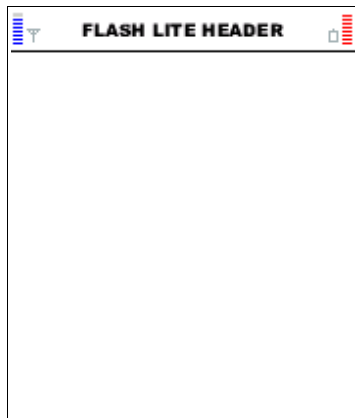


## Niveles de cobertura y batería

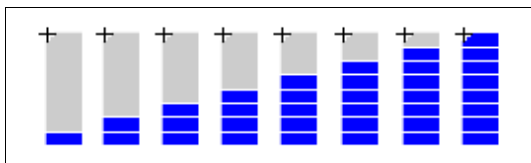
### Indicadores básicos en las aplicaciones

Dado que es vital aprovechar el máximo espacio disponible en la pantalla del dispositivo, prácticamente todas las aplicaciones realizadas en Flash Lite se ejecutan a pantalla completa. En ocasiones además es conveniente mostrar en nuestra aplicación un indicador del nivel de batería y cobertura, manteniendo informado así al usuario.

Se va a realizar una cabecera donde aparecerán estos niveles y ya de paso el nombre de la aplicación, que se mostrará en un campo de texto.



Para ello, creamos un nuevo movieClip (*indicador*) y dibujamos un rectángulo que será el fondo del indicador. A continuación creamos ocho keyframes que van a representar los ocho posibles estados del nivel. Podemos hacer más keyframes si queremos más precisión, pero para esta práctica es más que suficiente. El nivel vendrá representado por un pequeño rectángulo que convertiremos en movieClip (*lcolor*), para cambiarle posteriormente el color mediante ActionScript y poder usar el mismo símbolo tanto para la cobertura como para la batería. A continuación se muestra lo que hay que dibujar en cada uno de los ocho keyframes:



En cada fotograma vamos aumentando la altura del movieClip *lcolor* y vamos dibujando unas líneas blancas para indicar el nivel de carga.

En cada uno de los ocho keyframes debemos añadir esta línea para detener el movieClip:

```
stop();
```

Una vez creado arrastramos el indicador desde la librería hasta el escenario, y le damos el nombre de instancia *signal\_mc*.

Este movieClip debe estar actualizándose constantemente, para reaccionar según el hardware del teléfono, que es el que calcula el nivel de carga de la batería y la cobertura. Eso se hace con el evento *onEnterFrame*:

```
signal_mc.onEnterFrame = function():Void {  
    // detección de cobertura  
};
```

Lo que hará este *callback* es calcular un número comprendido entre 1 y 8 en función del nivel de señal recibido y el nivel máximo de señal que puede recibir el dispositivo, que en cada uno es distinto. Ese número servirá para ir al fotograma del movieClip *signal\_mc*.

```
signal_mc.onEnterFrame = function():Void {  
var aux:Number = Math.ceil(fscommand2 →  
("GetSignalLevel") / fscommand2 →  
("GetMaxSignalLevel") * 8 );  
this.gotoAndStop(aux);  
};
```

Para crear el indicador de batería, arrastramos nuevamente desde la biblioteca el movieClip *indicador* y le damos el nombre *batt\_mc*. El código necesario es similar, añadiendo además la instrucción para colorear de rojo el movieClip *lcolor*, que estaba dentro de *indicador*.

```
batt_mc.onEnterFrame = function():Void {  
var aux:Number = Math.ceil(fscommand2 →  
("GetBatteryLevel") / fscommand2 →  
("GetMaxBatteryLevel") * 8 );  
this.gotoAndStop(aux);  
var pinta:Color = new Color(this.lcolor);  
pinta.setRGB(0xFF0000);  
};
```

Esta aplicación sólo puede comprobarse físicamente en el dispositivo, pero el emulador devuelve unos valores de referencia para ver si hemos programado correctamente lo que buscábamos.

Para hacer reutilizable esta aplicación, habría que crear una clase y usar el API de dibujo de ActionScript para representar los niveles de cada indicador.