

## Herramientas: Formatos gráficos. Postcript y SVG. Parte II. El formato SVG.

### 1. Introducción.

El formato SVG fue creado por la w3c como formato para realizar gráficos en páginas web. Usa el formato XML y es manejado por programas de diseño gráfico como Inkscape. Algunos navegadores como Firefox, Konqueror y IExplorer tienen extensiones que permiten visualizar estas imágenes directamente. También existen applets escritos en Java que permiten insertar las imágenes SVG dentro de los navegadores.

Lo cómodo de este formato es que se puede generar de forma automática un dibujo usando Octave y después editarlo con algún programa de edición como puede ser Inkscape o Gimp. Así, por ejemplo, podríamos generar una gráfica con Octave y añadirle textos con Inkscape indicando los puntos importantes de la gráfica. Este formato también es generado por otros programas como Gnuplot.

Los ejemplos que se han creado en este documento han sido traducidos a formato png e insertados en el documento usando Gimp.

### 2. Un dibujo sencillo.

Los dibujos en formato SVG se pueden editar usando cualquier editor de texto, como Emacs o Kwrite. Abrimos un editor y escribimos lo siguiente:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <rect x="1" y="1" width="198" height="198" fill="blue" stroke="red" stroke-width="2" />
  <line x1="100" y1="100" x2="200" y2="200" stroke="#ff0000" stroke-width="1" />
  <text x="50" y="50" font-family="Times" font-size="24" fill="white" >
    Hello world
  </text>
</svg>
```

Lo guardamos como ej1.svg.

Si nuestro navegador web soporta el formato SVG sólo hay que hacer doble click sobre el archivo. En el caso de no soportarlo, se puede usar un programa de edición como Inkscape o Gimp. Se tendrá la siguiente imagen:



Se estudian cada una de las partes del dibujo anterior:

La primera línea hay que copiarla tal cual, indica que se trata de un fichero XML

```
<?xml version="1.0" standalone="yes"?>
```

Después se indica que se va a iniciar el dibujo el tamaño del dibujo serán 200 puntos de ancho (width="200") por 200 puntos de alto (height="200"). Con version="1.1" se indica la versión del formato SVG que se usará.

Aquí sólo se va a usar la versión 1.1 de SVG por lo que este parámetro no se debería de cambiar.

```
<svg width="200" height="200" version="1.1">
```

Con rect se dibuja un rectángulo. Su esquina superior izquierda va a estar en la posición x="1" y="1" y va a tener un ancho de width="198" y un alto de height="198". Se rellenará de color azul fill="blue" y la línea del contorno será de color rojo stroke="red" con una anchura de 2 pixels stroke-width="2". Los parámetros de color de relleno indicados anteriormente se pueden aplicar a otras figuras.

```
<rect x="1" y="1" width="198" height="198" fill="blue" stroke="red" stroke-width="2" />
```

Ahora se dibuja una línea recta. Para ello se usa line. La línea comienza en el centro de la imagen x1="100" y1="100" y acaba en la esquina inferior derecha x2="200" y2="200". Su color es rojo stroke="#ff0000" y la anchura de la línea es de 1 pixel stroke-width="1". Los colores se pueden introducir usando su nombre en

inglés o usando la notación de RGB que se explicará más adelante.

```
<line x1="100" y1="100" x2="200" y2="200" stroke="#ff0000" stroke-width="1" />
```

Se dibuja un texto usando text. El texto comenzará en las coordenadas x="50" y="50" y usarán el tipo de letra Times font-family="Times". El tamaño del texto será de 24 pixels font-size="24" y su color será blanco fill="white".

```
<text x="50" y="50" font-family="Times" font-size="24" fill="white" >
  Hello world
</text>
```

Por último se marca el final del dibujo con </svg>.

```
</svg>
```

### 3. Colores, rellenos y contornos.

La mayoría de los objetos gráficos de SVG permiten colores para rellenarlos, así como ponerles líneas de contorno.

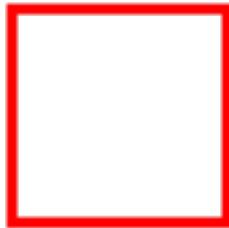
Los colores se pueden especificar usando el nombre del color en inglés o usando el formato RGB. Para usar el formato RGB se pone un sostenido (#) y a continuación se coloca en formato hexadecimal las proporciones de rojo, verde y azul que lleva dicho color. Por ejemplo, el color rojo será #ff0000, el color azul #0000ff, el color blanco #ffffff, o se pueden dar otras combinaciones de colores #1099aa.

Para rellenar una figura de un color se usará la propiedad fill. Así fill="blue" rellenará la figura de color azul.

Para no rellenar una figura de ningún color se usará fill="none". Por ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <rect x="50" y="50" width="100" height="100" fill="none" stroke="red" stroke-width="5" />
</svg>
```

Dibujará un cuadrado sin relleno, pero con un contorno rojo:



Se pueden dibujar las líneas del contorno de una figura, como en el ejemplo anterior indicando stroke y el color. Para controlar el espesor de la línea de contorno no se usa stroke-width. Se usará stroke="none" si no se quiere dibujar ningún contorno alrededor de la figura. Por ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <rect x="50" y="50" width="100" height="100" fill="blue" stroke="none" />
</svg>
```

Dibujará un rectángulo azul sin contorno:



## 4. Figuras.

Entre las figuras que el formato SVG puede manejar se encuentran líneas, rectángulos, círculos, elipses, polilíneas, textos,... A continuación se ponen ejemplos de algunos tipos.

### 4.1 Líneas.

Se dibujan usando `line`. Por ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <line x1="50" y1="50" x2="150" y2="150" stroke="black" stroke-width="3"/>
</svg>
```

Dibujarán una línea que comienza en `x1="50" y1="50"`, acaba en `x2="150" y2="150"`, de color negro `stroke="black"` y espesor 3 `stroke-width="3"`.

### 4.2 Rectángulos.

Se dibujan usando `rect`. Por ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <rect x="1" y="1" width="198" height="198" fill="blue" stroke="red" stroke-width="2" />
</svg>
```

Se dibuja un rectángulo. Su esquina superior izquierda va a estar en la posición `x="1" y="1"` y va a tener un ancho de `width="198"` y un alto de `height="198"`. Se rellenará de color azul `fill="blue"` y la línea del contorno será de color rojo `stroke="red"` con una anchura de 2 píxeles `stroke-width="2"`.

### 4.3 Círculos.

Se dibujan usando `circle`. Por ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <circle cx="100" cy="100" r="50" fill="blue" stroke="red" stroke-width="2" />
</svg>
```

Se dibuja un círculo. Su centro va a estar en la posición `cx="100" cy="100"` y tiene un radio `r="50"`. Se rellenará de color azul `fill="blue"` y la línea del contorno será de color rojo `stroke="red"` con una anchura de 2 píxeles `stroke-width="2"`.

### 4.4 Elipses.

Se dibujan usando `ellipse`. Por ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <ellipse cx="100" cy="100" rx="50" ry="90" fill="blue" stroke="red" stroke-width="2" />
</svg>
```

Se dibuja una elipse. Su centro va a estar en la posición `cx="100" cy="100"` y los semiejes de la elipse tienen una longitud de `rx="50" ry="90"`. Se rellenará de color azul `fill="blue"` y la línea del contorno será de color rojo `stroke="red"` con una anchura de 2 píxeles `stroke-width="2"`.

## 4.5 Polilíneas.

Se usará path para realizar el dibujo de polilíneas. Este comando es algo complejo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <path d="M 10 10 L 190 10 L 100 190" fill="none" stroke="blue" stroke-width="1" />
</svg>
```

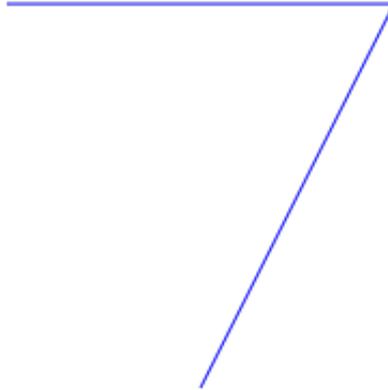
Cuando se usa path, debe indicar en el parámetro d la forma de la polilínea. Así en este caso se ha escrito: d="M 10 10 L 190 10 L 100 190"

M 10 10 significa que el punto de comienzo será (10, 10).

L 190 10 significa que a partir de la posición anterior se va a dibujar una línea hasta (190,10).

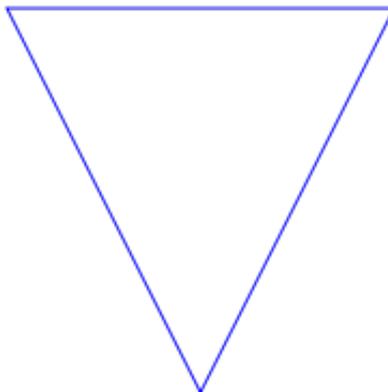
L 100 190 significa que a partir de la posición anterior se va a dibujar una línea hasta (100, 190).

El dibujo final será:



Usando z al final del parámetro d se puede indicar que se cierre la figura. Por ejemplo, si se le añade la z a la figura anterior:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <path d="M 10 10 L 190 10 L 100 190 z" fill="none" stroke="blue" stroke-width="1" />
</svg>
```



## 4.6 Textos.

Se usará text para insertar un texto:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <text x="50" y="50" font-family="Times" font-size="24" fill="blue" >
    Hello world
  </text>
</svg>
```

El texto comenzará en las coordenadas x="50" y="50" y usarán el tipo de letra Times font-family="Times".

El tamaño del texto será de 24 pixels font-size="24" y su color será azul fill="blue". Entre <text...> y

</text> se debe escribir el texto. En este caso se escribirá "Hello world".

## 4.7 Imágenes.

Se pueden insertar imágenes de los formatos jpeg y png dentro de una imagen SVG. Para ello se usa image.

Por ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <image x="50" y="50" width="64" height="64" href="icono_qtoctave2.png"/>
</svg>
```

Abrirá la imagen icono\_qtoctave2.png href="icono\_qtoctave2.png", en la posición x="50" y="50", con una anchura y alturas de width="64" height="64".

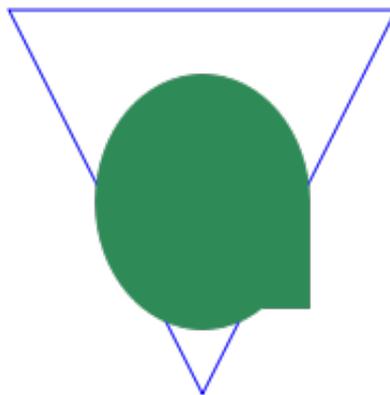
## 5. Grupos.

En la práctica puede ser práctico agrupar una serie de figuras. A todas ellas se les pueden dar los mismos atributos de color,... y cuando se editan con programas de edición como Inkscape se comportan como una sola figura, por lo que, en el caso de manejar una figura compleja puede ser útil.

Para crear un grupo se usa g. Dentro de los parámetros de g se puede poner un nombre al grupo usando id y aplicar atributos que se usarán en todas las figuras que estén entre <g...> y </g>. Por ejemplo:

```
<?xml version="1.0" standalone="yes"?>
<svg width="200" height="200" version="1.1">
  <path d="M 10 10 L 190 10 L 100 190 z" fill="none" stroke="blue" stroke-width="1" />
  <g id="Name" fill="seagreen" >
    <ellipse cx="100" cy="100" rx="50" ry="60" />
    <rect x="100" y="100" width="50" height="50"/>
  </g>
</svg>
```

La figura generada quedará:



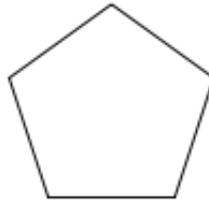
## 6. Usando SVG dentro de Octave.

Vamos a ver algunos ejemplos de uso de SVG dentro de Octave.

La siguiente función crea un polígono regular con los lados que se le indiquen:

```
function ej8(n)
  out=fopen("ej8.svg", "w");
  fprintf(out, '<?xml version="1.0" standalone="yes"?>\n');
  fprintf(out, '<svg width="200" height="200" version="1.1">\n');
  fprintf(out, '<path d="M 100 50');
  for k=1:n
    fprintf(out, "L %f %f", 100+50*sin(k/n*2*pi), 100-50*cos(k/n*2*pi) );
  endfor
  fprintf(out, '" fill="none" stroke="black" stroke-width="1"/>\n');
  fprintf(out, '</svg>\n');
  fclose(out);
endfunction
```

Si se ejecuta con ej8(5) se dibujará un pentágono:



El siguiente ejemplo dibuja una figura de Lissajous. A la función hay que pasarle la razón entre las frecuencias y la diferencia de fase:

```
function ej9(w, phase)
out=fopen("ej9.svg", "w");
fprintf(out, '<?xml version="1.0" standalone="yes"?>\n');
fprintf(out, '<svg width="200" height="200" version="1.1">\n');
fprintf(out, '<path d="M %f %f', 200, 100-100*sin(phase) );
n=1000;
for k=1:n
fprintf(out, " L %f %f", 100+100*cos(k/n*2*pi), 100-100*sin(w*k/n*2*pi+phase) );
endfor
fprintf(out, '" fill="none" stroke="black" stroke-width="1"/>\n');
fprintf(out, '</svg>\n');
fclose(out);
disp("Ok");
endfunction
```

Si se ejecuta con  $ej9(2, \pi/4)$  se obtiene la siguiente figura:

