

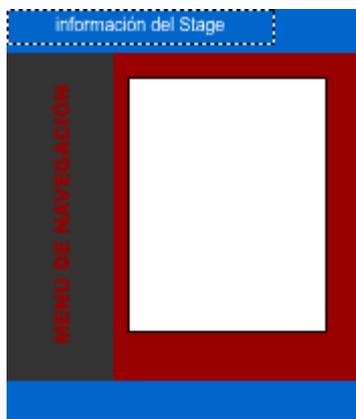
## Diagramación de contenidos

### Distribución de los elementos en la aplicación

La diagramación consiste básicamente en la distribución de elementos en un espacio. En el diseño gráfico se refiere a la maquetación de un dibujo, mientras que en el diseño Web se refiere al *layout* de las páginas. En una aplicación informática estaremos refiriéndonos, por tanto, a la distribución de los elementos visuales en pantalla. Pero, ¿qué pasa si el tamaño del espacio se modifica? Esto es algo muy común en la Web, donde las páginas pueden visualizarse desde diferentes resoluciones de monitor y diferentes tamaños de ventana. En los dispositivos móviles este problema se acentúa aún más, encontrándonos con infinidad de resoluciones “estándar” según cada fabricante.

Con ActionScript podemos realizar tres tipos de diagramación de contenidos, a saber: elástica, estática y líquida. Para controlarla se emplea la clase `Stage`, que es la encargada del escenario (la ventana de la aplicación).

Vamos a colocar dos barras horizontales de anchura igual al ancho del escenario y de altura 20 pixels. Una estará en la parte superior de la pantalla (`bSup`) y otra en la inferior (`bInf`). En el centro de la aplicación habrá un campo de texto (`campoTexto`) *selectable* y *HTML*, y un menú de navegación (`miMenu`) ficticio. En la esquina superior izquierda colocaremos un campo de texto (`miTexto`) que mostrará las dimensiones del escenario.



### Diagramación elástica

Esta es la diagramación por defecto al crear una aplicación en Flash Lite. El contenido de la ventana se escala proporcionalmente al tamaño de la misma, lo cual es interesante hasta cierto punto — por cuestiones de rendimiento y apariencia básicamente. El escenario se centra en la pantalla y se escala proporcionalmente a las dimensiones de la misma. Podemos especificar tres formas distintas de escalar el contenido, a saber:

**showAll** – Es el valor por defecto. Se muestra todo el contenido proporcionalmente, sin distorsiones y manteniendo el aspect ratio (la relación ancho/alto) del escenario. Si la pantalla tiene un aspect ratio distinto del aspect ratio del escenario, pueden aparecer bordes a los lados de la ventana de la aplicación.

```
Stage.scaleMode = "showAll";
```

**noBorder** – Escala todo el contenido sin distorsiones, manteniendo el aspect ratio y sin bordes, a costa de un ligero cropping (recortado en anchura o altura).

```
Stage.scaleMode = "noBorder";
```

**exactFit** – Escala todo el contenido hasta ajustarlo totalmente al tamaño de la ventana sin mantener el aspect ratio, produciéndose distorsiones.

```
Stage.scaleMode = "exactFit";
```

Cuando todo el contenido de la aplicación se escala, estrictamente se pierde algo de rendimiento. Pero en ocasiones no es tan significativo como para obviar esta característica de las aplicaciones basadas en gráficos vectoriales.

### Diagramación estática

Se consigue escribiendo este código:

```
Stage.scaleMode = "noScale";
```

De esta forma, al variar el tamaño de la ventana todo el contenido conservará su tamaño original y su posición en coordenadas.

### Diagramación líquida

Esta diagramación, la más interesante, hace que los elementos de la pantalla se acomoden automáticamente a las dimensiones de la ventana de la aplicación, modificando su tamaño. Esto se consigue situando el origen de coordenadas de la aplicación en la esquina superior izquierda e indicando que el escenario no se escale, con las siguientes líneas de código:

```
Stage.align = "TL";
```

```
Stage.scaleMode = "noScale";
```

Ahora hay que detectar el cambio de tamaño de la ventana y definir una función que actúe en consecuencia:

```
var sListener:Object = new Object();  
sListener.onResize = reubica;  
Stage.addListener(sListener);  
function reubica():Void {  
    // nuestra función  
};
```

En la función deberemos decir qué elementos van a modificar su posición y el espacio que van a ocupar. Por ejemplo:

```
function reubica():Void {  
    miTexto.text = "Ventana: " + Stage.width + " x " + Stage.height;  
    bSup._width = Stage.width;
```

```
bInf._width = Stage.width;  
bInf._y = Stage.height - bInf._height;  
miMenu._height = Stage.height -      →  
bSup._height - bInf._height;  
miMenu._x = 0;  
miMenu._y = bSup._height;  
campoTexto._height = Stage.height - 60;  
campoTexto._width = Stage.width - 70;  
};
```

De esta forma, podemos publicar una vez nuestra aplicación y visualizar un layout líquido en la pantalla de cualquier otro dispositivo móvil.