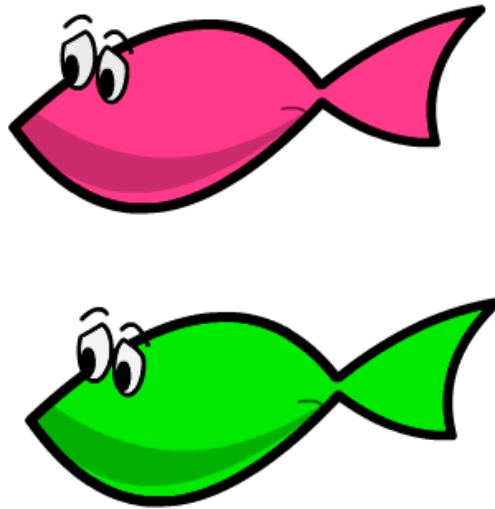


## Unidad 20. Los filtros (VIII)

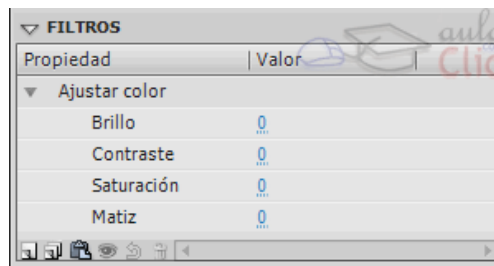
### 20.8. El filtro Ajustar color

Utilizando este filtro podremos cambiar todas las características correspondientes al color de la imagen:



Sólo tiene una opción: **.matrix**.

En ella escribiremos una **matriz de 20 elementos (4x5)** que indicarán la cantidad de cada color, su intensidad, brillo, saturación y contraste.



Podrás modificar el filtro **Ajustar Color** escribiendo el siguiente código:

```
1 import flash.filters.ColorMatrixFilter;
2
3 var miMatriz:Array = [1, 0, 0, 0, 100,
4                       0, 1, 0, 0, 100,
5                       0, 0, 1, 0, 100,
6                       0, 0, 0, 1, 0];
7
8 var ajustaColor:ColorMatrixFilter = new ColorMatrixFilter(miMatriz);
9
10 miClip.filters = [ajustaColor];
```

### 20.9. Acceder a los filtros de un elemento

Para modificar los filtros asociados a un objeto tendremos que acceder a su propiedad **filters**.

Esta propiedad almacena una matriz que no puede ser editada directamente. Así que si queremos modificar los filtros que afectan a un objeto tendremos que darle una nueva matriz a la propiedad y desechar la anterior.

Así, si queremos añadir un nuevo filtro al objeto **miClip** deberemos escribir lo siguiente:

```
1  var sombra:DropShadowFilter = new DropShadowFilter();
2
3  var nuevaMatriz:Array = miClip.filters;
4  nuevaMatriz.push(sombra);
5
6  miClip.filters = nuevaMatriz;
```

Como puedes ver, lo que hacemos es guardar los filtros existentes en una matriz **nuevaMatriz**, introducimos el nuevo filtro en ella con el método **push()** y luego le asignamos los filtros que contiene **miMatriz** al clip **miClip**.

Como esta propiedad se trata de una matriz también podemos acceder a los filtros de esta forma:

```
1  var nuevaMatriz:Array = miClip.filters;
2
3  nuevaMatriz[0].blurX = 15;
4  nuevaMatriz[0].blurY = 15;
5
6  miClip.filters = nuevaMatriz;
```

Descargamos los filtros en **nuevaMatriz**, y ahora accedemos al primer filtro almacenado y cambiamos sus propiedades **blurX** y **blurY**. Una vez modificadas, volvemos a volcar el contenido de **nuevaMatriz** en **miClip.filters**.

Fácil, ¿verdad? El único problema que tiene este método es que **deberás recordar el lugar que ocupa cada filtro en la matriz**. Recuerda que el método **push()** inserta un elemento en una matriz siempre en **último** lugar.

 [Ejercicios del Tema 20.](#)

 [Prueba evaluativa del Tema 20.](#)

[Tweet](#)



Marzo-2011

Pág. 20.8

<<

Índice

>>



Cursos de Informática gratuitos

Cursos de aulaClic

Access

CorelDraw

Dreamweaver

Excel

Fotografía

Internet

Páginas Web

Photoshop

PowerPoint

SQL Server

Selección de  
Tutoriales

Android

Base de datos

C++

Directorios de  
tutoriales

...

Linux

Metodología y  
algoritmos

OpenOffice

Photoshop

...

- [Artículos de aulaClic](#)
- [AulaClic en YouTube](#)
- [Apps aulaClic](#)
- [Cursos de](#)

