

Contenido

SVG (SCALABLE VECTOR GRAPHICS).....	2
1. CONCEPTO	2
2. VENTAJAS.....	2
3. CARACTERÍSTICAS.....	2
4. CÓMO EMPEZAMOS.....	2
5. ESTRUCTURA DE LA ETIQUETA PARA HACER UN SVG.....	3
6. FORMAS BÁSICAS	4

SVG (SCALABLE VECTOR GRAPHICS)

1. CONCEPTO

El SVG se puede traducir como gráficos basados en vectores escalables, es decir es un formato gráfico en XML (lenguaje de marcado de texto extendido) para crear archivos vectoriales en 2D con un lenguaje de etiquetas como el HTML. Entre sus posibilidades podemos:

1. Formas de vectores gráficos como líneas, polígonos, círculos, elipses, entre otras.
2. Imágenes y texto.

2. VENTAJAS

Algunas ventajas son:

- ✓ No pierde la calidad si se le hace un zoom.
- ✓ Se puede escalar
- ✓ Se muestra en forma progresiva, no hay que esperar que se cargue todo el archivo
- ✓ Se le pueden aplicar estilos con CSS

3. CARACTERÍSTICAS

Comparados con los mapas de bits tienen las siguientes características:

Un gráfico en PNG podría ocupar el triple de un SVG

La nitidez del SVG siempre será mejor que un mapa de bits

4. CÓMO EMPEZAMOS

Para realizar el gráfico debemos emplear un programa editor de desarrollo (ambiente integrado de desarrollo), en este caso se utiliza el Sublime Text.

Gusanita

Es necesario para cada proyecto desarrollarlo dentro de la plantilla que se muestra en la figura 1, esta es traída automáticamente al comenzar un proyecto, pero si no es así, se puede obtener de la siguiente manera:

Dar clic en la pestaña **View**, opción **Syntax**, activar la opción de **HTML**. Luego para activar la plantilla se da clic en **Tools**, opción **Snippets** y se selecciona **Snippets html**.

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Actividad</title>
5  </head>
6  <body>
7
8  Dentro del body se escriben las otras etiquetas
9
10
11 </body>
12 </html>
```

Figura. 1 Plantilla HTML

5. ESTRUCTURA DE LA ETIQUETA PARA HACER UN SVG

Dentro del body siempre debemos comenzar con la siguiente etiqueta, que tendrá la siguiente estructura:

```
<svg version="1.1" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink"
x="0"
y="0"
width="100%"
height="100%"
viewBox="0 0 400 400">
```

Figura. 2 Estructura de la etiqueta SVG

- **Version:** Actualmente estamos en la 1.1.
- **Namespaces "xmlns" y "xmlns:xlink"**, algo que necesita el lenguaje de marcado XML. El segundo, que pone xmlns:xlink es para referenciar archivos dentro del SVG, como imágenes, para poner enlaces y si lo dejamos de poner suele dar muchos problemas con Mozilla, porque se pierden las imágenes, no funcionan los links, etc.
- **Atributo "width" y "height"** por defecto en píxeles, pero lo podemos poner en cualquier unidad, como centímetros, picas, etc. Hacen referencia al ancho y alto respectivamente.
- **Coordenadas X e Y**, empiezan siempre en la esquina superior izquierda.
- **"Viewbox"**, el primer caracter representa la coordenada horizontal, el segundo la vertical, el tercero el ancho y el cuarto el alto. Como se puede apreciar en el código de la etiqueta anterior, representan

los mismos valores de coordenadas y tamaño de la imagen, pero podemos alterarlos para indicar qué es lo que vamos a ver exactamente en la imagen de todo lo que se represente en el código del SVG.

6. FORMAS BÁSICAS

- ✓ **Rectángulo (Rectangle):** Los atributos **x** e **y** son coordenadas horizontal y vertical donde lo situamos, siendo la referencia la esquina superior izquierda. Luego asignamos altura y anchura, colores de relleno y de borde y el grosor de la línea del borde.

Si se desea con los bordes redondeados, se añade añadimos **rx="4"**, siendo 4 el radio del redondeado. Ver la figura número 3.



Figura. 3 Código para el rectángulo y resultado

- ✓ **Círculo (circle):** Todos los atributos de los distintos tags para formas básicas son bastante parecidos. En este caso **cx** y **cy** son las coordenadas x e y del círculo, sin embargo, en este caso la referencia a situar es el centro del círculo. El atributo **r** es el radio. Ver la figura número 4.

```
<div>  
  <svg version="1.1" xmlns="http://www.w3.org/2000/svg"  
    xmlns:xlink="http://www.w3.org/1999/xlink"  
    x="0"  
    y="0"  
    width="100%"  
    height="100%"  
    viewBox="0 0 400 400">  
  
    <circle cx="150" cy="100" r="40" fill="#16a085" stroke=" #000" strokewidth="4" />  
  
  </svg>  
</div>
```

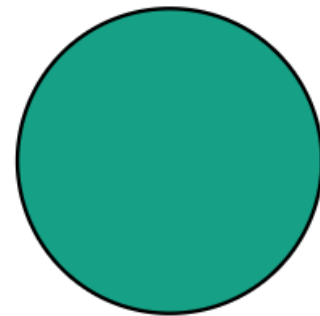


Figura. 4 SVG para el círculo

- ✓ **Elipse (ellipse):** Es muy parecido al círculo, solo que en una elipse definimos dos radios: uno horizontal y otro vertical, que pueden tener valores distintos. Ver la figura numero 5

```
<div>  
  <svg version="1.1" xmlns="http://www.w3.org/2000/svg"  
    xmlns:xlink="http://www.w3.org/1999/xlink"  
    x="0"  
    y="0"  
    width="100%"  
    height="100%"  
    viewBox="0 0 400 400">  
  
    <ellipse cx="100" cy="100" rx="30" ry="80" fill="#16a100" stroke=" #000" strokewidth="4"  
    />  
  
  </svg>  
</div>
```

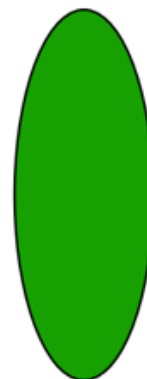


Figura. 5 SVG para la elipse.

- ✓ **Polígono (Polygon):** Se utiliza para dibujar formas con 3 o más caras, se debe definir sus puntos vértices. Cada par de valores en el atributo **points** indica uno de los vértices del polígono. Ver la figura número 6.

```
<div>
  <svg version="1.1" xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    x="0"
    y="0"
    width="100%"
    height="100%"
    viewBox="0 0 400 400">
    <polygon points="200,100,250,180,160,210" fill="#B4045F" stroke="#000" strokewidth="4" />
    <polygon points="300,200,350,280,260,310, 150,305, 200, 200" fill="#BF00FF" stroke="#000" strokewidth="4" />
  </svg>
</div>
```

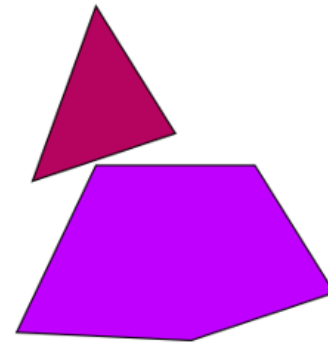


Figura. 6 SVG para el Polígono

✓ **Línea (line):** Dibuja líneas simples. Se indica la esquina del principio y fin con sus coordenadas. (Nota: sobraron comas en el código) corregir son de apar: 200,100 250,180 160,210

```
<div>
  <svg version="1.1" xmlns="http://www.w3.org/2000/svg"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    x="0"
    y="0"
    width="100%"
    height="100%"
    viewBox="0 0 400 400">
    <line x1="50" y1="50" x2="150" y2="50" stroke="#0101DF" strokewidth="3" />
  </svg>
</div>
```



Figura. 7. SVG para la línea simple

Estilo de líneas:

Podemos poner un remate al final de la línea con el atributo **stroke-linecap** con los siguientes los valores

- stroke-linecap="round": la línea empieza y termina redondeada
- stroke-linecap="square": la línea empieza y termina cuadrada
- stroke-linecap="butt": la línea se queda como está

```
<line x1=90 y1=900 x2=300 y2=900 stroke="purple" stroke-width=5 />  
<line x1=90 y1=950 x2=300 y2=950 stroke="red" stroke-width=10 stroke-linecap="round" />  
<line x1=90 y1=1000 x2=300 y2=1000 stroke="blue" stroke-width=10 stroke-linecap="square" />  
<line x1=90 y1=1050 x2=300 y2=1050 stroke="pink" stroke-width=20 stroke-linecap="butt" />
```



El atributo **stroke-dasharray** permite hacer líneas discontinuas. Se colocan un par de números donde el primero indica la longitud de la línea que se ve y el segundo la longitud que está oculta.

```
<line x1=90 y1=1100 x2=300 y2=1100 stroke="navy" stroke-width=5 stroke-dasharray="4,4" />  
<line x1=90 y1=1150 x2=300 y2=1150 stroke="cyan" stroke-width=5 stroke-dasharray="10,10" />  
<line x1=90 y1=1200 x2=300 y2=1200 stroke="purple" stroke-width=5 stroke-dasharray="15,6,2,6,2,6" />
```



✓ **Polilínea (polyline):** una línea formada por varias líneas seguidas

```
<div>  
  <svg version="1.1" xmlns="http://www.w3.org/2000/svg"  
    xmlns:xlink="http://www.w3.org/1999/xlink"  
    x="0"  
    y="0"  
    width="100%"  
    height="100%"  
    viewBox="0 0 400 400">  
    <polyline points=20,40,40,40,80,80,80,80,120" fill="none" stroke="#0101DF" strokewidth="10" />  
  </svg>  
</div>
```

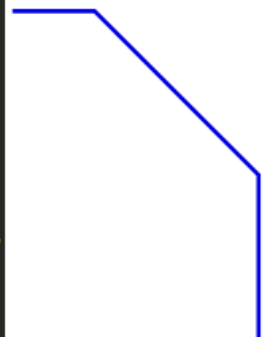


Figura. 8. SVG para la polilínea

Gusanita

ACTIVIDAD

Realizar las siguientes figuras cambiando su:

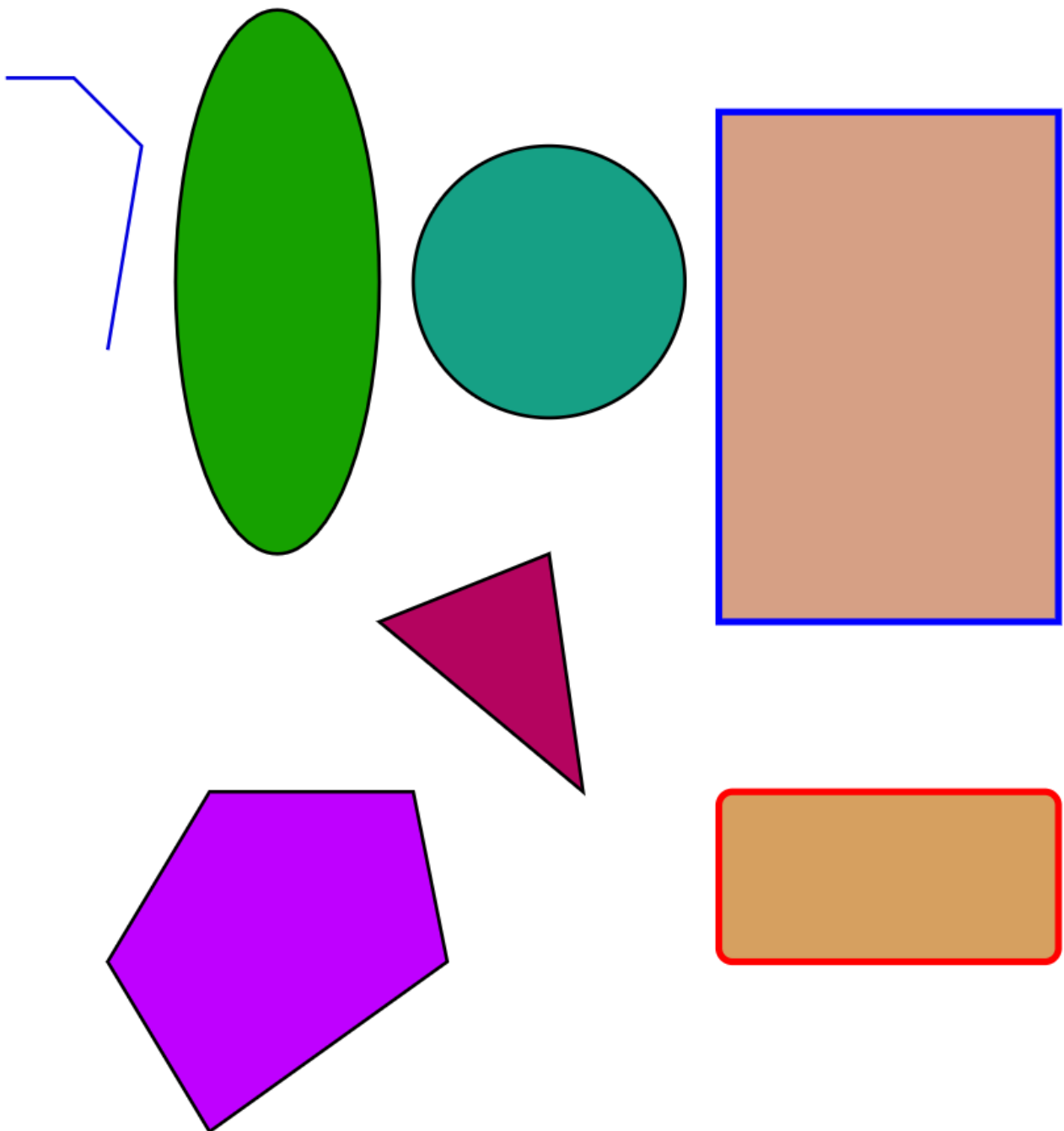
color (fill) de fondo,

color de línea

ancho de línea

Alto y ancho

Enviar por correo colocando su nombre en el código HTML




```
<svg version="1.1" xmlns="http://www.w3.org/2000/svg"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  x="0"
  y="0"
  width="100%"
  height="100%"
  viewBox="0 0 400 400">

  <rect x="230" y="50" width="100" height="150" fill="#d6a085" stroke-width="2" stroke=
    "blue" />

  <rect x="230" y="250" width="100" height="50" fill="#d6a060" stroke-width="2" stroke=
    "red" rx="4"/>

  <circle cx="180" cy="100" r="40" fill="#16a085" stroke= "#000" stroke-width="4" />

  <ellipse cx="100" cy="100" rx="30" ry="80" fill="#16a100" stroke="#000" stroke-width=
    "4" />

  <polygon points="130,200 180,180 190, 250" fill="#B4045F" stroke="#000" stroke-width="
    4" />

  <polygon points="50,300 80,350 150,300 140,250 80,250" fill="#BF00FF" stroke="#000"
    stroke-width="4" />

  <polyline points="20,40,40,40,60,60,50,120" fill="none" stroke="#0101DF" stroke-width="
    10" />

</svg>
</div>

</body>
</html>
```

7. RUTAS

Es un trazado de una o varias líneas que se hacen una detrás de la otra, debemos indicar el movimiento del lápiz de un punto a otro. Se hace con la etiqueta **path** y el atributo **d**. Se debe tener en cuenta como se ha visto antes el color de la línea (**stroke**), el ancho de la línea (**stroke-width**) y color del relleno (**fill**).

Al trazar la ruta el lápiz o puntero tiene unas coordenadas determinadas, las cuales vienen indicadas siempre por el último punto que hemos marcado. Los comandos que se ponen como valor del atributo **d** son una serie de letras que indican un tipo de trazado, así:

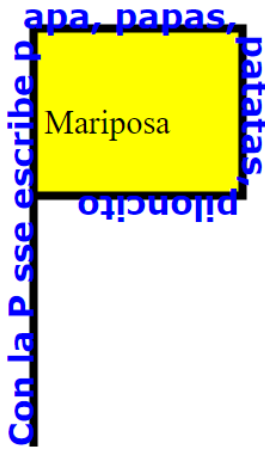
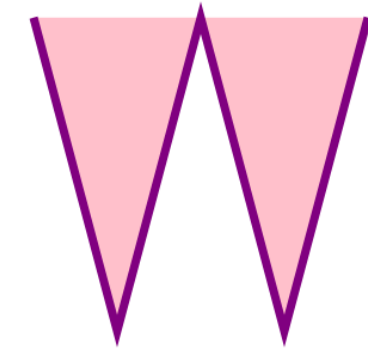
- ✓ **M**: el puntero se sitúa en la posición indicada
- ✓ **L**: Traza línea recta
- ✓ **H**: traza línea recta horizontal
- ✓ **V**: traza línea recta vertical
- ✓ **C**: traza línea curva Bézier
- ✓ **Q**: traza línea curva cuadrática
- ✓ **A**: traza un arco elíptico
- ✓ **Z**: cierra la ruta actual

EJERCICIOS

Líneas

- ✓ Se empieza posicionando el puntero al principio de la ruta (**M 20 25**).
- ✓ Seguimos indicando una serie de líneas con el comando **l** (ele minúscula) (**l 40 150 l 40 -150 l 40 150 l 40 -150**). En cada una de ellas se indican las coordenadas relativas "x" e "y" respecto de la anterior.
- ✓ Después se posiciona el puntero en el último punto en el que está para que éste no vuelva al inicio al cerrar la ruta (**m 0 0**).
- ✓ Y por último cerramos la ruta (**z**).

Gusanita



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>SVG</title>
5 </head>
6 <body>
7
8 <svg width=1000 height=1500>
9 <path d="M 20 25
10   l 40 150
11   l 40 -150
12   l 40 150
13   l 40 -150
14   M 0 0 z"
15   stroke="purple" stroke-width=4 stroke-width=4 fill="pink"/>
16
17
18 <path id="pota" d="M 20 400
19   l 0 -200
20   h 100
21   v 80
22   h -100
23   M 0 0 z"
24   stroke="black" stroke-width=4 stroke-width=4 fill="yellow"/>
25
26 <text x=25 y=250>Mariposa</text>
27 <text fill="blue">
28   <textPath xlink:href="#pota" style="font: bold 1em verdana"> Con la P sse escribe papa,
29   papas, patatas, piloncito</textPath>
30 </text>
31
32 <path id="potal" d="M 20 600
33   l 30 -150
34   l 26 80
35   h -44
36   h 44
37   l 22 70
38   M 0 0 z"
39   stroke="black" stroke-width=4 stroke-width=4 fill="green"/>
40 <text fill="red">
41   <textPath xlink:href="#potal" style="font: bold 1em verdana"> Hola mundo, estoy aqui y
42   me quedo, que tal</textPath>
43 </text>
44 </svg>
45 </body>
46 </html>
```

```
<!DOCTYPE html>
<html>
<head>
  <title>SVG</title>
</head>
<body>

<svg width=1000 height=1500>
<path d="M 20 25
      l 40 150
      l 40 -150
      l 40 150
      l 40 -150
      M 0 0 z"
      stroke="purple" stroke-width=4 stroke-width=4 fill="pink"/>/>

</svg>
</body>
</html>
```



Gusanita

<https://desarrolloweb.com/articulos/que-es-svg.html>