

APENDICE E
OTROS OBJETOS SVG DOM

En este apéndice describiremos otros atributos, elementos y objetos SVG DOM que sirven para el manejo de documentos SVG. Al final de este apéndice se muestra un ejemplo de un documento SVG. Para más información se puede consultar en libros de SVG [SVG Un-2002] o en la página principal de SVG [SVG-2003].

Atributos y elementos del elemento SVG

Atributo “viewport”

Cuando un elemento SVG está anidado dentro de otro, el elemento SVG anidado abre un nuevo “viewport”. Esto tiene un número de efectos dependiendo de cómo el elemento SVG del documento y el elemento SVG anidado son usados. Los atributos de X e Y en el elemento SVG no tienen ningún significado si se agregan a un elemento SVG que no está anidado. Por otro lado si dicho elemento se encuentra anidado, las coordenadas X e Y indican donde empezará su “viewport” en relación con el del elemento SVG padre.

Elemento title

El elemento de title (título) provee información acerca del elemento SVG dentro del cual está anidado. El elemento title debe estar anidado dentro del elemento SVG del documento, en cuyo caso el visualizador del SVG escribirá el texto del elemento title en la barra de estado del navegador o del visualizador.

Elemento defs

El elemento `defs` contiene definiciones. Tales definiciones típicamente serán referenciadas y reusadas varias veces por otros elemento en el documento SVG. Los elementos dentro del elemento `defs` no son dibujados, pero pueden ser referenciados por otros elementos posteriormente en el documento.

Elemento symbol

El elemento `symbol` define objetos gráficos que pueden ser usados después en el documento SVG a través del elemento `use`. Este elemento no es dibujado en la pantalla, sino que es la instancia de un elemento símbolo, creada usando el elemento `use` el cual si es dibujado en la pantalla. El elemento `symbol` puede tener los atributos `viewbox` y `preserveAspectRatio`, que pueden ser usados para escalar o ajustar a un tamaño predeterminado.

Elemento “use”

Es común en un grafico tener varias formas similares dentro del mismo. Para ello nos sirve el elemento `use`, que nos permite hacer referencia a otros elementos, y nos permite ahorrar código y aligerar el documento. Este elemento solo puede hacer referencia a código que se encuentra dentro del mismo documento SVG.

Otros Objetos SVG DOM

Objeto GetSVGDocument

Es usado para regresar un documento SVG usando código script. Este objeto no tiene constantes ni propiedades específicas y tiene el siguiente método:

- `getSVGDocument ()` – el documento regresa un valor de tipo `SVGDocument`. El método no toma argumentos.

Objeto SVGElement

Este objeto corresponde a un elemento en el código SVG. Este objeto tiene las propiedades y métodos de los objetos `SVGElement`, `SVGURIReference`, `SVGTests`, `SVGLangSpace`, `SVGExternalResourceRequired`, `SVGStylable`, `SVGTransformable` y el elemento `EventTarget` de los eventos de DOM nivel 2. Este objeto no tiene métodos, pero tiene la siguiente propiedad:

- `target` – de tipo `SVGAnimatedString`.

Objeto SVGAngle

Es usado para especificar el tipo de unidades usadas para un ángulo específico. Tiene las siguientes constantes asociadas a él:

- `SVGAngle.SVG_ANGLETYPE_UNKNOWN` – es de tipo `short` y tiene un valor de 0.
- `SVGAngle.SVG_ANGLETYPE_UNSPECIFIED` – es de tipo `short` y tiene un valor de 1.

- `SVGAngle.SVG_ANGLETYPE_DEG` – es de tipo short y tiene un valor de 2.
- `SVGAngle.SVG_ANGLETYPE_RAD` – es de tipo short y tiene un valor de 3.
- `SVGAngle.SVG_ANGLETYPE_GRAD` – es de tipo short y tiene un valor de 4.

El objeto tiene las siguientes propiedades:

- `unitType` – de tipo unsigned short.
- `value` – de tipo flotante.
- `valueInSpecifiedUnits` – de tipo flotante.
- `valueAsString` – de tipo DOMString.

Tiene los siguientes métodos:

- `newValueSpecifiedUnits (unitType, valueInSpecifiedUnits)` – el método regresa un valor de tipo void. El argumento `unitType` es de tipo unsigned short, el argumento `valueInSpecifiedUnits` es de tipo flotante.
- `convertToSpecifiedUnits (unitType)` – el método regresa un valor de tipo void. El argumento `unitType` es de tipo unsigned short.

Objeto SVGCursorElement

Este objeto tiene las propiedades y métodos de los objetos `SVGElement`, `SVGURIReference`, `SVGTTests` `SVGExternalResourcesRequired`. El objeto no tiene ningún método específico, pero tiene las siguientes propiedades:

- `x` - de tipo `SVGAnimatedLength`.
- `y` - de tipo `SVGAnimatedLength`.

Objeto SVGDefsElement

Corresponde al elemento defs del documento SVG en el SVG DOM. Tiene todos los métodos y propiedades de los siguientes objetos: SVGElement, SVGTests, SVGLangSpace, SVGExternalResourcesRequired, SVGStylable, SVGTransformable y EventTarget de los eventos DOM nivel 2.

Objeto SVGDescElement

Corresponde al elemento desc del documento SVG en el SVG DOM. Tiene todas las propiedades y métodos de los siguientes objetos: SVGElement, SVGLangSpace, SVGStylable.

Objeto SVGDocument

Corresponde a la entidad documento de un documento SVG. Tiene los métodos y propiedades del objeto Document DOM nivel 2 y el objeto DocumentEvent DOM nivel 2. No tiene métodos específicos, pero si tiene propiedades específicas que son las siguientes:

- title – de tipo DOMString.
- referrer – de tipo DOMString.
- domain - de tipo DOMString.
- URL - de tipo DOMString.
- rootElement – un objeto de tipo SVGSVGElement.

Objeto SVGElementInstance

El elemento use tiene un objeto `SVGElementInstance` correspondiente en SVG DOM. Si hay múltiples elementos use un objeto `SVGElementInstancelist` tiene varios objetos `SVGElementInstance` asociados. Tiene los métodos y propiedades del objeto `EventTarget` de SVG DOM nivel 2. No tiene métodos específicos, pero tiene las siguientes propiedades específicas:

- `correspondingElement` – de tipo `SVGElement`.
- `correspondignUseElement` - de tipo `SVGUseElement`.
- `parentNode` – de tipo `SVGElementInstance`.
- `childNodes` – de tipo `SVGElementInstanceList`.
- `firstChild` – de tipo `SVGElementInstance`.
- `lastChild` – de tipo `SVGElementInstance`.
- `previuosSilbling` – de tipo `SVGElementInstance`.
- `Nextsibling` – de tipo `SVGElementInstance`.

Objeto `SVGElementInstanceList`

Las instancias de un elemento u objeto definidas dentro de elementos `defs` tienen un objeto `SVGElementInstanceList` correspondiente en SVG DOM. Tiene la siguiente propiedad:

- `length` – de tipo `unsigned long`.

Tiene el siguiente método específico:

- `item (index)` – el método regresa un valor de tipo `SVGElementInstance`. El argumento `index` es de tipo `unsigned long`.

Objeto `SVGEvent`

Tiene los métodos y propiedades del objeto `event` de DOM nivel 2. No tiene propiedades o métodos específicos.

Objeto `SVGExternalResourcesRequired`

No tiene ningún método específico, pero tiene la siguiente propiedad específica:

- `externalResourceRequired` – de tipo `SVGAnimatedBoolean`.

Objeto `SVGGElement`

Corresponde al elemento `g` de un documento SVG, en SVG DOM. Tiene todas las propiedades y métodos de los siguientes objetos: `SVGElement`, `SVGTests`, `SVGLangSpace`, `SVGExternalResourcesRequired`, `SVGStylable`, `SVGTransformable` y `eventTarget` de los eventos DOM nivel 2.

Objeto `SVGLangSpace`

Este objeto guarda información correspondiente a los atributos `xml:lang` y `xml:space`.

No tiene métodos específicos pero tiene las siguientes propiedades específicas:

- `xmlLang` – de tipo `DOMString`.
- `xmlspace` - de tipo `DOMString`.

Objeto SVGLengthList

Tiene la siguiente propiedad:

- `numberOfItems` – de tipo `unsigned long`.

Tiene los siguientes métodos:

- `clear ()` – regresa un valor de tipo `void`. No tiene argumentos.
- `Initialize (newItem)` – regresa un valor de tipo `SVGLength`. El argumento `newItem` es de tipo `SVGLength`.
- `getItem (index)` – regresa un valor de tipo `SVGLength`. El argumento `index` es de tipo `unsigned long`.
- `insertItemBefore (newItem, index)` – regresa un valor de tipo `SVGLength`. El argumento `newItem` es de tipo `SVGLength` y el argumento `index` es de tipo `unsigned long`.
- `replaceItem (newItem, Index)` – regresa un valor de tipo `SVGLength`. El argumento `newItem` es de tipo `SVGLength` y el argumento `Index` es de tipo `unsigned long`.
- `removeItem (index)` – regresa un valor de tipo `SVGLength`. El argumento `index` es de tipo `unsigned long`.
- `appendItem (newItem)` – el método regresa un valor de tipo `SVGLength`. El argumento `newItem` es de tipo `SVGLength`.

Objeto SVGLocatable

Tiene las siguientes propiedades:

- `nearestViewportElement` – de tipo `SVGElement`.
- `farthestViewportElement` – de tipo `SVGElement`.

Tiene los siguientes métodos:

- `getBBox ()` – regresa un valor de tipo `SVGRect`. No toma argumentos.
- `getCTM ()` – regresa un valor de tipo `SVGMatrix`. No toma argumentos.
- `getScreenCTM ()` – regresa un valor de tipo `SVGMatrix`. No toma argumentos.
- `getTransformToElement (element)` – regresa un valor de tipo `SVGMatrix`. El argumento `element` es de tipo `SVGElement`.

Objeto `SVGMetadataElement`

Tiene las propiedades y métodos del objeto `SVGElement`.

Objeto `SVGNumber`

Ni tiene métodos específicos, pero tiene la siguiente propiedad:

- `value` – de tipo flotante.

Objeto `SVGNumberList`

Tiene la siguiente propiedad:

- `numberOfItems` – de tipo `unsigned long`.

Tiene los siguientes métodos:

- `clear ()` – regresa un valor de tipo `void`. No toma argumento alguno.

- `Initialize (newItem)` – regresa un valor de tipo `SVGNumber`. El argumento `newItem` es de tipo `SVGNumber`.
- `getItem (index)` – regresa un valor de tipo `SVGNumber`. El argumento `index` es de tipo `unsigned long`.
- `insertItemBefore (newItem, index)` – regresa un valor de tipo `SVGNumber`. El argumento `newItem` es de tipo `SVGNumber` y el argumento `index` es de tipo `unsigned long`.
- `replaceItem (newItem, index)` – regresa un valor de tipo `SVGNumber`. El argumento `newItem` es de tipo `SVGNumber` y el argumento `index` es de tipo `unsigned long`.
- `removeItem (index)` – regresa un valor de tipo `SVGNumber`. El argumento `index` es de tipo `unsigned long`.
- `appendItem (newItem)` – regresa un valor de tipo `SVGNumber`. El argumento `newItem` es de tipo `SVGNumber`.

Objeto `SVGRect`

No tiene métodos específicos, pero tiene las siguientes propiedades:

- `x` – de tipo flotante.
- `y` - de tipo flotante.
- `width` - de tipo flotante.
- `height` - de tipo flotante.

Objeto `SVGScriptElement`

Es la representación SVG DOM de un elemento script en un documento SVG. Tiene las propiedades y métodos de los siguientes objetos: `SVGElement`, `SVGURIReference`, `SVGExternalResourcesRequired`. No tiene métodos específicos, pero tiene la siguiente propiedad específica:

- `type` - de tipo `DOMString`.

Objeto `SVGStringList`

Tiene las siguientes propiedades:

- `numberOfItems` - de tipo `unsigned long`.

Tiene los siguientes métodos:

- `clear ()` – regresa un valor de tipo `void`. No toma argumentos.
- `Initialize (newItem)` – regresa un valor de tipo `DOMString`. El argumento `newItem` es de tipo `DOMString`.
- `getItem (index)` – regresa un valor de tipo `DOMString`. El argumento `index` es de tipo `unsigned long`.
- `insertItemBefore (newItem, index)` – regresa un valor de tipo `void`. El argumento `newItem` es de tipo `DOMString`, el argumento `index` es de tipo `unsigned long`.
- `replaceItem (newItem, index)` – regresa un valor de tipo `DOMString`. El argumento `newItem` es de tipo `DOMString`, el argumento `index` es de tipo `unsigned long`.
- `removeItem (index)` – regresa un valor de tipo `DOMString`. El argumento `index` es de tipo `unsigned long`.

- `appendItem (newItem)` – regresa un valor de tipo `DOMString`. El argumento `newItem` es de tipo `DOMString`.

Objeto `SVGSymbolElement`

Es la representación SVG DOM del elemento `symbol` en un documento SVG. Tiene todos los métodos y propiedades de los siguientes objetos: `SVGElement`, `SVGLangSpace`, `SVGExternalResourcesRequired`, `SVGStylable`, `SVGFirToViewBox` y el objeto `EventTarget` de DOM nivel 2.

Objeto `SVGTests`

Tiene las siguientes propiedades específicas:

- `requiredFeatures` - de tipo `SVGStringList`.
- `requiredExtensions` – de tipo `SVGStringList`.
- `systemLanguage` - de tipo `SVGStringList`.

Tiene el siguiente método específico:

- `hasExtension (extension)` – regresa un valor de tipo booleano. Es argumento extensión es de tipo `DOMString`.

Objeto `SVGTitleElement`

Es la representación SVG DOM del elemento `title` de un documento SVG. Tiene todos los métodos y propiedades de los siguientes objetos: `SVGElement`, `SVGLangSpace` y `SVGStylable`.

Objeto SVGURIReference

No tiene métodos específicos, pero tiene la siguiente propiedad específica:

- href. – de tipo SVGAnimatedString.

Objeto SVGUseElement

Es la representación SVG DOM del elemento use de un documento SVG. Tiene los métodos y propiedades de los siguientes objetos: SVGElement, SVGURIReference, SVGTests, SVGLangSpace, SVGExternalResourceRequired, SVGStylable, SVGTransformable y el objeto eventTarget de de DOM nivel 2. No tiene métodos específicos, pero tiene las siguientes propiedades específicas:

- x – de tipo SVGAnimatedLength.
- y - de tipo SVGAnimatedLength.
- width - de tipo SVGAnimatedLength.
- height - de tipo SVGAnimatedLength.
- instanceRoot – de tipo SVGElementInstance.
- animatedInstanceRoot - de tipo SVGElementInstance.

Objeto SVGViewElement

Es la representación SVG DOM del elemento view en un documento SVG. Tiene las propiedades y métodos de los siguientes objetos: SVGElement, SVGExternalResourceRequired, SVGFitToViewBox y SVGZoomAndPan. No tiene métodos específicos, pero tiene la siguiente propiedad específica:

- viewTarget – de tipo SVGStringList.

Objeto SVGViewSpec

Tiene los métodos y propiedades de los siguientes objetos: SVGZoomAndPan y SVGFitToViewBox. No tiene métodos específicos, pero si tiene las siguientes propiedades específicas:

- transform – de tipo SVGTransformList.
- viewTarget – de tipo SVGElement.
- viewBoxString – de tipo DOMString.
- preserveAspectRatioString – de tipo DOMString.
- transformString - de tipo DOMString.
- viewTargetString – de tipo DOMString.

Objeto SVGZoomAndPan

Es la representación SVG DOM del valor del atributo ZoomAndPan en el elemento SVG de un documento SVG. Tiene las siguientes constantes asociadas:

- SVGZoomAndPan.SVG_ZOOMANDPAN_UNKNOWN – esta constante es de tipo short y tiene un valor de 0.
- SVGZoomAndPan.SVG_ZOOMANDPAN_DISABLE – esta constante es de tipo short y tiene un valor de 1.

- `SVGZoomAndPan.SVG_ZOOMANDPAN_MAGNIFY` – esta constante es de tipo `short` y tiene un valor de 2.

No tiene métodos específicos, pero tiene la siguiente propiedad específica:

- `zoomAndPan` – de tipo `unsigned short`.

Objeto `SVGZoomEvent`

Tiene las propiedades y métodos del objeto `UIEvent` de DOM nivel 2. No tiene métodos específicos, pero tiene las siguientes propiedades:

- `zoomRectScreen` – de tipo `SVGRect`.
- `previousScale` – de tipo flotante.
- `previousTranslate` – de tipo `SVGPoint`.
- `newScale` – de tipo flotante.
- `newTranslate` – de tipo `SVGPoint`.

Ejemplo SVG

El ejemplo que se muestra a continuación, es el documento SVG para generar dos polígonos de la capa Parques. En la Figura E se muestra el resultado gráfico.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE svg SYSTEM "svg11.dtd">
<svg id="allMap" onload="on_load(evt)" zoomAndPan="disable" width="800"
height="800" viewBox="0 0 800 800">
  <g visibility="visible" id="PARQUES.xml">
    <g id="Polygons_PARQUES.xml" style="fill: #83b23a; stroke: #4f6e1f;
stroke-width: 1;">
      <text visibility="hidden" id="descPolygon_1_PARQUES.xml">&lt;tr&gt;
&lt;td&gt;OBJECTID: &lt;/td&gt; &lt;td&gt;1&lt;/td&gt;
&lt;/tr&gt;&lt;tr&gt; &lt;td&gt;ID: &lt;/td&gt; &lt;td&gt;0&lt;/td&gt;
&lt;/tr&gt;</text>
      <path id="Polygon_1_PARQUES.xml" onclick="click(evt)"
onmouseover="over(evt)" onmouseout="out(evt)" d="M 799.47 525.45 L 797.61
524.33 L 795.57 522.24 L 791.40 519.83 L 788.57 517.43 L 786.64 516.04 L
784.94 514.84 L 781.51 511.73 L 778.85 509.71 L 776.85 508.21 L 772.24
505.37 L 770.08 504.43 L 765.54 502.22 L 760.15 499.37 L 756.12 496.44 L
753.86 496.05 L 751.42 496.21 L 749.20 496.54 L 747.20 496.91 L 745.68
497.24 L 744.38 497.67 L 743.17 498.33 L 741.61 499.29 L 739.86 500.69 L
738.16 503.35 L 736.57 507.19 L 735.03 510.63 L 733.59 512.62 L 731.28
514.76 L 727.48 517.51 L 721.98 520.58 L 715.74 523.12 L 710.35 525.23 L
705.28 526.82 L 701.22 527.77 L 695.64 529.17 L 693.04 530.89 L 690.62
532.48 L 690.27 532.79 L 690.01 533.34 L 689.86 533.82 L 689.88 534.35 L
690.47 536.19 L 691.95 539.25 L 692.36 540.25 L 693.35 541.30 L 708.78
555.46 L 716.18 562.47 L 717.46 559.78 L 717.57 554.54 L 717.74 551.72 L
719.88 549.45 L 721.78 547.96 L 723.52 547.54 L 725.39 547.53 L 726.91
547.98 L 728.72 548.96 L 730.80 550.07 L 733.06 551.15 L 736.99 551.99 L
738.26 552.30 L 740.85 552.89 L 747.68 553.88 L 750.58 553.86 L 752.61
553.48 L 754.80 549.19 L 756.70 546.06 L 757.81 544.26 L 761.08 542.49 L
764.36 539.42 L 765.83 534.31 L 767.73 527.71 L 768.99 527.10 L 770.40
526.38 L 771.19 526.52 L 771.75 526.63 L 772.29 526.72 L 773.00 526.88 L
775.14 528.21 L 776.50 531.37 L 776.98 536.44 L 779.26 539.65 L 781.43
540.25 L 783.78 539.99 L 787.73 538.74 L 790.63 537.47 L 793.72 534.97 L
795.98 533.39 L 797.90 529.81 L 800.0 525.95 L 799.47 525.45 Z" />
      <text visibility="hidden" id="descPolygon_2_PARQUES.xml">&lt;tr&gt;
&lt;td&gt;OBJECTID: &lt;/td&gt; &lt;td&gt;2&lt;/td&gt;
&lt;/tr&gt;&lt;tr&gt; &lt;td&gt;ID: &lt;/td&gt; &lt;td&gt;0&lt;/td&gt;
&lt;/tr&gt;</text>
      <path id="Polygon_2_PARQUES.xml" onclick="click(evt)"
onmouseover="over(evt)" onmouseout="out(evt)" d="M 682.42 554.63 L 681.31
555.19 L 678.57 555.72 L 678.12 555.92 L 678.05 556.28 L 677.97 556.80 L
678.35 557.30 L 678.85 558.02 L 679.17 558.77 L 680.30 560.07 L 681.12
561.63 L 682.47 562.88 L 684.63 564.37 L 687.26 566.15 L 688.06 566.80 L
689.01 568.77 L 689.58 569.93 L 690.55 571.01 L 690.76 571.54 L 690.86
573.15 L 690.91 574.31 L 690.57 575.84 L 690.26 576.54 L 689.54 577.98 L
690.04 579.16 L 690.89 580.80 L 691.83 580.08 L 692.99 579.71 L 694.52
578.92 L 695.85 578.34 L 697.29 577.77 L 698.82 577.33 L 700.22 576.89 L
702.19 575.60 L 704.02 572.99 L 704.74 571.52 L 705.55 569.82 L 706.00
```

```

568.89 L 707.65 568.04 L 710.12 566.74 L 712.47 565.48 L 715.41 564.08 L
716.09 562.68 L 702.28 549.97 L 693.14 541.71 L 691.71 539.66 L 690.59
537.46 L 689.95 537.42 L 689.03 537.41 L 688.06 537.78 L 687.47 538.11 L
686.91 538.67 L 686.50 539.55 L 686.22 541.12 L 684.87 545.40 L 685.01
546.63 L 684.03 547.56 L 683.57 547.20 L 683.28 546.94 L 682.90 546.65 L
682.11 546.05 L 681.53 545.69 L 680.83 545.71 L 680.30 546.41 L 680.13
546.75 L 680.64 548.69 L 681.89 551.56 L 683.19 554.22 L 682.42 554.63
Z" />
</g>
</g>

<script language="JavaScript"><![CDATA[
var label = "";
var svgstyle;
var style = "";
var style2 = "";
function on_load(e){
  svgdoc = e.target.getOwnerDocument();
  svgroot = svgdoc.getDocumentElement();
  var data = svgdoc.createTextNode("");
  var text = svgdoc.createElement("text");
  text.setAttribute("style", "stroke:black;stroke-
width:1;fill:black;font-size:20;font-family:arial;");
  text.setAttribute("id", "label");
  text.setAttribute("x", "0");
  text.setAttribute("y", "20");
  text.appendChild(data);
  svgroot.appendChild(text);
  var x;
}
function over(e) {
  target = e.currentTarget;
  var id = target.id;
  if (id!="") {
  label = svgdoc.getElementById("label");
  label.firstChild.setData(id);
  svgstyle = e.currentTarget.style;
  style = svgstyle.getPropertyValue('fill');
  style2= svgstyle.getPropertyValue('stroke');
  svgstyle.setProperty ('stroke', '#d43d2b');
  if (id.charAt(0)!='L'&&id.charAt(1)!='L')
    { svgstyle.setProperty ('fill', '#d43d2b');}
  }
}
function out(e) {
  var label = svgdoc.getElementById("label");
  label.firstChild.setData(" ");
  var svgstyle = e.currentTarget.style;
  target = e.currentTarget;
  var id = target.id;
  var
idG=id.substr(0,id.indexOf('_'))+"s"+id.substr(id.lastIndexOf('_'),id.len
gth);
  var gNode=svgdoc.getElementById(idG);
  svgstyle.setProperty('fill',gNode.style.getPropertyValue('fill'));

svgstyle.setProperty('stroke',gNode.style.getPropertyValue('stroke'));

```

```

    }
    function click (evt){
        target = evt.currentTarget;
        var id = target.id;
        label = svgdoc.getElementById("desc"+id);
        var info=label.firstChild.nodeValue;
        newwindow2=
open('', 'name', 'scrollbars=1,resizable=1,height=300,width=270');
        var tmp = newwindow2.document;
        tmp.write("<html> <head> <title> Informacion descriptiva </title>
<style type=\"text/css\"> <!--.datos { font-family: \"Trebuchet MS\",
Arial, Verdana, sans-serif; font-size: 12px; font-weight: normal; color:
#333333; background-color: #EEEEEE; }--> </style> </head> <body> <table
width=\"100%\" border=\"0\" cellpadding=\"0\" cellspacing=\"0\"
class=\"datos\"> <tr bgcolor=\"#FF9999\" > <td><strong>Property
</strong> <td><strong>Value</strong></tr> <tr> <td height=\"3\"
colspan=\"2\" bgcolor=\"#FFFFFF\"></td> </tr>");
        tmp.write(info);
        tmp.write("</table> </body> </html>");
        tmp.close();
    }
]]></script>
</svg>

```



Figura E. Ejemplo gráfico de SVG.