



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



**MASTER EN COMUNICACIONES Y  
DESARROLLO DE SERVICIOS MÓVILES**  
2009-2010

INTERFACES GRÁFICAS MULTIMEDIA  
Desarrollo de aplicaciones





## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- Interactividad y navegación
- Texto
- Multimedia
- Comandos del dispositivo
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA



Este será el tema más extenso y aplicado de la asignatura. Se trabajará con el IDE Flash para realizar diferentes aplicaciones multimedia en dispositivos móviles.



## Contenido



MUCOM 2009-2010

- **Introducción** ►
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- Interactividad y navegación
- Texto
- Multimedia
- Comandos del dispositivo
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





# Introducción



MUCOM 2009-2010



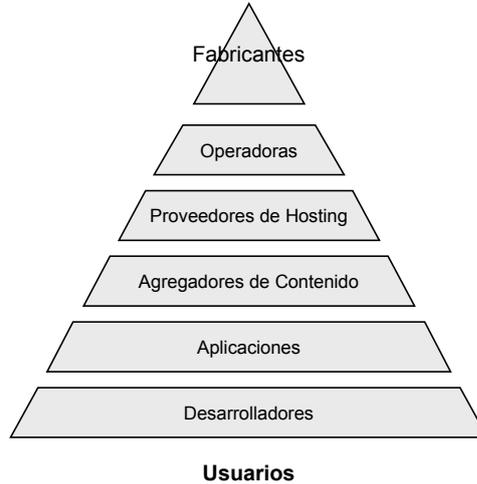
INTERFACES GRÁFICAS MULTIMEDIA



ActionScript permite el control total sobre la interfaz de las aplicaciones, lo cual lo convierte en la herramienta ideal para el desarrollo de aplicaciones multimedia. En esta imagen se muestran capturas de pantalla de algunas aplicaciones realizadas con este software.



- El ecosistema móvil



El ecosistema móvil es un modelo de negocio relativamente nuevo, sujeto por ello a diversos (y rápidos) cambios. Cuando se desarrollan aplicaciones en este ecosistema, dicho modelo de negocio es bastante distinto del empleado en software de escritorio o en aplicaciones Web. Por ello, es interesante entender los diferentes jugadores que forman parte del mismo.

Básicamente cuando un desarrollador crea una aplicación, éste suele venderla a un agregador de contenidos, el cual la promociona a través de su canal de distribución. Dichos canales de distribución suelen accederse desde la Web y, además, pertenecer a una(s) operadora(s). Finalmente, en la cumbre de la pirámide se encuentran los fabricantes, ya que son ellos los que deciden el grado de integración de las aplicaciones con su hardware.

Existe una tendencia actual a crear ecosistemas cerrados, tales como la App Store o el Android Market, en la que son los propios fabricantes los encargados de aprobar y entregar al usuario las aplicaciones que crean los desarrolladores.



## Introducción



MUCOM 2009-2010

- **Determinar el tipo de contenido**

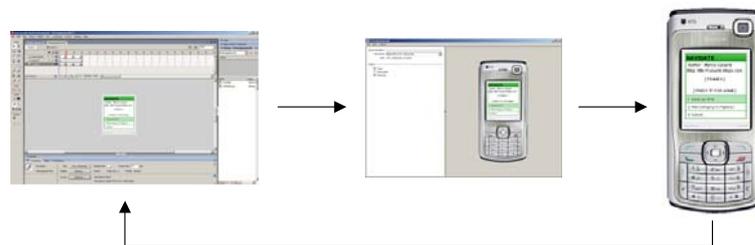
Características del dispositivo para el que vamos a desarrollar.  
Aplicaciones Flash permitidas (standalone, screensaver, alarm, etc.).

- **Preparar el entorno de desarrollo**

Ventana aplicación (escenario), frame rate, opciones de publicación, colores.  
Creación de plantillas para automatizar este proceso.

- **Flujo de trabajo**

Proceso iterativo: desarrollo → testeo en emulador → implementación real en el dispositivo.



INTERFACES GRÁFICAS MULTIMEDIA



Existe gran variedad de dispositivos móviles con gran variedad de resoluciones de pantalla, colores y distintos formatos de vídeo y audio. Estos factores son sólo algunos de los que van a restringir el diseño de nuestra aplicación. Es más, algunos dispositivos permiten emplear aplicaciones Flash Lite como salvapantallas, alarmas o tonos de llamada, lo cual puede limitar las características y uso de dicha aplicación – típicamente un salvapantallas no suele tener acceso a las conexiones de red o descarga de datos, por ejemplo. Flash Lite es la herramienta perfecta para aplicaciones con funcionalidades sencillas como por ejemplo relojes, microjuegos o FMAs (Flexible Messaging Areas: aplicaciones informativas y de merchandising). Existe una lista de dispositivos que soportan Flash Lite en la Web de Adobe: [http://www.adobe.com/mobile/supported\\_devices/](http://www.adobe.com/mobile/supported_devices/).

Una vez que decidimos empezar un nuevo proyecto o archivo individual de Flash Lite, lo primero que hay que hacer es configurar las propiedades del escenario: tamaño (ancho y alto en pixels), color de fondo, frame rate, opciones de publicación y el dispositivo contra el que vamos a probar la aplicación. En este curso emplearemos el teléfono Nokia N70, cuya resolución de pantalla es de 176x208 px, tiene 262144 colores (18 bits), un procesador a 220 MHz y 2 MB de RAM.

El desarrollo de aplicaciones bajo Flash Lite es un proceso iterativo que consta principalmente de tres fases: el desarrollo de la aplicación en el IDE Flash, su testeo en el emulador y su implementación real en el dispositivo. El emulador de Flash Lite nos permitirá comprobar el funcionamiento de prácticamente toda nuestra aplicación. Efectivamente, no será hasta que transfiramos esa aplicación al dispositivo móvil cuando podamos ver cómo se comporta realmente nuestro desarrollo — de momento el emulador no puede simular, por ejemplo, la velocidad del procesador del dispositivo o su profundidad de color. Por ello, una vez comprobada físicamente la aplicación podremos refinar posteriormente su diseño en el entorno de desarrollo Flash.

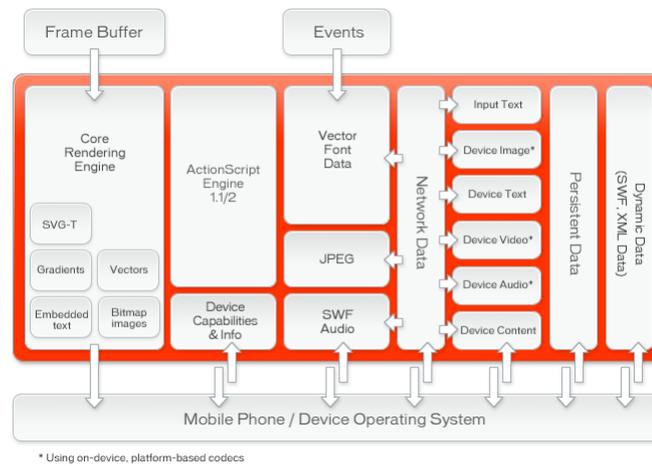


## Introducción



MUCOM 2009-2010

- Diagrama de arquitectura



INTERFACES GRÁFICAS MULTIMEDIA



El componente principal de Flash Lite es el motor de renderizado, compuesto de elementos visuales que pueden mostrarse en la pantalla. Se complementa con el componente ActionScript, que procesa código y eventos, lo que permite la interactividad dinámica en la aplicación.

También existen componentes especializados para procesar y visualizar otro tipo de datos, como es el caso de imágenes y vídeo, o bien datos de dispositivos, como la señal de red o el nivel de batería. La clase final de componentes interactúa directamente con las capacidades de procesamiento de los mismos dispositivos con el fin de optimizar el rendimiento general.



- Versiones de Flash Lite

Características	Flash Lite 1.x	Flash Lite 2.x	Flash Lite 3.x
Flash Player	4	7	8
ActionScript	1	2	2
Device communication	sólo en 1.1	sí	sí
Player Size (KB)	275	450	374
Min. RAM (KB)	64	128	128

Las primeras versiones de Flash Lite estaban basadas en el Flash Player 4, el cual reproducía contenidos desarrollados con ActionScript 1. Teniendo en cuenta que Flash Lite llegó al mercado cuando el Flash Player iba por su versión 7, esto supuso una involución para los desarrolladores otrora más que acostumbrados a la sintaxis de ActionScript 2. En Macromedia Adobe se dieron cuenta de esta molesta circunstancia y en menos de dos años estaba disponible Flash Lite 2.0. Desde esta versión, Flash Lite es básicamente una adaptación del Flash Player 7 para ser usada específicamente en teléfonos y otros dispositivos móviles. No incluye todas las funcionalidades del Flash Player 7 (como por ejemplo acceder de forma nativa a una cámara) pero dispone de otras que sólo pueden emplearse en este tipo de dispositivos (véase el envío de SMS). La nueva versión de Flash Lite (la 3) está basada en Flash Player 8.

Las características más importantes de Flash Lite, las que han impulsado a este software a la categoría de herramienta para desarrollar aplicaciones, son las que a continuación se comentan.



## Introducción



MUCOM 2009-2010

- **Formato de aplicaciones**

SWF nativo (por defecto)

Empaquetado (ej. SIS en Symbian)

- **Canales de distribución**

Bluetooth

Cable

Tarjetas SD/MMC

OTA (Over The Air)

Infrarrojos

WiFi

...

INTERFACES GRÁFICAS MULTIMEDIA



El formato de las aplicaciones creadas con Flash Lite es el SWF, aunque es posible empaquetarlas como programas nativos del sistema operativo en el que se ejecutan (ej. SIS en Symbian o APP en iPhone).

Para distribuir las aplicaciones existen numerosas formas, si bien las únicas que permitirán monetizar nuestro trabajo son las que empleen los canales de los agregadores de contenidos (ej. Android Market, OTA).



## Contenido



MUCOM 2009-2010

- Introducción
- **Características de Flash Lite 2.x ►**
- Características de Flash Lite 3.x
- Interactividad y navegación
- Texto
- Multimedia
- Comandos del dispositivo
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





## Características de Flash Lite 2.x



MUCOM 2009-2010

- **ActionScript 2**  
Compatibilidad con AS1 (muchos dispositivos llevan aún preinstalado Flash Lite 1.1).
- **Soporte para vídeo**  
Sólo formatos soportados por el dispositivo.  
Puede incrustarse en la aplicación o cargarse externamente.
- **Imagen y sonido**  
Formatos del dispositivo.  
No se admite streaming de audio.
- **Almacenamiento persistente de datos**  
Shared Objects ≈ Cookies.

INTERFACES GRÁFICAS MULTIMEDIA



### **ActionScript**

Con Flash Lite 2.0 podemos emplear la sintaxis de ActionScript 1 o 2 indistintamente, aunque evidentemente es recomendable usar la segunda por motivos de facilidad de uso, mejor gestión de errores y rendimiento de nuestra aplicación (aunque este último motivo es discutible).

### **Soporte para vídeo**

Es posible reproducir cualquier formato de vídeo que esté soportado nativamente por el dispositivo. Es el propio terminal móvil el que decodifica el vídeo y lo renderiza en el reproductor Flash. Tenemos dos posibilidades para usar vídeo en las aplicaciones: incluyéndolo como un dato más de la aplicación (con lo que el tamaño final de la misma aumentará sensiblemente) o cargándolo externamente (ej. desde la tarjeta de memoria o desde la red). Los protocolos externos admitidos son HTTP y RTSP.

### **Imagen y sonido**

Se puede cargar cualquier imagen y/o sonido que esté soportado nativamente por el dispositivo. Los ficheros de audio externos deben cargarse totalmente en memoria antes de poder ser reproducidos.

### **Almacenamiento de datos**

Para guardar datos en el dispositivo Flash Lite emplea los Shared Objects, unos ficheros de texto plano similares a las cookies de los navegadores de Internet. La clase `SharedObject` permite leer y escribir en este tipo de ficheros, y posee ciertas restricciones como por ejemplo no poder ser compartido con otros SWF y un tamaño máximo del fichero – dependiendo de la marca y modelo de dispositivo.



## Características de Flash Lite 2.x



MUCOM 2009-2010

- **Sincronización de sonidos del dispositivo**  
Animaciones con sonidos externos (`_forceframerate`).
- **XML nativo**  
Clases `XML` y `XMLNode`, modeladas según DOM Level 1.
- **Texto mejorado**  
Codificación Unicode – un código único para cada carácter.  
Recomendación para intercambio de ficheros de texto externos: UTF-8.  
Algunas etiquetas HTML permitidas (muchas son obsoletas).

INTERFACES GRÁFICAS MULTIMEDIA



### Sincronización de sonidos del dispositivo

Antes sólo era posible sincronizar el audio nativo de Flash con las animaciones de la línea de tiempo, puesto que los sonidos específicos del dispositivo son reproducidos por él y no por Flash Lite. Ahora disponemos de la propiedad global `_forceframerate` con la cual podremos saltar fotogramas para mantener sincronizada una animación con un sonido del dispositivo (por ejemplo MIDI o SMAF).

### Procesado de XML

En FL 1 y 1.1 no había soporte para archivos XML, por lo que había que trabajar con ficheros de texto plano que sólo albergaban pares de `variable=valor`, sin ningún tipo de estructura. En ocasiones es cómodo trabajar así, pero en aplicaciones más complejas puede ser inabordable. En FL 2 tenemos las clases `XML` y `XMLNode`, modeladas según la recomendación del W3C DOM Level 1.

### Soporte mejorado para texto

Ahora el texto es codificado como Unicode: cada carácter tiene asignado un único código. Así, los caracteres se representan de forma abstracta, y deja la representación visual (tamaño, dimensión, fuente o estilo) a merced del propio dispositivo. Unicode define dos métodos de mapeo o de localización de caracteres: La codificación UTF (Unicode Transformation Format) y la codificación UCS (Universal Character Set). Se recomienda emplear la codificación estándar para el intercambio de texto en Unicode: el UTF-8 (utiliza de 1 a 4 bytes por cada punto de código, siendo relativamente compacto y compatible con ASCII).

Para dar formato al texto (alineación, tipografía, etc.) podemos emplear etiquetas HTML o la clase `TextFormat`. Las hojas de estilo (CSS) sólo están soportadas en FL 3.1.



## Características de Flash Lite 2.x



MUCOM 2009-2010

- **Soporte para teclas adicionales**

Flash Lite 2.0 funciona con teclados QWERTY y hasta 12 Soft keys.

- **En Flash Lite 2.1...**

Textos de entrada (input) editables directamente desde la aplicación.

Clase `XMLSocket` para comunicaciones – conexión persistente con el servidor.

Herramientas de desarrollo para la plataforma BREW (USA).

INTERFACES GRÁFICAS MULTIMEDIA



### **Soporte para teclas adicionales**

Las teclas de navegación estándar en todo dispositivo móvil son las numéricas, las de dirección y las soft keys. Flash Lite 2.0 soporta teclados QWERTY y hasta 12 soft keys.

### **Características de Flash Lite 2.1**

La más interesante desde el punto de vista de la interfaz es que los textos de entrada (input) pueden ser editados directamente en la aplicación y no a través del diálogo genérico del dispositivo.

Sin embargo la gran novedad es que se pueden emplear sockets (comunicaciones persistentes entre cliente y servidor), a través de la clase `XMLSocket`.

Se incluyen herramientas para desarrollar aplicaciones en la plataforma BREW (Binary Runtime Environment for Wireless) de Qualcomm Inc., muy extendida en Estados Unidos.



## Características de Flash Lite 2.x



MUCOM 2009-2010

- **Requisitos de hardware**

Conforme avanza Flash Lite, los requerimientos suelen ser más exigentes.

Espacio libre para instalar el reproductor	450 KB
CPU	200 MHz, 32 bit data bus
Memoria RAM	128 KB
Size-to-Heap ratio	1:15

Fuente: <http://www.adobe.com/products/flashlite/productinfo/overview/datasheet.pdf>

INTERFACES GRÁFICAS MULTIMEDIA



### Requisitos del dispositivo

Para poder ejecutar contenido Flash Lite en un dispositivo móvil, éste debe cumplir al menos estas características:

- 450 KB de espacio libre para instalar el reproductor
- CPU a 200 MHz, 32 bit data bus
- 128 KB de memoria RAM
- Size-to-Heap ratio de 1:15 (caso más desfavorable)

Con Flash Lite 1 era necesario un procesador ARM de tan solo 50 MIPS (algo más de 30 MHz), pero conforme se ha ido mejorando el reproductor Flash, la CPU se ha vuelto más exigente.

El Size-to-Heap ratio se refiere al consumo aproximado de memoria RAM. Por ejemplo, para ejecutar una aplicación que ocupa en disco 100 KB, se estima que se podrían ser necesarios 1.5 MB de RAM en el peor de los casos.



## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- **Características de Flash Lite 3.x ►**
- Interactividad y navegación
- Texto
- Multimedia
- Comandos del dispositivo
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





## Características de Flash Lite 3.x



MUCOM 2009-2010

- **Requisitos del dispositivo**

Mismo hardware que para Flash Lite 2.

Sin embargo, no soportado en teléfonos "antiguos" (ej: Symbian S60 2<sup>rd</sup> Edition).

- **Novedades**

Basado en Flash Player 8 (no incluye todo el API).

Soporte de vídeo nativo de Flash (FLV): rotaciones, overlays, etc.

Mayor rendimiento (20%) y renderizado (30%) que Flash Lite 2.

- **En Flash Lite 3.1...**

Soporte de vídeo H.264.

Aceleración por hardware.

Conexiones con Flash Media Server.

INTERFACES GRÁFICAS MULTIMEDIA



### Características de Flash Lite 3

Los requisitos de hardware son los mismos que FL 2, si bien por ahora sólo está disponible para dispositivos de la plataforma Symbian S60 3rd edition.

La gran novedad es que incluye soporte para vídeo nativo de Flash (formato FLV), con lo que es posible aplicar todo tipo de efectos sobre los objetos de vídeo. En FL 2 los vídeos son renderizados en una capa por encima de la aplicación, y son gestionados por el propio dispositivo.

Al estar basado en el Flash Player 8, es posible visualizar gran cantidad de sitios web, siempre que el dispositivo disponga de un Flash Player en su navegador web, que prácticamente son todos los dispositivos Symbian S60 3rd edition.

El rendimiento y la potencia de renderizado han sido ligeramente optimizados con respecto a FL 2.



## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- **Interactividad y navegación ►**
- Texto
- Multimedia
- Comandos del dispositivo
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





## Interactividad y navegación



MUCOM 2009-2010

- **Modos de interacción**  
Teclado, stylus, pantallas táctiles y similares.
- **Navegación por defecto: Teclado**  
Basada en botones, campos de texto, MovieClips y eventos.



INTERFACES GRÁFICAS MULTIMEDIA



Toda aplicación interactiva debe permitir al usuario determinar qué objeto está seleccionado (*focus*) en pantalla, navegar entre ellos y realizar acciones. Las aplicaciones realizadas con Flash Lite pueden interactuar con el usuario a través del teclado, stylus (a fin de cuentas se trata de un ratón) o pantallas táctiles. En este curso nos centraremos en la interacción por defecto, que es la que se realiza por medio del teclado. La navegación por defecto se basa en botones, campos de texto, MovieClips y eventos principalmente – aunque ya hemos visto que las posibilidades son muchísimo más amplias.



## Interactividad y navegación



MUCOM 2009-2010

- **Focus**

Es el indicador del elemento interactivo que está seleccionado.  
Por defecto es una caja de inclusión (*bounding box*) de color amarillo.



INTERFACES GRÁFICAS MULTIMEDIA



### Focus

Por defecto, Flash Lite dibuja un rectángulo amarillo alrededor de los objetos que permiten interacción. Este rectángulo se llama *caja de inclusión (bounding box)* y su propósito es mostrar claramente lo que está seleccionado; sólo aparece cuando se selecciona un elemento.

Por supuesto, este rectángulo puede desactivarse o configurarse mediante ActionScript, tanto de forma global (mediante `fscommand2`) como para un objeto en particular (mediante la propiedad `_focusrect`).



- **Teclas de navegación**

Estándar: numéricas, de dirección y soft keys.



### Teclas de navegación

Además del teclado numérico de todos los teléfonos (números del 0 al 9, \* y #), prácticamente todos los dispositivos móviles poseen un teclado de navegación (*navigation keypad*) que les permite seleccionar los ítems de la pantalla y dos (o más) *soft keys*, unas teclas configurables por el usuario.

El típico teclado de navegación tiene las cuatro teclas de dirección (arriba, abajo, izquierda y derecha) y una de selección, normalmente situada en el centro de las teclas de dirección.



## Interactividad y navegación



MUCOM 2009-2010

- **Soft keys**

Totalmente personalizables mediante el API `fscommand2`.

Constantes: `ExtendedKey.SOFT1` y `ExtendedKey.SOFT2`



INTERFACES GRÁFICAS MULTIMEDIA



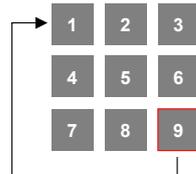
### Soft keys

Para asignar un nombre a las soft keys del dispositivo móvil hay que usar el comando `SetSoftKeys` mediante `fscommand2`. Si la aplicación está ejecutándose a pantalla completa no podremos ver el nombre que hemos asignado a las soft keys. En ese caso tendremos que crear dos campos de texto que las representen, informando así al usuario. En ActionScript podemos usar las constantes `ExtendedKey.SOFT1` y `ExtendedKey.SOFT2` para interactuar con las soft keys izquierda y derecha respectivamente. A la hora de probar la aplicación en el emulador de dispositivos, las teclas `RePág` y `AvPág` se corresponden con `SOFT1` y `SOFT2` respectivamente.

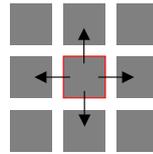


## • Modos de navegación

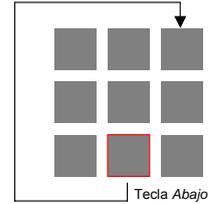
Two-way, four-way y four-way with wraparound.



Two-way  
(arriba/abajo)



Four-way  
(4 direcciones)



Four-way with wraparound  
(solución híbrida)

### Modos de navegación

En las aplicaciones de escritorio el focus entre objetos se realiza mediante las teclas Tab y Mayúscula+Tab. Este focus solamente existe si el objeto permite interacción. Así, un simple logotipo no podrá ser seleccionado a menos que se le asigne algún evento. En los dispositivos móviles el focus se aplica mediante el teclado de navegación.

Flash Lite soporta tres modos de navegación:

- **two-way** Sólo emplea dos teclas (arriba y abajo), de forma análoga a la navegación en escritorio mediante las teclas Mayúscula+Tab y Tab respectivamente. El orden de navegación por defecto es de izquierda a derecha y de arriba a abajo, y por supuesto se puede configurar mediante ActionScript. Cuando el último elemento de la pantalla tiene el focus, al pulsar la tecla *abajo* el focus pasa al primero de los elementos.
- **four-way** Es similar al uso de las teclas de dirección en el teclado de escritorio. Si no existe ningún objeto en la dirección de la tecla pulsada, el focus no cambia.
- **four-way with wraparound** Es una combinación de los dos modos de navegación anteriores. Si no existe ningún objeto en la dirección de la tecla pulsada, el focus cambia al siguiente elemento en el orden lógico de navegación.



- **Recomendaciones de usabilidad**

Tener en pantalla al menos dos objetos que puedan recibir focus.

Si desactivamos el focus debemos incluir pistas visuales para los elementos interactivos.

Orden de navegación (precedencia) controlable mediante `tabIndex`.

Ojo con las tipografías!

### **Recomendaciones de usabilidad**

Es aconsejable tener al mismo tiempo en pantalla dos o más elementos que puedan recibir focus. Si solamente existe un objeto con focus el usuario puede sentirse bloqueado en la interfaz gráfica de la aplicación. De igual forma, si nuestra pantalla solamente tiene un botón es aconsejable emplear entonces la pulsación de una tecla.

Si se desactiva el rectángulo de focus, es conveniente dotar de algún signo visual de focus a los elementos que posean alguna interacción. En los botones esto se realiza de forma automática al definir el estado *over*. En textos de entrada o MovieClips bastan unas pocas líneas de código para añadir esta usabilidad a la aplicación.

Podemos asignar el focus inicial de la interfaz a un determinado objeto para guiar al usuario y minimizar así el número de teclas pulsadas. Si además reforzamos el guiado por la aplicación con algo de texto explicativo, la experiencia de uso mejorará notablemente.

El alineado de los objetos en la pantalla es importante. Si estos se encuentran muy desordenados el usuario no podrá intuir correctamente cuál es el orden de precedencia (*tabIndex*) del focus. Para facilitar la tarea podemos incluir pistas visuales.



- **Eventos de teclado**

Dos tipos: al pulsar una tecla y al soltarla.

Tecla del dispositivo		Código AS y Key (ASCII)	
Selección		Key.ENTER (13)	
Arriba		Key.UP (14)	
Abajo		Key.DOWN (15)	
Izquierda		Key.LEFT (1)	
Derecha		Key.RIGHT (2)	
Soft key izquierda		ExtendedKey.SOFT1 (0)	
Soft key derecha		ExtendedKey.SOFT2 (0)	
0	6	48 (48)	54 (54)
1	7	49 (49)	55 (55)
2	8	50 (50)	56 (56)
3	9	51 (51)	57 (57)
4	*	52 (52)	56 (42)
5	#	53 (53)	51 (35)

### Eventos de teclado

Los dos eventos que podemos emplear al trabajar con la clase `Key` son `onKeyDown` (al pulsar una tecla) y `onKeyUp` (al liberarla). Cada dispositivo móvil responde a un determinado tipo de teclas, si bien de forma estándar existen unos códigos que a continuación se presentan en esta tabla.



## • Eventos de botones

onPress	onRelease	onReleaseOutside	onRollOver
onRollOut	onSetFocus	onKillFocus	onKeyUp
onKeyDown	onDragOver*	onDragOut*	

## • Eventos de ratón

onMouseMove*	onMouseDown*	onMouseUp*	
--------------	--------------	------------	--

\* Sólo si el dispositivo tiene ratón o *stylus* (`System.capabilities.hasMouse == true`)

### Eventos de botones

Flash Lite soporta los mismos eventos de teclado que la versión de escritorio Flash, pero algunos de ellos sólo están disponibles para dispositivos que tengan *stylus* o ratón, como por ejemplo `onDragOut`. A modo de recordatorio, estos son los eventos que se pueden escuchar de un botón:

- **onPress** Se pulsa la tecla de Selección, cuando un botón tiene focus.
- **onRelease** Se deja de pulsar la tecla de Selección, cuando un botón tiene focus.
- **onReleaseOutside** Se libera el *stylus* fuera del botón que se había pulsado antes.
- **onRollOver** Invocado cuando se pasa por encima del botón con focus.
- **onRollOut** Invocado cuando se sale del botón con focus.
- **onDragOver** Se pulsa el *stylus* fuera del botón y se pasa por encima del mismo.
- **onDragOut** Se pulsa el *stylus* sobre el botón, y se libera fuera del mismo.
- **onKeyUp** Cuando se libera una tecla sobre el botón que tiene focus.
- **onKeyDown** Cuando se pulsa una tecla sobre el botón que tiene focus.
- **onSetFocus** Cuando un objeto recibe focus.
- **onKillFocus** Cuando se elimina el focus de un objeto.



## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- Interactividad y navegación
- **Texto** ►
- Multimedia
- Comandos del dispositivo
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





## Texto



MUCOM 2009-2010

- **Generalidades**

Tres tipos de texto: estáticos, dinámicos e inputs (de entrada).  
Uso de fuentes del dispositivo vs. tipografías comerciales.  
Codificación Unicode.  
Formato mediante etiquetas HTML (básicas, sin CSS) o la clase `TextFormat`.  
Muy dependiente de los PPI (*Pixels Per Inch*) de la pantalla.

INTERFACES GRÁFICAS MULTIMEDIA



### Generalidades

Existen tres tipos de texto en Flash Lite: estáticos, dinámicos y de entrada (*input*). Los textos estáticos son los que no se modifican durante el transcurso de la aplicación. Los textos dinámicos pueden modificar su contenido mediante ActionScript. Los textos del tipo input se utilizan para recibir información a través del teclado del dispositivo.

Podemos escribir texto usando las fuentes del dispositivo o incrustar nuestras propias tipografías. En el primer caso el texto no se suaviza, lo cual es recomendable para facilitar la legibilidad de largos párrafos. En el segundo caso podemos decidir si suavizar o no la fuente tipográfica. Las fuentes del dispositivo no pueden usarse como máscara ni enmascararse o cambiar su opacidad.

El texto en FL 2.x es Unicode, lo cual permite representar cualquier idioma si la tipografía escogida posee los caracteres correspondientes.

Se puede dar formato usando etiquetas HTML o mediante las propiedades de la clase `TextFormat` con ciertas limitaciones. En el primer caso sólo podemos modificar el tipo de fuente (*face*), color (*color*), tamaño (*size*), negrita (*b*) o cursiva (*i*), y emplear las etiquetas de párrafo (*p*) y salto de línea (*br*).



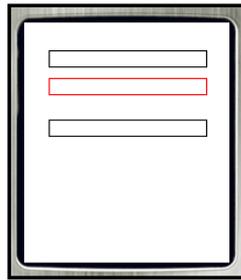
## Texto



MUCOM 2009-2010

- **Textos de entrada**

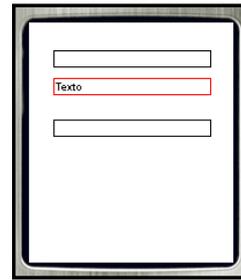
En FL <= 2 aparece el cuadro de diálogo genérico del dispositivo al introducir datos.  
A partir de FL 2.1 se hace directamente en la aplicación (*inline text input*).



Texto de entrada con Focus



Cuadro de diálogo genérico



Datos introducidos

INTERFACES GRÁFICAS MULTIMEDIA



### Textos de entrada

En FL 2.1 se puede modificar el contenido de un texto de tipo input directamente sobre la aplicación. En dispositivos que tengan instalada una versión anterior de Flash Lite, al modificar el contenido de un texto de entrada, aparecerá el cuadro de diálogo genérico del dispositivo y la aplicación se detendrá hasta que se cierre dicho cuadro de diálogo — bien pulsando la soft key izquierda (*Ok* por defecto) o la derecha (*Cancel* por defecto).

Para poder escribir en un texto de entrada es necesario que dicho texto reciba focus y se pulse la tecla de Selección.



## Texto

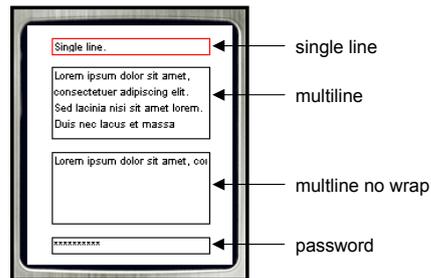


MUCOM 2009-2010

- **Textos de entrada**

Tipos: *single line*, *multiline*, *multiline no wrap*, *password*.

Es posible restringir el tipo de datos a introducir y el número de caracteres.



INTERFACES GRÁFICAS MULTIMEDIA



### Textos de entrada

Este tipo de texto, al no ser estático, puede configurarse mediante estas opciones:

- **Single line** El texto sólo puede ocupar una línea.
- **Multiline** El texto puede ocupar tantas líneas como queramos.
- **Multiline no wrap** El texto ocupa una sola línea a menos que haya un retorno de carro.
- **Password** Los caracteres se muestran como asteriscos mientras se escribe.

Se puede limitar el número de caracteres introducidos y mostrar o no un borde alrededor del cuadro de texto. Otra característica es poder restringir el tipo de datos a escribir en un texto de entrada (mediante el comando `SetInputTextType` o la propiedad `restrict`), aunque siempre es recomendable escribir algo de código para validación.

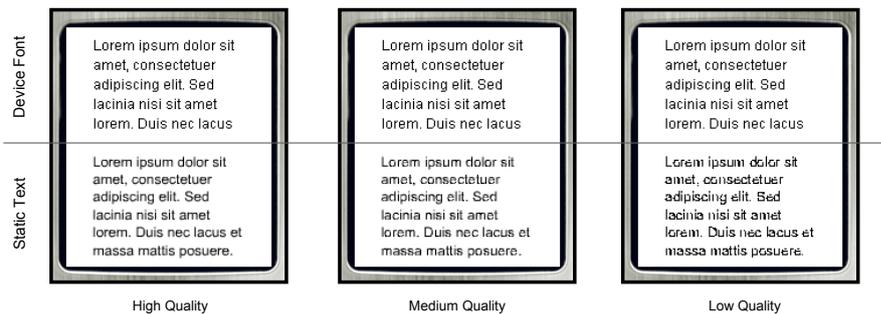


## • Renderizado del texto

Las fuentes del dispositivo (`_sans`, `_serif` y `_typewriter`) no se suavizan.

*Bitmap text* (no anti-alias) vs. *Antialias for animation*.

Tres niveles de calidad global: HIGH, MEDIUM y LOW.



### Renderizado del texto

Ya adelantamos anteriormente que podemos escribir texto usando las fuentes del dispositivo o incrustar nuestras propias tipografías. En el primer caso podemos escoger entre tres tipos distintos de fuentes genéricas: `_serif` (con remates), `_sans` (sin remates) y `_typewriter` (monoespaciadas de ancho fijo, como las antiguas máquinas de escribir). Flash Lite asociará automáticamente uno de estos tres tipos genéricos de tipografía con la más adecuada del dispositivo. Por ejemplo, la fuente `_sans` se corresponde con la sans serif, arial, helvetica y similares.

Para no suavizar una tipografía personalizada deberemos escoger la opción de *bitmap text* (no *anti-alias*). Esto permite la legibilidad de largos párrafos de texto y sobre todo con tamaños de letra pequeños (9 puntos o menos). Si suavizamos el texto con la opción *antialias for animation* el texto se representa como vectores, consumiendo más memoria, porque hay que crear los caracteres a incluir, y emborronando los tamaños de letra pequeños. Podemos decidir qué caracteres incrustar para reducir el tamaño de la aplicación.

Flash Lite admite tres niveles de calidad de renderizado: LOW, MEDIUM y HIGH. Los movieClips disponen además de un cuarto nivel de calidad: BEST. La propiedad `_quality` es aplicada de forma local, mientras que el comando `SetQuality` afecta de forma global. A mayor calidad, mejor suavizado de los gráficos vectoriales y mayor tiempo de procesado. Por defecto, el nivel de calidad es el medio, pero es fácilmente configurable mediante ActionScript. Podemos ir alternando la calidad de renderizado temporalmente, en función de lo cargada (computacionalmente hablando) que esté nuestra aplicación — por ejemplo durante una animación muy compleja.

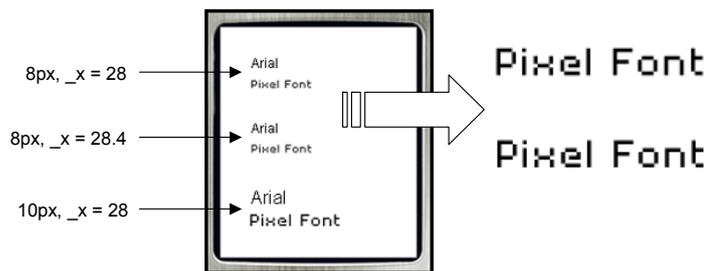


- **Pixel fonts**

Hechas únicamente con pixels blancos y negros, y sin suavizar.

Están diseñadas para un tamaño de fuente determinado.

Hay que ubicar los campos de texto en valores enteros de coordenadas (si no, empeora la legibilidad).



### Pixel fonts

Son un tipo de fuentes tipográficas que están hechas únicamente con pixels blancos y negros, y sin suavizar. La diferencia con la opción *bitmap text* de Flash es que las pixel fonts sólo tienen un tamaño adecuado para los textos (normalmente 8 puntos).

Cuando trabajemos con alguna pixel font hay que procurar ubicar el texto en valores enteros de coordenadas. Si las coordenadas del texto son números decimales, el motor de renderizado puede hacerlas borrosas.



## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- Interactividad y navegación
- Texto
- **Multimedia** ►
- Comandos del dispositivo
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





## Multimedia



MUCOM 2009-2010

- **Sonido: del IDE Flash o del dispositivo**  
Son gestionados por Flash Player o por el propio dispositivo móvil, respectivamente.
- **Vídeo: gestionado por el dispositivo**  
Sólo pueden reproducirse los formatos que reconozca el terminal.  
Flash Lite 3 acepta archivos FLV.
- **Imagen: ficheros externos o en la aplicación**  
Procurar ficheros externos para reducir el tamaño de la aplicación y su mantenimiento.

INTERFACES GRÁFICAS MULTIMEDIA



Flash Lite puede trabajar con dos tipos de sonido: nativos del IDE o propios del dispositivo. Los primeros son reproducidos directamente por el Flash Player, mientras que los segundos los reproduce el mismo dispositivo móvil. La ventaja de trabajar con sonidos del IDE Flash es que van a escucharse en cualquier aparato, pues son gestionados por el Flash Player. La ventaja de trabajar con sonidos del dispositivo es que podemos emplear otro(s) formato(s) de mayor calidad, pero algunos aparatos no podrán reproducirlo(s).

A la hora de incluir vídeo en las aplicaciones Flash Lite, el contenido es gestionado por el propio dispositivo. Esto es así en aras de mantener los ficheros SWF lo más pequeños posible. Debemos trabajar entonces con formatos de vídeo estándar, tales como 3GP o MPEG-4, o de lo contrario no todos los dispositivos podrán visualizar el vídeo de nuestra aplicación.

Con las imágenes ocurre exactamente lo mismo que con el sonido: podemos incluir las imágenes directamente en nuestra aplicación – aumentando con ello el tamaño del archivo SWF – o cargarlas externamente. Cuando elijamos esta segunda opción, debemos conocer qué formatos de imagen puede decodificar el dispositivo. Para ello se recomienda usar la propiedad `capabilities.imageMIMETypes` de la clase `System`.



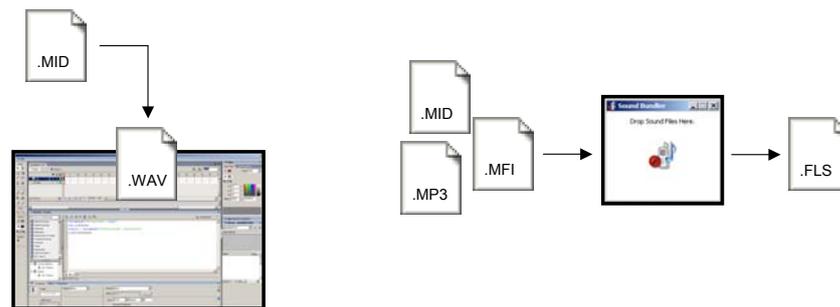
- **Sonidos del dispositivo**

*Proxy sound*: enlace de un formato que reconoce Flash con el original del dispositivo.

Encapsulación de varios formatos en un solo archivo FLS.

Clase `SoundObject`. Streaming de audio no soportado.

Sincronización con la línea de tiempo mediante `_forceframerate`.



INTERFACES GRÁFICAS MULTIMEDIA



## Sonidos del dispositivo

Para incluir un sonido nativo del dispositivo móvil en nuestra aplicación, tenemos que importar al IDE un sonido que Flash reconozca (WAV, MP3 o AIFF) y enlazarlo con el sonido del dispositivo (MIDI, OGG, etc). Este método se conoce como *proxy sound*. A la hora de publicar la aplicación, el sonido que reconocía el IDE se reemplaza por el que reconoce el dispositivo. Durante la ejecución de la aplicación, es el dispositivo móvil el que decodifica el sonido.

Existe la posibilidad de encapsular varios formatos de audio (del mismo sonido) en un único fichero con la extensión FLS. De esta forma, cada dispositivo reproducirá el formato que pueda decodificar dentro de ese paquete de ficheros de audio. Para encapsular distintos formatos de audio podemos usar la aplicación Sound Boundler (de momento sólo disponible en entornos Windows).

Para cargar ficheros de audio externos mediante ActionScript emplearemos la clase `SoundObject`. Estos sonidos del dispositivo que se cargan externamente no pueden reproducirse a medida que se van descargando (streaming). Todos los sonidos en Flash Lite son tratados como *event sounds*, esto es, se reproducen cuando se han descargado completamente. Por esta razón no podemos sincronizar, estrictamente hablando, una animación con un sonido del dispositivo. Empleando la propiedad global `_forceframerate`, el Flash Player se salta fotogramas con tal de mantener el frame rate que hayamos definido, por lo que es una aproximación casi perfecta para sincronizar sonidos del dispositivo con la línea de tiempo de la aplicación.



- **Sonidos del IDE Flash**

Formatos: WAV, AIFF o MP3.

Reproducibles en todos los dispositivos móviles.

Sincronizables con la línea de tiempo.

Desventaja: el tamaño de la aplicación aumenta.

- **Frecuencias de muestreo**

5, 11, 22 y 44 KHz por defecto.

Algunos terminales soportan 8 y 16 KHz.

### **Sonidos del IDE Flash**

La gran ventaja de trabajar con sonidos nativos de Flash Lite es que, además de que podrán ser escuchados en cualquier dispositivo móvil, pueden funcionar como *event sound* y/o sincronizarse con la línea de tiempo. La desventaja es que el tamaño de la aplicación aumenta.

Las frecuencias de muestreo que exporta Flash Lite son 5, 11, 22 o 44 KHz. Sin embargo, muchos dispositivos móviles no soportan estos ratios. Por ejemplo, los teléfonos Nokia de la Serie 60 sólo pueden reproducir sonidos a 8 o a 16 KHz. En Flash 8 podemos habilitar esta frecuencia de muestreo para todos los sonidos de nuestra aplicación o para uno en concreto.



## • Vídeos del dispositivo

Formato estándar en teléfonos móviles: 3GPP (video H.263 y audio AMR).

Codecs en otros dispositivos móviles: MPEG (1, 2 y 4), H.264, WMV, RM.

Propiedad `System.capabilities.videoMIMETypes` para ver los formatos soportados.

Pueden incluirse en la aplicación o cargarse externamente (mejor URLs relativas):

Notación	Significado
<code>./</code>	Directorio actual.
<code>../</code>	Directorio padre.
<code>/</code>	Acceso a directorios hijo.

### Vídeos del dispositivo

Para incluir vídeo ya vimos que es el propio dispositivo el que se encarga de decodificarlo. Desde la versión 2 de Flash Lite, es posible controlar el vídeo mediante ActionScript e incluso encapsularlo en la aplicación. Esta última opción implica un notable incremento en el tamaño del fichero SWF, pero es la opción que ofrece la mejor portabilidad y reproducción entre dispositivos, porque no se usan ficheros externos. Existe la posibilidad de crear un contenedor *video object* al que asignar la ejecución de varios vídeos de la Biblioteca.

Si no lo hemos hecho previamente, podemos codificar nuestro vídeo en uno de los formatos estándar para aplicaciones móviles mediante el asistente del IDE Flash.

Para cargar el vídeo desde la red o desde la tarjeta de memoria del dispositivo hay que indicar la URL absoluta o relativa. No se recomienda emplear el protocolo `file://` para cargar vídeos porque no todos los dispositivos móviles lo soportan. En su lugar es más adecuado trabajar con paths relativos, mediante esta nomenclatura:

- `./` para referirse al directorio actual
- `../` para el directorio padre (encima del actual)
- `/` para acceder a un directorio por debajo del actual

Para determinar qué formatos de vídeo soporta un dispositivo en concreto, podemos usar la propiedad `System.capabilities.videoMIMETypes`, la cual devuelve una matriz con el formato `video/type` (por ejemplo, `video/3gpp`).



- **Limitaciones de vídeo del dispositivo**

Rotaciones, distorsiones y, a veces, escalado.

No sincronizable con la línea de tiempo.

No se puede manipular directamente el volumen de audio.

Se renderiza en una capa por encima de la aplicación.

En FL 3 existe el formato FLV.

### **Limitaciones de vídeo del dispositivo**

Debido a que la inclusión de vídeo en Flash Lite sólo fue posible a partir de la versión 2, existen ciertas limitaciones que habrá que tener en cuenta a la hora de desarrollar la aplicación:

- No podemos rotar o distorsionar el vídeo, ya que estas operaciones de transformación son muy intensivas. No todos los dispositivos móviles soportan el escalado del vídeo.
- No podemos sincronizar vídeo con la línea de tiempo ni controlar el volumen del audio.
- El vídeo, al estar gestionado por el dispositivo, se renderiza directamente encima de nuestra aplicación. Por ello, no podemos mezclarlo ni componerlo con ninguna imagen o texto de la aplicación.

Sin embargo, desde FL 3 es posible trabajar con el formato nativo de Flash Video (FLV), el cual permite un control total sobre los vídeos (rotaciones, overlays, etc.).



- **Imágenes del dispositivo**

Propiedad `System.capabilities.imageMIMETypes` para ver los formatos soportados.  
Carga externa mediante `loadMovie()`.

### **Imágenes del dispositivo**

Desde FL 1.1, es posible cargar ficheros SWF externos mediante la función `loadMovie()`. En FL 2 podemos emplear esta función para cargar imágenes soportadas por el dispositivo. La propiedad `System.capabilities.imageMIMETypes` devuelve una matriz con los formatos de imagen que puede visualizar el dispositivo, en la forma `image/type` (por ejemplo, `image/png`).



- **Limitaciones con imágenes**

Como máximo cinco funciones `loadMovie()` en el mismo fotograma.

Diez operaciones simultáneas.

Soluciones:

- dividir el código en fotogramas
- usar intervalos de ejecución
- emplear clases creadas para este fin

### **Limitaciones con imágenes**

Por cuestiones de rendimiento y optimización de la aplicación, Flash Lite solamente puede contener cinco llamadas a la función `loadMovie()` en el mismo fotograma, y ejecutar diez operaciones al mismo tiempo. Para evitar estas limitaciones podemos dividir el código en distintos fotogramas, emplear intervalos de ejecución mediante el método `setInterval()`, o usar clases especializadas que abstraen este proceso (ej. el framework Feather).



## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- Interactividad y navegación
- Texto
- Multimedia
- **Comandos del dispositivo** ►
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





## Comandos del dispositivo



MUCOM 2009-2010

- API `fscommand2`

Permite interactuar con características del dispositivo: cobertura, vibración, etc.  
API `fscommand` para ejecutar aplicaciones (admite múltiples argumentos).

- El método `getURL`

Hacer una llamada: `getURL("tel:123456789");`

Mandar un SMS: `getURL("sms:123456789?body=hola, qtal?");`

→ Enviar un MMS: `getURL("mms:123456789?body=hola, qtal?");`

Mandar un email: `getURL("mailto:tal@serv.com?subject=soy yo&body=...");`

No se pueden adjuntar archivos directamente.

INTERFACES GRÁFICAS MULTIMEDIA



### Comandos del dispositivo

Flash Lite dispone de una serie de métodos para comunicarse (de forma limitada) con el hardware del dispositivo.

La función `fscommand` existe también en las versiones de escritorio del Flash Player, pero en Flash Lite tiene la particular novedad de aceptar múltiples argumentos. Y esta es precisamente una base robusta para ejecutar todo tipo de aplicaciones externas, ampliando las funcionalidades de Flash Lite.

Mediante el método `getURL` podemos enviar un mensaje SMS, MMS, email o hacer una llamada de teléfono, previa autorización del terminal – la aplicación solicita confirmación por parte del usuario.



- API fscommand2

FullScreen	GetBatteryLevel	GetTimeZoneOffset
Quit	GetMaxBatteryLevel	GetTotalPlayerMemory
SetQuality	SetSoftKeys	GetFreePlayerMemory
StartVibrate	ResetSoftKeys	GetLanguage
StopVibrate	GetDateDay	GetDeviceID
GetVolumeLevel	GetDateWeekDay	GetPlatform
GetMaxVolumeLevel	GetDateMonth	GetDevice
GetSignalLevel	GetDateYear	GetSoftKeyLocation
GetMaxSignalLevel	GetLocaleShortDate	Launch
GetNetworkConnectStatus	GetLocaleLongDate	Escape
GetNetworkRequestStatus	GetLocaleTime	UnEscape
GetNetworkStatus	GetTimeHours	SetInputTextType
GetNetworkName	GetTimeMinutes	ExtendBackLightDuration
GetPowerSource	GetTimeSeconds	SetFocusRectColor

En esta tabla se recogen las funciones más interesantes del API fscommand2.



## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- Interactividad y navegación
- Texto
- Multimedia
- Comandos del dispositivo
- **Emulador de dispositivos** ►
- Adobe Device Central
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





## Emulador de dispositivos



MUCOM 2009-2010

- Testeo de la aplicación
- En la práctica:

Hay que probar la aplicación físicamente en el dispositivo antes de distribuirla.  
No todas las instrucciones son simulables (ej. `fscommand2`).

INTERFACES GRÁFICAS MULTIMEDIA



El emulador de Flash Lite nos permitirá comprobar el funcionamiento de prácticamente toda nuestra aplicación. Efectivamente, no será hasta que transfiramos esa aplicación al dispositivo móvil cuando podamos ver cómo se comporta realmente nuestro desarrollo; el emulador no puede simular, por ejemplo, la profundidad de color de la pantalla del dispositivo o la velocidad del procesador – aunque en las nuevas versiones del emulador sí que se contempla esta última opción. Por ello, una vez comprobada físicamente la aplicación podremos refinar a posteriori su diseño.

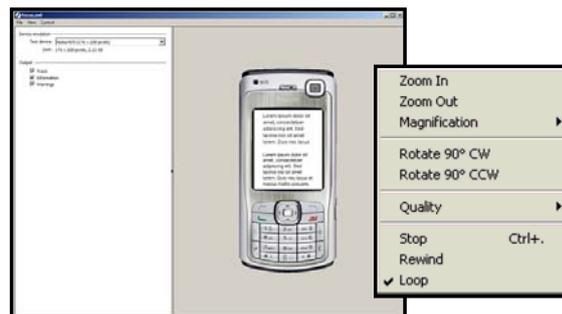


- **Funcionalidades**

Pantalla de configuración: modelo de dispositivo y tipo de contenido. Amplia BBDD.

Pantalla de testeo: advertencias, info del dispositivo, salidas `trace()`.

Interacciones con el emulador: zoom, rotaciones, cambio de calidad, control de la película.



### Funcionalidades

Este emulador de dispositivos consiste en dos partes: la pantalla de configuración y la pantalla de testeo. En la primera indicaremos el (o los) modelo(s) de dispositivo(s) que vamos a probar y el tipo de aplicación que vamos a realizar – salvapantallas, tono de llamada, standalone, etc. Este emulador dispone de una amplia base de datos con los distintos formatos multimedia que soporta cada modelo, comportamientos y capacidades para conectarse a Internet, etc. Por otra parte, la pantalla de testeo nos mostrará el funcionamiento de la aplicación. El panel de Salida nos informa, a medida que ejecutamos la aplicación, de las advertencias, comandos no soportados e información del dispositivo, además de mostrar las líneas de `trace()`.

Podemos rotar la pantalla del dispositivo emulado y hacer varios niveles de zoom sobre la aplicación. Este zoom no tiene nada que ver con el realizado por el Flash Player del dispositivo, mucho más preciso y de mayor calidad.

Dependiendo del tipo y modelo de dispositivo, tendremos que trabajar con un tamaño determinado de escenario en pixels. Si por ejemplo ejecutamos una aplicación realizada para la consola PSP en un teléfono Nokia N70, Flash Lite escalará el contenido para ajustarlo a esta nueva (y menor) resolución de pantalla.



## Emulador de dispositivos



MUCOM 2009-2010

- **Características no simulables**

Comandos `fscommand2` exclusivos de los dispositivos móviles.

- **Características indirectamente simulables**

Listar las variables y objetos de la aplicación a medida que van empleándose.

Tiempo de descarga de la aplicación desde Internet a diferentes tasas de conexión.

Analizar la carga del contenido gráfico en memoria a medida que se ejecuta.

Mostrar las zonas de redibujado de la aplicación.

Control de la película más preciso.

INTERFACES GRÁFICAS MULTIMEDIA



### **Características no simulables**

El emulador de dispositivos no puede simular todos los comandos disponibles en los dispositivos móviles, como por ejemplo hacer llamadas o enviar SMS: otra razón por la cual debemos probar nuestro desarrollo físicamente en el dispositivo. Los comandos que no pueden simularse son los propios de un terminal móvil, tales como devolver el nivel de señal, estado de la batería o activar la vibración.

Sin embargo, existen algunas características que no pueden simularse con el emulador de dispositivos pero sí al publicar los SWF en la versión de escritorio. Estas son:

- Listar las variables y objetos que están utilizándose en la aplicación.
- Simular el tiempo de descarga de la aplicación desde Internet.
- Analizar la carga del contenido gráfico en memoria a medida que se ejecuta.
- Mostrar las zonas de redibujado de la aplicación.
- Controlador la ejecución del SWF como si de una película de vídeo se tratase.



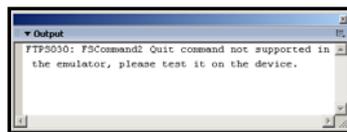
- **Mensajes del emulador**

Aparecen en el panel de Salida: FTPA####, FTPE#### y FTFS####.

- **Mensajes del dispositivo**

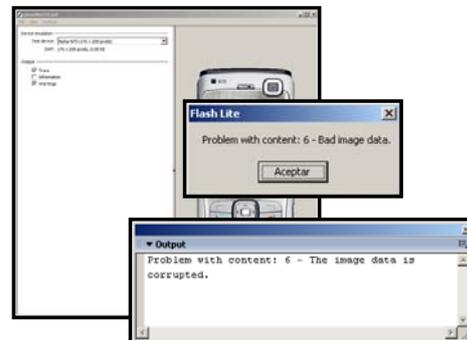
Aparecen en una ventana de diálogo genérica del dispositivo.

También aparecen en el emulador.



Top 3:

- Out Of Memory
- Bad Data
- ActionScript Stuck



### Mensajes de error

Existen dos tipos de mensajes de error: los que ocurren en el emulador y los que ocurren en el dispositivo. Los del primer tipo proporcionan información sobre la aplicación, y aparecen en el panel de Salida. Los del segundo tipo aparecen como una ventana de diálogo con un número que identifica el tipo de error, a saber:

- *Out of memory* (1) – Flash Lite asigna a cada SWF 1 MB de espacio para usar por defecto. Si este límite es superado aparece este error.
- *Stack limit reached* (2) – Se ha llenado la memoria disponible para reproducir la aplicación.
- *Corrupted SWF* (3) – El fichero SWF no es válido.
- *ActionScript stuck* (4) – Ha tardado en ejecutarse demasiado tiempo el código ActionScript, y en consecuencia se detiene la aplicación.
- *Bad image data* (6) – El fichero SWF contiene alguna imagen que no puede decodificar ni Flash Lite ni el dispositivo, o es un archivo dañado.
- *Bad sound data* (7) – El sonido no lo puede decodificar ni Flash Lite ni el dispositivo, o se trata de un archivo dañado.
- *Root movie unloaded* (8) – El SWF de `_level0` ha sido reemplazado por otro fichero SWF.
- *GetURL string too long* (10) – El texto de la dirección a la que se desea acceder tiene demasiados caracteres.
- *Not enough memory to decompress image* (11) – No queda memoria disponible para que la aplicación decodifique una imagen determinada.
- *Bad SVG data* (12) – Se ha intentado cargar un fichero SVG dañado.
- *Stream loading out of memory* (13) – Se ha intentado cargar un fichero muy grande desde Internet y el dispositivo se ha quedado sin memoria.



## Emulador de dispositivos



MUCOM 2009-2010

- **Información del dispositivo**

Genera un informe con las características técnicas.

Más información en la clase `System.capabilities`.

INTERFACES GRÁFICAS MULTIMEDIA



### Información del dispositivo

El emulador de Flash Lite permite listar todas las características del dispositivo contra el que se está probando la aplicación, de acuerdo a su base de datos de dispositivos que soportan Flash Lite. Esta base de datos se va actualizando regularmente, a medida que los nuevos aparatos que soportan Flash Lite van llegando al mercado. Dichos parámetros aparecen en la ventana de Salida al testear la aplicación en el emulador, al seleccionar la opción *Output Information*.



## Emulador de dispositivos



MUCOM 2009-2010

- **Tipos de contenido**

Cada dispositivo permite determinados usos de los ficheros SWF.

Address Book	Alarm	Browser	Calling History
Calling Screen	Chaku Flash	Data Box	Data Folder
Icon Menu	Image Viewer	Incoming Call	Mailer
Multimedia	My Picture	OpenEMIRO	Screen Saver
SMIL Player	<b>Standalone Player</b>	Standby Screen	Sub LCD
UILauncher	Wake Up Screen		

INTERFACES GRÁFICAS MULTIMEDIA



### Tipos de contenido

Cada dispositivo móvil permite varios usos para las aplicaciones realizadas con Flash Lite. En [http://www.adobe.com/mobile/supported\\_devices/](http://www.adobe.com/mobile/supported_devices/) puede consultarse información al respecto. Sin embargo, dado que la frecuencia de actualización de dicha lista no es muy habitual, se recomienda consultar en su lugar el repositorio <http://wurfl.sourceforge.net>.

Los tipos de contenido que pueden realizarse con Flash Lite son los mostrados en esta tabla, si bien los fabricantes pueden extenderlos en un futuro.



## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- Interactividad y navegación
- Texto
- Multimedia
- Comandos del dispositivo
- Emulador de dispositivos
- **Adobe Device Central ►**
- Integración y extensiones

INTERFACES GRÁFICAS MULTIMEDIA





## Adobe Device Central



MUCOM 2009-2010

- **La evolución del emulador de dispositivos**  
Más dispositivos soportados, más especificaciones técnicas.
- **Simulación con más posibilidades**  
Condiciones de uso del terminal, de la red móvil, rendimiento, memoria...
- **Cada dispositivo tiene un perfil asociado**  
Los perfiles de dispositivo son ficheros XML.

INTERFACES GRÁFICAS MULTIMEDIA



### Adobe Device Central

Device Central es la evolución del anterior emulador de dispositivos de Macromedia. Se trata, pues, de una biblioteca de perfiles para una gran variedad de teléfonos móviles y dispositivos electrónicos regularmente actualizada. Estos perfiles cuentan con información más detallada sobre las características técnicas de cada dispositivo. Adobe planea ampliarlos conforme vayan apareciendo nuevos modelos en el mercado mediante actualizaciones.

El flujo de trabajo de creación de contenido móvil se simplifica, pues es posible previsualizar y probar contenidos en un mayor número de dispositivos y diferentes condiciones de uso.

La interacción ofrece más posibilidades, imitando de forma más fidedigna las condiciones del mundo real. Por ejemplo: mostrar información del uso de memoria que hace nuestra aplicación y su rendimiento, modificar el nivel de la batería, la intensidad de la señal, o simular diferentes condiciones de visualización, como un indicador de tiempo de espera retroiluminado o reflejos solares.

Los perfiles de dispositivos se definen mediante archivos XML, por lo que podemos modificarlos a voluntad – por ejemplo, el Nokia N70 no aparecía como capaz de reproducir contenido Flash Lite 2, lo cual es erróneo.



## Contenido



MUCOM 2009-2010

- Introducción
- Características de Flash Lite 2.x
- Características de Flash Lite 3.x
- Interactividad y navegación
- Texto
- Multimedia
- Comandos del dispositivo
- Emulador de dispositivos
- Adobe Device Central
- Integración y extensiones ►

INTERFACES GRÁFICAS MULTIMEDIA





## Integración y extensiones



MUCOM 2009-2010



INTERFACES GRÁFICAS MULTIMEDIA



Vistas las limitadas posibilidades de Flash Lite a la hora de interactuar a bajo nivel con el hardware de los dispositivos, existen numerosa posibilidades para ampliar funcionalidades:

**Kunerilite** – <http://www.kunerilite.net>

API muy completa para acceder a la Cámara, GPS, Acelerómetro...

**SWF2SIS** – <http://www.blueskynorth.com/swf2sis>

**SWF2GO** – <http://www.orison.biz/products/swf2go>

Creación de instaladores, agrupando varios archivos en la misma aplicación.

**Flyer** – <http://www.flyerframework.org>

Framework para ampliar las capacidades de Flash Lite.

**GenialSiS** – <http://www.socci.eu/English>

Firmar SIS y SISX 3rd Edition. Se apoya en .NET 3.5.

**Jarpa** – <http://www.i2tecnologia.com.br/jarpa>

Distribuir contenido Flash Lite en archivos JAR.

**SWXml** – <http://www.swxformat.org>

Parseador genérico de ficheros XML.

**Janus** – <http://www.janus-flash.com>

Creación de instaladores añadiendo funcionalidades a Flash Lite.

**PAMP** – <http://opensource.nokia.com/projects/PAMP>

Personal Apache, MySQL y PHP para Symbian.

**Project Capuchin** – <http://developer.sonyericsson.com>

Emplear el API de J2ME en Flash Lite.

**Layout Manager / Feather Framework** – <http://opensource.blocketpc.com>

Gestión de la diagramación de contenidos. / Conjunto de clases optimizadas.