

# VCMG: Generador de mapas conceptuales vectoriales

Yuri Vizcarra González, Alberto Pacheco González

Departamento de Estudios de Posgrado,  
Instituto Tecnológico de Chihuahua  
México  
apacheco@itchiihuahua.edu.mx,  
yavizcar@hotmail.com

**Resumen.** Se describe una herramienta desarrollada en PHP para generar dinámicamente imágenes vectoriales de mapas conceptuales hipertexto a partir de presentaciones Web de ExpoVision.

## 1 Introducción

### 1.1 Presentaciones Web

La herramienta VCMG forma parte de una plataforma para asistir a la generación de presentaciones ultraligeras por Internet (Web Browser based) denominada ExpoVision.

### 1.2 Mapas conceptuales (MC)

MC es una técnica desarrollada en el departamento de Educación de la Universidad de Cornell en los 70's.

Los MC constituyen una respuesta a la teoría del aprendizaje significativo como parte del programa denominado "Aprender a Aprender". Asimismo, el término "aprendizaje significativo" de David Ausubel involucra asimilar conceptos nuevos mediante su inclusión en estructuras cognitivas ya existentes.

Se han convertido en un elemento muy importante en los planes de perfeccionamiento de los sistemas de enseñanza y han extendido su uso a otras esferas de la actividad humana en las que la gestión del conocimiento ocupa un lugar preponderante.

Un ejemplo de ello, es su uso en educación a distancia o educación semi-presencial, donde es fundamental el estudio independiente y el autoaprendizaje por parte de los estudiantes.

Los MC representan, simultáneamente, una estrategia de aprendizaje, un método para *captar lo más significativo de un tema* y un recurso esquemático para *representar un conjunto de significados conceptuales*, incluidos en una estructura de proposiciones.

### 1.3 SVG (Scalable Vector Graphics)

SVG es una representación basada en XML[3] de objetos geométricos vectoriales en 2D que pueden ser desplegados y manipulados mediante funciones propias SVG o scripts como *Javascript/ECMAScript, PHP* [2], etc. además de poseer las características de ser eficientes, versátiles, portables, ligeros e interactivos en la Web.

Es considerado un estándar abierto [4]-[6] amparado por el Consorcio Web (*W3C*) [7] además de ser el resultado del esfuerzo de compañías tales como: *Adobe, HP, IBM, Microsoft, Netscape, SUN Microsystems* y *Microsoft Visio* [15].

Fue propuesta como estándar gráfico 2D para la *Web* en Septiembre de 2001 [4], sucediéndole la especificación versión 1.1 en Noviembre de 2002 [5]. Actualmente se encuentra en desarrollo el draft de la especificación 1.2 [6].

*SVG* posee las siguientes cualidades:

- Código fuente compacto y portable.
- Código fuente editable en cualquier editor de texto (*i.e.* notepad).
- Provee escalamiento de imagen con mucha mayor resolución que archivos *GIF*.
- Permite ampliaciones-reducciones de imágenes (*zooming*) sin pérdida de resolución.
- Permite la generación de imágenes complejas mediante el uso de transformaciones tales como escalamientos y rotaciones.
- Es capaz de crear y manejar elementos gráficos basados en líneas, curvas, imágenes y textos.
- Permite agrupar, animar, filtrar, recortar (*clipping*), enmascarar (*masking*), y trasladar elementos.
- Puede ser generado mediante servlets de *Java, PHP* y *JSP*.
- Basado en texto estructurado con sintáxis *XML*, soporta *hiperenlaces* de *XLink* [12], hojas de estilos *CSS* [13] y *código embebido* de *JavaScript* [1].
- Soporta compactación *GZIP*, *zooming* y *panning* sin necesidad de recargar imágenes.
- Más compacto, rápido de transferir y versátil para transformar y manipular objetos que usando imágenes de *bitmaps*.
- Su independencia del dispositivo permite adaptar imagen a las capacidades del mismo, tales como resolución, tamaño, colores, etc. en computadoras personales, dispositivos móviles e impresoras.
- Hay diversidad de proveedores y supera ciertas deficiencias de tecnologías previas como *SGML* [8], *DrawML*, *HTML*, *VML* y *Flash* entre otros.
- Permite crear dinámicamente imágenes y manipularlas vía *JavaScript*.
- Gran versatilidad al interoperar con otros estándares *W3C* tales como *CSS*, *DOM* [17], *SMIL* y *JavaScript*.
- Gratuito, solo es necesario *Adobe SVG viewer* [16].
- Requiere conocimientos básicos de geometría y trigonometría.

## 2 Desarrollo

### 2.1 Funcionamiento

La herramienta *VCMG* genera mapas conceptuales a partir de presentaciones Web de *ExpoVision*. Los pasos que se siguen para la generación dinámica *Server-based* de los mapas conceptuales, como imágenes vectoriales *SVG*, son los siguientes:

1. Cuando un usuario consulta una presentación Web dentro del servidor de *ExpoVision*, aparece un hiperenlace dentro de la tabla de contenido de la presentación que invoca a *VCMG* para generar un mapa conceptual. Por ejemplo, tenemos la siguiente línea:

```
<embed src="map.php?f=ejemplo" width="900" height="600" type="image/svg+xml" />
```

2. *VCMG* (Script de PHP) verifica la existencia de un archivo auxiliar (z-file), que contiene un resumen de la presentación, generado en una consulta anterior, si no existe lo genera (paso 3). Si existe, verifica si la fecha de la presentación es más reciente que del archivo z-file auxiliar, si está actualizado pasa al paso 4.

3. Genera archivo auxiliar z-file. Se consulta el documento de la presentación y se extraen los elementos de texto necesarios para elaborar posteriormente un mapa conceptual. Para obtener dichos elementos se procede a: limpiar el documento de tags de HTML, espacios innecesarios, símbolos especiales y comandos del intérprete del generador de presentaciones Web de *ExpoVision*; derivar nombre corto de la presentación (concepto central); obtener títulos cortos de cada lámina (conceptos subordinados); generar resumen de láminas por temas (concatenar y compactar contenidos de láminas bajo un mismo título, además de señalar tablas, listados y figuras); derivar las relaciones entre los conceptos. Cabe mencionar que para realizar el análisis sintáctico anterior se utilizan intensivamente expresiones regulares de PHP. Por último, se graba en el servidor el archivo de texto auxiliar z-file que resume la presentación.

4. Generación dinámica de la imagen *SVG*: se usa una plantilla *SVG* como esqueleto de un mapa conceptual, dicha plantilla está embebida en un script de PHP. Se consulta el archivo auxiliar z-file, se analiza y construye un arreglo asociativo para representar los nodos y enlaces del mapa conceptual. Conforme se procesa el archivo z-file se registra e indexa cada elemento de texto, e.g. coordenadas para cada elemento gráfico, texto para cada nodo, relación, hiperenlaces y cuadros pop-up de texto. De acuerdo con el número de nodos resultante se procede a ubicar la información en los elementos gráficos *SVG* correspondientes.

Finalmente, aparece una nueva ventana en el navegador Web del usuario con una imagen que visualiza el mapa conceptual correspondiente para la presentación Web consultada.

5. El usuario puede interactuar con la imagen *SVG* del *MC* de distintas formas: al ubicar el cursor del ratón sobre un nodo, aparece un cuadro *pop-up* semi-opaco con un

resumen de las láminas asociadas a dicho concepto. Si el usuario se desplaza a otro lugar (fuera del nodo actual), el cuadro de texto desaparece gradualmente. En cambio, si el usuario da *click* sobre el nodo, la ventana del navegador que contiene la presentación *Web* despliega la lámina correspondiente para dicho concepto-nodo. Por otra parte, si selecciona un enlace nodo-a-nodo aparece un texto para describir la relación que existe entre un concepto y otro. Opcionalmente, el usuario puede ampliar/reducir la imagen, grabarla o imprimirla.

## 2.2 Componentes de la aplicación

A continuación se describe de manera visual cada uno de los principales componentes del gráfico *SVG* para un caso específico de un mapa conceptual que consta de cuatro nodos únicamente. Cada uno de sus elementos será explicado brevemente de manera alfabética.

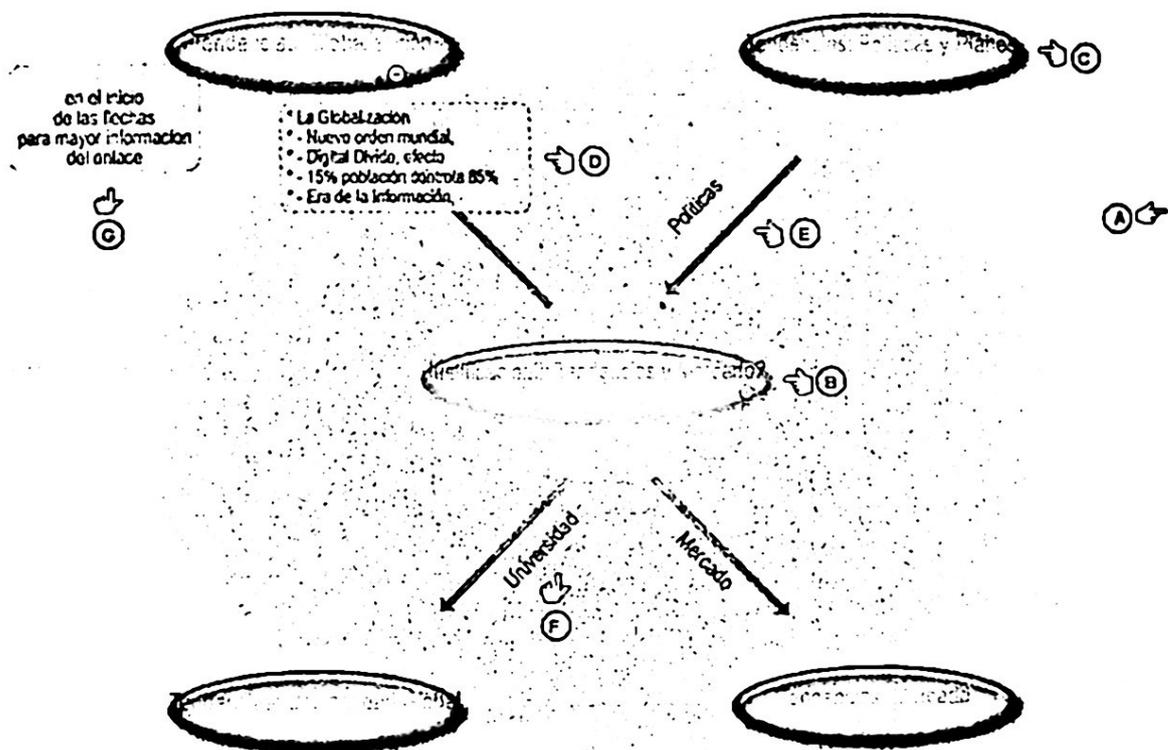


Fig. 1. Mapa Conceptual de Salida en Navegador Web

### 2.2.1 Gráfico vectorial (SVG) de salida

La figura 1 muestra la imagen completa generada tal y como aparece en el navegador Web del cliente.

### 2.2.2 Tema principal

Ubicado en el centro del gráfico, la figura 2 muestra la porción grafica donde aparece el nodo central (1) y arriba del nodo (2) se muestra el cuadro pop-up con la

información sobre el tema y los autores mostrada a partir de un evento "mouseover" (pose del cursor sobre el tema central).

Implementado mediante la etiqueta:

```
<g onmousedown="go(<?= $ID["#n{$n}pg"] ?>)"
opacity="0.5">
```

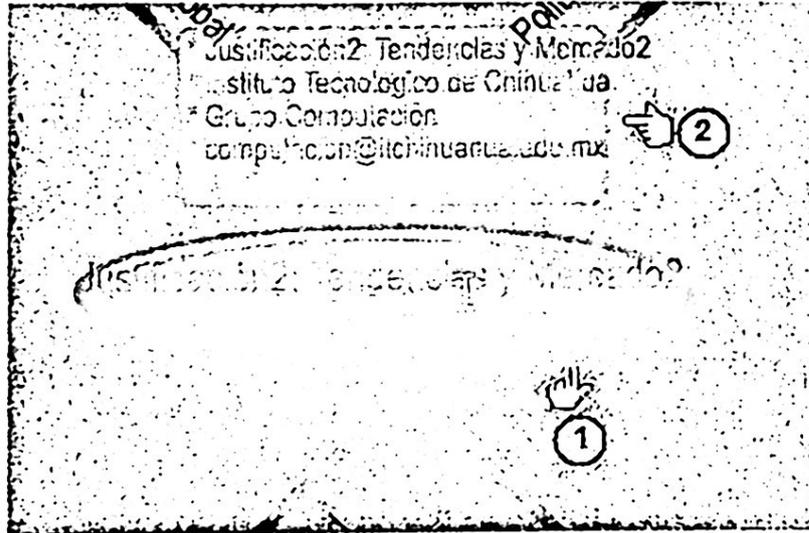


Fig. 2. Tema central e información adicional

### 2.2.3 Conceptos de tema principal

Los nodos externos, mostrados en la figura 3, representan conceptos del tema principal relacionados entre si mediante palabras enlace (1) y la capacidad de mostrar al usuario texto completo referente al tema mediante un evento de ratón (mousedown) (2).

Implementado mediante:

```
<g onmousedown="go(0)" opacity="0.5">
```

### 2.2.4 Información adicional del concepto

La visualización de esta información es activada por el usuario mediante el evento del ratón "mouseover", el puntero es colocado sobre el concepto.

La figura 3 (2) provee un recuadro de fondo de opacidad media (*translúcido*) con información adicional sobre el concepto en cuestión implementado mediante un cambio de un atributo CSS para, por ultimo, cambiar a opacidad nula (desvanecerse) en un período de 2 seg. (*Animación*) después de haber retirado el usuario el ratón del concepto (*mouseout*).

Implementado mediante:

```
<set attributeName="opacity" attributeType="CSS" begin="
n<?=$n?>.mouseover" to="1" fill="freeze" /> <animate
attributeName="opacity" begin="n<?=$n?>.mouseout"
from="1" to="0" dur="2s" fill="freeze" />
```

En la figura 4 se muestra otra característica importante. El hecho que cuando aparecen recuadros de información, si éste aparece sobre algún otro texto (*e.g. palabras enlace*)

dicho texto se desvanecerá durante el tiempo que esté visible el recuadro de información adicional en pantalla, esto con el fin de proveer mayor legibilidad.

Implementado mediante:

```
<animate attributeName="opacity" begin="n6.mouseover"
from="<?=$VIS['n6vis']?>" to="0" dur="1s"
fill="freeze" />
```

```
<animate attributeName="opacity" begin="n6.mouseout"
from="0" to="<?=$VIS['n6vis']?>" dur="1s"
fill="freeze" />.
```

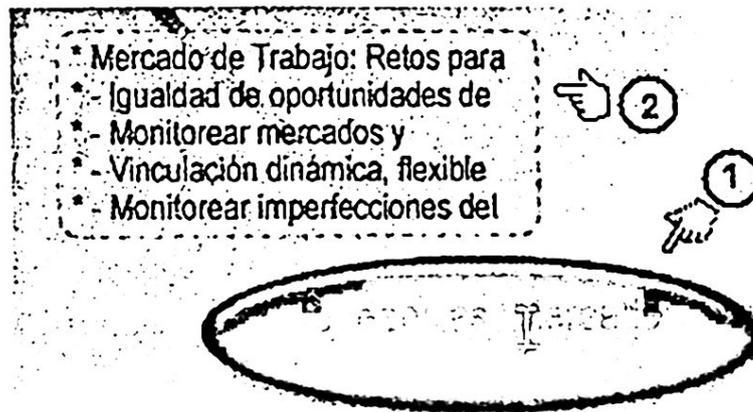


Fig. 3. Conceptos del tema e información adicional del concepto

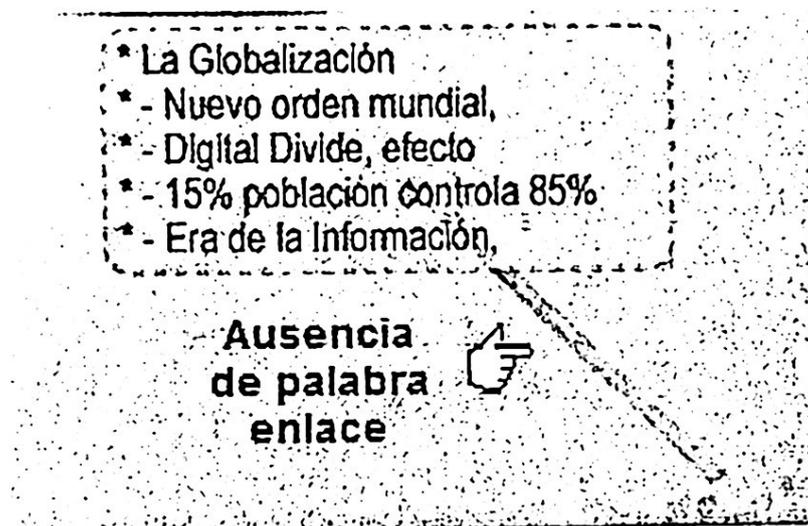


Fig. 4. Ocultamiento de palabras enlace

### 2.2.5 Relaciones nodo-a-nodo

Define brevemente la relación del concepto con respecto al tema principal mediante la dirección de su flecha.

Es posible conocer la relación mediante la identificación de palabras reservadas en el texto de entrada mediante análisis sintáctico implementado mediante expresiones regulares en PHP como se muestra:

```
If ( preg_match("/^%pg (\S+) (\S+)
([^:]+): (.+)$/S", $x, $t) ) {
    $t[2]=="to"? "#ArrowLeft": "#ArrowRight";
```

Además posee una manera muy particular de diferenciar su relación mediante diferenciación por color como se muestra en la figura 5, implementada mediante uso de hojas de estilo CSS.

```
<g id="ArrowRight">
    <path d="M 140 0 148 8 140 16 140 11 0 11 0 5 140
5" style="stroke-width:1; opacity:.4; fill:#909" />
</g>
<g id="ArrowLeft">
    <path d="M 0 8 8 0 8 5 148 5 148 11 8 11 8 16"
style="stroke-width:1; opacity:.4; fill:#900" />
</g>
```

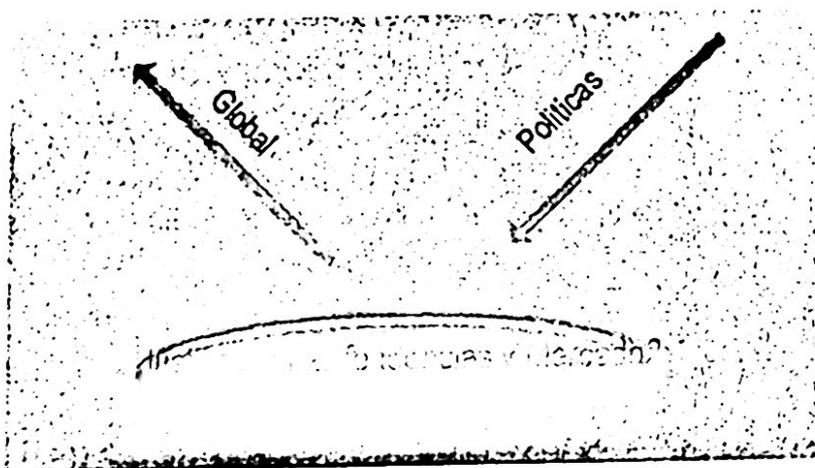


Fig. 5. Diferenciación de color en enlaces

### 2.2.6 Palabras enlace

Son las palabras que sirven para unir los conceptos y señalar el tipo de relación existente entre ambos [13].

Estas palabras son obtenidas mediante análisis sintáctico del texto de entrada. Implementado mediante:

```
if( preg_match("/^%pg (\S+) (\S+)
([^:]+): (.+)$/S", $x, $t) ) {
    $ID[$nId."Short"] = $t[3];
```

Las palabras son implementadas en el mapa conceptual mediante breves descripciones de texto alineadas sobre el “enlace nodo-a-nodo”, siendo estos permanentemente mostrados.

### 2.2.7 Información adicional de la palabra enlace

Son obtenidos también mediante análisis sintáctico e implementadas en el mapa conceptual mediante descripciones mas detalladas de la palabra enlace.

Implementado mediante:

```
if( preg_match("/^%pg (\S+) (\S+)
([^:]+):(.+)$/S", $x, $t)) $ID[$nId."Long"] = $t[4];
```

Son mostrados únicamente si el usuario realiza un evento de ratón “Mouse down” sobre el “hyperArrow” y desvanecidos cuando el botón del mouse es soltado (mouseup) como se indica a continuación en el siguiente código SVG.

```
<g id="n6linklong" opacity="0">
  <rect x="90" y="385" rx="8px" ry="8px" width="200"
height="32" class="tipce3" />
  <text x="100" y="405" class="d"> <?=$LONG['n6'] ?>
</text>
  <set attributeName="opacity" attributeTpe="CSS"
begin="n6link.mousedown" to="<?=$VIS['n6vis']?>"
fill="freeze" />
  <animate attributeName="opacity" begin="n6link.mouseup"
from="<?=$VIS['n6vis'] ?>" to="0" dur="1s"
fill="freeze" /> </g>
```

## 3 Conclusiones

La herramienta multiplataforma VCMG está completamente implementada en tecnologías basadas en estándares abiertos W3C y GNU además de poder ser utilizada en cualquier navegador Web.

Es capaz de generar imágenes hipermedia interactivas de mapas conceptuales de manera autónoma y dinámica a partir de presentaciones tipo Power Point.

Al aplicar mapas conceptuales como soporte para la generación de presentaciones Web se pretende demostrar su viabilidad y utilidad para:

- Auxiliar en la elaboración de presentaciones más didácticas y efectivas.
- Automatizar y facilitar la generación de resúmenes visuales de presentaciones.
- Visualizar y navegar de una manera más intuitiva en una presentación.
- Favorecer el aprendizaje significativo por su representación visual del conocimiento.

## Referencias

1. Javascript : The Definitive Javascript Resource. (2004)  
<http://www.javascript.com/>
2. PHP: Homepage Oficial de PHP. (2004)  
<http://www.php.net>
3. XML: Homepage Oficial de XML. (2003)  
<http://www.w3.org/XML>
4. W3C Recommendation : SVG 1.0 Specification. (Sep. 2001)  
<http://www.w3.org/TR/SVG>
5. W3C Recommendation : SVG 1.1 Specification . (Jan.2003)  
<http://www.w3.org/TR/SVG11>
6. W3C Working Draft : SVG 1.2 Specification. (Apr. 2003)  
<http://www.w3.org/TR/SVG12>
7. W3C : Sitio Oficial de World Wide Web Consortium. (2003)  
<http://www.w3.org>
8. W3C : Overview of SGML Resources. (2003)  
<http://www.w3.org/MarkUp/SGML/>
9. Pontificia Universidad Javeriana : Mapas conceptuales de Notoria (1997)  
<http://venus.javeriana.edu.co/qualitas/mayo97.htm>
10. Febles J., Estrada V.: Aplicaciones de la Inteligencia Artificial. Primera Edición, Parte III, Mapas Conceptuales. Pág. 400. Ed. Univ. de Guadalajara, México (2002)
11. Cuevas A.: Propuesta de Aplicación de los Mapas Conceptuales en un modelo Pedagógico Semipresencial. Instituto Superior Politécnico "José Antonio Echeverría" Cuba.
12. W3C : "XML Linking Language (XLink). (2003)  
<http://www.w3.org/XML/Linking>
13. W3C : CSS Level 2 Specification (CSS2). ( May 1998)  
<http://www.w3.org/TR/REC-CSS2>.
14. Pacheco A. : Servidor de ExpoVision (2004)  
<http://pantera.itch.edu.mx/apacheco/expo>
15. Campesato O. : Fundamentals of SVG Programming: Concepts to Source Code. Ch.R. Media, (2004)
16. Adobe : SVG Viewer. (2001)  
<http://www.adobe.com>.
17. Le Hors A. et al.(2000) "Document Object Model (DOM) Level 2 Core Specification," (2000).  
<http://www.w3.org/TR/DOM-Level-2-Core>