

VIDEOAPLICACIONES

PIMobile



GUÍA DE DESARROLLO

Cestel 

INTRODUCCIÓN

El presente documento es una guía de desarrollo de aplicaciones multimedia para teléfonos 3G, basadas en el entorno PIMobile de CESTEL.

ESQUEMA GENERAL DE FUNCIONAMIENTO DE PIMOBILE

PIMobile es un módulo software residente en la plataforma 3G de CESTEL. La misión de PIMobile es que las llamadas a él enrutadas por DNIS, ANIS, transferencia desde otro tipo de videoaplicación o cualquier otro criterio sean en realidad la retransmisión de la imagen y audio visible en un navegador web que accede a una determinada navegación por URL de servidores web conectados a Internet.

Por ello, toda aplicación PIMobile tiene una URL asociada como punto de inicio, a la cual PIMobile accede actuando de manera idéntica a un navegador web, y retransmite al teléfono móvil la imagen y sonido que dicho navegador web mostraría. Esta URL de inicio se configura mediante las herramientas de administración de la plataforma 3G.

A partir de mostrar la imagen obtenida de la URL inicial, PIMobile gestiona la conversión de teclas pulsadas en el teléfono (dígitosDTMF) a URL's a recuperar y mostrar en el móvil 3G, es decir, los dígitos pulsados equivalen a clicks sobre los enlaces de cada página web recuperada y luego mostrada en el teléfono.

Pimobile mantiene un navegador Internet Explorer para realizar esta tarea para cada llamada de móvil 3G que atiende de manera simultánea, por lo cual en cada momento mantiene una matriz de navegadores a los cuales asigna y desasigna las sesiones originadas por cada llamada.

Por lo tanto, PIMobile es capaz de realizar su función siempre que los sitios web a los que se accede estén preparados para funcionar bajo Internet Explorer, y en caso de emplearse en las páginas accedidas algún objeto adicional o embebido en una página web (por ejemplo un visualizador de videos o similar), éste debe estar previamente registrado en el servidor que alberga el software PIMobile.

Una aplicación PIMobile no reside en realidad en plataforma 3G ni en el propio espacio o entorno de PIMobile, sino que en realidad es una aplicación web que reside en un servidor web (o servidores web enlazados a nivel lógico entre sí mediante pulsación de enlaces) convencional.

Debido a que el destino final de la aplicación 3GB no es un navegador web, sino un teléfono móvil, las aplicaciones web deben reunir algunos requisitos que se enumeran en el siguiente capítulo, pero solo en cuanto al formato mostrado y a la captura de comandos, no en cuanto a las herramientas y tecnologías de desarrollo en sí.

Es decir, una aplicación PIMobile puede desarrollarse con cualquier herramienta o procedimiento habitual en desarrollo web: JSP, PHP, ASP, ASPX, FLASH, Javascript... etc

Cualquier programador capacitado en alguna de estas herramientas, lo está a su vez en el desarrollo para PIMobile sin más que seguir y tener en cuenta las cuestiones enumeradas y explicadas en este manual.

Por lo tanto, en el resto de este manual se supone que el lector está familiarizado con los conceptos y conocimientos de uso general en desarrollo de aplicaciones web: URL's, enlaces, marcos o frames, elementos HTML, javascript etc

REQUISITOS DE UNA APLICACIÓN WEB PIMOBILE

Cualquier aplicación web es *susceptible de ejecutarse* en el entorno PIMobile si cumple unos requisitos que enumeraremos a continuación.

¿Qué quiere decir “susceptible de ejecutarse” en PIMobile? Significa que será correctamente visualizada/escuchada en un teléfono móvil 3G y que mediante la simple pulsación de las teclas DTMF del teléfono se podrá navegar (es decir, enlazar URL's) por ella.

Por lo tanto esto equivale a decir que una aplicación web es “certificada PIMobile”.

Los requisitos principales son los siguientes, y para cada uno de ellos realizaremos unos comentarios y sugerencias para facilitar la reconversión de aplicaciones web ya existentes con el mínimo esfuerzo.

- Tamaño de pantalla
- Gestión de teclado
- Compatibilidad Internet Explorer 6 (IE6)
- Ejecución en una sola ventana o iframe

TAMAÑO DE PANTALLA

La pantalla de un móvil tiene el tamaño del standard QCIF es decir 176 pixels de anchura por 144 de altura (mayor anchura que altura, aunque en determinados teléfonos pudiera parecer que la dimensión vertical es mayor)

Por lo tanto si la aplicación web emplea más espacio en pantalla, solo se mostrarán los de 176x144 pixels de la esquina superior izquierda... de la información que mostraría un navegador internet explorer.

Si por ejemplo el punto de entrada de nuestra aplicación PIMobile fuese

<http://www.cestel.es>

y la imagen visualizada en un navegador Internet Explorer 6 (IE6) para dicha URL fuese la siguiente



PIMobile sólo retransmitiría al teléfono esta imagen de 176x144 correspondiente a la esquina superior izquierda



Sin embargo, si fuese necesario y la aplicación web accedida estuviese preparada, PIMobile dispone de la imagen completa y mediante Javascript podría realizarse un scroll para mostrar zonas inicialmente ocultas y fuera de la ventana de 176x144 mostrada.

Es decir, podríamos mostrar en la pantalla del móvil, por ejemplo, la zona recuadrada en rojo en la imagen siguiente. Cómo conseguir esto es objeto de estudio más adelante. Algunas aplicaciones del scroll de la zona visualizada serían:

- mover por la imagen de un mapa
- hacer scroll vertical en una pantalla web "larga", como por ejemplo las de una publicación electrónica de un diario.



En cualquier caso, lo habitual será mostrar al usuario en su teléfono una página web de tamaño QCIF, como en este menú de ejemplo



O una imagen de video o foto adaptada a ese tamaño como en estos ejemplos de video y webcam.



Webcam



Video de un Telediario

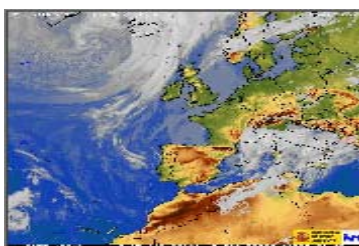


Imagen de satélite

Una buena práctica podría ser incluir el código HTML de las páginas a desarrollar para una aplicación PIMobile en un elemento DIV de tamaño igual al de QCIF. De esta forma, durante el desarrollo, si las páginas son invocadas desde un navegador de PC veremos solamente la parte que se vería en el teléfono, y si diseñamos una página no adecuada al tamaño QCIF, rápidamente nos daríamos cuenta solo con mostrarla en un navegador convencional de PC

Consideremos el siguiente ejemplo

```

<html>

  <head>
    <style type="text/css">
      td {font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
10px; color: #171717;}
      input {font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
10px; color: #171717;}
      .marcoQCIF {
        overflow: auto;
        position: absolute;
        top: 0px;
        left: 0px;
        width: 176px;
        height: 144px;
        border: 1px solid black;
        font-family: Verdana;
        font-size: 10;
        background-color: #ffffff;
        text-align: center;
        vertical-align: middle;
      }
    </style>
  </head>

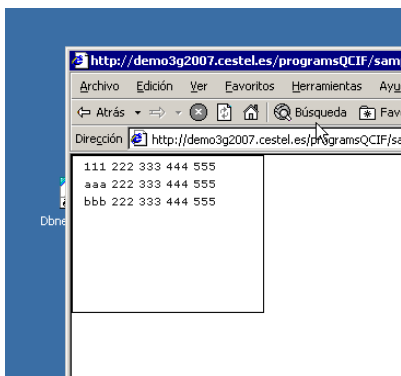
  <body topmargin="0" leftmargin="0">
    <div class="marcoQCIF">
      <table width=160>
        <tr>
          <td>111 222 333 444 555 </td>
        </tr>
        <tr>
          <td>aaa 222 333 444 555 </td>
        </tr>
        <tr>
          <td>bbb 222 333 444 555 </td>
        </tr>
      </table>
    </div>
  </body>
</html>

```

Al elemento DIV se le ha dado mediante CSS un borde para que sea visible su límite. El estilo aplicado al DIV puede incluirse en un fichero CSS, para ser reutilizado en las diferentes páginas que compongan el website PIMobile. Si invocamos a esta página cuya URL es

<http://www.pimobile.es/samples/demodiv1.htm>

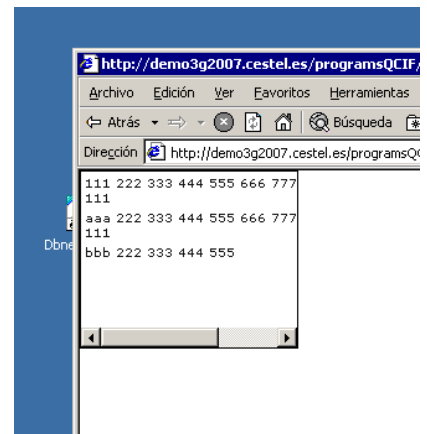
desde un navegador de PC, veremos en pantalla



El borde negro del DIV marca la zona equivalente a la pantalla del móvil. En este caso como se ve el móvil vería el contenido completo

Sin embargo si aumentamos la anchura de la tabla y añadimos más texto como por ejemplo cambiando el código de la tabla a

```
<table width=260>
  <tr>
    <td>111 222 333 444 555 666 777 888 999 000 111
  </td>
  </tr>
  <tr>
    <td>aaa 222 333 444 555 666 777 888 999 000
111</td>
  </tr>
  <tr>
    <td>bbb 222 333 444 555 </td>
  </tr>
</table>
```



El resultado es el de la imagen: ya no se muestra todo el texto en el recuadro (aparecen incluso barras de scroll) y el resultado por lo tanto sería inadecuado también para un teléfono móvil.

El desarrollador web para PIMobile por lo tanto debe tener en mente que el espacio del que dispone por medios “naturales” (sin contar con zoom y scroll, aunque como decimos es viable) es reducido y por ello

- el texto a incluir debe ser limitado en extensión
- las imágenes/vídeos deben tener suficiente calidad a tamaño QCIF

Por ello a la hora de adaptar o desarrollar una videoaplicación 3G será necesario definir claramente una estrategia y usabilidad que haga viable y ágil la presentación visual en pantalla.

GESTIÓN DE TECLADO

En una aplicación web convencional (destinada al navegador de un PC, con resolución mínima de 640x480), una vez que el navegador visualiza una página, es posible hacer click sobre los enlaces asociados a textos o imágenes, lo cual provoca que el navegador solicite la URL asignada al enlace clickado, y se muestre una nueva página web resultante.

Esta es la manera habitual de trabajar, pero en las videoaplicaciones 3G, por la propia naturaleza de las aplicaciones 3G para móviles no es posible más que enviar pulsaciones de dígitos DTMF (números 0 a 9, tecla * tecla #).

De manera simulada, al igual que cuando tecleamos el texto de un mensaje SMS, podemos también enviar caracteres de texto y especiales: por ejemplo tres pulsaciones rápidas de la tecla 2 equivalen a la letra C. En cualquier caso, se trata de pulsaciones sobre el teclado DTMF del teléfono.



Sin embargo desde una aplicación web convencional es posible también conseguir que los enlaces de una página se activen mediante pulsación de dígitos numéricos o de la tecla * ó #. Para ello se debe alterar mínimamente la página, añadiendo código Javascript, identificando cada enlace a considerar y asignándole una tecla.

Por ejemplo, es posible conseguir que el siguiente enlace (se muestra su código HTML) se pueda activar en una aplicación web convencional, y además en PIMobile, mediante una pulsación de teclado.

```
<a href="http://www.pimobile.es">Ir a web de PIMobile</a>
```

Para ello existen fundamentalmente dos métodos:

- el primero que denominaremos “gestión directa” permite que mediante javascript se pueda recibir directamente el código de la tecla pulsada y actuar en consecuencia: por ejemplo, para activar un enlace de la página, o hacer scroll, zoom o cualquier otra acción realizable vía javascript. Es adecuado cuando se quieren conseguir efectos complejos, como zoom, scroll, redimensionamiento de imágenes, activa o desactivar capas...
- el segundo, que denominaremos “gestión por enlaces”, facilita asignar, con muy poco código, opciones numéricas a enlaces presentes en la página web. Es el más apropiado para menús o páginas web que contengan varios enlaces y cuya misión sea fundamentalmente servir como paso intermedio a otras.

Como “evolución” de la gestión directa, es posible también gestionar el teclado para la captura de datos en un formulario. Para esto PIMobile proporciona una biblioteca Javascript, que será analizada en el capítulo [Formularios](#).

GESTIÓN DIRECTA DE TECLADO

1º) añadir en la página, en el elemento <BODY> el atributo

```
onkeypress="keyAction()"
```

Esto implica que ante cualquier pulsación de teclado (sea numérica o no...) el navegador llamará a la función indicada, sin realizar más acción al respecto.

2º) definir la función anterior, que atenderá por lo tanto las pulsaciones de teclado. Dentro de esta función es posible recuperar la tecla pulsada, que estará codificada en base a la tabla que adjuntamos. Aunque en una aplicación web convencional esta función recibiría las pulsaciones de cualquier tecla, en una videoaplicación solo se reciben las de dígitos DTMF, por lo cual nos limitaremos en la función a considerar estas teclas

Keycode	Dígito DTMF
48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
42	*
35	#

Ejemplo de una función de atención al teclado

```

<script>
function keyAction() {
    var key = String.fromCharCode(window.event.keyCode)
    switch( key ) {
        case '0': // codigo de atención a la tecla 0
            // por ejemplo activar el primer enlace de la página...
            document.links[0].click()
            break
        case '1': // codigo de atención a la tecla 1
            // por ejemplo navegar a una URL...
            location = 'http://www.cestel.es?index=1'
            break
        case '2': // codigo de atención a la tecla 2
            // por ejemplo redimensionar una imagen
            document.images[0].width=176
            break
        case '*': // tecla *
            // por ejemplo volver a la página anterior
            history.back()
            break
        default: // no atender a las otras teclas
            break
    }
    // acabar anulando el evento
    window.event.keycode = null
}
</script>

```

GESTIÓN DE TECLADO POR ENLACES

1º) incluir el código javascript del gestor de teclado "3G Navigation Handler" incluyendo en la cabecera de la página

```

<script src="http://www.pimobile.es/samples/lib/cestel.3g.NavHandler.js">
</script>

```

(La ruta mostrada apunta al servidor web de PIMobile, aunque el archivo puede hacerse residir con el resto de archivos de la aplicación web, para hacer ajustes propios)

2º) definir las asignaciones de teclas mediante código javascript como el del ejemplo siguiente

```

<html>
<head>
  <script src="http://www.pimobile.es/samples/lib/ceste1.3g.NavHandler.js">
  </script>
</head>
<body topmargin="0" leftmargin="0">
  <table width="176">
    <tr>
      <td align="center" width="176">
        <br>
        Elija idioma / Choose language<br>
        <br>
        <br><br>
        <br>
        <a href="menu1.htm" id="A1">Menu 1</a>
        <a href="menu2.htm" id="A2">Menu 2</a>
        <a href="menu3.htm" id="A3">Menu 3</a>
        <a href="menu4.htm" id="AA">Salir</a>
      </td>
    </tr>
  </table>

  <script>
    var commands = new Array();
    // asociaciones enlace opción
    commands["1"]="A1";
    commands["2"]="A2";
    commands["3"]="A3";
    commands["*"]="AA";
    // no olvidar esta línea
    window.document.onkeypress = navigationHandler;
  </script>
</body>
</html>

```

En el ejemplo anterior, la tecla '1' haría navegar a la dirección relativa a la URL actual 'menu1.htm', y la tecla '*' a 'menu4.htm'.

Como se observa, el único código javascript necesario es el que define las asociaciones tecla-enlace. Lo único adicional a tener en cuenta es asignar un identificador a cada enlace: (... id="A3"... en el ejemplo es la asignación de identificador al tercer enlace)

COMPATIBILIDAD INTERNET EXPLORER 6

Como se ha explicado en el esquema general de funcionamiento, las videoaplicaciones 3G se ejecutarán en el servidor PIMobile mediante un objeto Internet Explorer. Actualmente la versión empleada es la 6.0 por lo cual toda videoaplicación 3G PIMobile debe visualizarse y funcionar correctamente en dicha versión.

Basta acceder a la aplicación desarrollada desde un PC dotado de dicha versión para certificar que el funcionamiento será correcto en PIMobile.

EJECUCIÓN EN UNA SOLA VENTANA O IFRAME

Las videoaplicaciones 3G son la retransmisión de la imagen de una sola ventana de un navegador Internet Explorer. Por ello mensajes de alerta (como los alerts de Javascript) o las ventanas tipo pop-up no se muestran en el teléfono, por lo que deben evitarse.

Especialmente deben evitarse las de tipo alert de Javascript, pues el usuario no las podría cerrar mediante ninguna tecla de su teléfono, con lo cual la aplicación PIMobile quedaría bloqueada. Por ello este tipo de ventanas, empleadas ocasionalmente para depuración deben emplearse solo en versiones previas o de depuración de las aplicaciones y destinadas a ejecutarse como aplicaciones convencionales (en un PC)

PROGRAMACIÓN AVANZADA DE VIDEOAPLICACIONES 3G

En este capítulo se explica como conseguir que las videoaplicaciones 3G puedan realizar determinadas acciones no habituales en una aplicación web convencional pero que en un entorno 3G son especialmente útiles o necesarias.

INCLUSIÓN DE OBJETOS

Al igual que en una aplicación convencional, es posible incrustar llamadas a objetos mediante tags HTML de la forma

```
<object classid="CLSID:xxxx-xxxx-....."
        type="application/x-oleobject" id="foto"
        width="176" height="133"
        codebase="....." >
<param name="..." value="...">
<param name="..." value="...">
</object>
```

Se debe tener en cuenta que el área asignada al objeto debe ser más pequeña de lo que sería normal en una aplicación convencional. Este tamaño del área debe ser igual o inferior al estándar QCIF (salvo que vayamos a hacer scroll). El tamaño se controla con los atributos WIDTH y HEIGHT del tag OBJECT.

En cualquier caso, se debe verificar cómo reacciona la imagen que muestre el objeto al ser redimensionada, ya que quizás aparezcan franjas de relleno en las zonas superior, inferior y/o laterales, o el objeto ni siquiera admita mostrarse con el tamaño adaptado a nuestras necesidades.

Otra cuestión importante es que *el objeto debe estar registrado previamente en el servidor PIMobile* si queremos que se cargue en el navegador Internet Explorer y se visualice correctamente en el teléfono. Por lo tanto, no se podrán emplear objetos que en aplicaciones web convencionales requieran una descarga previa de algún elemento ActiveX, Java o similar. Solo se podrán emplear si previamente se solicita a PIMobile el registro en el servidor del objeto correspondiente.

Los dos tipos de objeto más frecuentes (Windows Media y Flash) ya están instalados en el servidor PIMobile, por lo que se pueden emplear directamente y funcionarán en videoaplicaciones 3G de la misma manera que lo harían en una aplicación web convencional.

En ambos tipos de objeto (y en otros que pudieran emplearse) hay que prestar atención a:

- Activar el modo autoarranque de la reproducción, si se trata de un vídeo, ya que por regla general será difícil convertir la pulsación de una tecla del teléfono a un click en una zona determinada del objeto que lance la reproducción.
- Desactivar la visualización del panel de control que suelen llevar estos objetos (botones play, stop, pausa etc...) ya que también será difícil emular los clicks en determinada posición del panel de control para conseguir la parada arranque etc.

Sin embargo, los objetos suelen ser controlables vía Javascript (mediante eventos). En la sección de gestión directa del teclado hemos visto que es posible traducir una pulsación de teclado a código Javascript y ese código podría ser válido para parar, arrancar, etc. la reproducción de videos en un objeto de este tipo.

A continuación se comentan dos ejemplos de inclusión de objetos Windows Media y Flash, sin que ello sea una guía exhaustiva de parámetros y posibilidades de estos objetos.

Para ello consultar:

Flash:

http://kb.adobe.com/selfservice/viewContent.do?externalId=tn_12701

Windows Media:

<http://msdn2.microsoft.com/en-us/library/ms983653.aspx>

EJEMPLO DE INCLUSIÓN DE OBJETO FLASH

```
<object id="Player" width="176" height="194" valign="top"
      classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000">
  <param name="allowScriptAccess" value="sameDomain" />
  <param name="menu" value="false" />
  <param value="/comunes/multimedia/video/1_a/video_1_a.swf?update=200704261431"
name="movie" />
  <param
value="url_fv=%2Fcomunes%2Fplayer%2Fmm.php%3Fxfref%3D20070524elpepunac_5.Ves
&referrer_fv=http%3A/www.elpais.com/videos/espana/Acebes/acusa/Zapatero/reabrir/herid
as/pasado/generar/division/elpvid/20070524elpepunac_5/Ves/&memory_fv=0&autoplay_fv=
1&id_fv=842983&pagead_fv=" name="FlashVars" />
  <param value="best" name="quality" />
  <param value="opaque" name="wmode" />
  <param value="#ffffff" name="bgcolor" />
</object>
```

EJEMPLO DE INCLUSIÓN DE OBJETO WINDOWS MEDIA VIDEO

Resaltados los parámetros fundamentales

- Autostart = indica que se debe arrancar directamente la reproducción
- Rate = actúa sobre la velocidad
- Uimode = indica si se debe mostrar el panel de control
- URL = URL de la fuente del video (formato http, mms, rtsp...)

```
<object classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6"
      type="application/x-oleobject"
      id="foto"
      width="176" height="133"
      standby="Loading content..."
codebase="http://activex.microsoft.com/activex/controls/mplayer/en/nsmp2inf.cab#Version=
5,1,52,701">
  <param name="autoStart" value="true"/>
  <param name="balance" value="0"/>
  <param name="currentPosition" value="0"/>
  <param name="enableContextMenu" value="false"/>
  <param name="enabled" value="false"/>
  <param name="fullScreen" value="false"/>
  <param name="mute" value="false"/>
  <param name="playCount" value="1"/>
  <param name="rate" value="0.5"/>
  <param name="uiMode" value="none"/>
  <PARAM NAME="windowlessVideo" VALUE="true">
  <PARAM NAME="URL"
VALUE="mms://stream.farreachtech.com/newsmyrnabeachcam">
  <param name="volume" value="70"/>
</object>
```

FORMULARIOS

En las aplicaciones web convencionales basadas en HTML es frecuente el uso de formularios, definidos mediante tags FORM, INPUT y SELECT.

Para completar los campos de texto se emplea el teclado alfabético en un PC, y para activar, desactivar o listar los campos radio, checkbox y select se emplea el ratón y las teclas de cursor.

Como se ha comentado en el capítulo de gestión del teclado, mediante pulsación de dígitos DTMF es posible simular la de los caracteres alfabéticos y especiales (comas, punto, espacio etc), por lo tanto será también posible completar los campos de texto de formularios que se presenten en videoaplicaciones 3G.

Para ello se suministra un fichero de código Javascript, que incluido en el código de la página HTML del formulario a manejar permite que desde el teléfono se rellenen sus datos y se valide cada campo y el formulario en su conjunto

Un ejemplo básico sería el siguiente, accesible bajo

<http://www.pimobile.es/samples/demo.htm>

```

<html>
  <head>
    <script src="forms.js"></script>
    <!-- control the text aspect (size, color...) with CSS -->
    <style type="text/css">
      td {font-family: Verdana, Arial, Helvetica, sans-serif; font-size:
10px; color: 171717;}
      input {font-family: Verdana, Arial, Helvetica, sans-serif; font-
size: 10px; color: 171717;}
    </style>
  </head>

  <body topmargin="0" leftmargin="0" onload="focusField( 'form', 'field1' )">
    <form name="fform" action="action.htm">
      <table>
        <tr>
          <td colspan=2 align=center>Sample form</td>
        </tr>
        <tr>
          <td>Field1</td>
          <td><input type=text name=field1 id=field1 size=4
maxlength=4
onkeypress="tecla(kLetAlfa,false)" tabindex=1></td>
        </tr>
        <tr>
          <td>Field2</td>
          <td><input type=text name=field2 id=field2 size=10
maxlength=10
onkeypress="tecla(kLetAlfaNum,false)" tabindex=2></td>
      </table>
    </form>
  </body>
</html>

```

```

        </tr>
        <tr>
            <td>Field3</td>
            <td><input type=text name=field3 id=field3 size=8
maxlength=8
onkeypress="tecla(kLetNum,true)" tabindex=3>##=Submit</td>
        </tr>
        <tr>
            <td colspan=2>Fill every field, ##=Next field</td>
        </tr>
    </table>
</form>
</body>
</html>

```

Puntos a resaltar en el código anterior y que constituyen las condiciones para conseguir un formulario válido para una videoaplicación 3G:

- Se incluye una llamada al código Javascript del archivo *forms.js*. En este archivo reside el código de las funciones *focusField* y *tecla*.
- El formulario solo incluye campos input de tipo *text*, aunque pudieran incluirse también de tipo *hidden*
- En el TAG *body* se incluye una llamada para situar el foco en un determinado campo del formulario, que pudiera no ser el primero. Los parámetros son

function focusField(nform, nfield)

nform = nombre del formulario sobre el que centrar el foco, si es null o inexistente será el primero que exista en la página

nfield = nombre del campo a activar en el formulario, si es null o inexistente será el primero

- Cada input debe contener un atributo que defina su nombre dentro del formulario (name) y otro su identificador de elemento en el DOM (Document Object Model) de la página (id).
- Cada input debe contener un atributo *tabindex* que defina la secuencia de "salto" entre campos, es decir el primero tendrá valor *tabindex=1*, el segundo *tabindex=2* etc
- Cada input debe indicar la función *tecla* para ser ejecutada en cada pulsación de tecla en ese campo, es decir, en el atributo *onkeypress*. Los parámetros son

function tecla(arrayValores, validateField, validateFunction)

arrayValores = nombre del array con los caracteres asignados a cada tecla DTMF. Por ejemplo la asignación de los caracteres a,b,c y 2 es la habitual para la tecla 2 de un teléfono móvil. Se define así la secuencia de caracteres que se mostrarán en el campo en pulsaciones rápidas y sucesivas de una tecla. Por ejemplo 3 pulsaciones sobre la tecla 2 del teléfono harían aparecer en el campo el carácter 2. Existen cuatro arrays predefinidos, cuyos nombres se pueden indicar aquí

kLetAlfaNum = letras, números y espacio (tecla 0)

kLetNum = números

kLetAlfa = letras y espacio (tecla 0)

kLetNumBlank = números y espacio(tecla 0)

validateField = true si el campo valida el formulario mediante dos pulsaciones rápidas de la tecla #

validateFunction = si el campo es de validación, y se pulsan dos # consecutivos y rápidos, función Javascript que revisa el contenido y del formulario para verificar su coherencia. Si encuentra un error, debe devolver el texto a mostrar al usuario como advertencia. Si devuelva null o una cadena vacía, se valida el formulario

- En cada campo sobre el que en la función *tecla* no se ha definido que sea validador del formulario (segundo parámetro = false, null o inexistente) la tecla # provoca el paso al siguiente campo, en el orden definido por los atributos *tabindex*.
- En los campos validadores, una doble pulsación rápida de la tecla # valida el formulario (tras ejecutarse la función de validación, si procede). Una sola # y una pausa provoca el paso al campo siguiente en el orden definido por los atributos *tabindex*. Otro carácter pulsado tras el primer # sin dejar pausa, mantiene el foco en el mismo campo y muestra el carácter pulsado

El aspecto del formulario de ejemplo, en la pantalla del teléfono móvil (o en la de un PC convencional) sería el siguiente

Sample form

Field1

Field2

Field3 #

Fill every field, #=Next field

SCROLL Y ZOOM

El navegador Internet Explorer ejecutado por PIMobile para cada llamada de móvil recupera el contenido completo de la pantalla de cada URL a la que navega, aunque como se he explicado solo muestra el recuadro de 176x144 pixeles de la esquina superior izquierda.

Por ello una videoaplicación 3G mediante Javascript puede conseguir desplazar dicho recuadro para mostrar otras zonas de la imagen completa de la página web cargada en el navegador Internet Explorer.

Esto puede ser de utilidad en determinadas aplicaciones que pretendan mostrar una imagen (fija o video) que en tamaño QCIF no se aprecie con suficiente detalle, por ejemplo

- webcams
- mapas
- gráficos
-

Otra posibilidad interesante es la de redimensionar imágenes (o incluso objetos) cuyo tamaño original no sea adecuado a QCIF, es decir hacer zoom sobre ellos. En combinación con el scroll esto facilita el inspeccionar al detalle una imagen de tamaño considerablemente mayor al de QCIF, por ejemplo una webcam fija o de video

El ejemplo básico sería el siguiente, accesible bajo

<http://www.pimobile.es/samples/demoscroll.htm>

Se muestra, actualizada en tiempo real, la imagen de una webcam estática (imagen tomada cada cierto tiempo) de una estación de esquí

```
<html>
  <head>
    <style type="text/css">
      td {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;
color: #171717;}
      input {font-family: Verdana, Arial, Helvetica, sans-serif; font-size: 10px;
color: #171717;}
    </style>
  </head>

  <body onload="loaded()"
    onkeypress="keyStroke()" topmargin="0" leftmargin="0">

    <!-- help screen, command 0 hides/unhides -->
    <div width="100" height="80" id="help" style="font:10px/12px
Arial,Helvetica,sans-serif;
border:solid 1px #666666;
width:100px; height:80px; padding:1px; position:absolute; z-
index:100; visibility:visible;
color:#333333;
top:15px; left:15px; background-color:#ffffcc; layer-
background-color:#ffffcc;">

      0=help on/off<br>
      9=exit<br>
      7=zoom+<br>
      3=zoom-<br>
```

```

                2,8,6,4=move up/down/right/left
            </div>

            <!-- lower permanent options menu -->
            <div width="176" height="8" id="help1" style="font:10px
    Arial,Helvetica,sans-serif; border:solid 1px
            #666666;
            width:175px; height:15px; padding:1px; position:absolute; z-
    index:100; visibility:visible; color:#333333;
            top:129px; left:0px; background-color:#ffffcc; layer-
    background-color:#ffffcc;">
            <span id=txtcamara1><b>0 Help 7/3 Zoom+/- 2,8,6,4
    Move</b><br></span>
            </div>

            <script src=scroll_zoom.js></script>
            <script>
                // start position ( coords of the point to be shown at screen
    center)
                var x = 100
                var y = 200

                // URL of the image to show
                var imageURL = 'http://www.valdesqui.es/cam/valdemartin.jpg'
                // URL to exit (command 9)
                var urlQuit = null
                document.write('<br><br><br>' )

                var windowHeight = 176
                var windowHeight = 144
                var windowHeightHalf = parseInt(windowHeight/2)
                var windowHeightHalf = parseInt(windowHeight/2)
                var menuHeight = 15 // height of the permanent menu

                //real screen height, considering bottom menu
                var actualScreenHeight = windowHeight-menuHeight

                // pixels from top for permanente menu (width of menu=144-
    topMenu)
                var topMenu = windowHeight-menuHeight
                // pixels from left for permanente menu (width of menu=144-
    topMenu)
                var leftMenu = 0

                // max Zoom factor allowed
                var topZoom = 8
                // starting zoom level
                var zoom = 1
                // pixels for every zoom level increment
                var pixZoom = 80
                // pixels for every scroll incremente
                var pixScroll = 30

                // if true, the image if initially fitted to 176x144 ("full screen")
                var startFit = true
                // allow fit ( command 5 )
                var allowFit = true
    
```

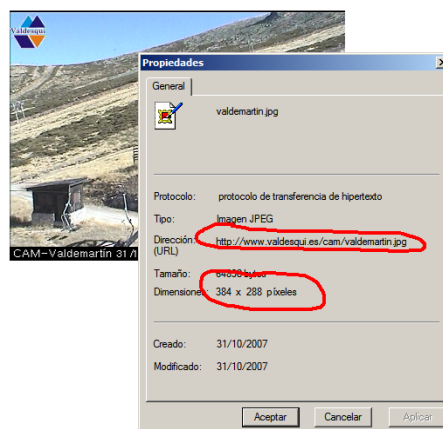
```
// allow move to home position (upper left 0,0) (command 1)
var allowHome = true

</script>
</body>
</html>
```

La imagen original, extraída de la URL <http://www.valdesqui.es/cam/valdemartin.jpg> sería la siguiente



El tamaño real es considerablemente mayor al QCIF, como se observa



El resultado sobre la pantalla de un teléfono móvil que acceda a una videoaplicación 3G que muestre la página de ejemplo sería el mostrado aquí.

Como se observa, se muestra el contenido ajustado a la pantalla del móvil, según se solicita en el código Javascript de la página (**startFit = true**)



También sería posible mostrar la imagen en su tamaño real, pero centrada por ejemplo en unas determinadas coordenadas. Esto puede ser, por ejemplo, útil en un plano. Para ello se haría

`startFit = false`

`x = ...`

`y = ...`

donde x e y serían las coordenadas donde se debe centrar la visualización

en el ejemplo mostrado x=100, y=200



El archivo javascript

`<script src=scroll_zoom.js></script>`

Contiene el código necesario para la gestión de teclado, y permite las siguientes posibilidades (numéricas como corresponde a los dígitos DTMF)

3, 7 = Zoom +/-

2,8,4,6 = Arriba/Abajo/Izquierda/Derecha (scroll de la imagen si no está ajustada a la pantalla)

0 = Mostrar ayuda de comandos

9 = Salir (navegar a otra URL)

La cantidad de pixels de zoom o scroll en cada incremento o movimiento se pueden determinar en el código HTML



Otra posibilidad es manejar un objeto, por ejemplo un reproductor de video Media Player. Si el objeto admite el redimensionamiento mediante la manera habitual en Javascript (modificando sus propiedades width y height) el comportamiento será similar al estudiado para el ejemplo anterior de la webcam estática.

En la dirección

<http://www.pimobile.es/samples/demoscrollobject.htm>

Se puede ver un ejemplo de la webcam de una autopista. Ha bastado cambiar, en el código del ejemplo anterior

```
// URL of the image to show
var imageURL = 'http://www.valdesqui.es/cam/valdemartin.jpg'
document.write('<br><br><br>' )
por

// URL of the image to show
var imageURL = 'mms://stream2.terra.es/wm2live2'
document.write(' <object classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6" +
"type="application/x-oleobject" +
'id="foto" +
'width="176" height="134" +
'standby="Loading content..." +
'codebase="http://activex.microsoft.com/activex/controls/mplayer/en/nsmp2inf.cab#Versio
n=5,1,52,701">' +
'<param name="autoStart" value="true"/>' +
'<param name="balance" value="0"/>' +
'<param name="currentPosition" value="0"/>' +
'<param name="enableContextMenu" value="false"/>' +
'<param name="enabled" value="false"/>' +
'<param name="fullScreen" value="false"/>' +
'<param name="mute" value="false"/>' +
'<param name="playCount" value="1"/>' +
'<param name="rate" value="1.0"/>' +
'<param name="uiMode" value="none"/>' +
'<param name="windowlessVideo" VALUE="true">' +
'<param name="URL" value="' + imageURL + "'>' +
'<param name="volume" value="70"/>' +
'</object>' )
```

SIMULADORES

Es posible emplear simuladores para mostrar en un PC lo que se vería en un móvil 3G al acceder a una videoaplicación 3G. Actualmente hay dos que pueden ser de utilidad, uno de ellos para la fase de desarrollo de la aplicación y otro para la de despliegue y verificación completa.

No se debe olvidar que también es posible acceder a cualquier aplicación PIMobile desde un PC dotado de Internet Explorer sin más que escribir directamente la URL de entrada. En la esquina superior izquierda del navegador se mostraría lo que se vería en el teléfono.

SIMULADOR VIDEOAPLICACIONES 3G

Desarrollado y suministrado por CESTEL, se trata de un navegador Internet Explorer con aspecto de un móvil 3G y con el tamaño de pantalla ajustado al standard QCIF. Se puede escribir directamente la URL a acceder, y hacer click sobre las teclas numéricas para enviar los comandos a la aplicación.

Tiene la ventaja de poder probar las videoaplicaciones 3G sin necesidad de desplegarlas en la plataforma donde se ubique PIMobile. Es decir, tiene aplicación especialmente en las fases de desarrollo previas al despliegue, sin necesidad de acceso alguno por videollamada.



SIMULADOR X-LITE



Suministrado por CounterPath

(<http://www.counterpath.com>) se trata de un softPhone es decir de un teléfono software adecuado para servicios de voz sobre IP.

Mediante él es posible acceder a la dirección SIP (dirección IP) del Video Call Center donde reside PIMobile y lanzar una videollamada, idéntica a la que realizaría un teléfono móvil 3G pero sin acceder mediante la red UMTS de un proveedor de servicios móviles.

Tiene la ventaja respecto al otro simulador que permite probar la aplicación completa, ya instalada en el Video Call Center y configurada para su acceso desde su terminal real, un teléfono móvil. Por ello tiene su mayor utilidad para verificar el correcto funcionamiento de los despliegues y videoaplicaciones 3G

PUESTA EN FUNCIONAMIENTO DE VIDEOAPLICACIONES 3G

Las videoaplicaciones 3G al ser realmente aplicaciones Web NO RESIDEN EN LA PLATAFORMA DONDE ESTÉ UBICADO EL SERVIDOR PIMOBILE, sino en el/los servidor/es web que albergan las aplicaciones accedidas a través de las URL de sus páginas.

Por lo tanto el despliegue de una videoaplicación 3G consiste en

- 1) verificar que los servidores web a los que se vaya a acceder tengan correctamente desplegada la aplicación.
- 2) verificar que dichos servidores son accesibles, para las URL's, direcciones IP y puertos en cuestión desde Internet, o al menos desde la IP del servidor PIMobile.
- 3) solicitar a PIMobile un punto de entrada desde su plataforma, facilitando la URL de entrada de la aplicación. Es decir, la plataforma realizará, si verifica que la aplicación es compatible PIMobile, la asignación de un número telefónico de acceso a la URL en cuestión y definirá el número de conexiones simultáneas admitidas.

Una vez puesta en explotación, debe notarse que la disponibilidad de una videoaplicación 3G y su tiempo de respuesta, por lo tanto, depende no solo de la plataforma y de Pimobile, sino del/de los servidor/es web que realmente contienen la videoaplicación 3G.