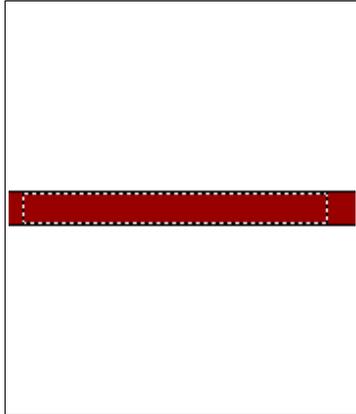


# Salvapantallas

## Fecha e intervalos de ejecución

Esta aplicación mostrará la fecha y hora en un movieClip que irá posicionándose verticalmente de forma aleatoria por la pantalla.

Partiendo del fichero **plantillaN70 fla**, crearemos un campo de texto llamado `texto_txt` y dibujaremos un rectángulo de fondo, convirtiendo ambos objetos en el movieClip `hf_mc`.



### Fecha y hora

La clase `Date` contiene todos los métodos necesarios para esta práctica (obtener día, mes, año y hora). Debemos tener en cuenta estas consideraciones:

- El día de la semana se devuelve como número, comprendido entre 0 (Domingo) y 6 (Sábado). Habrá que transformarlo en un formato más legible.
- El mes se devuelve como número entre 0 (Enero) y 11 (Diciembre).
- Para actualizar sucesivamente la fecha y hora habrá que crear un bucle `onEnterFrame`.

Vamos a mostrar la fecha con este formato:

```
Lun 12/12/2007 · 18:45:22
```

En primer lugar crearemos una lista con los días en español, que relacionaremos posteriormente con el día devuelto de la semana:

```
var dias:Array = new Array("Dom", "Lun", →  
"Mar", "Mie", "Jue", "Vie", "Sab");
```

El primer índice del vector es el 0, que se corresponde con el Domingo según el método `getDay()` de la clase `Date`.

Para obtener la fecha completa y poder actualizarla constantemente definiremos este bucle:

```
hf_mc.onEnterFrame = function():Void {  
    var fecha:Date = new Date();  
    var diasem:Number = fecha.getDay();  
    var mes:Number = fecha.getDate();
```

```
    var dia:Number = fecha.getMonth();  
    var anyo:Number = fecha.getFullYear();  
    var hora:Number = fecha.getHours();  
    var min:Number = fecha.getMinutes();  
    var seg:Number = fecha.getSeconds();  
  
    this.texto_txt.text = dias[diasem] + →  
    " " + mes + "/" + (dia+1) + "/" + anyo →  
    + " · " + hora + ":" + min + ":" + seg;  
};
```

El formato lo podemos mejorar visualmente poniendo un cero delante de aquellos números que sean de una sola cifra. Tan solo hay que definir una función que devuelva un número con dicho formato:

```
function lead(n:Number):String {  
    var result:String = "";  
    if (n < 10) result += "0";  
    result += n.toString();  
    return result;  
};
```

Y aplicarlo al texto anterior:

```
this.texto_txt.text = dias[diasem] + →  
" " + lead(mes) + "/" + lead(dia+1) + →  
"/" + anyo + " · " + lead(hora) + ":" + →  
lead(min) + ":" + lead(seg);
```

Así, si por ejemplo la fecha es esta:

```
Lun 4/3/2007 · 18:5:2
```

Obtendríamos este formato:

```
Lun 04/03/2007 · 18:05:02
```

### Posicionado aleatorio

Vamos a emplear un intervalo de ejecución, el cual funcionará cada tres segundos y se borrará tras haberse ejecutado mil veces. Vamos a definir también una variable que funcionará como contador y otra que proporcione un identificador del intervalo de ejecución, para poder borrarlo más tarde.

```
var intervalId:Number;  
var count:Number = 0;  
var maxCount:Number = 1000;  
var cadaCuanto:Number = 3000;
```

La función para reposicionar el movieClip es:

```
function reposiciona():Void {  
    hf_mc._y = Math.round(Math.random() →  
    * (Stage.height - this._height));  
  
    if(count >= maxCount) →  
    clearInterval(intervalId);
```

```
count++;  
};
```

Para llamarla a intervalos regulares se emplea esta notación:

```
intervalId = setInterval(this, →  
"reposiciona", cadaCuanto);
```

### Posicionado aleatorio con deceleración

De nuevo, vamos a mejorar visualmente nuestro salvapantallas, aplicando un efecto de *easing*, que no es más que una transición con velocidad variable. La clase Tween contiene muchos tipos de transiciones con efectos de deceleración.

Importamos dicha clase y los efectos de easing:

```
import mx.transitions.Tween;  
import mx.transitions.easing.*;
```

Y sólo hay que cambiar la línea donde definíamos la propiedad `_y` en la función `reposiciona` por:

```
var pos:Number = Math.round( →  
Math.random() * (Stage.height - →  
this._height) );  
  
new Tween(hf_mc, "_y", Regular.easeOut, →  
hf_mc._y, pos, 1, true);
```