

03-09b Filtros SVG

Los filtros se usan definiéndolos en el general del svg con un identificador y después aplicándolos

```
<svg>  
  <filter id="identificador">  
    <feGaussianBlur in="SourceGraphic" stdDeviation="5" />  
  </filter>  
  
  <Elemento svg (por ej. circle filter="url(#identificador)">  
</svg>
```

Encadenado efectos

Los atributos **in** y **result** son fundamentales para entender cómo funcionan los **filtros en SVG**, porque permiten **encadenar efectos** dentro de un mismo <filter>.

Atributo in

Indica **cuál es la imagen de entrada** que un efecto (como <feGaussianBlur>, <feOffset>, <feBlend>, etc.) debe usar.

Las entradas más comunes son:

Valor de in	Qué significa
SourceGraphic	El elemento original sin filtrar.
SourceAlpha	Solo el canal alfa del elemento original.
BackgroundImage	La imagen del fondo (no siempre disponible).
BackgroundAlpha	Canal alfa del fondo.
FillPaint	El color de relleno.
StrokePaint	El color del trazo.
result de otro filtro	(Ej: "blurred") Entrada generada previamente por un filtro.

Ejemplo simple:

```
<feGaussianBlur in="SourceGraphic" stdDeviation="4" />
```

Esto difumina directamente el elemento original.

Atributo result

Permite **nombrar la salida** de un filtro para poder usarla más adelante como entrada (in) en otro filtro.

Sirve para encadenar pasos.

Ejemplo:

```
<feGaussianBlur in="SourceGraphic" stdDeviation="5" result="blurred" />
```

```
<feOffset in="blurred" dx="10" dy="10" result="shadow" />
```

Aquí:

- La primera operación guarda su salida como **blurred**
- La segunda toma esa salida para generar **shadow**



3. Encadenamiento completo (esquema visual)

SourceGraphic

|

v

[feGaussianBlur] -- result="blurred"

|

v

[feOffset] -- in="blurred" -- result="shadow"

|

v

[feBlend] -- mezcla de SourceGraphic + shadow

4. Ejemplo completo de filtro SVG usando in y result

```
<svg width="200" height="200">
```

```
<defs>
```

```
<filter id="sombra">
```

```
<!-- 1. Difuminar la imagen original -->
```

```
<feGaussianBlur in="SourceAlpha" stdDeviation="4" result="blur" />
```

```
<!-- 2. Desplazar el difuminado -->
```

```
<feOffset in="blur" dx="4" dy="4" result="desplazado" />
```

```
<!-- 3. Combinar sombra + elemento original -->
```

```
<feBlend in="SourceGraphic" in2="desplazado" mode="normal" />
```

```
</filter>
```

```
</defs>
<rect x="40" y="40" width="100" height="100" fill="tomato" filter="url(#sombra)" />
</svg>
```

Resumen

- **in** = qué imagen usar como entrada.
- **result** = nombre que le das a la salida para usarla después.
- Son esenciales para **construir pipelines de efectos** dentro de un `<filter>`.

Filtros SVG y opciones de configuración

<feGaussianBlur>

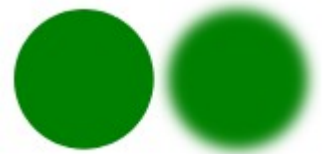
La primitiva de filtro SVG `<feGaussianBlur>` desenfoca la imagen de entrada en la cantidad especificada en `stdDeviation`, que define la curva de campana.

// Ejemplo 1

```
<svg
width="230"
height="120"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">

<filter id="blurMe">
<feGaussianBlur in="SourceGraphic" stdDeviation="5" />
</filter>

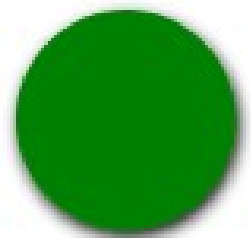
<circle cx="60" cy="60" r="50" fill="green" />
<circle cx="170" cy="60" r="50" fill="green" filter="url(#blurMe)" />
</svg>
```



// Ejemplo 2

```
<svg width="120"
height="120"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">

<filter id="dropShadow">
```





Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web
IES Playamar

```
<feGaussianBlur in="SourceAlpha" stdDeviation="3" />
<feOffset dx="2" dy="4" />
<feMerge>
<feMergeNode />
<feMergeNode in="SourceGraphic" />
</feMerge>
</filter>

<circle cx="60" cy="60" r="50" fill="green" filter="url(#dropShadow)" />
</svg>
```

<feImage>

La primitiva de filtro SVG <feImage> obtiene datos de imagen de una fuente externa y proporciona los datos de píxeles como salida (lo que significa que si la fuente externa es una imagen SVG, se rasteriza).

// Ejemplo 3

```
<svg
viewBox="0 0 200 200"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
width="200"
height="200">
<defs>

<filter id="image">
<feImage href="playa.png" />
</filter>

</defs>

<rect x="10%" y="10%" width="80%" height="80%" style="filter:url(#image);">
<animate attributeName="width" from="10%" to="80%" dur="1s" begin="0" end="1"> </animate>
<animate attributeName="width" from="80%" to="10%" dur="1s" begin="1" end="2"> </animate>
</rect>
<text x="50" y="180" font-size="20"> Esto es un Svg</text>
</svg>
```



<feMerge>

<feMergeNode>

El elemento SVG <feMerge> permite aplicar efectos de filtro simultáneamente en lugar de secuencialmente. Esto se logra mediante otros filtros que almacenan su salida a través del atributo result y luego acceden a ella en un elemento secundario <feMergeNode>.

feMergeNode toma el resultado de otro filtro para que lo procese su padre <feMerge>.

// Ejemplo 4

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink"
width="200" height="200" >

<filter id="feOffset" x="-40" y="-20" width="100" height="200">

  <feOffset in="SourceGraphic" dx="60" dy="60" />
  <feGaussianBlur in="SourceGraphic" stdDeviation="5" result="blur2" />

  <feMerge>
    <feMergeNode in="blur2" />
    <feMergeNode in="SourceGraphic" />
  </feMerge>
</filter>
<rect x="40" y="40" width="100" height="100" style="stroke: #000000; fill: green; filter: url(#feOffset);"
/>
<rect x="40" y="40" width="100" height="100" style="stroke: #000000; fill: green;" />
</svg>
```

<feTurbulence>

La primitiva de filtro SVG <feTurbulence> crea una imagen utilizando la función de turbulencia Perlin. Permite la síntesis de texturas artificiales como nubes o mármol. La imagen resultante llenará toda la subregión primitiva del filtro.

BaseFrequency="0.125"	→ Granularidad de la textura de turbulencia
numOctaves="2"	→ iteraciones sobre la textura o complejidad

// Ejemplo 5

```
<svg width="200" height="200" viewBox="0 0 220 220" xmlns="http://www.w3.org/2000/svg">

<filter id="displacementFilter">
<feTurbulence
type="turbulence" baseFrequency="0.05"
numOctaves="2" result="turbulence" />

<feDisplacementMap
in2="turbulence" in="SourceGraphic"
scale="50" xChannelSelector="R"
yChannelSelector="G" />

</filter>

<circle cx="100" cy="100" r="100" style="filter: url(#displacementFilter)" />

</svg>
```



// Ejemplo 6

```
<svg width="200" height="200" viewBox="0 0 200 200" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<defs>
<radialGradient id="star_grad">
<stop offset="0%" style="stop-color: #dffda5;"/>
<stop offset="50%" style="stop-color: orange;"/>
<stop offset="100%" style="stop-color: #eddfa6;"/>
</radialGradient>
<filter id="star_filter" x="0" y="0" width="100%" height="100%">
<b>feTurbulence id="color2" baseFrequency="0.03, 0.02" seed="2" numOctaves="4"
type="fractalNoise" stitchTiles="noStitch" result="ripples" />
<feComposite result="comp" id="feComposite_def" in = "SourceGraphic" in2 = "ripples"
operator="out"/>
</filter>
</defs>

<rect id="feConvolveMatrix_image" x="1" y="1" width="200" height="200" fill="white" />
<circle cx="50%" cy="50%" r="35%" fill="url(#star_grad)" style="filter: url(#star_filter);"/>
</svg>
```

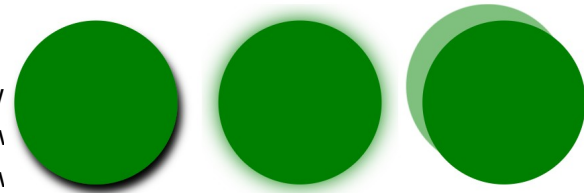


<feDropShadow>

La primitiva de filtro SVG <feDropShadow> crea una sombra paralela de la imagen de entrada. Sólo se puede utilizar dentro de un elemento <filter>.

// Ejemplo 7

```
<svg viewBox="0 0 30 10" xmlns="http://www.w3.org/2000/svg">
  <defs>
    <filter id="shadow"> <feDropShadow dx="0.2" dy="0.4" stdDeviation="0.2" /> </filter>
    <filter id="shadow2"> <feDropShadow dx="0" dy="0" stdDeviation="0.5" flood-color="green" />
    </filter>
    <filter id="shadow3">
      <feDropShadow dx="-0.8" dy="-0.8" stdDeviation="0"
        flood-color="green" flood-opacity="0.5" />
    </filter>
  </defs>
  <circle cx="5" cy="50%" r="4" style="fill:green; filter:url(#shadow) />
  <circle cx="15" cy="50%" r="4" style="fill:green; filter:url(#shadow2) />
  <circle cx="25" cy="50%" r="4" style="fill:green; filter:url(#shadow3) />
</svg>
```



<fePointLight>

La primitiva de filtro <fePointLight> define una fuente de luz que permite crear un efecto de luz puntual. Puede usarse dentro de una primitiva de filtro de iluminación: <feDiffuseLighting> o <feSpecularLighting>.

// Ejemplo 8

```
<svg width="200" height="200" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink">
  <defs>
    <filter id="spotlight">
      <feSpecularLighting result="spotlight" specularConstant="1.5"
        specularExponent="80" lighting-color="white">
        <fePointLight x="50" y="50" z="220" />
      </feSpecularLighting>
      <feComposite
        in="SourceGraphic" in2="spotlight" operator="arithmetic" k1="0" k2="1" k3="1"
        k4="0" />
    </filter>
  </defs>
  <image href="playa.png" x="10%" y="10%" width="80%" height="80%"
    style="filter:url(#spotlight);" />
</svg>
```


<feBlend> y <feFlood>

La primitiva de filtro SVG <feFlood> llena la subregión del filtro con el color y la opacidad definidos por color de inundación y opacidad de inundación.

La primitiva de filtro SVG <feBlend> compone dos objetos juntos regidos por un determinado modo de fusión. Esto es similar a lo que se conoce con el software de edición de imágenes cuando se combinan dos capas. El modo está definido por el atributo de modo.

// Ejemplo 9

```
<svg  
width="200" height="200" xmlns="http://www.w3.org/2000/svg"  
xmlns:xlink="http://www.w3.org/1999/xlink">  
  
  <defs>  
    <filter id="filtroverde">  
  
      <feFlood  
        result="floodFill" x="0" y="0"  
        width="100%" height="100%"  
        flood-color="blue"  
        flood-opacity="1" />  
  
      <feBlend in="SourceGraphic" in2="floodFill" mode="multiply" />  
    </filter>  
  </defs>  
  
  <image href="playa.png" x="10%" y="10%" width="80%" height="80%"  
    style="filter:url(#filtroverde);" />  
</svg>
```



Practica

Ejercicio 1. Crea todos los ejemplos que se encuentran en el documento

Ejercicio 2. Modifica el ejemplo 1 para hacer un blur sobre una elipse roja irregular con desviación 10

Ejercicio 3. Modifica el ejemplo 2 para aplicar el mismo ejemplo de la una sombra blur sobre una rectángulo rojo.

Ejercicio 4. Modifica el ejemplo 3 para insertar una imagen de superman que vuele de izquierda a derecha

Ejercicio 5. Modifica varios ejemplos del ejemplo 5 para variar el tipo de turbulencias.

Ejercicio 6. Modifica el ejemplo 4 para insertar los filtros del ejemplo 2 y el ejemplo 5 a la vez

Ejercicio 7. Modifica el ejemplo 7 sobre sombras para que quede así

