

## Unidad 20. Los filtros (I)

### 20.1. Introducción

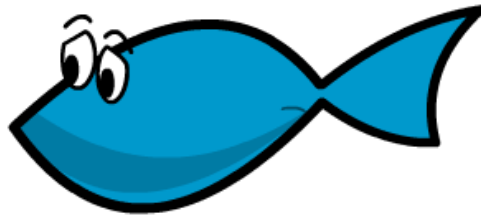


Los filtros son una potente herramienta que abre nuevas posibilidades en nuestras animaciones Flash.

Su uso aumentará la capacidad de mejorar el aspecto y funcionamiento de nuestros proyectos, aunque **sólo están disponibles para los clips de película, los botones y el texto**.

En este tema veremos los filtros disponibles, su uso y cómo manejarlos mediante el panel Propiedades, el Editor de movimiento y sobre todo mediante ActionScript.

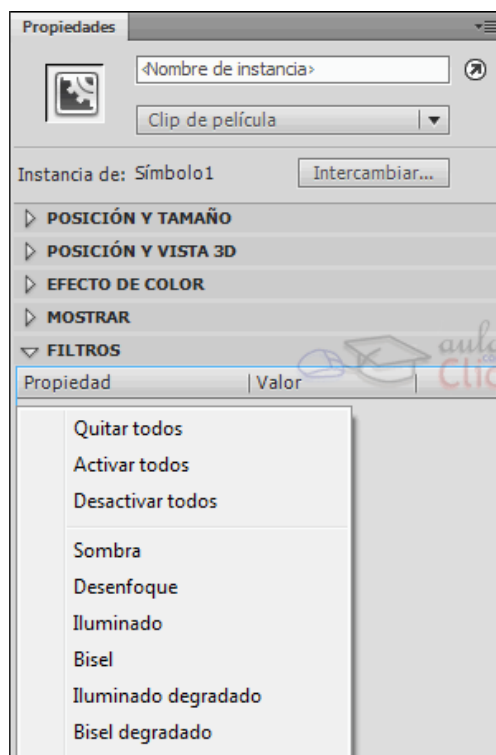
Para ilustrar su funcionamiento mostraremos ejemplos de cómo afectan los filtros a esta imagen:

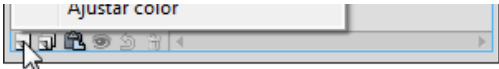


Antes de empezar resaltaremos una cosa. En ActionScript **los colores se indican mediante una cadena de 8 caracteres** con un componente hexadecimal: **0xRRVVAA**. Las dos primeras serán siempre las mismas 0x, las dos siguientes corresponderán a la cantidad de color rojo, las dos siguientes de verde y las dos últimas de azul.

De este modo **0x000000** sería el color negro (la ausencia de color), y **0xFFFFFF** el color blanco (con el rojo, azul y verde al máximo).

Veremos principalmente cómo crear los filtros mediante ActionScript, por ser la opción más potente. Pero también puedes crear los filtros desde el panel de **Propiedades**, pulsando el icono **Nuevo** en la sección **Filtros**.





**Nota:** Antes de utilizar el filtro, observarás que lo importamos con la sentencia `import flash.filters.tipoFiltro;`. Esto no es necesario si trabajamos en el panel **Acciones**, pero sí si lo hacemos en otro archivo, por ejemplo, en una clase.

[Tweet](#)



Marzo-2011

Pág. 20.1



Cursos de Informática gratuitos

Cursos de aulaClic

- |              |               |
|--------------|---------------|
| Access       | Internet      |
| CorelDraw    | Páginas Web   |
| Dreamweaver  | Photoshop     |
| Excel        | PowerPoint    |
| Fotografía   | SQL Server    |
| Google Drive | Windows 10    |
| HTML         | Word          |
| Illustrator  | Más cursos... |

Selección de Tutoriales

- |                           |                          |
|---------------------------|--------------------------|
| Android                   | Linux                    |
| Base de datos             | Metodología y algoritmos |
| C++                       | OpenOffice               |
| Directorios de tutoriales | Photoshop                |
| Hardware                  | PHP                      |
| HTML                      | Python                   |
| Java                      | XML/XHTML                |
| Javascript                | Más tutoriales...        |

- [Artículos de aulaClic](#)
- [AulaClic en YouTube](#)
- [Apps aulaClic](#)
- [Cursos de colaboradores](#)
- [Cursos Creative Commons](#)
- [Foro de aulaClic](#)
- [Preguntas más frecuentes](#)
- [Usuarios registrados](#)

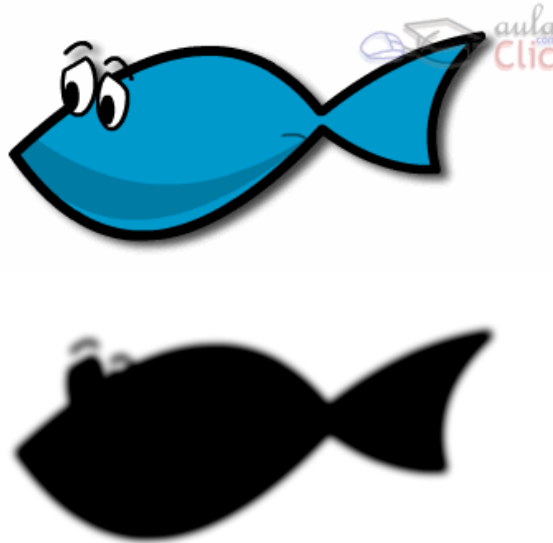
Síguenos en: [Facebook](#) , [G+](#)    [Sobre aulaClic](#)    [Política de Cookies](#)

© aulaClic S.L. Todos los derechos reservados. Prohibida la reproducción por cualquier medio.

## Unidad 20. Los filtros (II)

### 20.2. El filtro Sombra

Utilizando este filtro podremos añadirle una sombra a la imagen:

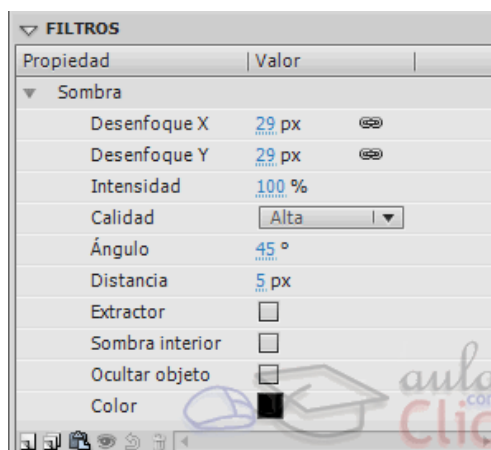


En la imagen de la izquierda vemos la imagen original más la sombra, en la imagen de la derecha sólo se ve la sombra.

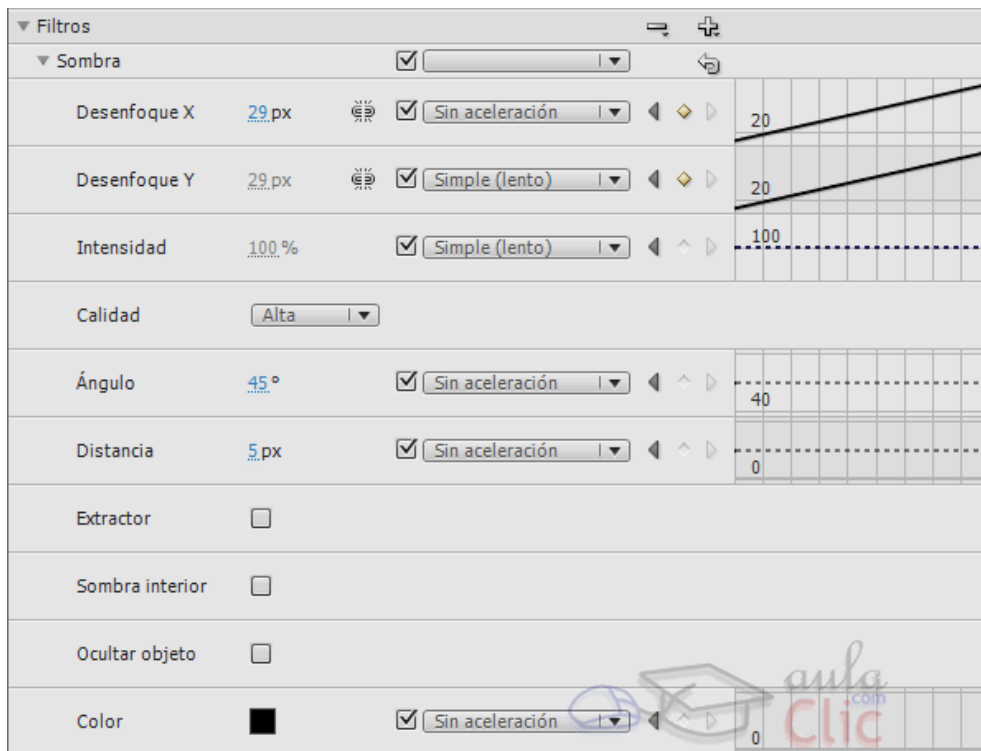
Puedes aplicar este filtro y todos los demás desde el panel **Propiedades**:

Sus opciones son las siguientes:

- **Desenfoco X**: determina la cantidad de desenfoque horizontal en píxeles.
- **Desenfoco Y**: determina la cantidad de desenfoque vertical en píxeles.
- **Intensidad**: determina la intensidad de la sombra (de 0 a 100).
- **Calidad**: número de veces que se aplica el filtro. Puede tomar los valores **baja**, **media** y **alta**.
- **Ángulo**: determina el ángulo de la sombra.
- **Distancia**: determina la distancia de la sombra al objeto en píxeles.
- **Extractor**: booleano, activa el filtro extractor, hace que el objeto se vuelva del color del fondo.
- **Sombra interior**: booleano, hace que la sombra sea interior.
- **Ocultar Objeto**: booleano, esconde el objeto dejando sólo la sombra.
- **Color**: determina el color de la sombra.



Como hemos dicho, también podemos aplicar los filtros desde el **Editor de movimiento**, tendríamos las mismas opciones pero con las gráficas propias del editor de movimiento:



● Si aplicamos el filtro desde **Action Script**, los parámetros son los mismos pero el nombre hay que darlo en inglés, los valores correspondientes son los siguientes (además del valor alpha):

- **.blurX**: determina la cantidad de desenfoco horizontal.
- **.blurY**: determina la cantidad de desenfoco vertical.
- **.strength**: determina la intensidad de la sombra.
- **.quality**: número de veces que se aplica el filtro. Puede tomar los valores **1**, **2** ó **3**. Calidad **baja**, **media** y **alta**, respectivamente.
- **.angle**: determina el ángulo de la sombra.
- **.distance**: determina la distancia de la sombra al objeto.
- **.knockout**: booleano, activa el filtro extractor, hace que el objeto se vuelva del color del fondo.
- **.inner**: booleano, hace que la sombra sea interior.
- **.hideObject**: booleano, esconde el objeto dejando sólo la sombra (segunda imagen).
- **.color**: determina el color de la sombra (escrito **0xRRVVAA**).
- **.alpha**: determina la transparencia de la sombra (de 0 a 1).

Podremos utilizar el filtro **Sombra** escribiendo el siguiente código:

```
1 import flash.filters.DropShadowFilter;
2
3 var sombra:DropShadowFilter = new DropShadowFilter();
4 sombra.strength = 5;
5 sombra.alpha = 0.50;
6 sombra.color = 0x000000;
7
8 ...
9
10 miClip.filters = [sombra];
```

También podríamos crear la sombra directamente con todas sus propiedades en el constructor de este modo:

```
1 var sombra:DropShadowFilter = new DropShadowFilter(distance, angle, color, alpha, blur
2
3 miClip.filters = [sombra];
```

Así en el constructor deberemos sustituir cada parámetro por el valor que queremos darle.

Puedes omitir los parámetros que quieras, pero ten en cuenta que siempre se interpretarán los valores que des en este orden hasta que no encuentre más.

● Para practicar la carga y modificación de filtros te aconsejamos realizar el [Ejercicio Carga y modificación de filtros](#).

[Tweet](#)



Marzo-2011

Pág. 20.2

<<

Índice

>>



Cursos de Informática gratuitos

Cursos de aulaClic		Selección de Tutoriales		
Access	Internet	Android	Linux	• <b>Artículos</b> de aulaClic
CorelDraw	Páginas Web	Base de datos	Metodología y algoritmos	• <b>AulaClic</b> en YouTube
Dreamweaver	Photoshop	C++	OpenOffice	• <b>Apps</b> aulaClic
Excel	PowerPoint	Directorios de tutoriales	Photoshop	• <b>Cursos</b> de colaboradores
Fotografía	SQL Server	Hardware	PHP	• <b>Cursos</b> Creative Commons
Google Drive	Windows 10	HTML	Python	• <b>Foro</b> de aulaClic
HTML	Word	Java	XML/XHTML	• <b>Preguntas</b> más frecuentes
Illustrator	Más cursos...	Javascript	Más tutoriales...	• <b>Usuarios</b> registrados

Síguenos en: Facebook , G+ Sobre aulaClic Política de Cookies

## Unidad 20. Los filtros (III)

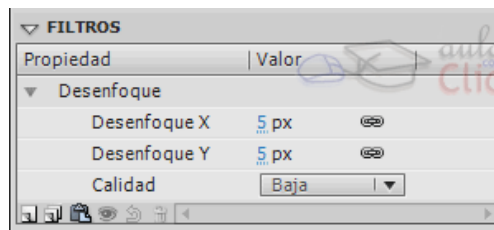
### 20.3. El filtro Desenfoque

Utilizando este filtro podremos hacer que la imagen se muestre desenfocada:



Sus opciones son las siguientes:

- **.blurX** (Desenfoque X): determina la cantidad de desenfoque horizontal.
- **.blurY** (Desenfoque Y): determina la cantidad de desenfoque vertical.
- **.quality** (Calidad): número de veces que se aplica el filtro. Puede tomar los valores **1**, **2** ó **3**. Calidad **baja**, **media** y **alta**, respectivamente.



Podremos utilizar el filtro **Desenfoque** escribiendo el siguiente código:

```
1 import flash.filters.BlurFilter;
2
3 var desenfocar:BlurFilter = new BlurFilter ();
4 desenfocar.blurX = 5;
5 desenfocar.blurY = 5;
6 desenfocar.quality= 3;
7
8 miClip.filters = [desenfocar];
```

También podremos crear el desenfoque directamente con todas sus propiedades en el constructor, de este modo:

```
1 var desenfocar:BlurFilter = new BlurFilter(blurX, blurY, quality);
2
3 miClip.filters = [desenfocar];
```

Así en el constructor deberemos sustituir cada parámetro por el valor que queremos darle.

Puedes omitir los parámetros que quieras, pero ten en cuenta que siempre se interpretarán los valores que des en este orden hasta que no encuentre más.

[Tweet](#)





**Cursos de Informática gratuitos**

#### Cursos de aulaClic

Access  
CorelDraw  
Dreamweaver  
Excel  
Fotografía  
Google Drive  
HTML  
Illustrator

Internet  
Páginas Web  
Photoshop  
PowerPoint  
SQL Server  
Windows 10  
Word  
Más cursos...

#### Selección de Tutoriales

Android  
Base de datos  
C++  
Directorios de tutoriales  
Hardware  
HTML  
Java  
Javascript

Linux  
Metodología y algoritmos  
OpenOffice  
Photoshop  
PHP  
Python  
XML/XHTML  
Más tutoriales...

- **Artículos de aulaClic**
- **AulaClic en YouTube**
- **Apps aulaClic**
- **Cursos de colaboradores**
- **Cursos Creative Commons**
- **Foro de aulaClic**
- **Preguntas más frecuentes**
- **Usuarios registrados**

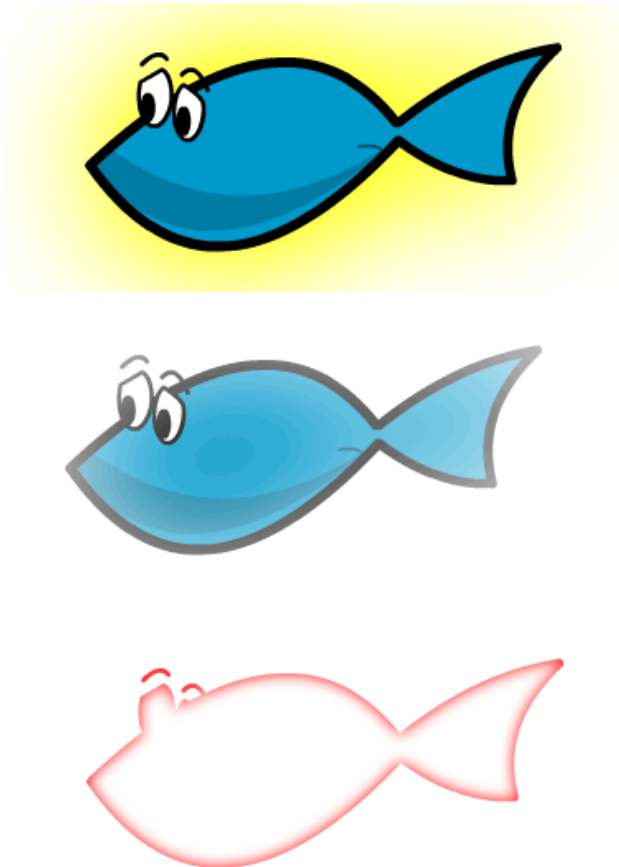
Síguenos en: [Facebook](#) , [G+](#) [Sobre aulaClic](#) [Política de Cookies](#)

© aulaClic S.L. Todos los derechos reservados. Prohibida la reproducción por cualquier medio.

## Unidad 20. Los filtros (IV)

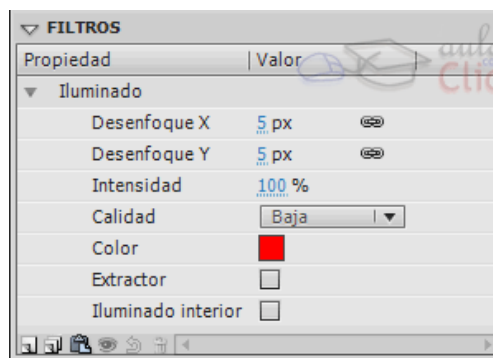
### 20.4. El filtro Iluminado

Utilizando este filtro podremos añadirle un efecto de iluminación a la imagen:



Sus opciones son las siguientes:

- **.alpha**: determina la transparencia de la iluminación (de 0 a 1).
- **.blurX**: determina la cantidad de desenfocado horizontal.
- **.blurY**: determina la cantidad de desenfocado vertical.
- **.color**: determina el color de la iluminación (escrito **0xRRVVAA**).
- **.inner**: booleano, hace que la iluminación sea interior.
- **.knockout**: booleano, activa el filtro extractor, hace que el objeto se vuelva del color del fondo.
- **.quality**: número de veces que se aplica el filtro. Puede tomar los valores **1**, **2** ó **3**. Calidad **baja**, **media** y **alta**, respectivamente.
- **.strength**: determina la intensidad de la iluminación.



Podremos utilizar el filtro Iluminado escribiendo el siguiente código:



```

1  import flash.filters.GlowFilter;
2
3  var iluminado:GlowFilter = new GlowFilter();

4
5  iluminado.quality = 3;
6  iluminado.alpha = 0.50;
7  iluminado.color = 0x000000;
8
9  ...
10
11  miClip.filters = [iluminado];

```

También podríamos crear la iluminación directamente con todas sus propiedades en el constructor, de este modo:

```

1  var iluminado:GlowFilter = new GlowFilter(color, alpha, blurX, blurY, strength, qualit
2
3  miClip.filters = [iluminado];

```

Así en el constructor deberemos sustituir cada parámetro por el valor que queremos darle.

Puedes omitir los parámetros que quieras, pero ten en cuenta que siempre se interpretarán los valores que des en este orden hasta que no encuentre más.

[Tweet](#)



Marzo-2011

Pág. 20.4

<<

Índice

>>



Cursos de Informática gratuitos

#### Cursos de aulaClic

Access  
CorelDraw  
Dreamweaver  
Excel  
Fotografía  
Google Drive  
HTML  
Illustrator

Internet  
Páginas Web  
Photoshop  
PowerPoint  
SQL Server  
Windows 10  
Word  
Más cursos...

#### Selección de Tutoriales

Android  
Base de datos  
C++  
Directorios de tutoriales  
Hardware  
HTML  
Java  
Javascript

Linux  
Metodología y algoritmos  
OpenOffice  
Photoshop  
PHP  
Python  
XML/XHTML  
Más tutoriales...

- **Artículos de aulaClic**
- **AulaClic en YouTube**
- **Apps aulaClic**
- **Cursos de colaboradores**
- **Cursos Creative Commons**
- **Foro de aulaClic**
- **Preguntas más frecuentes**
- **Usuarios registrados**

Síguenos en: [Facebook](#) , [G+](#)

[Sobre aulaClic](#)

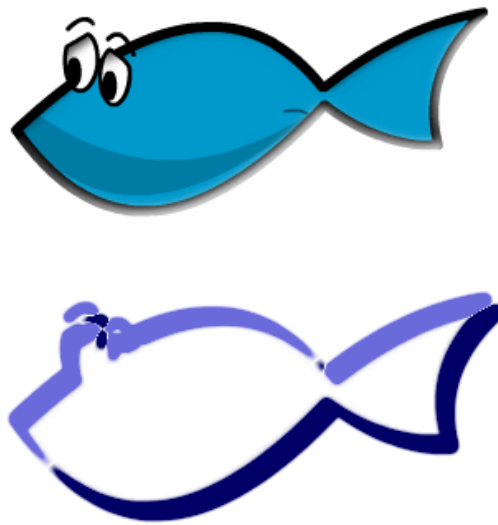
[Política de Cookies](#)

© aulaClic S.L. Todos los derechos reservados. Prohibida la reproducción por cualquier medio.

## Unidad 20. Los filtros (V)

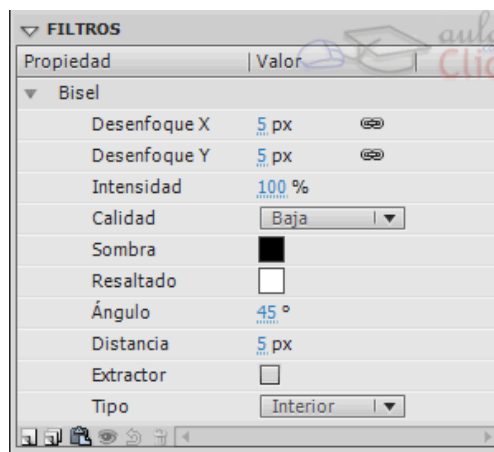
### 20.5. El filtro Bisel

Utilizando este filtro podremos añadirle un bisel a la imagen:



Sus opciones son las siguientes:

- **.angle**: determina el ángulo del bisel.
- **.blurX**: determina la cantidad de desenfocado horizontal.
- **.blurY**: determina la cantidad de desenfocado vertical.
- **.distance**: determina la distancia que abarcará el bisel en el objeto.
- **.highlightAlpha**: determina la transparencia del color de resaltado (bisel).
- **.highlightColor**: determina el color del resaltado (escrito **0xRRVVA**).
- **.knockout**: booleano, activa el filtro extractor, hace que el objeto se vuelva del color del fondo.
- **.quality**: número de veces que se aplica el filtro. Puede tomar los valores **1**, **2** ó **3**. Calidad **baja**, **media** y **alta**, respectivamente.
- **.shadowAlpha**: determina la transparencia del color de sombra del bisel.
- **.shadowColor**: determina el color de la sombra del bisel (escrito **0xRRVVA**).
- **.strength**: determina la intensidad del bisel.
- **.type**: indica el tipo de bisel a aplicar. Puede tomar los valores **inner**, **outer** y **full**. **Interior**, **exterior** y **completo**, respectivamente.



Podrás utilizar el filtro **Bisel** escribiendo el siguiente código:

```
1 import flash.filters.BevelFilter;
2
```

```

3  var bisel:BevelFilter = new BevelFilter();
4  bisel.strength = 5;
5  bisel.angle = 45;
6  bisel.shadowColor = 0x000000;
7
8  ...
9
10 miClip.filters = [bisel];

```

También podrías crear el bisel directamente con todas sus propiedades en el constructor de este modo:

```

1  import flash.filters.BevelFilter;
2
3
4
5  var bisel:BevelFilter = new BevelFilter(distance, angle, highlightColor, highlightAlpha);
6
7
8
9  miClip.filters = [bisel];

```

Así en el constructor deberemos sustituir cada parámetro por el valor que queremos darle.

Puedes omitir los parámetros que quieras, pero ten en cuenta que siempre se interpretarán los valores que des en este orden hasta que no encuentre más.

[Tweet](#)



Marzo-2011

Pág. 20.5

<<

Índice

>>



Cursos de Informática gratuitos

#### Cursos de aulaClic

Access	Internet
CorelDraw	Páginas Web
Dreamweaver	Photoshop
Excel	PowerPoint
Fotografía	SQL Server
Google Drive	Windows 10
HTML	Word
Illustrator	Más cursos...

#### Selección de Tutoriales

Android	Linux
Base de datos	Metodología y algoritmos
C++	OpenOffice
Directorios de tutoriales	Photoshop
Hardware	PHP
HTML	Python
Java	XML/XHTML
Javascript	Más tutoriales...

- **Artículos** de aulaClic
- **AulaClic** en YouTube
- **Apps** aulaClic
- **Cursos** de colaboradores
- **Cursos** Creative Commons
- **Foro** de aulaClic
- **Preguntas** más frecuentes
- **Usuarios** registrados

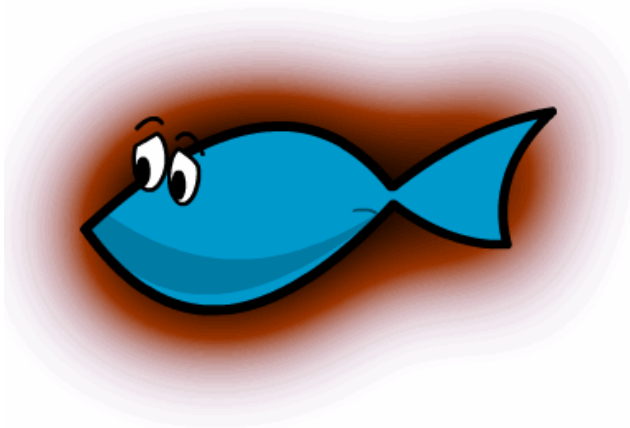
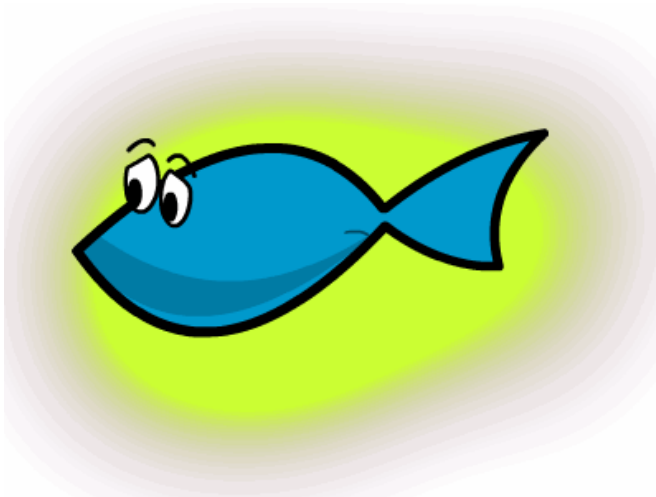
Síguenos en: [Facebook](#) , [G+](#) [Sobre aulaClic](#) [Política de Cookies](#)

© aulaClic S.L. Todos los derechos reservados. Prohibida la reproducción por cualquier medio.

## Unidad 20. Los filtros (VI)

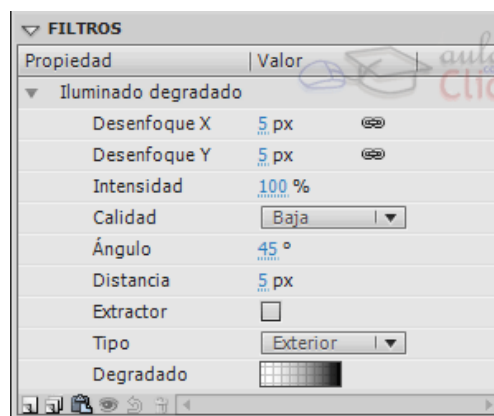
### 20.6. El filtro Iluminado degradado

Utilizando este filtro podremos añadirle una iluminación (como en el filtro anterior) a la imagen, pero con la característica de que esta iluminación estará compuesta por un degradado:



Sus opciones son las siguientes:

- **.alphas**: determina matriz de valores de transparencia alfa (de 0 a 1) para los colores correspondientes de la matriz **colors**.
- **.angle**: determina el ángulo de la iluminación.
- **.blurX**: determina la cantidad de desenfocado horizontal.
- **.blurY**: determina la cantidad de desenfocado vertical.
- **.colors**: conjunto de colores que definen el degradado (escritos **0xRRVVAA**).
- **.distance**: determina la distancia de la iluminación al objeto.
- **.knockout**: booleano, activa el filtro extractor, hace que



el objeto se vuelva del color del fondo.

- **.quality**: número de veces que se aplica el filtro. Puede tomar los valores **1**, **2** ó **3**. Calidad **baja**, **media** y **alta**, respectivamente.
- **.ratios**: matriz de proporciones de distribución de color para los colores correspondientes de la matriz **colors** (de **0** a **255**).
- **.strength**: determina la intensidad de la iluminación.
- **.type**: indica la colocación de la iluminación. Puede tomar los valores **inner**, **outer** y **full**. **Interior**, **exterior** y **completo**, respectivamente.

Podrás utilizar el filtro **Iluminado Degradado** escribiendo el siguiente código:

```
1 import flash.filters.GradientGlowFilter;
2
3 var iluminado:GradientGlowFilter = new GradientGlowFilter();
4
5 iluminado.angle = 45;
6 iluminado.colors = [0xFF0000, 0x00FF00, 0x0000FF];
7 iluminado.type = "inner";
8
9 ...
10
11 miClip.filters = [iluminado];
```

También podrías crear la iluminación directamente con todas sus propiedades en el constructor de este modo:

```
1 var iluminado:GradientGlowFilter = new GradientGlowFilter(distance, angle, colors, alpha);
2
3 miClip.filters = [iluminado];
```

Así en el constructor deberemos sustituir cada parámetro por el valor que queremos darle.

Puedes omitir los parámetros que quieras, pero ten en cuenta que siempre se interpretarán los valores que des en este orden hasta que no encuentre más.

[Tweet](#)



Marzo-2011

Pág. 20.6

<<

Índice

>>



Cursos de Informática gratuitos

Cursos de aulaClic

Access

Internet

CorelDraw

Páginas Web

Dreamweaver

Photoshop

Selección de  
Tutoriales

Android

Base de datos

C++

Linux

Metodología y  
algoritmos

• Artículos  
aulaClic

• AulaClic

de

en

16/5/2018

Curso gratis de Flash CS5. aulaClic. 20 - Los filtros

Excel

Fotografía

Google Drive

HTML

Illustrator

PowerPoint

SQL Server

Windows 10

Word

Más cursos...

Directorios de tutoriales

Hardware

HTML

Java

Javascript

OpenOffice

Photoshop

PHP

Python

XML/XHTML

Más tutoriales...

YouTube

• Apps aulaClic

• Cursos de colaboradores

• Cursos Creative Commons

• Foro de aulaClic

• Preguntas más frecuentes

• Usuarios registrados

Síguenos en: [Facebook](#) , [G+](#) [Sobre aulaClic](#) [Política de Cookies](#)

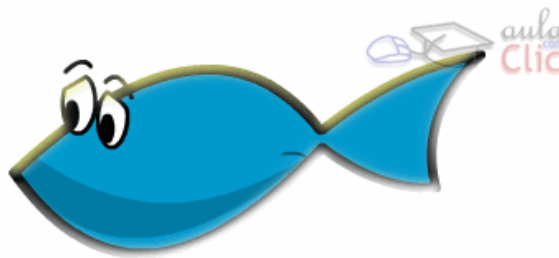
© aulaClic S.L. Todos los derechos reservados. Prohibida la reproducción por cualquier medio.



## Unidad 20. Los filtros (VII)

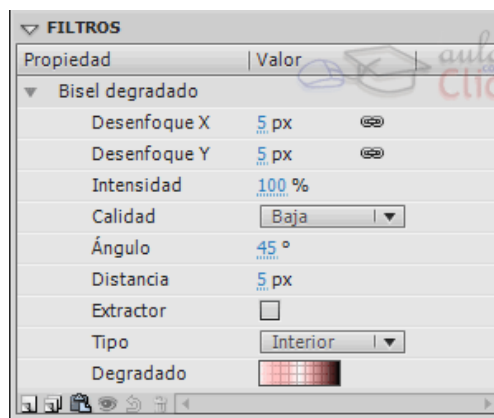
### 20.7. El filtro Bisel degradado

Igual que el filtro que hemos visto anteriormente, este filtro es capaz de mostrar un bisel sobre el objeto, pero añadiendo un degradado a la forma de éste:



Sus opciones son las siguientes:

- **.alphas**: determina matriz de valores de transparencia alfa (de 0 a 1) para los colores correspondientes de la matriz **colors**.
- **.angle**: determina el ángulo del bisel.
- **.blurX**: determina la cantidad de desenfocado horizontal.
- **.blurY**: determina la cantidad de desenfocado vertical.
- **.colors**: conjunto de colores que definen el degradado (escritos **0xRRVVA**).
- **.distance**: determina la distancia que abarcará el bisel en el objeto.
- **.knockout**: booleano, activa el filtro extractor, hace que el objeto se vuelva del color del fondo.
- **.quality**: número de veces que se aplica el filtro. Puede tomar los valores **1**, **2** ó **3**. Calidad **baja**, **media** y **alta**, respectivamente.
- **.ratios**: matriz de proporciones de distribución de color para los colores correspondientes de la matriz **colors** (de **0** a **255**).
- **.strength**: determina la intensidad del bisel.
- **.type**: indica la colocación del bisel. Puede tomar los valores **inner**, **outer** y **full**. **Interior**, **exterior** y **completo**, respectivamente.



Podremos utilizar el filtro **Bisel Degradado** escribiendo el siguiente código:

```
1 import flash.filters.GradientBevelFilter;
2
3 var bisel:GradientBevelFilter = new GradientBevelFilter();
4 bisel.angle = 90;
5 bisel.colors = [0xFF0000, 0x00FF00, 0x0000FF];
6 bisel.type = "full";
7
8 ...
9
10 miClip.filters = [bisel];
```

También podremos crear el bisel directamente con todas sus propiedades en el constructor de este modo:

```

1  import flash.filters.GradientBevelFilter;
2
3  var bisel:GradientBevelFilter = new GradientBevelFilter(distance, angle, colors, alpha
4
5  miClip.filters = [bisel];

```

Así en el constructor deberemos sustituir cada parámetro por el valor que queremos darle.

Puedes omitir los parámetros que quieras, pero ten en cuenta que siempre se interpretarán los valores que des en este orden hasta que no encuentre más.

[Tweet](#)


Marzo-2011

Pág. 20.7



Cursos de Informática gratuitos

#### Cursos de aulaClic

Access  
CorelDraw  
Dreamweaver  
Excel  
Fotografía  
Google Drive  
HTML  
Illustrator

Internet  
Páginas Web  
Photoshop  
PowerPoint  
SQL Server  
Windows 10  
Word  
Más cursos...

#### Selección de Tutoriales

Android  
Base de datos  
C++  
Directorios de tutoriales  
Hardware  
HTML  
Java  
Javascript

Linux  
Metodología y algoritmos  
OpenOffice  
Photoshop  
PHP  
Python  
XML/XHTML  
Más tutoriales...

- **Artículos de aulaClic**
- **AulaClic en YouTube**
- **Apps aulaClic**
- **Cursos de colaboradores**
- **Cursos Creative Commons**
- **Foro de aulaClic**
- **Preguntas más frecuentes**
- **Usuarios registrados**

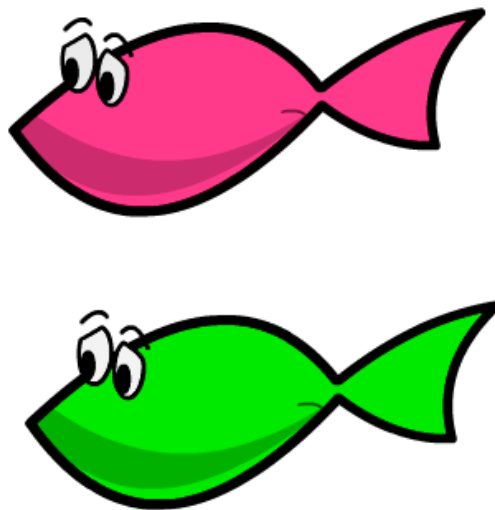
Síguenos en: [Facebook](#), [G+](#) [Sobre aulaClic](#) [Política de Cookies](#)

© aulaClic S.L. Todos los derechos reservados. Prohibida la reproducción por cualquier medio.

## Unidad 20. Los filtros (VIII)

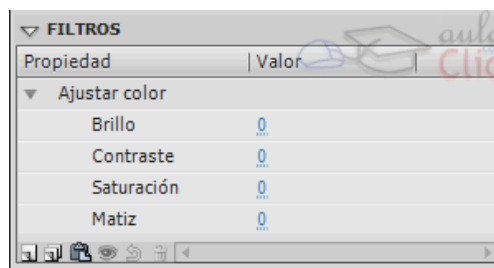
### 20.8. El filtro Ajustar color

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	Internet
CorelDraw	Páginas Web
Dreamweaver	Photoshop
Excel	PowerPoint
Fotografía	SQL Server

Selección de Tutoriales

Android	Linux
Base de datos	Metodología y algoritmos
C++	OpenOffice
Directorios de tutoriales	Photoshop
...	...

• [Artículos de aulaClic](#)

• [AulaClic en YouTube](#)

• [Apps aulaClic](#)

• [Cursos de](#)

