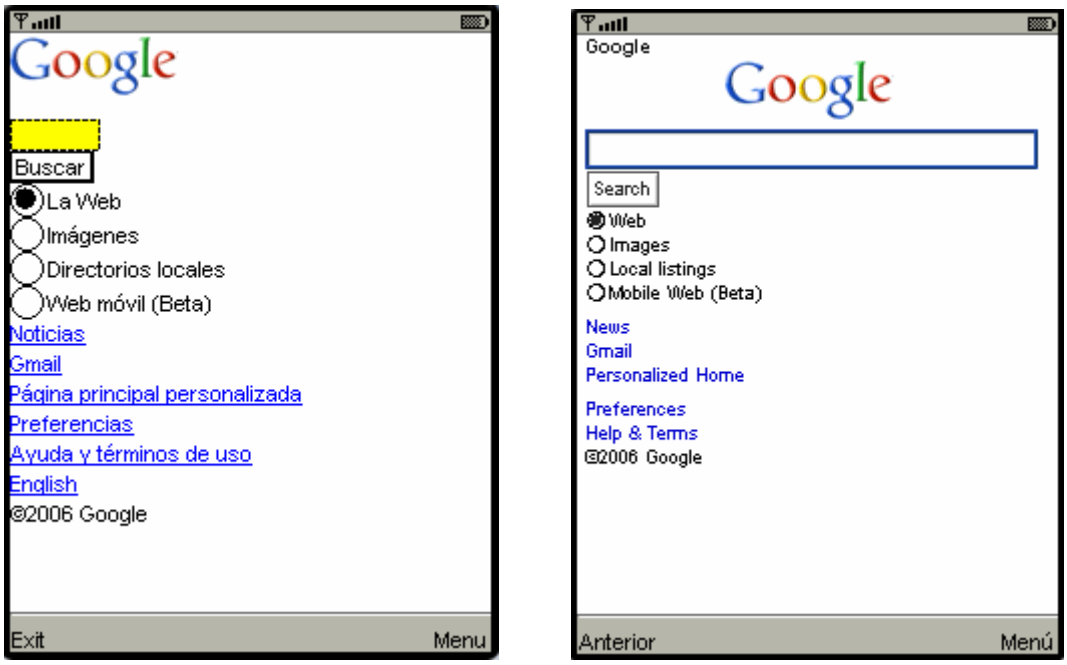


Figura 9.5 Google escrito en XHTML visto por WebViewer y Opera Mini Basic.

9.3.5 Conclusiones

La siguiente tabla muestra un resumen del apartado. Indica los lenguajes de marca que soportan los microbrowsers analizados.

	WML	**HTML**	**XHTML**
jBrowser	Sí	No	No
WebViewer	No	Sí	Sí
Opera Mini	Sí	Sí	Sí

Tabla 9.1 Lenguajes de marcas soportado

En cuanto a navegadores web para ordenadores de sobremesa, Internet Explorer 6.0 y Firefox 1.5 no soportan WML, únicamente leen HTML. En el caso de Firefox existe un plugin para permitirle interpretar WML. El navegador Opera 9.0 (versión para escritorio de Opera software) sí soporta ambos lenguajes.

9.4 Disponibilidad del MIDlet

Los microbrowsers suelen estar desarrollados por empresas con fines comerciales. Pueden estar sujetos a pago o disponibles durante un periodo de prueba. El precio de estos programas oscila entre 15€ y 40€. Aunque también existen algunos microbrowsers gratuitos, como es habitual en los navegadores para ordenadores de sobremesa. De los modelos analizados se puede decir lo siguiente:

- jBrowser: Versión de prueba para 60 días.
- WebViewer: Libre acceso.
- Opera-Mini: Libre acceso.

9.5 Tamaño de carga de las páginas.

La baja capacidad de procesamiento de los dispositivos móviles limita el tamaño de las páginas que pueden leer. La solución pasa por crear páginas pequeñas o por que sea el microbrowser quien las reduzca.

Para tener una idea de cuál es el tamaño adecuado para este tipo de dispositivos, el grupo de investigación, adscrito al W3C, *Best Practises Working Group* ha definido un contexto estándar, el *Default Delivery Context*, con las limitaciones que pueden esperarase de un móvil. Este grupo recomienda que el tamaño máximo de las páginas no exceda de los 20Kb.

El tamaño de las páginas WML oscila entre 1 y 5 Kb. Este lenguaje se creó para ser empleado en dispositivos inalámbricos en 1999, cuando la capacidad de los aparatos menor por lo no presentan problemas a los modelos actuales.

El tamaño de las páginas HTML está entre 10 y 250 Kb. Este lenguje no está pensado para dispositivos móviles, de ahí que algunos de sus contenidos no cumplan las recomendaciones del *Default Delivery Context* y por esta razón pueda crear problemas para los microbrowsers. El contenido informativo de las páginas ocupa sólo entre 1 y 30 Kb, el resto se trata de etiquetas para darle formato al texto, por lo que el microbrowser tiene margen para alterar la apariencia de la página respetando el contenido.

Otra manera de reducir el tamaño de carga de las páginas es hacer un tratamiento de compresión de imágenes, aunque a costa de perjudicar su resolución.

Los tamaños máximos de páginas que pueden leer los microbrowsers analizados son:

- jBrowser 6Kb.
- WebViewer Estimado en unos 400Kb.
- Opera Mini Límite no encontrado.

JBrowser está muy limitado en cuanto a tamaño de página debido a que usa WML. Cada vez que intenta leer una página de mayor tamaño del que puede interpretar muestra el mensaje de error "*Receive response is too large to process, and can't be processed.*".

El límite de WebViewer se encontró al ver el contenido de un archivo .ZIP, no una página web. No se ha encontrado ninguna página HTML lo suficientemente grande como para no poder ser abierta por WebViewer o por Opera Mini.

A continuación se muestra el proceso de carga de la página web del diario deportivo Marca, http://www.marca.es, en los microbrowsers. Se ha elegido esta página por su gran tamaño tanto de texto como de imágenes.

WebViewer muestra una barra de estado en la que se puede ver el proceso de la carga. En primer lugar interpreta el texto y luego las imágenes como puede verse en la siguiente figura. El proceso suele durar entre 1 y 2 segundos.

Figura 9.6 Proceso de carga de una web en WebViewer.

Opera Mini realiza la carga de texto e imágenes conjuntamente y permite detener el proceso. Usa un servidor externo que emplea algoritmos para reducir la cantidad bytes transmitidos.

Opera Mini Advanced puede emplear algoritmos aún más sofisiticados para reducir más el número de bytes a costa de perjudicar la calidad de las imágenes.

En la siguiente figura se representa la carga de la página web del diario Marca por parte de Opera Mini en versión Basic y Advanced. En este segundo caso sólo se muestra el contenido de la página una vez acabada la carga completa.

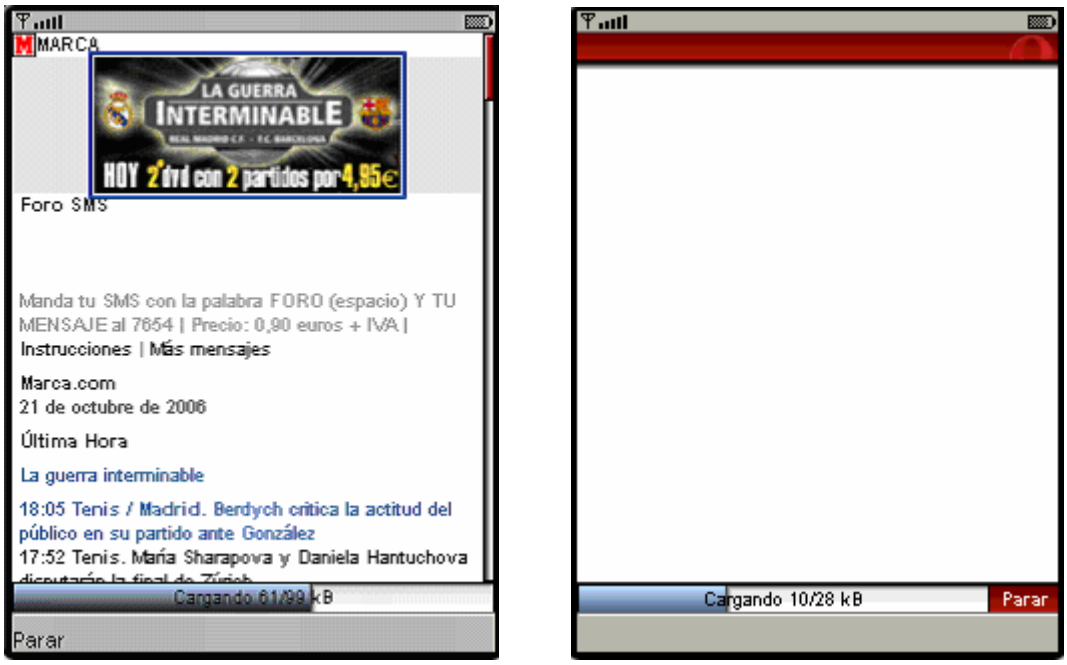


Figura 9.7 Proceso de carga de una web en Opera Mini Basic y Advanced.

En la siguiente tabla se recoge el tamaño cargado por Firefox 1.5 (este navegador se utilizará de referencia porque respeta el tamaño original de las páginas), WebViewer, Opera Mini Basic y Advanced para visualizar tres páginas HTML elegidas de ejemplo. No se incluye jBrowser 1.0.4 porque trabaja con otro tipo de lenguaje de marcas, WML, que no presenta problemas de tamaño por estar diseñado para dispositivos de reducida capacidad.

	www.esi.us.es		http://portal.us.es		www.marca.es	
	Texto	**Imagen**	**Texto**	**Imagen**	**Texto**	**Imagen**
Firefox 1.5	47,2 Kb	84,5 Kb	22 Kb	262 Kb	78,3 Kb	169 Kb
WebViewer	20 Kb	41 Kb	8 Kb	54 Kb	30 Kb	61 Kb
Opera Mini Basic		33 Kb		35 Kb		99 Kb
Opera Mini Advanced		16 Kb		17 Kb		28 Kb

Tabla 9.2 Tamaño de carga de páginas HTML.

Los datos indicados en la tabla anterior quedan representos gráficamente en la siguiente figura:

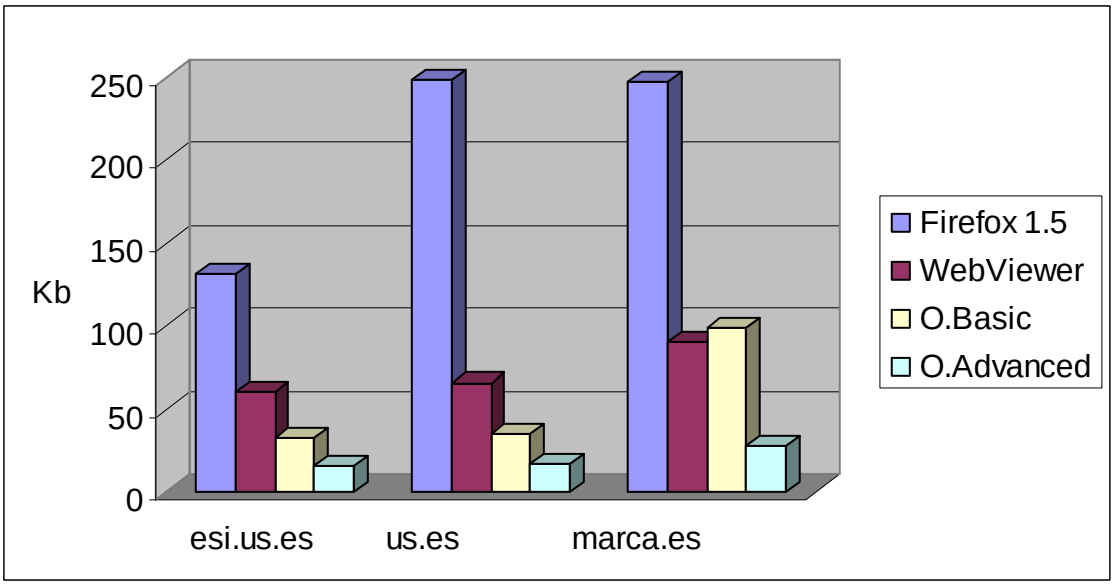


Figura 9.8 Tamaño de carga de páginas HTML

9.6 Uso de memoria

9.6.1 Introducción

Todo programa necesita ocupar recursos del dispositivo para funcionar. En este apartado se estudia el consumo de memoria de los microbrowsers, medido en bytes. Se han analizado varios momentos de su ejecución:

- El instante de carga de la aplicación.
- Estado de reposo.
- Navegación por páginas web.

Para obtener los datos de consumo se ha usado la monitorización de memoria (*Memory Monitor Extension*) disponible en el emulador *Sun Java Wireless Toolkit*. La siguiente tabla recoge los valores recogidos durante los instantes de carga y de reposo de los microbrowsers. Durante la carga se produce una exigencia de memoria del orden de cuatro veces superior al reposo. Se observa también que todas las versiones tienen consumos similares.

	Carga	**Reposo**
jBrowser	527000	144000
WebViewer 3.2	526000	91300
WebViewer 4.0	525000	118000
Opera-Mini Basic	538000	126000
Opera-Mini Advanced	552000	140000

Tabla 9.3 Uso de memoria.

Durante la navegación web se consume una cantidad de memoria proporcional al tamaño de la página. Otras acciones como la de pulsar botones del teléfono o el uso de scripts también incrementan el consumo.

El punto crítico para un correcto funcionamiento del microbrowser es la situación de consumo máximo. Si el dispositivo la soporta, se garantiza el perfecto funcionamiento del microbrowser en cuanto a recursos de memoria. Las distintas versiones estudiadas tienen sus consumos máximos en los siguientes momentos:

- En jBrowser se produce en el momento de carga. El tamaño reducido de las páginas WML hace que el consumo durante la navegación sea menor.
- En WebViewer 3.2 y Opera Mini Basic el consumo durante la navegación no suele elevarse respecto del momento de la carga ni siquiera en accesos a páginas de gran tamaño.
- La capacidad de WebViewer 4.0 de abrir documentos con otras extensiones como .ZIP se aprovechó para encontrar un límite en torno a un consumo de 2.020.000 de bytes en el que el emulador deja de funcionar correctamente. En cuanto a navegación web presenta el mismo consumo que la versión WebViewer 3.2.
- Opera Mini Advanced es el microbrowser que consume más memoria durante la navegación web. Está diseñado para dispositivos de perfil MIDP 2.0, que poseen mayor capacidad de memoria, así que puede permitirse usar algoritmos complejos que demandan mucha memoria. Se han observado consumos en torno a los 2.000.000 bytes, aunque sin llegar a saturar en ninguna ocasión.

En la siguiente gráfica se muestra el consumo en el instante de carga del MIDlet y el de reposo comparado con el tope encontrado en 2.020.000 bytes sobre el cual el emulador deja de funcionar. Se obtienen valores muy parecidos entre todos los microbrowsers

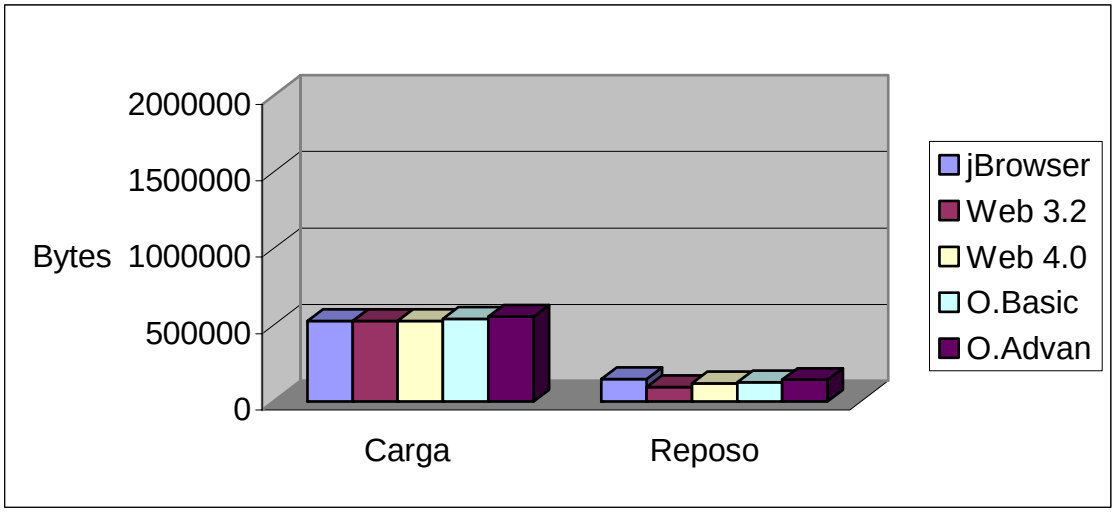


Figura 9.9 Consumos de memoria respecto al máximo.

En los siguientes apartados se muestran curvas de consumo obtenidas durante diferentes accesos a Internet desde cada microbrowser. Las gráficas se consiguieron con la utilidad *Memory Monitor Extension* del emulador *Sun Java Wireless Toolkit 2.5 Beta*.

9.6.2 JBrowser

JBrowser usa el lenguaje WML. El tamaño de estas páginas es pequeño, por lo que no plantea problemas de consumo de memoria del dispositivo. En la siguiente gráfica se muestra la curva de consumo al arrancar el programa, el pico más pronunciado, y al realizar el acceso a 3 páginas, que provocan picos muy reducidos.

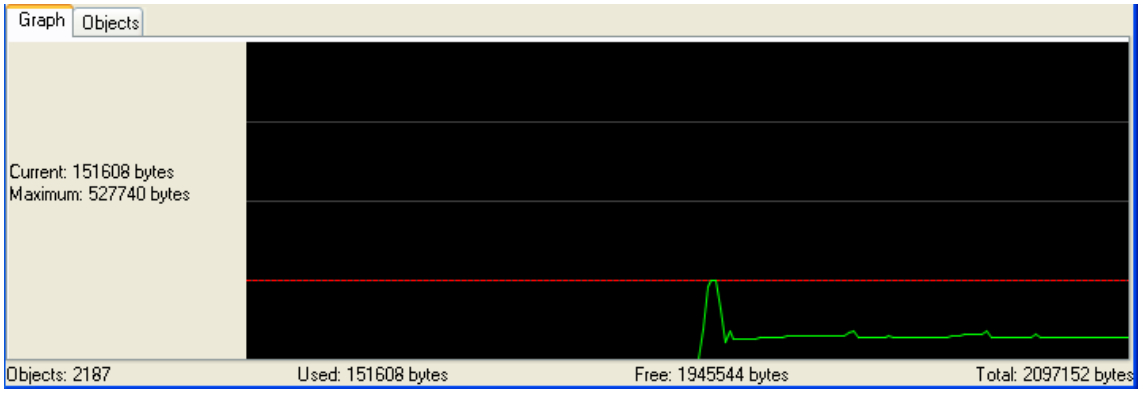


Figura 9.10 Curva de memoria ocupada por Jbrowser

9.6.3 WebViewer 3.2 y 4.0

Estos microbrowsers trabajan con el lenguaje HTML. El consumo de memoria es proporcional al tamaño de la página visitada y es mucho mayor al de WML. La gráfica representa el acceso a cuatro páginas web HTML:

- www.google.es
- http://portal.us.es
- www.esi.us.es
- http://www.tejedoresdelweb.com/307/article-1061.html

Las cimas aparecen en los momentos de navegación web, los valles durante los periodos intermedios. La tendencia ascendente de las cimas se debe al uso del teclado para introducir la siguiente dirección URL.

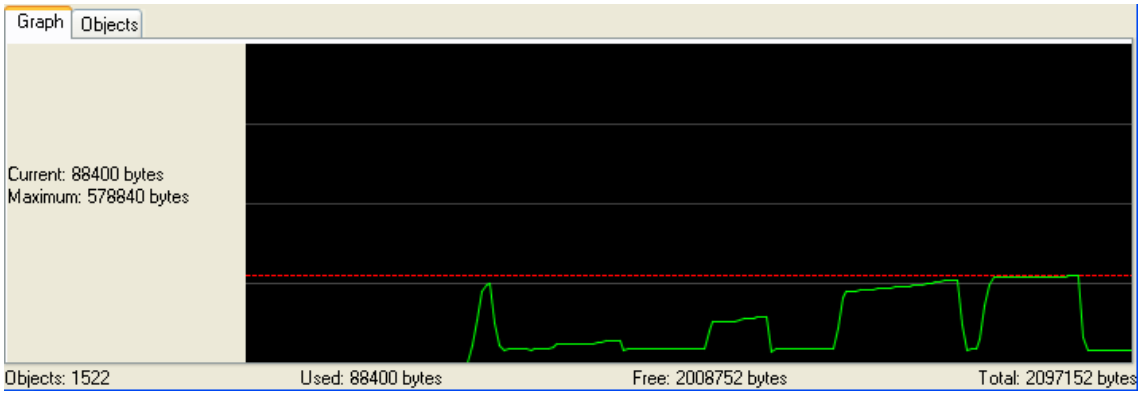


Figura 9.11 Curva de memoria ocupada por WebViewer 3.2

La versión WebViewer 4.0 tiene un consumo muy similar a WebViewer 3.2 como se ve en la siguiente gráfica donde se muestra el acceso a las mismas páginas que la anterior versión, con la excepción de que la primera página fue el buscador Google, pero en su versión personalizada.

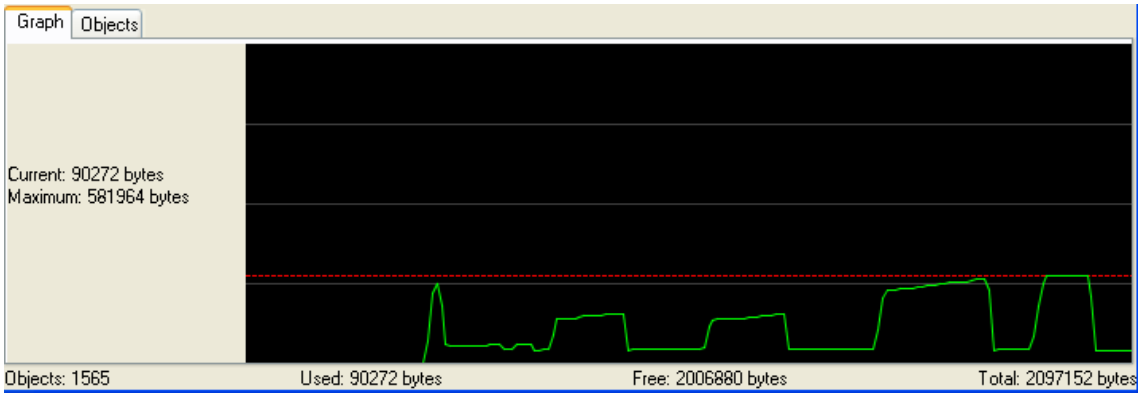


Figura 9.12 Curva de memoria ocupada por WebViewer 4.0 sin javascripts

WebViewer 4.0 puede soportar javascripts. Si se elige habilitar su uso aumenta considerablemente el tiempo de acceso a las páginas, incluso de aquellas que no utilizan javascript. En la siguiente gráfica puede verse el consumo de las mismas páginas del ejemplo anterior. En cuanto al consumo de memoria sí se mantiene.

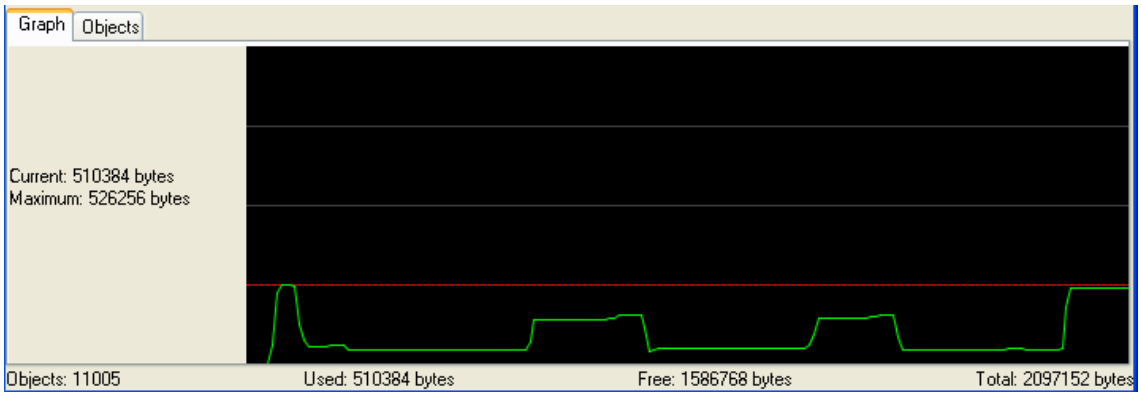


Figura 9.13 Curva de memoria ocupada por WebViewer 4.0 con javascripts

Para estudiar el consumo de memoria de los javascript se ha utilizado una página que los emplea, como es http://www.tejedoresdelweb.com/307/article-1061.html. La gráfica obtenida se explica al detalle a continuación:

- En primer lugar hay un pico de consumo por la carga del microbrowser.
- La primera curva ascendente se debe a la utilización del teclado para introducir la dirección URL.
- El primer valle es el periodo que tarda el microbrowser en reproducir la web con la opción de ver javascripts habilitada.
- La cima se produce durante la navegación web. Tiene una curva ascendente en su parte final debido al uso del teclado para cambiar la configuración del microbrowser para que dejen de aceptarse los javascripts.
- El siguiente valle es mucho más reducido. Esto muestra que la velocidad de acceso es mayor al no aceptar javascript.
- La cima final representa el consumo sin ejecutar los javascript, que como puede verse no ha variado.

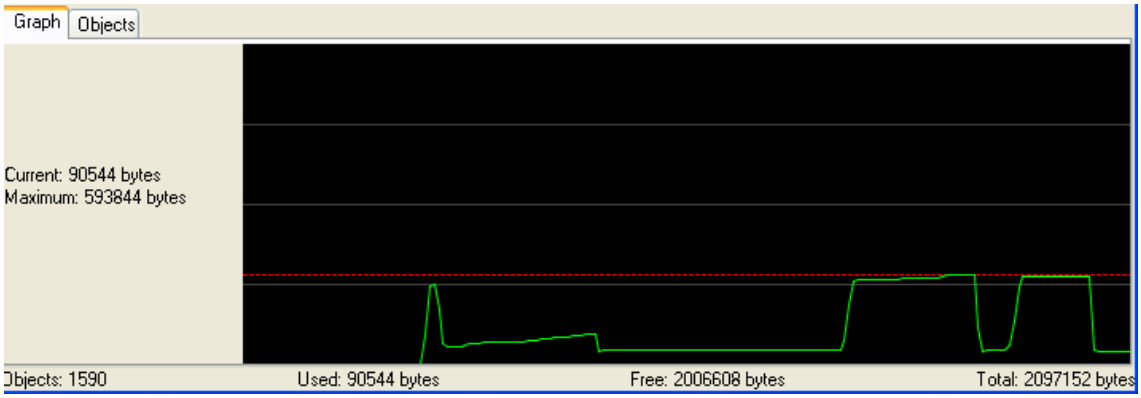


Figura 9.14 Curva de memoria ocupada por WebViewer 4.0 sin y con javascripts

WebViewer 4.0 es capaz de ver el contenido de los archivos con extensión .ZIP, pero no descomprimirlos. El archivo http://apache.rediris.es/tomcat/tomcat5/v5.5.20/bin/apache-tomcat-5.5.20.zip supone abrir 209 Kb debido a los nombres de los archivos comprimidos en esa carpeta. Esto supuso un consumo de 2.000.000 bytes de memoria. El archivo .ZIP http://apache.rediris.es/httpd/httpd-2.2.3-win32-src.zip supone abrir 450 KB. Este caso saturó la memoria disponible del emulador con un consumo de 2.020.000 bytes.

La siguiente gráfica muestra la saturación por exceso consumo de memoria del dispositivo hasta tal punto del cese de su funcionamiento.

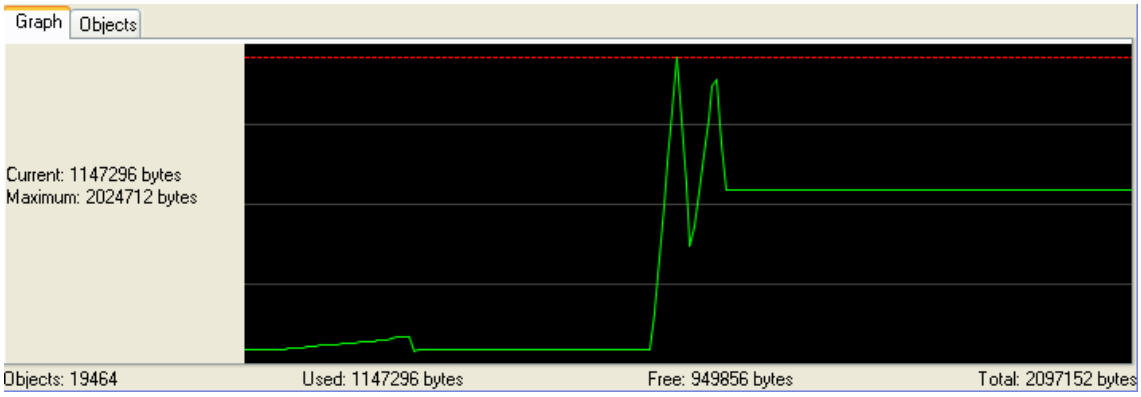


Figura 9.15 Curva de memoria de una saturación.

9.6.4 Opera Mini Basic y Advanced

Las dos versiones de Opera Mini usan un servidor externo que emplea una serie de algoritmos para reducir el tráfico de datos.

Para las pruebas de consumo de Opera Mini Basic se ha accedido a tres de las páginas que se emplearon en el caso de WebViewer:

- www.google.es
- http://portal.us.es
- www.esi.us.es.

En la siguiente gráfica puede comprobarse que el consumo de Opera Mini Basic es muy reducido gracias a su menor tráfico de datos.

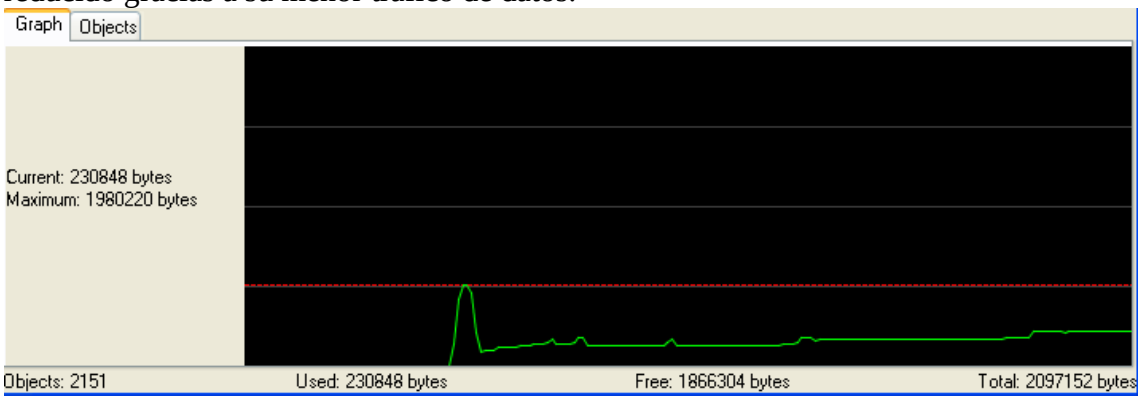


Figura 9.16 Curva de memoria ocupada por Opera Mini Basic

Para las pruebas de consumo de Opera Mini Advanced se han visualizado las páginas:

- www.google.es
- http://portal.us.es
- www.esi.us.es
- www.google.es nuevamente.

Este microbrowser presenta un comportamiento especial, no disminuye el consumo de memoria en los periodos sin navegación, el segundo acceso a la misma página consume más que la primera vez, por tanto el consumo no es lineal al tamaño de carga y tiene el mayor consumo de todos a pesar de tener el menor flujo de datos.

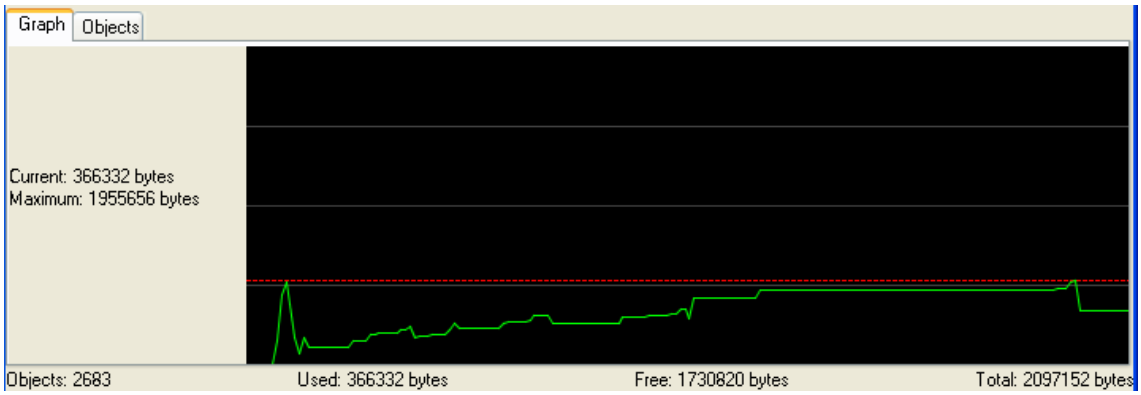


Figura 9.17 Curva de memoria ocupada por Opera Mini Advanced

Esta utilización de recursos tan diferente puede deberse a que está diseñado para dispositivos del perfil MIDP 2.0, que poseen una capacidad de memoria mucho mayor, de modo que no se preocupa de minimizar el consumo.

En la siguiente gráfica se muestra que la curva de consumo de Opera Mini Advanced no es lineal. Otra peculiaridad es que el consumo durante la navegación web supera al producido en la carga de la aplicación.

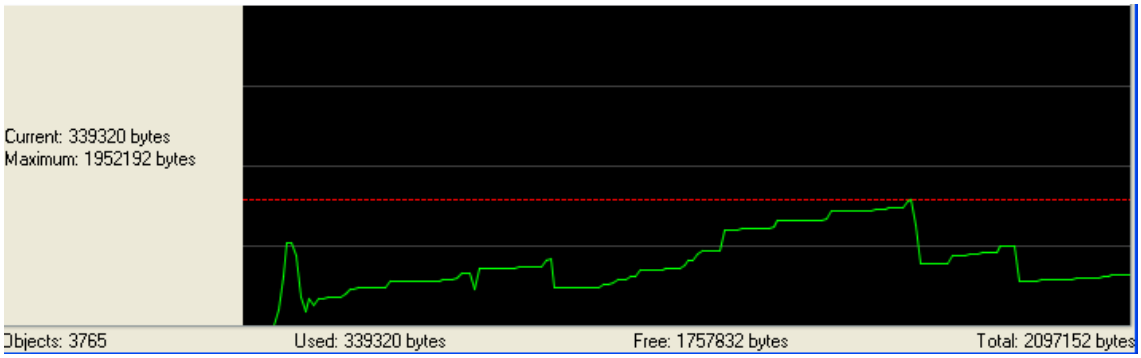


Figura 9.18 Curva de memoria ocupada por Opera Mini Advanced de mayores consumos.

9.7 Velocidad de proceso

En este apartado se estudia el tiempo que tarda un microbrowser en abrir una página web. Es un dato que depende de varios factores:

- Habilidad del microbrowser.
- Velocidad del procesador del móvil.
- Estado de la red.
- Estado del servidor que sirve las páginas.
- Etc.

Al realizarse las pruebas en el mismo entorno y momento de ejecución parecidos, se igualan todos los factores excepto el correspondiente a la habilidad del microbrowser. De ahí que pueda realizarse esa comparación.

La siguiente tabla muestra el tiempo que tardó cada microbrowser en abrir algunas páginas y también cuánto tardaban en desechar una conexión. Los valores son aproximados por las razones comentadas anteriormente.

Como jBrowser sólo usa páginas WML no puede ver algunas de las que se han usado de ejemplo. WebViewer 3.2 tarda lo mismo que WebViewer 4.0 cuando tiene los javascript deshabilitados. Lo que se ha comprobado es la diferencia entre tenerlos habilitados (“WebViewer con”) o no (“WebViewer con”).

	www.google.es	www.us.es	www.esi.us.es	Desechar conexión
jBrowser	2s	No disponible	No disponible	22s
WebViewer sin	6s	16s	20s	(Ver nota)
WebViewer con	25s	40s	50s	(Ver nota)
Opera Mini	2s	4s	4s	12s
Opera Adva	2s	4s	4s	12s

Tabla 9.4 Tiempos de carga.

Nota: WebViewer tarda unos cuatro minutos en desechar una conexión. Es un tiempo tan elevado que se ha sacado de la gráfica para no distorsionar el resto de resultados.

En la siguiente gráfica se representan los datos de la tabla anterior:

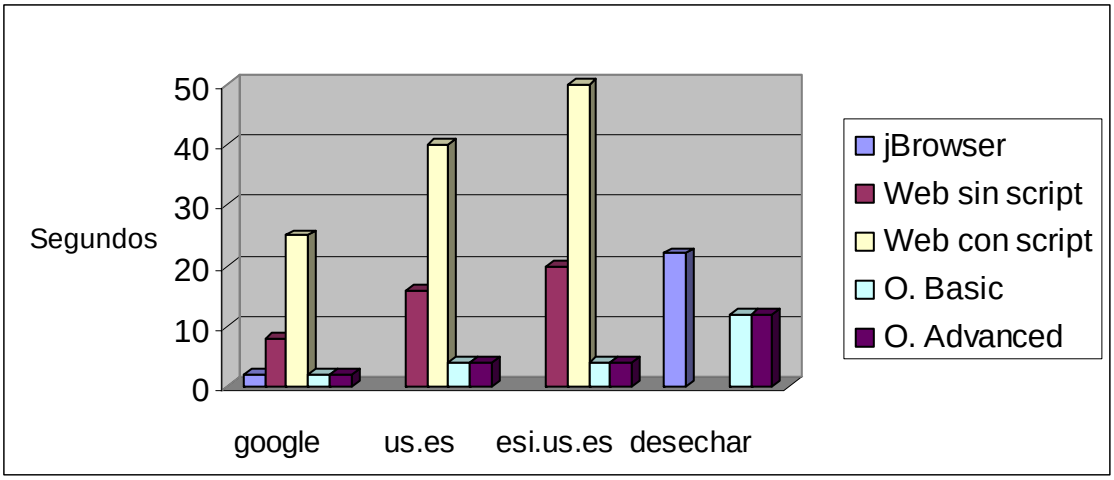


Figura 9.19 Tiempo de carga de página

9.8 Modo de navegación

Los teléfonos móviles no disponen de periféricos, como el ratón, que les facilite la navegación por Internet. Sólo disponen de un teclado alfanumérico reducido. En este apartado se explica la manera de desplazarse por las páginas que permite cada microbrowser.

- jBrowser UP, DOWN mueve de línea en línea.
- WebViewer UP, DOWN mueven de página en página. 1 y 9 mueven de línea en línea. Para moverse a la primera página se puede usar la tecla 3, y para la última la tecla 7.
- Opera Mini UP, DOWN mueven de línea en línea. Posee teclas de atajos para ejecutar opciones de forma inmediata.

JBrowser y Opera Mini muestran una barra en la parte derecha de la pantalla que indica la posición relativa dentro de la página web. Sirve a modo ilustrativo pero no para desplazarnos con ella. En Opera Mini el botón es de tamaño variable según la extensión de la página. En la siguiente figura se muestran estas barras de posicición de jBrowser y Opera Mini.

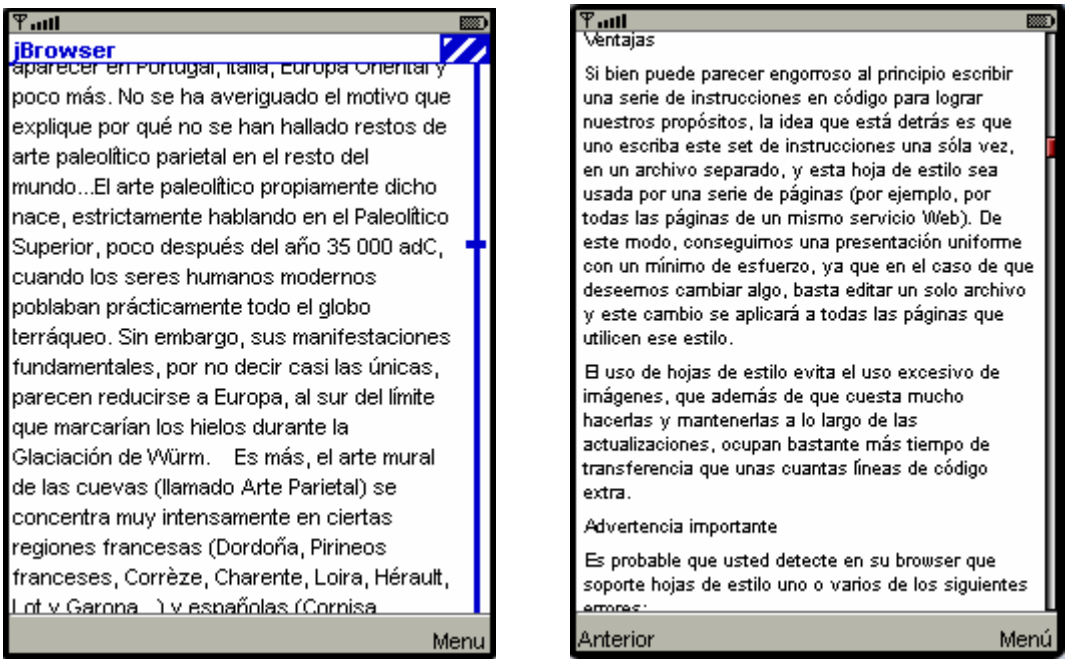


Figura 9.20 Barra de posición de jBrowser y de Opera-Mini

9.9 Web de fabricante

Todos los fabricantes de microbrowser tienen una web corporativa. En este apartado se realiza una valoración de la misma. Se ha puntuado del 0 al 10 la cantidad de información sobre el producto y el diseño. También se indica si permite la descarga del microbrowser y la fecha de su creación. En la tabla siguiente se muestran estos datos.

	Web	**Información**	**Diseño**	**Descargas**	**Año**
jBrowser	www.jataayusoft.com	8	8	No	2002
WebViewer	www.reqwireless.com.	5	3	Si	2005
Opera Mini	www.opera.com/products/mobile/operamini	7	9	Si	2006

Tabla 9.5 Web de los fabricantes de los microbrowsers.

Jataayusoft no permite en su página la descarga directa de sus productos, es necesario contactar con ellos para que nos los proporcionen. Para obtener su microbrowser se usó otra web, http://java.mob385.com/en/browseri.shtml, dedicada a la distribución de MIDlets de toda clase. Algunas páginas especializadas en ofrecer programas, como http://www.softonic.com, también ofrecen microbrowsers.

En la siguiente gráfica se muestran los datos de la tabla:

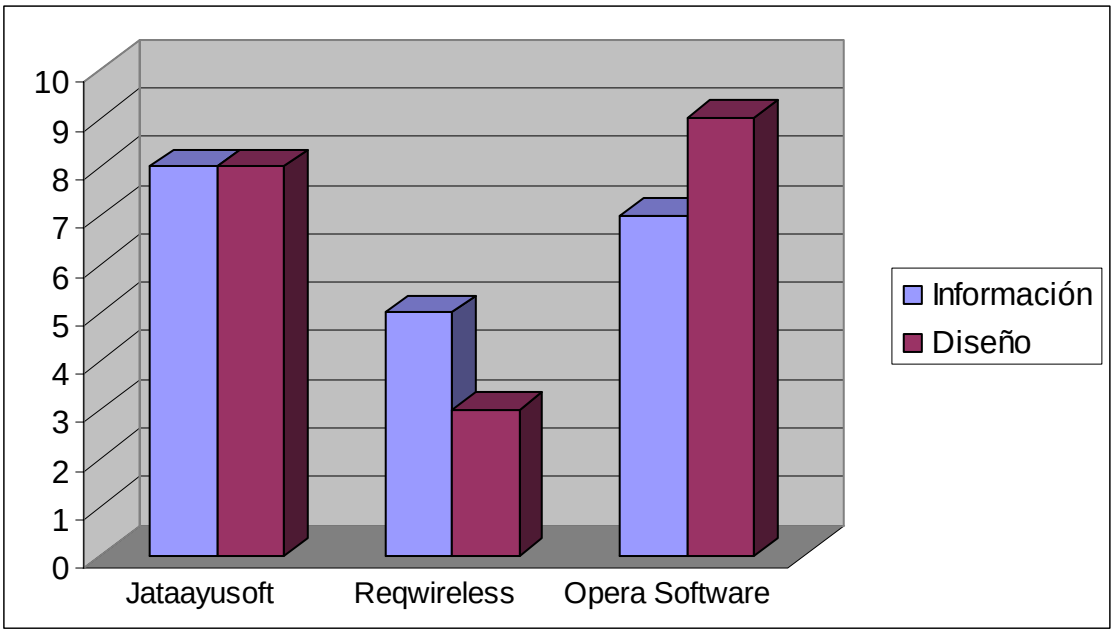


Figura 9.21 Valoración subjetiva de las webs de los fabricantes.

9.10 Compatibilidad con los dispositivos

Un teléfono que tenga soporte para J2ME podrá ejecutar las aplicaciones implementadas en este lenguaje. Además, algunos fabricantes proporcionan una lista de modelos de teléfonos móviles en los que han comprobado que pueden instalarse sus microbrowsers. Las listas de compatibilidad son las siguientes.

J.Browser 1.0.4:

- No ha sido especificada por el fabricante.

WebViewer 3.2 y 4.0:

- BenQ P30.
- Motorola A1000, A920, A925.
- Nokia 3230, 3600, 3620, 3620-AT&T, 3620-Cingular, 3620-TMobile, 3650, 3660, 3660-AT&T, 3660-Cingular, 3660-TMobile, 6260, 6600, 6620, 6630, 6670, 6680, 6681, 7610, 7650, 7710, 9200 Series Communicator, 9210 Communicator, 9290 Communicator, 9300, 9500, N-Gage, N-Gage QD, N-Gage-AT&T, N-Gage-Cingular, N-Gage-TMobile, Series 60.
- SendoX.
- Siemens SX1.
- Sony Ericsson P800, P900, P910

Opera Mini da una lista separada en marcas y modelos. La lista de marcas es:

- Alcatel
- Audiovox
- Benq-Siemens
- BlackBerry
- Dopod
- Generic
- LG
- Motorola
- Nokia
- O2
- Orange
- Palm
- Panasonic
- Pantech
- Qtek
- Sagem
- Samsung
- Sanyo
- Sharp
- Siemens
- Sony Ericsson
- i-mate
- Modelos sueltos de otras marcas.

9.11 Imágenes

Las páginas web además de texto suelen completar su información con imágenes. Existen varios formatos gráficos según el algoritmo de codificación que usen, con lo que conseguiran más o menos resolución. Dentro de cada uno pueden usarse más o menos bits para representar cada píxel, que es la unidad mínima de imagen. Esta característica se conoce como “profundidad de bit” y permite variar el número de colores disponible para la imagen.

Otro aspecto a tener en cuenta es el tamaño en bytes que ocupa la imagen, si este valor es muy grande puede provocar que el microbrowser no sea capaz de reproducirla en el dispositivo. Por esta razón algunas versiones permiten deshabilitar la carga de imágenes, por ejemplo WebViewer y Opera Mini.

9.11.1 Tipos de imágenes que existen en HTML

De acuerdo con las especificaciones de HTML 4.0 [40], los únicos formatos gráficos que se pueden utilizar son GIF, JPEG y PNG.

El formato gráfico GIF [41] debe sus siglas a los vocablos ingleses *Graphics Interchange Format*, lo que traducido al español es “formato de intercambio de gráficos”.

Las principales características de este formato gráfico son:

- Formato de mapa de bits.
- Color indexado en una paleta máxima de 256 colores.
- Compresión LZW, sin pérdida, que alcanza tasas 1:4 (hasta un 25% del tamaño original).

JPEG es un estándar de compresión de imágenes fijas (no existen los JPEG animados) desarrollado por el *Joint Photographic Experts Group* (JPEG) [42] de cuyas siglas viene su nombre, y cuya traducción directa al castellano es “grupo de expertos en fotografía”.

Las características de este formato gráfico son bastante diferentes a las del formato GIF, de ahí que estos 2 formatos sean los utilizados por el lenguaje HTML, ya que uno intenta complementar al otro y viceversa.

Las características del formato gráfico JPEG son:

- Formato gráfico de mapa bits.
- Soporte de color verdadero, también conocido por su homónimo inglés *true color* (24 bits).
- Algoritmo de compresión (con pérdida) que soporta altas tasas de empaquetado (1:20 y más).

PNG debe sus siglas a los vocablos ingleses *Portable Network Graphics*, cuya traducción directa al castellano es “gráficos de red portátiles”.

Las características principales del formato PNG son [43]:

- Formato de mapa de bits.
- Alta tasa de compresión sin pérdidas.
- Soporte de colores: color indexado, color verdadero de 48 bits, escala de grises de 16 bits.
- Sistema de entrelazado Adam7.
- Transparencia en color indexado.
- Transparencia de canal alfa en imágenes en color verdadero y escala de grises.
- Corrección gamma.
- Cromaticidad.

9.11.2 Otros formatos.

El formato BMP (Windows BitMaP) es probablemente el formato de fichero para imágenes en color más simple que existe. Aunque teóricamente permite compresión, en la práctica nunca se usa, y consiste simplemente en una cabecera y a continuación los valores de cada píxel, comenzando por la línea de más abajo y terminando por la superior, píxel a píxel de izquierda a derecha. Su única ventaja es su sencillez. Su gran desventaja es el enorme tamaño de los ficheros.

El formato *Wireless BitMaP* (WBMP) es una especificación para un formato simple de imagen de 1 bit (monocromo) que se utiliza en el lenguaje WML. Una imagen en este formato puede verse en la figura siguiente, en la que se compara BMP y WBMP.

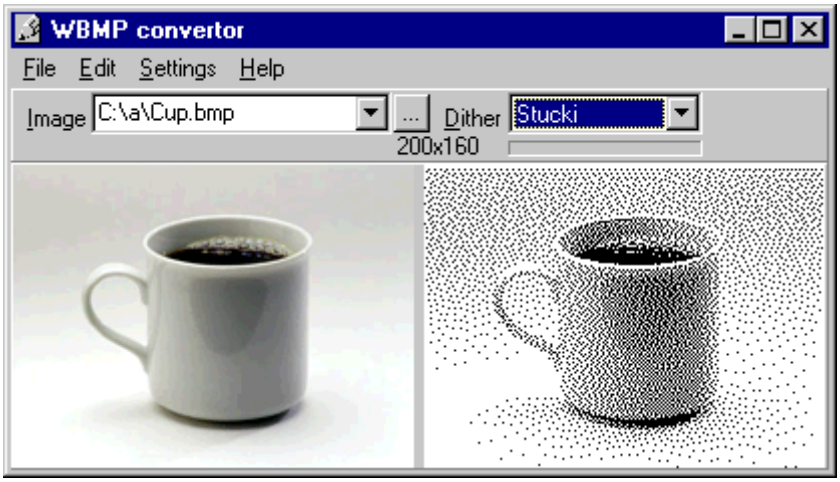


Figura 9.22 Imagen BMP y su equivalente WBMP

9.11.3 Imágenes de ejemplo

Se han realizado pruebas de visualización en los microbrowsers de los formatos WBMP, BMP, GIF, JPEG y PNG. Para ello se han usado las siguientes imágenes de ejemplo:

- **WBMP.** Imagen WBMP de 22 bytes extraída del CD que acompaña al libro Compendium HTML de Günter Born, 2001. Incluido en la bibliografía de esta memoria.

Figura 9.23 Imagen WBMP de ejemplo.

- **BMP.** Imagen BMP de 45.382 bytes con profundidad de 8 bits extraída de la página http://sprott.physics.wisc.edu/images/cat.bmp.

Figura 9.24 Imagen BMP de ejemplo.

- **GIF**. GIF normal y GIF transparente de 4.460 bytes con profundidad de 8 bits y GIF animado de 52.596 bytes con profundidad de 32 bits extraídos de la página http://www.conganat.org/iicongreso/comunic/008/gift.htm.

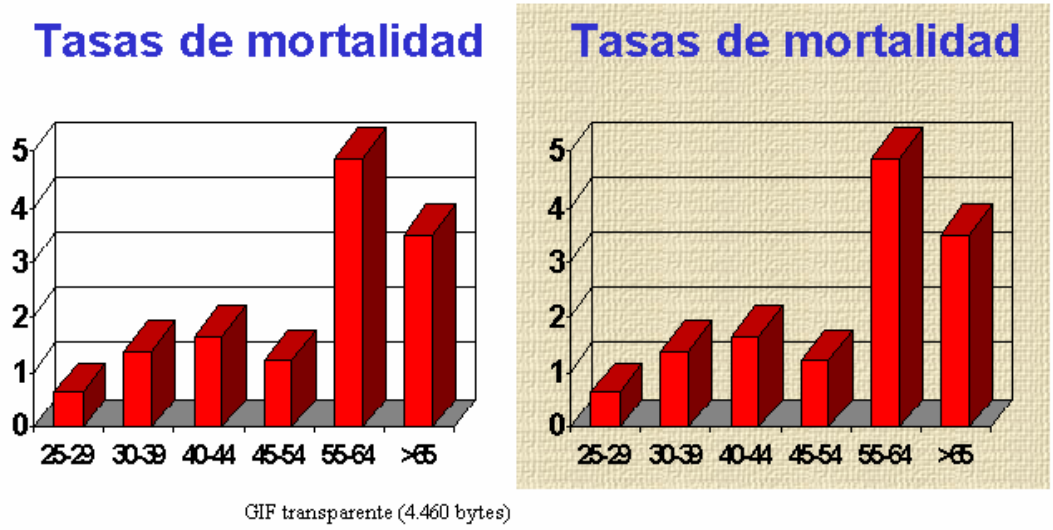


Figura 9.25 GIF normal y GIF transparente de ejemplo.

- **JPEG**. Imagen JPEG de 15.128 bytes con profundidad de 24 bits extraída de la página http://dpaso.blogspot.com/2005_05_01_dpaso_archive.html

Figura 9.26 Imagen JPEG de ejemplo.

- **PNG**. Imagen PNG de 4.287 bytes y profundidad de 24 bits e imagen PNG de 15.640 bytes y profundidad de 8 bits extraídos de la página http://www.conganat.org/iicongreso/comunic/008/png.htm

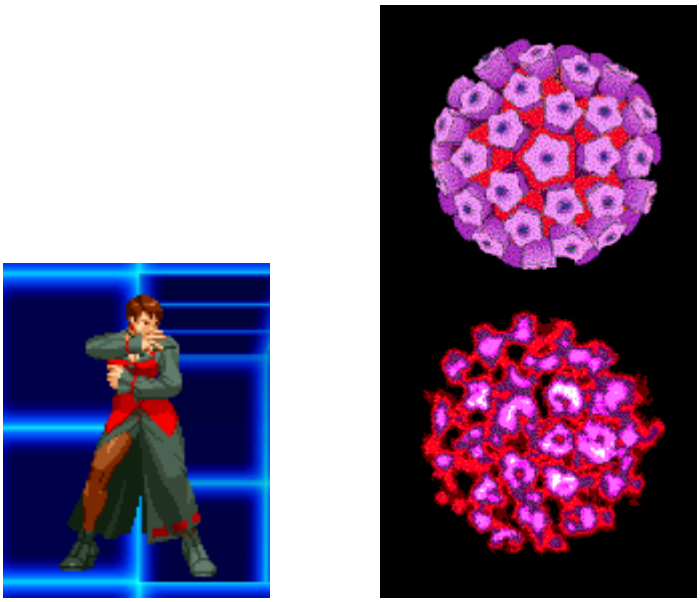


Figura 9.27 Imágenes PNG de ejemplo.

9.11.4 Prueba de las imágenes en los microbrowsers

9.11.4.1 Introducción

La metodología seguida para visualizar imágenes en los microbrowsers ha sido distinta para el lenguaje WML y el HTML.

- Para el lenguaje WML se han creado las páginas con los todos los formatos y se han alojado en el servidor local Tomcat. Este lenguaje no posee muchos ejemplos útiles en Internet y de esta manera pudieron realizarse todas las pruebas necesarias.
- Para HTML se han usado páginas de servidores externos porque se disponían de ejemplos de todos los formatos de imágenes estudiados.

9.11.4.2 WBMP

El siguiente código se usó para mostrar una imagen WBMP en 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="card1" title="WAP">
   <p>Una imagen <img src="uparrow.wbmp" alt="flecha"
      height="20" width="20" /></p>
  </card>
</wml>
```

Figura 9.28 Código WML con imagen WBMP.

La imagen uparrow.wbmp ocupa sólo 22 bytes. El resultado obtenido por jBrowser al leer el código anterior se muestra en la siguiente figura.

Figura 9.29 Imagen WBMP de 22 bytes en jBrowser

9.11.4.3 BMP

Este formato es soportado por WebViewer y Opera Mini. La imagen cat.bmp ocupa sólo 34 Kb en WebViewer por los 37 Kb de Opera Mini.

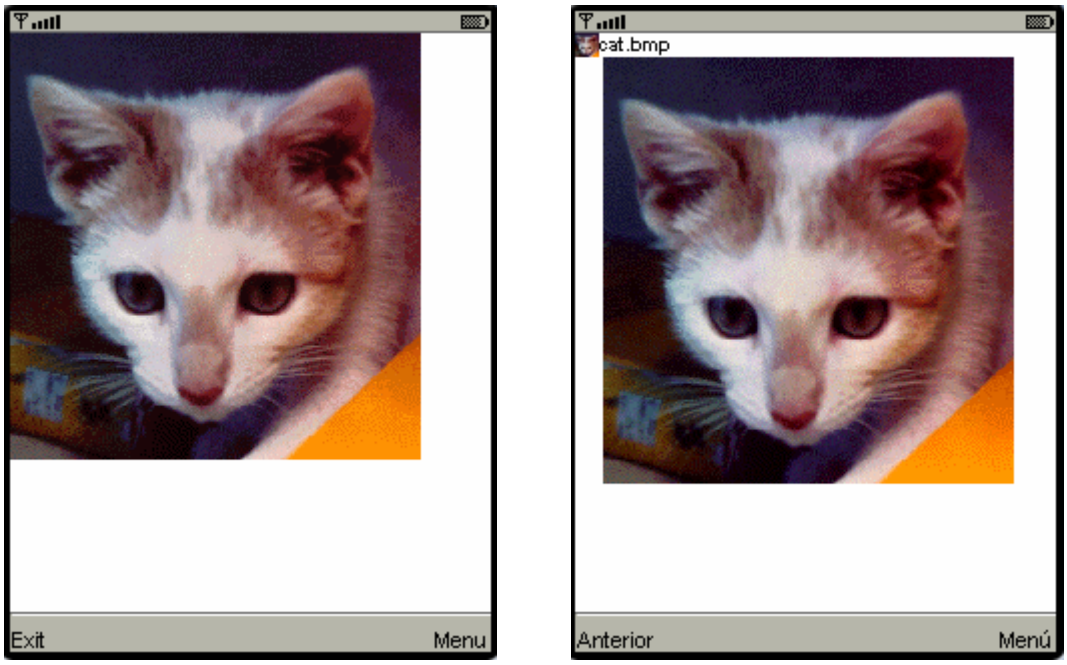


Figura 9.30 Imagen BMP en WebViewer y en Opera Mini Basic.

La siguiente figura muestra la imagen anterior en Opera Mini Advanced. Consigue reducir su tamaño a la mitad (17Kb) y sin pérdida aparente de calidad.

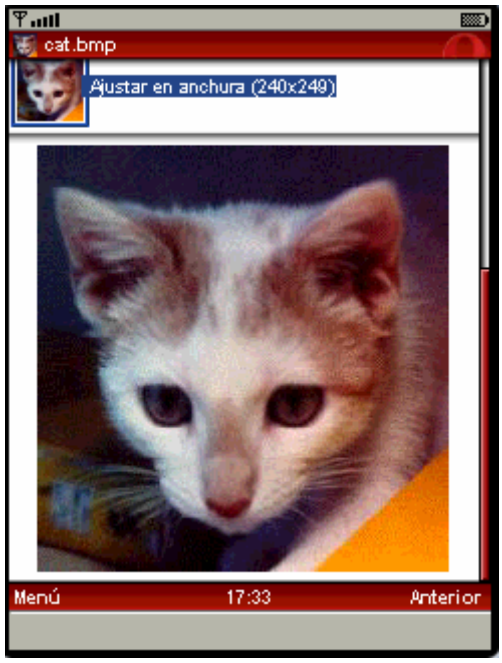


Figura 9.31 Imagen BMP en Opera Mini Advanced.

9.11.4.4 GIF

Este formato tiene varias variantes, GIF normal, transparente y animado. Webviewer muestra correctamente los GIF normales, sin el fondo los GIF transparentes y estáticos los GIF animados. Todo esto puede verse en la siguiente figura.

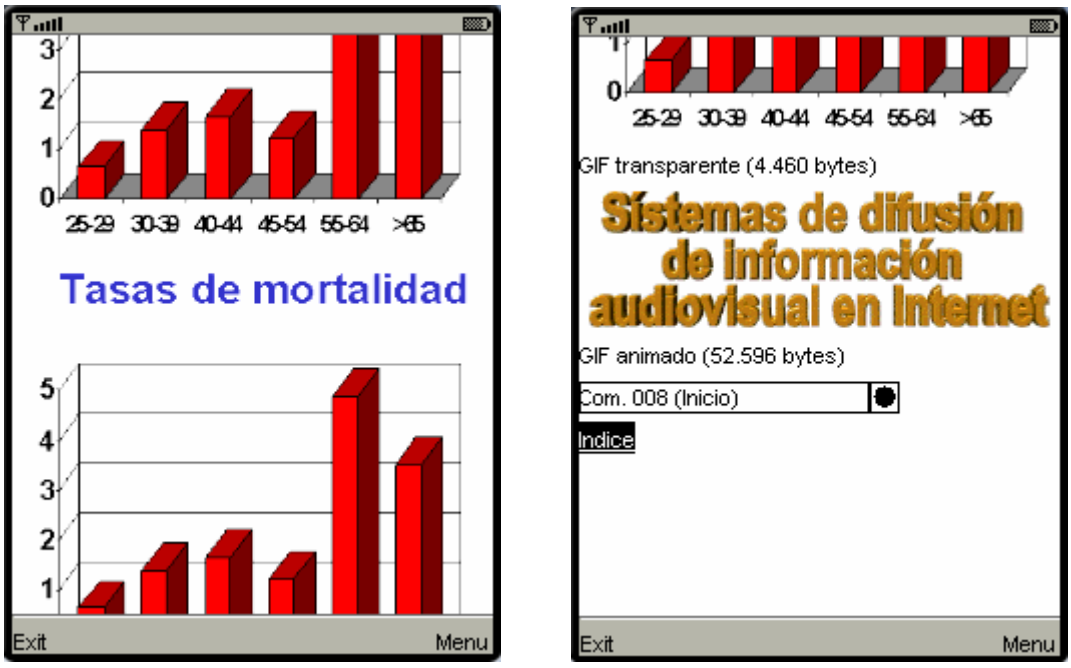


Figura 9.32 GIF normal, transparente y animado enWebViewer

En la siguiente figura se muestran los tres tipos de GIF tanto en Opera Mini Basic como en la versión Advanced. Como en el anterior microbrowser, los GIF transparentes se muestran sin su fondo y los animados se ven estáticos.

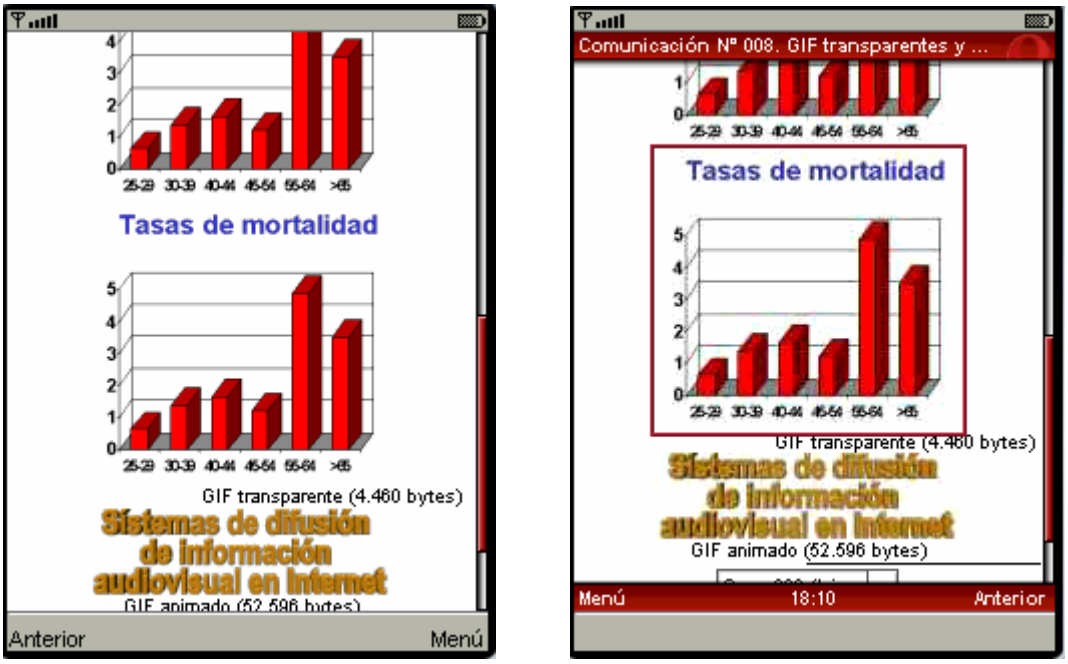


Figura 9.33 GIF normal, transparente y animado en Opera Mini Basic y Advanced.

Para comprobar si jBrowser es capaz de visualizar imágenes GIF se ha elegido un archivo GIF más pequeño debido a la limitación de 6KB en el tamaño de las páginas de este microbrowser. El GIF elegido se llama UP1.gif y ocupa sólo 71 bytes, se muestra en la siguiente figura:

Figura 9.34 Imagen GIF para jBrowser.

El código empleado para la prueba fue el siguiente:

```
<?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="card1" title="WAP">
   <p>Una imagen <img src="UP1.gif" alt="flecha"
      height="20" width="20" /></p>
  </card>
</wml>
```

Figura 9.35 Código WML con imagen GIF.

jBrowser no puede mostrar imágenes GIF y por eso muestra el texto alternativo indicado en el atributo alt de la etiqueta `<img>`, que sirve para estos casos, como se ve en la siguiente figura.

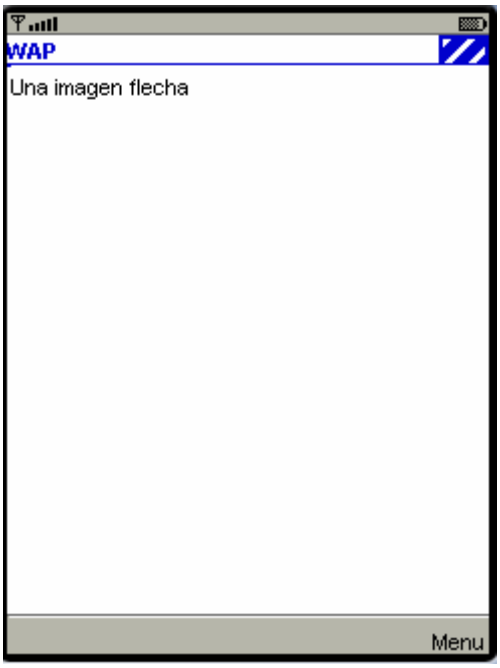


Figura 9.36 Imagen GIF no mostrada en jBrowser

9.11.4.5 JPEG

Este formato se caracteriza por permitir pérdida de calidad en favor de un tamaño más reducido. Es soportado por WebViewer y Opera Mini como puede verse en la siguiente figura. La imagen arcoiris2.jpg ocupa 24Kb en WebViewer por los 23Kb de Opera Mini, en ambas situaciones se pierde resolución respecto a la imagen original.

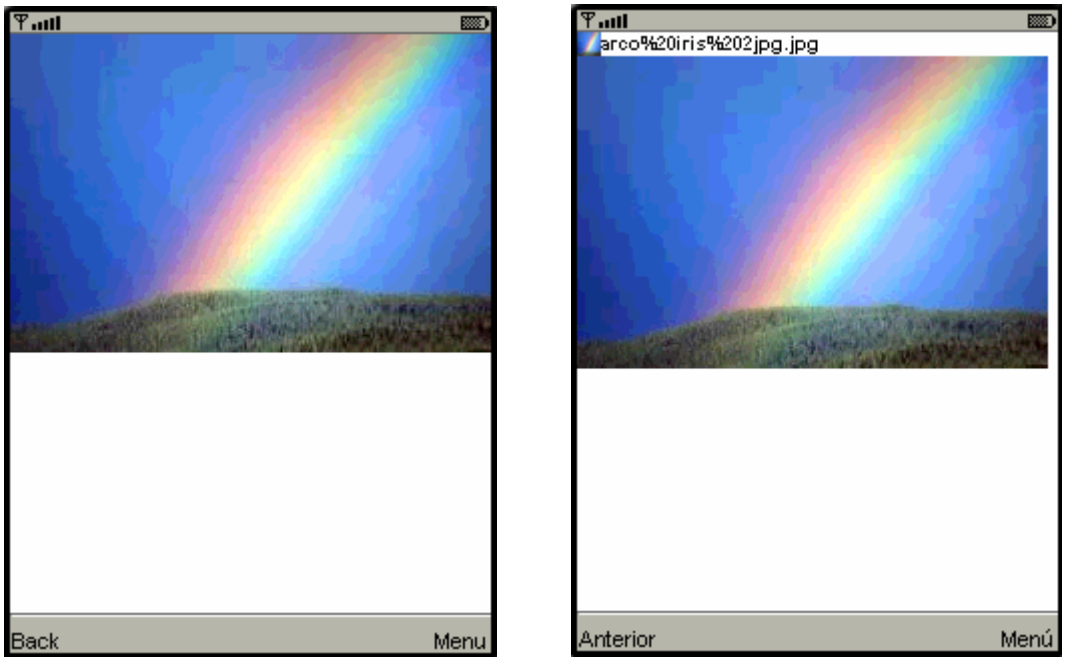


Figura 9.37 Imagen JPEG en WebViewer y en Opera Mini Basic.

La siguiente figura muestra esa misma imagen en Opera Mini Advanced, cuyo tamaño consigue reducir a algo menos de la mitad (11Kb), pero también con pérdidas de calidad.

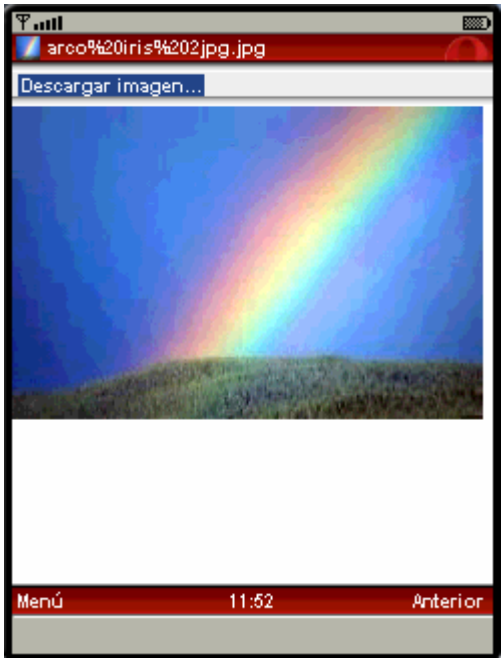


Figura 9.38 Imagen JPEG en Opera Mini Advanced.

9.11.4.6 PNG

Este formato es soportado por todos los microbrowsers. JBrowser sólo es capaz de mostrar hasta 6KB de imagen y por esta razón se usó un ejemplo diferente al resto de los microbrowsers. Se empleó luchador.png que ocupaba únicamente 4,18 KB. En la figura siguiente se ven imágenes PNG en jBrowser y WebViewer.

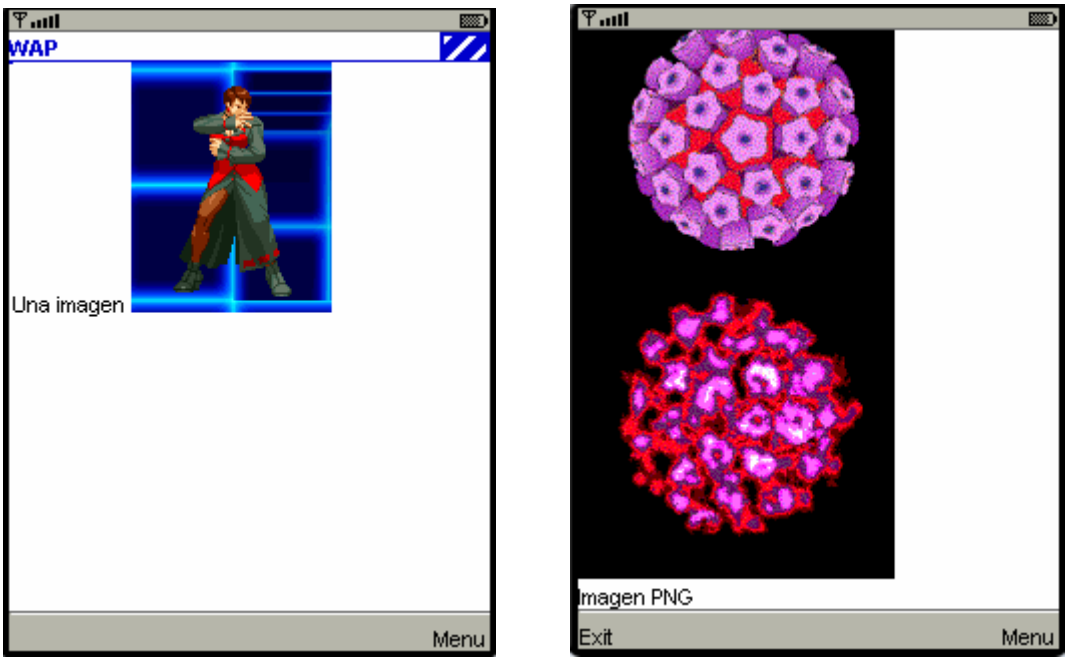


Figura 9.39 Imagen PNG para jBrowser e imagen PNG en WebViewer

El código usado para mostrar la imagen PNG en jBrowser fue el siguiente:

```
<?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="card1" title="WAP">
   <p>Una imagen <img src="luchador.png" alt="flecha"
      height="20" width="20" /></p>
  </card>
</wml>
```

Figura 9.40 Código WML con imagen PNG.

En la siguiente figura se muestra la imagen PNG en Opera Mini tanto Basic como Advanced. Estos microbrowser tienen la posibilidad de usar algoritmos de compresión de imágenes en un servidor externo. La versión Advanced usa algoritmos de mayor compresión a costa de una mayor pérdida de calidad.

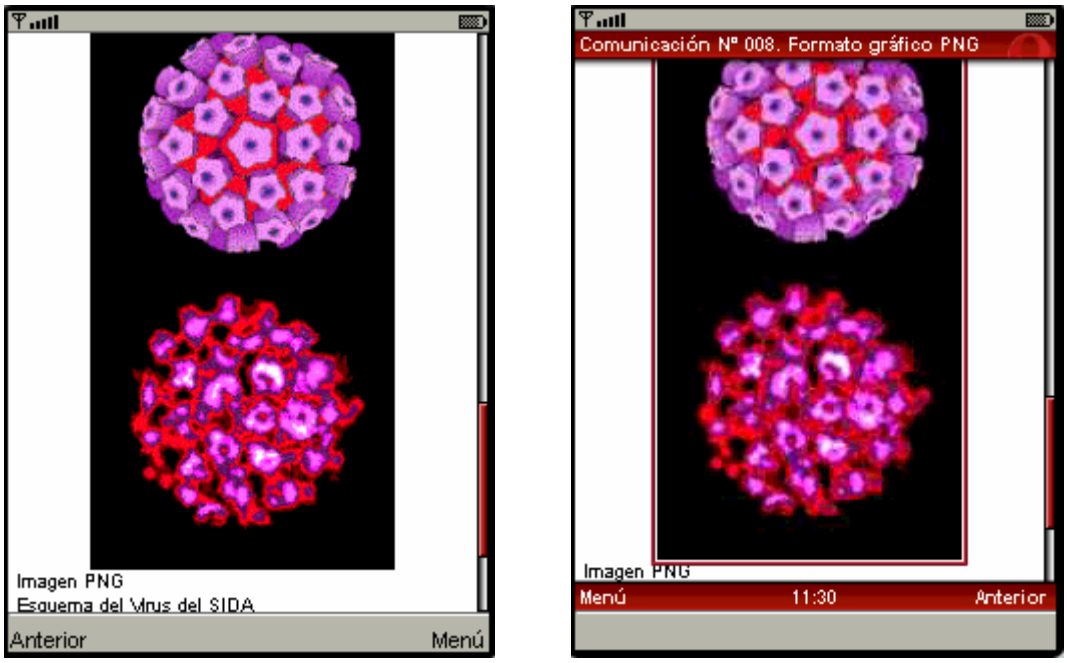


Figura 9.41 Imagen PNG en Opera Mini Basic y Opera Mini Advanced.

Para evitar esta pérdida de calidad el usuario puede configurar Opera Mini para que no ejecute el algoritmo de compresión marcando la casilla “Calidad de imagen máxima” dentro del menú “Ajustes” del microbrowser, como se explicó en el capítulo 7 de esta memoria “Microbrowsers”.

En la siguiente figura se muestra la imagen PNG anterior a calidad máxima en Opera Mini Basic y Advanced.

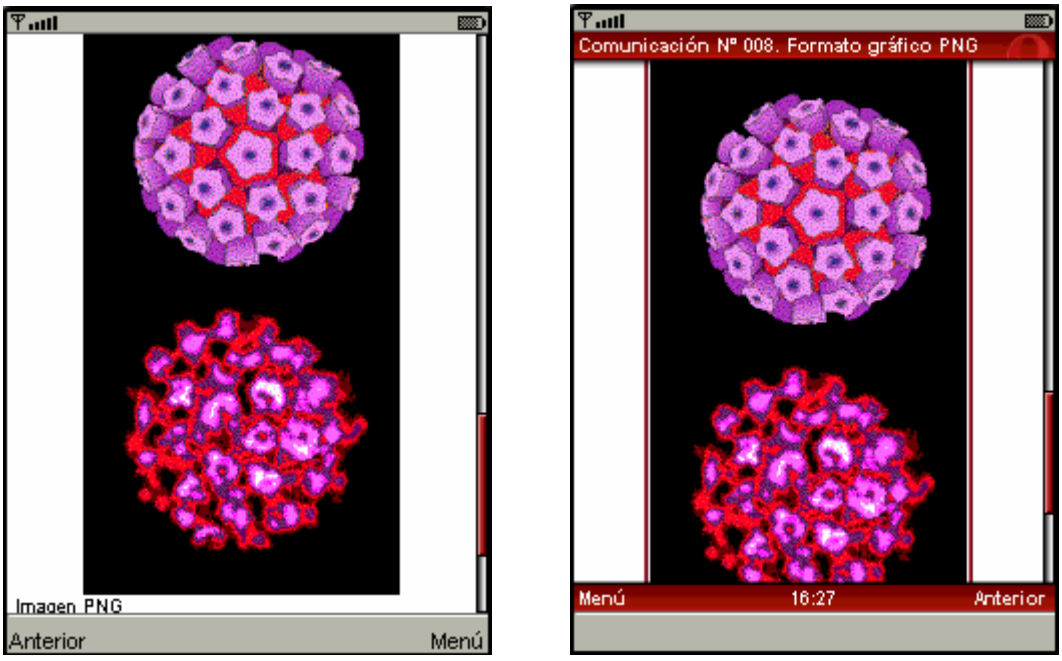


Figura 9.42 Imagen PNG en Opera Mini Basic y Advanced sin comprimir

9.11.5 Conclusiones

La siguiente tabla resume el tipo de formato soportado por cada microbrowser. En ella se indica el nivel de calidad con el que se ven las imágenes.

	WBMP	**BMP**	**PNG**	**GIF**	**JPEG**
jBrowser	Alta	No soportado	Alta	Alta	No
WebViewer	No soportado	Alta	Alta	Alta	Baja
Opera Mini Basic	Alta	Alta	Media/Alta	Alta	Baja
Opera Mini Advanced	Alta	Alta	Baja/Alta	Baja/Alta	Baja

Tabla 9.6 Formatos de imagen soportados por los microbrowsers

Los microbrowsers Opera Mini Basic y Advanced permiten al usuario cambiar la resolución con la que quiere ver la imagen, por eso los Formatos PNG y GIF indican dos tipos de resoluciones.

WebViewer y Opera Mini permiten desactivar totalmente la carga de imágenes en las páginas, mostrando sólo el texto. De esta manera se puede reducir mucho la tasa de transferencia y el tiempo de carga de las mismas.

9.12 Uso de scripts

Sobre las páginas especificadas por lenguajes de marcas podemos disponer de aplicaciones web. Son fragmentos de código que se ejecutan cuando se realizan ciertas peticiones o respuestas HTTP. Hay que distinguir entre:

- Aplicaciones en el lado del cliente: el cliente web es el encargado de ejecutarlas en la máquina del usuario. Son las aplicaciones tipo Java (applet) o javascript: el servidor proporciona el código de las aplicaciones al cliente y éste, mediante el navegador, las ejecuta. Es necesario, por tanto, que el cliente disponga de un navegador con capacidad para ejecutar aplicaciones (también llamadas scripts). Normalmente, los navegadores permiten ejecutar aplicaciones escritas en lenguaje javascript y java, aunque pueden añadirse más lenguajes mediante el uso de *plugins*. Las applets son más potentes que javascript, permiten abrir sus propias ventanas, crear botones, crear animaciones y abrir conexiones de red en el servidor local.

- Aplicaciones en el lado del servidor: el servidor web ejecuta la aplicación; ésta, una vez ejecutada, genera cierto código HTML; el servidor toma este código recién creado y lo envía al cliente por medio del protocolo HTTP.

Dado que estamos estudiando el funcionamiento de los microbrowsers, sólo estamos interesados en las aplicaciones del lado del cliente. Ninguno de ellos es capaz de ejecutar applets, pero sí algunos javascripts.

Para el lenguaje inalámbrico WML existen scripts específicos conocidos como WMLScript. Este lenguaje posee un conjunto de librerías de las que podemos echar mano para el desarrollo de las páginas web. Son Lang, Float, String, URL, WMLBrowser, y Dialogs.

Sin embargo jBrowser, el microbrowser especializado en WML estudiado en este proyecto, es incapaz de usar estos WMLScripts. Se comprobó intentando ejecutar el siguiente ejemplo [45]:

Este es el código del archivo helloWorldEg1.wml que invoca al script.

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.3//EN"
"http://www.wapforum.org/DTD/wml13.dtd">

<wml>
  <card id="card1" title="WMLScript Tutorial">
    <p>
      <a href="helloWorldEg1.wmls#helloWorld()">Run WMLScript</a><br/>
      $(message)
    </p>
  </card>
</wml>
```

Figura 9.43 Código WML con script.

Este es el contenido del archivo helloWorldEg1.wmls que contiene el script.

```
extern function helloWorld()
{
  WMLBrowser.setVar("message", "Hello World. Welcome to our WMLScript
tutorial.");
  WMLBrowser.refresh();
}
```

Figura 9.44 Contenido del WMLScript.

Al intentar ejecutar el WMLScript se obtuvo el mensaje de error: "*Data with content type 'text/vnd.wap.wmlscript' can't be displayed*" como puede verse en la siguiente figura.

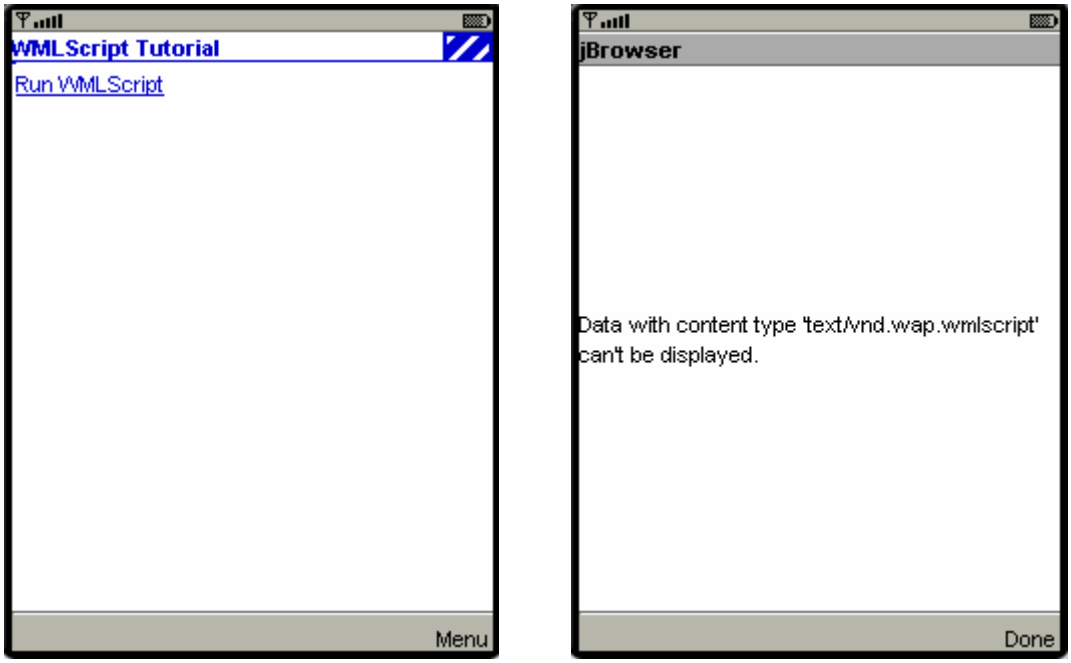


Figura 9.45 Scripts en jBrowser

WebViewer 3.2 es incapaz de mostrar tanto javascript como applets. La versión 4.0 sí permite habilitar el uso de javascript.

Opera Mini no ejecuta applets pero sí javascripts. Se ha comprobado usando los siguientes ejemplos, encontrados en las páginas:

- http://www.conganat.org/iicongreso/comunic/008/java.htm
- http://www.tejedoresdelweb.com/307/article-1061.html

Ejemplo en javascript que muestra el día, el mes y el año actual.

```
<script type="text/javascript">
var days = new Array( "Domingo", "Lunes", "Martes", "Miércoles",
"Jueves", "Viernes", "Sábado" );
var months = new Array( "Enero", "Febrero", "Marzo", "Abril", "Mayo",
"Junio", "Julio", "Agosto", "Septiembre", "Octubre", "Noviembre",
"Diciembre" );
var d = new Date();
var day   = d.getDate();
var month = d.getMonth();
var year  = d.getYear();
if( year < 1900 ) {
        year += 1900;
}
var hour  = d.getHours();
var min   = d.getMinutes();

document.write( '' + days[d.getDay()] + ', ' + day + ' de ' +
months[month] + ' de ' + year);
</script>
```

Figura 9.46 Código de un javascript.

El resultado de ejecutar el anterior javascript en WebViewer 3.2 y en 4.0 es que en el segundo caso funciona y por tanto muestra la fecha, como puede verse en la siguiente figura.

Figura 9.47 Javascript de la fecha en WebViewer 3.2 y 4.0.

Opera Mini Basic y Advanced también ejecutan el javascript que muestra la fecha de forma correcta, como puede verse en la siguiente figura.

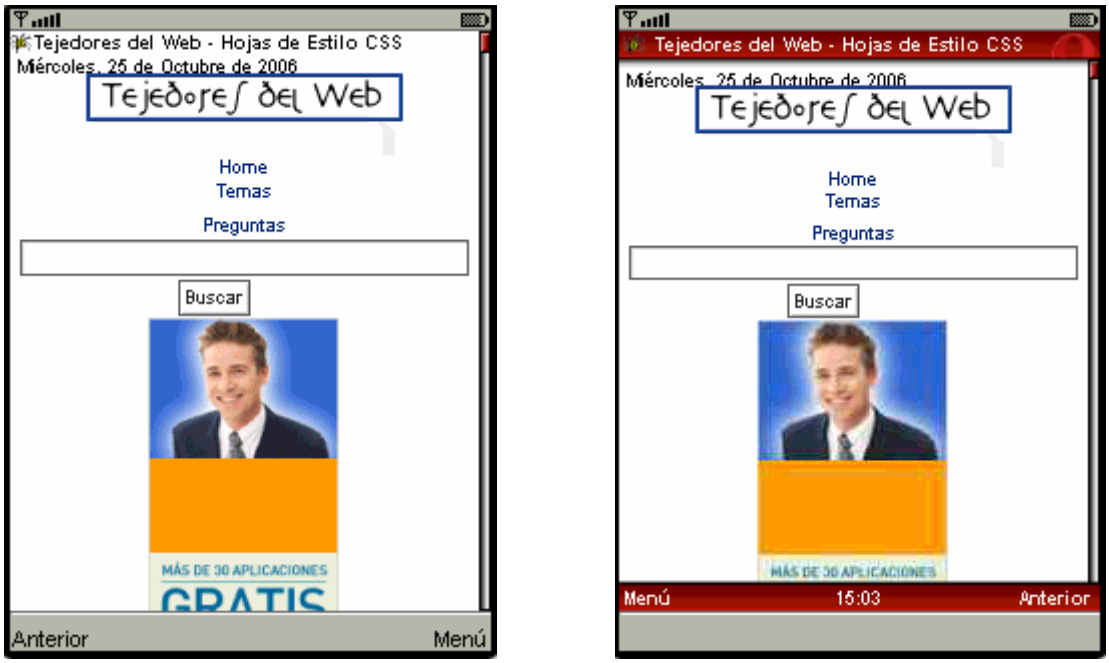


Figura 9.48 Javascript de la fecha en Opera Mini Basic y Advanced.

También se ha probado el código de una applet que muestra efectos en 3-D contenido en un documento HTML. Ningún microbrowser es capaz de mostrarlo.

```
  <applet code="TDMessage.class" align="baseline" width="550"
height="150" id="3D Message">
    <param name="3dsize" value="15">
    <param name="bgcolor" value="240,239,234">
    <param name="bold" value="true">
    <param name="delay" value="2000">
    <param name="delay3d" value="15">
    <param name="desc0" value="Sistemas de Difusión...">
    <param name="desc1" value="De Información Audiovisual...">
    <param name="desc2" value="En Internet">
    <param name="desc3" value="Aplicaciones en...">
    <param name="desc4" value="Anatomía Patológica">
    <param name="desc5" value="Luis Alfaro">
    <param name="desc6" value="María José Roca">
    <param name="font" value="Times New Roman">
    <param name="heading6" value="0">
    <param name="heading7" value="1">
    <param name="heading8" value="2">
    <param name="heading9" value="3">
    <param name="italic" value="false">
    <param name="movedelay" value="0">
    <param name="Notice" value="3D Message Effect, Copyright (c) 1996
OpenCube Technologies - Freeware">
    <param name="numofmessages" value="7">
    <param name="size" value="40">
  </applet>
```

Figura 9.49 Código de una applet.

9.13 Hojas de estilo

Una hoja de estilo es una colección de reglas que afecta a la apariencia de un documento. Estas normas se refieren al modo en que aparecerá un documento en pantalla cuando el usuario utilice un navegador gráfico, controlando por ejemplo el color, el fondo, tipo de fuente, apariencia del borde, márgenes, alineación, espacio entre caracteres, etc.

Se ha empleado la página http://www.tejedoresdelweb.com/307/article-1061.html para analizar las hojas de estilo en los microbrowsers. En su cabecera indica el uso de una hoja de estilo externa basda en CSS:

```
<style type="text/css"> @import "stylesheet-136.css"; </style>
```

También posee estilos internos a la página, como por ejemplo:

```
<body style="margin-top: 0px; margin-bottom: 0px; margin-left: 0px;
margin-right: 0px"
```

El atributo “class” permite aislar una regla de estilo y uno se puede referir a ella en la hoja de estilo como un subselector, dicha definición puede hallarse en otro documento de extensión CSS, que puede no estar accesible, o en el propio documento. El ejemplo usa el documento externo "stylesheet-136.css". El selector para el estilo del título “Hoja de Estilo CSS” es <td class="CompletoT1"> y para el párrafo anexo es <td class="CompletoPP">.

Como puede verse en la siguiente figura, WebViewer no soporta los estilos externos. Opera Mini sólo es capaz de mostrar el atributo que da el color de fondo, no muestra la alineación del texto ni la negrita ni otras muchas propiedades.

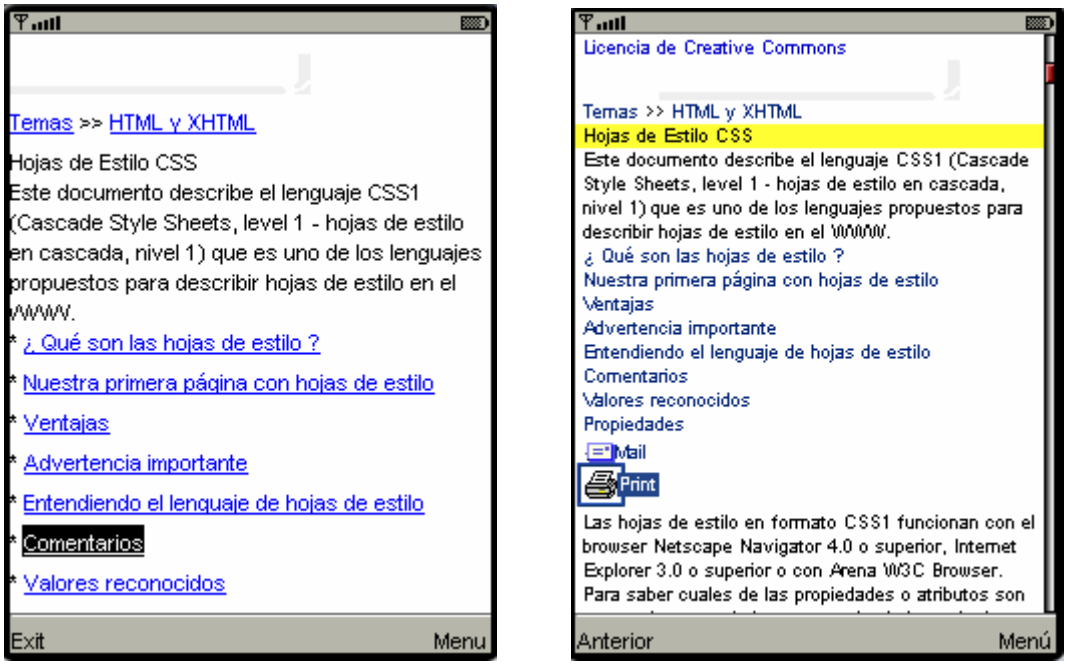


Figura 9.50 Estilos externos en microbrowsers.

Usando Firefox 1.5 vemos la correcta aplicación de los estilos.

Figura 9.51 Estilos externos en Firefox 1.5.

Ahora se va analizar un estilo interno de HTML, por ejemplo el que centra el enlace "Alojamiento Web" gracias a la siguiente línea:

```
<p style="text-align: center; " class="ADAlojaliaP">
```

Como puede verse en la siguiente figura WebViewer y Opera Mini no interpretan correctamente los estilos internos, a pesar de ser un caso muy simple, porque no muestra dicho enlace centrado.

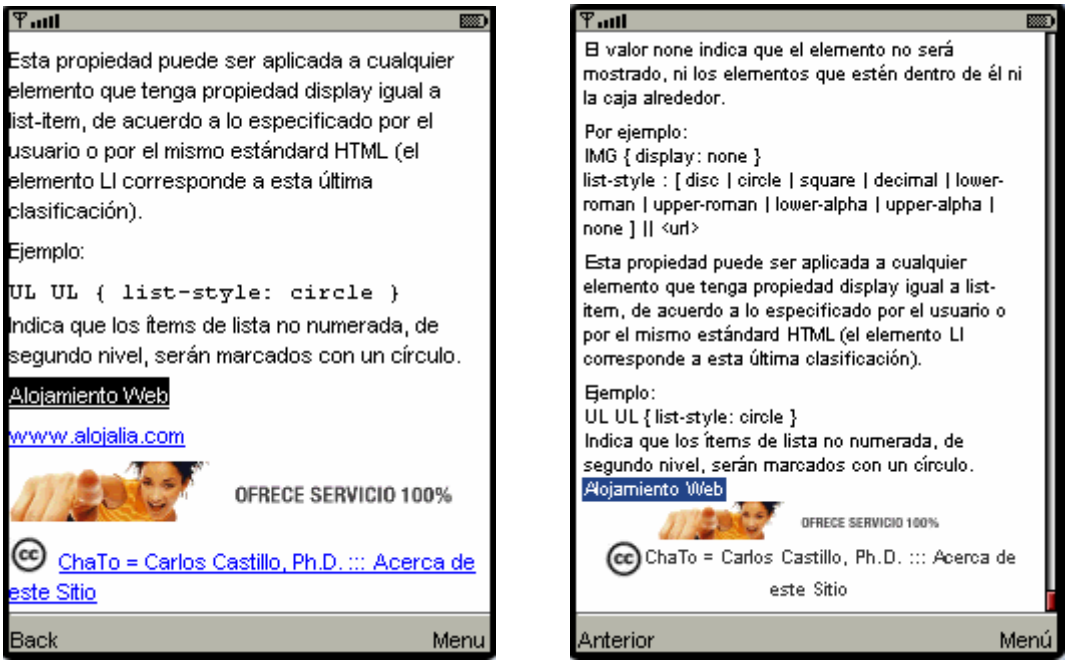


Figura 9.52 Estilos internos en microbrowsers.

Así es como mostraría correctamente centrada la línea “Alojamiento Web” el navegador Firefox 1.5.

Figura 9.53 Estilos internos en Firefox 1.5.

JBrowser tampoco usa hojas de estilo puesto que el lenguaje con el que trabaja, WML, carece de este concepto.

9.14 Uso del protocolo HTTP

El protocolo de transferencia de hipertexto (HTTP, *HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web (WWW) [46]. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con mensajes y otros similares. Existen dos versiones de este protocolo, la 1.0 y la 1.1. Todos los microbrowsers soportan ambas.

El protocolo HTTP es un protocolo sin estado; está basado en el modelo cliente-servidor: un cliente HTTP abre una conexión y realiza su solicitud al servidor, el cual responde generalmente con el recurso solicitado y la conexión se cierra. El puerto por defecto utilizado para el servicio WWW es el 80.

En las solicitudes van tres campos separados por un espacio en blanco: "Método Recurso VersiónDelProtocolo".

Existen tres verbos básicos (hay más, pero por lo general no se utilizan) que un cliente puede utilizar para dialogar con el servidor:

- GET, se utiliza para recoger cualquier tipo de información del servidor. Se utiliza siempre que se solicita una URL. Como resultado, el servidor HTTP envía el documento que le corresponde.
- HEAD, es igual que GET pero pide al servidor que le envíe sólo como la respuesta la cabecera. Es utilizado por gestores de caché de páginas o los servidores proxy.

- POST, funciona como HEAD pero además envía datos de información al servidor.

El protocolo HTTPS es la versión segura del protocolo HTTP [47]. El sistema HTTPS utiliza un cifrado basado en las *Secure Socket Layers* (SSL) para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP. Es utilizado principalmente por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas. El puerto estándar para este protocolo es el 443.

Para estudiar el intercambio de mensajes HTTP entre microbrowsers y servidores web, se ha empleado la utilidad *Network Monitor Extension* del emulador *Sun Java Wireless Toolkit 2.5 Beta*. Se han realizado una serie de operaciones y registrado los mensajes HTTP de la versión 1.1 que se intercambiaron. Estos son los resultados obtenidos:

- Conexión con respuesta correcta, jBrowser emplea GET, más simple, donde los demás usan POST:

  JBrowser
  HTTP GET /wml HTTP/1.1
  HTTP HTTP/1.1 200 OK

  WebBrowser
  HTTP POST /Web/HTTP/1.1
  HTTP HTTP/1.1 200 OK

  Opera
  HTTP POST /HTTP/1.1
  HTTP HTTP 1.1 200 OK

- JBrowser muestra el mesaje: “Receive response is too large to process, and can’t be processed.”, cuando es incapaz de cargar la página debido a su tamaño. Los mensajes HTTP son idénticos a los de un funcionamiento correcto:

  JBrowser
  HTTP GET / HTTP/1.1
  HTTP HTTP/1.1 200 OK

- Error por denegación de permiso por URL incorrecta:

  JBrowser
  HTTP GET /~rumeet/browser/index.xhtml HTTP/1.1
  HTTP HTTP/1.1 403 Forbidden

  WebViewer
  HTTP POST /Web/ HTTP/1.1
  HTTP HTTP/1.1 200 OK

  Opera
  HTTP POST /HTTP/1.1
  HTTP HTTP/1.1 200 OK

  En la siguiente figura se muestra cómo avisa el microbrowser al usuario de la aparición de este error:

Figura 9.54 Ejemplo de error HTTP (Error 403)

- Cuando existe un servidor que no llega a responder, se transmiten los siguientes mensajes. JBrowser no realiza intercambio de mensajes HTTP en este caso:

JBrowser	WebBrowser	Opera
Nada	HTTP POST /Web/HTTP/1.1	HTTP POST /HTTP/1.1
Nada	HTTP HTTP 1.1 200 OK	HTTP HTTP 1.1 200 OK

- WebViewer 4.0 es capaz de usar HTTPS usando la opción de certificado:

  HTTPS POST /Web/ HTTP/1.1
  HTTPS HTTP/1.1 200 OK

9.15 Visualizar otro tipo de archivos

Los microbrowsers están pensados para abrir páginas web. El microbrowser WebViewer 4.0 es capaz de abrir adicionalmente otro tipo de documentos de extensiones:

- DOC.
- PDF.
- ZIP.

Sólo puede abrir algunos archivos .DOC y PDF, y no lo hace de la misma manera que Microsoft Word y Acrobat Reader. Muchas veces lo único que muestra es el texto plano del documento. Existen otros formatos de compresión, TAR.GZ o RAR, pero no son soportados por WebViewer 4.0. Los ZIP no los puede descomprimir, sólo ver la lista de archivos que incluye.

Como ejemplo de archivo DOC se ha usado: www.wto.org/spanish/docs_s/legal_s/28-dsu.doc (que ocupa 125 Kb). Para los documentos PDF se ha usado el documento http://farmacia.ugr.es/ars/pdf/324.pdf. Algunos documentos no pudo abrirlos, como el www.iec.org/online/tutorials/acrobat/wap.pdf. En su lugar muestra el mensaje de error "*This PDF document is not viewable*" (este documento PDF no es visualizable).

La figura siguiente muestra documentos DOC y PDF abiertos por WebViewer 4.0:

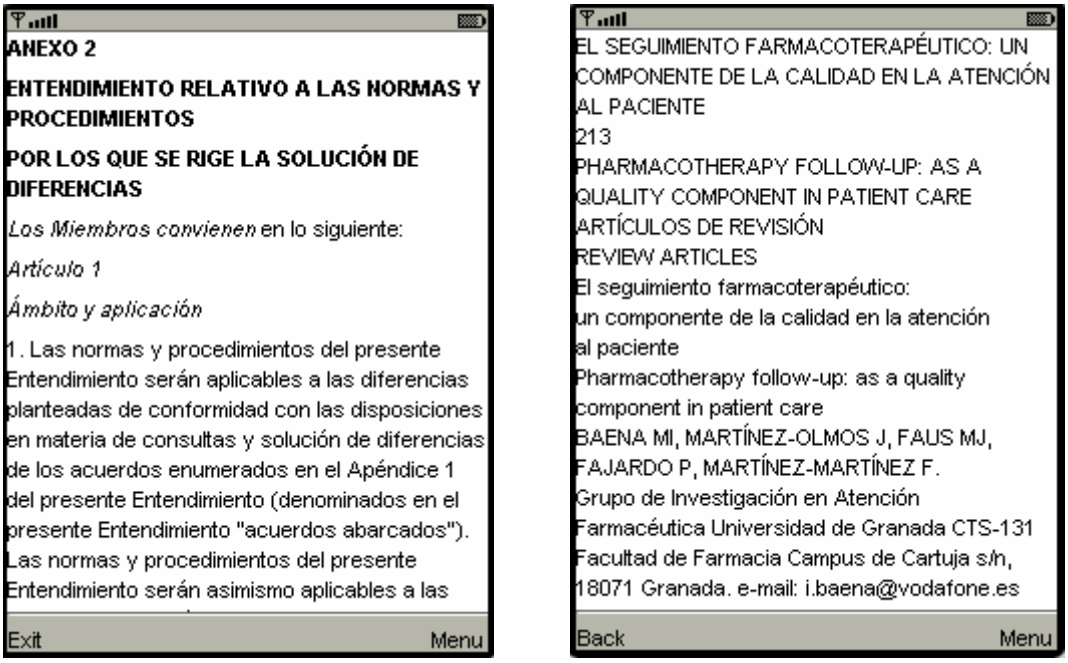
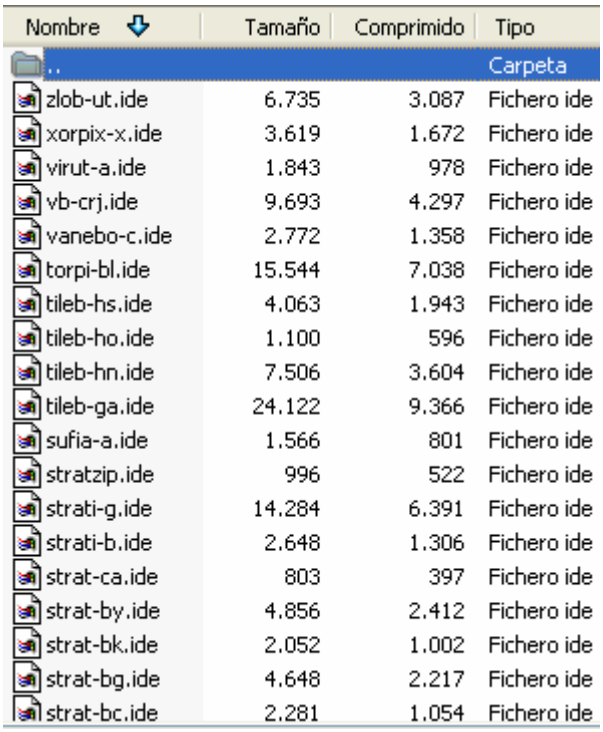


Figura 9.55 Documentos en formato DOC y PDF en WebViewer 4.0.

Como ejemplo de archivo .ZIP se ha usado:
http://www.sophos.com/downloads/ide/411_ides.zip.

En la siguiente figura se compara la información mostrada por WebViewer 4.0 y la de un programa de descompresión de archivos como es WinRar 3.5:



Nombre	Tamaño	Comprimido	Tipo
..			Carpeta
zlob-ut.ide	6.735	3.087	Fichero ide
xorpix-x.ide	3.619	1.672	Fichero ide
virut-a.ide	1.843	978	Fichero ide
vb-crj.ide	9.693	4.297	Fichero ide
vanebo-c.ide	2.772	1.358	Fichero ide
torpi-bl.ide	15.544	7.038	Fichero ide
tileb-hs.ide	4.063	1.943	Fichero ide
tileb-ho.ide	1.100	596	Fichero ide
tileb-hn.ide	7.506	3.604	Fichero ide
tileb-ga.ide	24.122	9.366	Fichero ide
sufia-a.ide	1.566	801	Fichero ide
stratzip.ide	996	522	Fichero ide
strati-g.ide	14.284	6.391	Fichero ide
strati-b.ide	2.648	1.306	Fichero ide
strat-ca.ide	803	397	Fichero ide
strat-by.ide	4.856	2.412	Fichero ide
strat-bk.ide	2.052	1.002	Fichero ide
strat-bg.ide	4.648	2.217	Fichero ide
strat-bc.ide	2.281	1.054	Fichero ide

Figura 9.56 Archivo .ZIP en WebViewer y en WINRAR 3.5

9.16 Uso de formularios

Los formularios son cajas de texto, botones, menús desplegables y otros métodos que permiten al usuario enviar información a través de las páginas web. Son útiles para implementar un servicio de educación a distancia.

Un ejemplo de formulario en WML 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" maxlenght="100" title="Email" name="Email" />
</fieldset>
1- Un motor de 8 cilindros en V, ¿ cuantos cigüeñales 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 9.57 Código WML con formulario.

La figura siguiente muestra cómo muestra esa página los microbrowsers que trabajan con WML, jBrowser 1.0.4 y Opera Mini

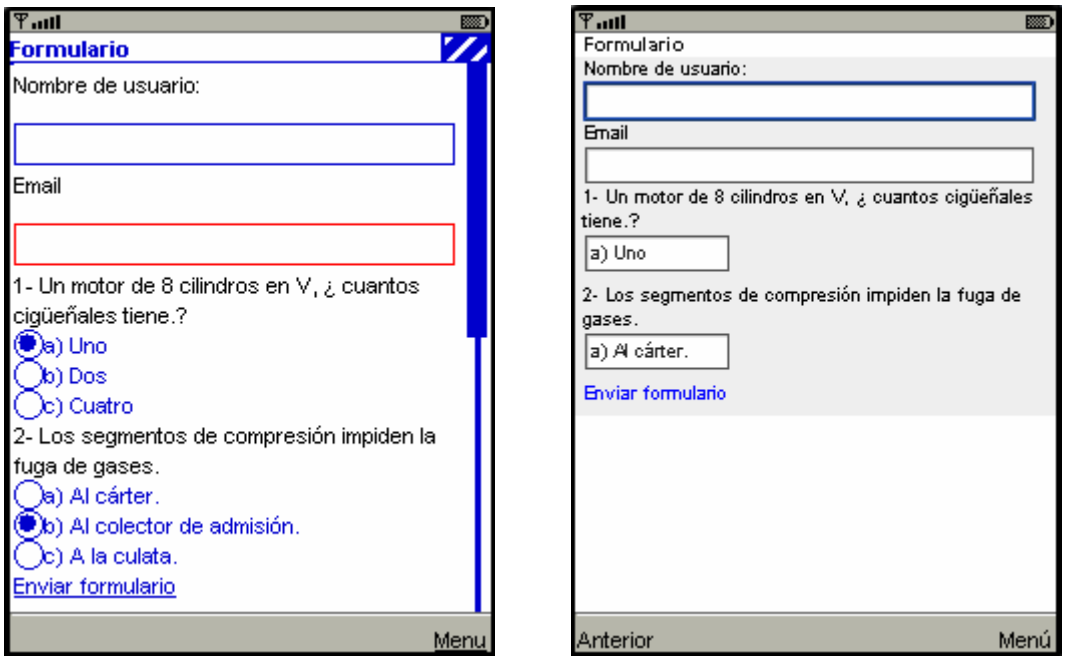


Figura 9.58 Formulario WML en jBrowser y Opera Mini.

En HTML se pueden hacer formularios como 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 9.59 Código HTML con formulario.

Este código se puede ejecutar en WebViewer y Opera Mini como se muestra en la siguiente figura.

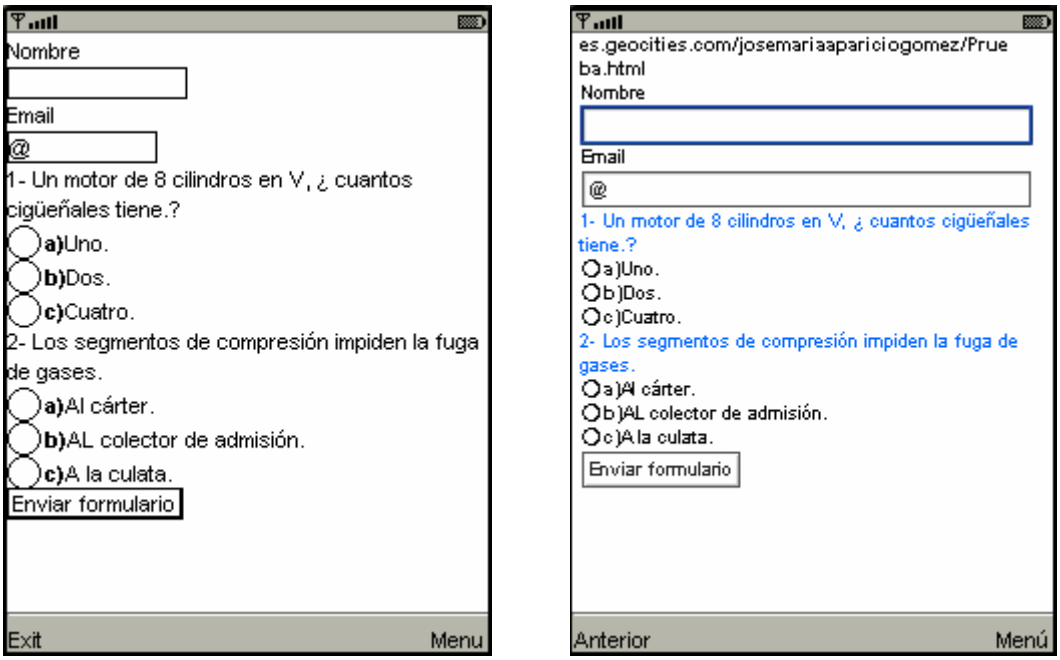


Figura 9.60 Formulario HTML en WebViewer y Opera Mini.

9.17 Otras opciones

En este último apartado se muestran algunas opciones típicas de los microbrowsers.

- **Marcadores**. Son palabras que el usuario define para sustituir a direcciones URL.
- **Caché**. Permite almacenar localmente una web para que no tenga que recurrir al servidor en los siguientes accesos.
- **Cookies**. Son archivos con cierta información sobre el usuario enviados por las páginas que se almacenan en el teléfono móvil
- **Descargas**. Algunos microbrowsers permiten almacenar archivos desde Internet.

	Marcadores	**Caché**	**Cookies**	**Descargas**
jBrowser	Hasta 10	Sí	No	No
WebViewer	Con directorios	Sí	Sí	No
Opera Mini Basic	Sí	Sí	No	No
Opera Mini Advanced	Sí	Sí	No	Sí

Tabla 9.7 Otras opciones.

Opera Mini Advanced es el único microbrowser que permite la descarga de archivos. La versión Basic tampoco puede, como se ve en la siguiente figura.

Figura 9.61 Opera Mini Advanced permite la descarga de los documentos.

Los resultados obtenidos en las pruebas realizadas nos han servido para comparar los diversos microbrowsers. A continuación se realiza una valoración de los resultados obtenidos.

9.18 Valoración global

El tamaño de los microbrowsers destinados al perfil MIDP 1.0 estaba limitado en torno a 60 Kb debido a la capacidad de los dispositivos. Opera Mini Advanced, pensado para MIDP 2.0, puede ocupar más espacio que sus antecesores, 95.8 Kb.

Debido a las limitaciones técnicas de estos dispositivos se diseñó un lenguaje de marcas propio, WML. Pero la mayoría de las páginas web de Interntet seguían estando especificadas únicamente en HTML, por lo que empezaron a desarrollarse microbrowsers que soportaran este lenguaje. Al no estar adaptado para ellos, para poder visualizarlo correctamente algunos hacen uso de servidores externos que transforman el código.

Los microbrowsers escritos en J2ME pueden conseguirse a través Internet de forma gratuita. La versión analizada de jBrowser indica un uso limitado a 60 días, pero pudo emplearse por un periodo de tiempo superior sin ningún problema.

Existen instituciones que aconsejan limitar el tamaño de las páginas web para que puedan ser soportadas por los dispositivos móviles. Los microbrowsers por su parte pueden reducir el tamaño de las páginas eliminando información adicional relativa al aspecto pero manteniendo el contenido.

Los datos reflejan una gran lentitud de WebViewer comparado con los demás. La versión 4.0, con la opción de ver javascript activada, es incluso excesivamente lento para manejarlo. Opera Mini Advanced es el más rápido de los que usan HTML porque tiene las menores tasas de transferencia de datos. JBrowser es el más rápido de todos porque el tamaño de las páginas WML es el menor.

Una de las mayores limitaciones de los dispositivos móviles es la dificultad para introducir datos. No disponen de ratón y su teclado es reducido. Algunos microbrowsers permiten usar combinaciones de teclas para moverse con mayor agilidad por las páginas.

La mayoría de los formatos de imágenes existentes en Internet se soportan también en el entorno inalámbrico y se ven con la misma calidad, a excepción de JPEG que presenta pérdidas en todos los microbrowsers. Opera Mini permite comprimir las imágenes en un servidor externo para reducir la tasa de transferencia, aunque perdiendo definición. La versión Advanced emplea algoritmos de compresión aún más exigentes que Basic, reduciendo aún más el tamaño de los archivos. Ningún microbrowser es capaz de mostrar fondos en las páginas, los GIF transparentes se ven como si fueran GIF opacos ni tampoco mostrar GIF animados en movimiento.

Algunas páginas contienen pequeños programas que se ejecutan mientras se están visualizando. Los más exigentes en recursos, las applets, no son soportadas por ningún microbroser. Otro tipo más sencillo, los scripts, sí funcionan. WML tiene su propio lenguaje para crear scripts, WMLscript, sin embargo jBrowser, que usa WML, no los soporta. En cuanto a HTML, WebViewer, en su versión 4.0, puede utilizar scripts, aunque activar esta función retarda el acceso a las páginas. Opera Mini, en sus dos versiones, también muestra scripts, en esta ocasión sin perjudicar la velocidad de carga.

El protocolo HTTP es el utilizado en Internet para transimitir los lenguajes de marcas. Todos los microbrowser lo soportan. jBrowser usa el comando GET, en la que los datos se transportan en el URI, donde los demás usan POST, que es más seguro.

Las hojas de estilo empleadas en HTML están muy mal soportadas por los microbrowsers analizados. Por eso algunas instituciones han tenido que especificar hojas de estilo propias para estos entornos inalámbricos como CSS Mobile Profile o WAP CSS.

WebViewer 4.0 tiene capacidad para abrir archivos con formatos como DOC o PDF aunque no correctamente. También puede ver el contenido de documentos ZIP, no así descomprimirlos, por lo que esta habilidad es poco útil.

Todos los microbrowsers soportan el uso de formularios. Esto se debe a que están especificados en los lenguajes de marcas, esta propiedad es muy útil para implementar un servicio de educación a distancia en dispositivos móviles.

Finalmente se vieron algunas opciones que tienen estos programas que resultan muy útiles, como usar caché o marcadores por ejemplo. La siguiente tabla condensa la información más importante proporcionada por este capítulo, algunos de los valores son aproximados:

	JBrowser 1.0.4	WebViewer 3.2	Web Viewer 4.0	Opera Mini Basic	Opera Mini Advanced
Tamaño	59.6 Kb	46.8 Kb	48.6 Kb	61.6 Kb	95.8 Kb
Lenguaje	WML	HTML	HTML	WML,HTML	WML,HTML
Tamaño de carga	~6Kb	~60Kb	~60Kb	~30Kb	~15Kb
Uso de memoria	Bajo	Medio	Medio	Bajo	Elevado
Velocidad	~2s	~15s	~40s	~3s	~3s
Calidad de imágenes					
WBMP	Alta	No disponible	No disponible	Alta	Alta
BMP	No disponible	Alta	Alta	Alta	Alta
PNG	Alta	Alta	Alta	Media/Alta	Baja/Alta
GIF	Alta	Alta	Alta	Alta	Baja/Alta
JPEG	No disponible	Baja	Baja	Baja	Baja
Scripts	No disponible	No disponible	Sí	Sí	Sí
Hojas de estilo	No disponible	No disponible	No disponible	Poco	Poco
HTTP	GET	POST	POST	POST	POST
Cookies	No disponible	Sí	Sí	No disponible	No disponible

Tabla 9.8 Comparativa de los microbrowsers.

9.19 Conclusiones

Como resumen, de cada microbrowsers se concluye lo siguiente:

- jBrowser es el microbrowser más limitado en cuanto contenido web por usar el lenguaje inalámbrico WML. Sus páginas tienen un tamaño muy reducido y son pobres gráficamente.
- WebViewer 3.2 y 4.0 presentan un rendimiento parecido salvo en el uso de scripts y la opción de abrir otros tipos de documentos. Los grandes defectos que tienen son la lentitud de navegación y la visualización distorsionada de las páginas HTML.
- Opera Mini es el microbrowser más completo, más reciente (2006), y diseñado por la compañía más potente, Opera Software. Sus dos versiones, Basic y Advanced, utilizan servidores externos y pueden emplear algoritmos de compresión para reducir la tasa de transferencia de datos a costa de perder calidad de imagen.