

TEXT CLASS REVIEW

TEMAS A TRATAR EN LA CUE

- Imágenes SVG.
- XML.
- Estructura de SVG.
- Elementos básicos que las componen.
- Integración de SVG.
- Optimización de SVG.

IMÁGENES SVG

O gráficos vectoriales escalables, corresponden a un formato de imágenes vectoriales bidimensionales, que pueden ser tanto animados como estáticos, y que están escritos en el formato XML (el cual explicaremos más adelante). A diferencia de otros formatos, como **.jpg** y **.png**, estos, como lo indica su nombre, son vectorizados, convirtiéndolos en escalables y con la posibilidad de aumentar su tamaño indefinidamente sin alterar la calidad. Es decir, son conjuntos de objetos geométricos que forman un diseño.

Actualmente, este es el formato más elegido por diseñadores gráficos, ya que permite replicar sus diseños en distintos tamaños, sin llegar a perder su nitidez, y además, los archivos de este tipo tienen un peso menor que otros como **.png**. Otra de sus ventajas, es que da la posibilidad de modificar su aspecto, ya sea: cambiar colores, insertar filtros o animar los diseños; tanto utilizando una interfaz gráfica (como Illustrator, Inkscape y Sketch), como alterando directamente el código XML.

XML

O “Extensible Markup Language”, es un metalenguaje de marcado que permite guardar y transportar datos, utilizando etiquetas y valores definidos por los desarrolladores, lo que lo convierte en una herramienta para definir lenguajes totalmente personalizados. Como se señaló anteriormente, las imágenes SVG están escritas en éste.

ESTRUCTURA DE SVG

Se define por algunos elementos que pueden o no, estar presentes dependiendo del editor con el que se exportó. Actualmente, editores como Sketch e Illustrator, han dejado de utilizar dichos elementos al estar migrando a SVG 2.0.

- **Declaración XML**

Es lo primero que podemos encontrar en el código, ésta permite definir la versión, el tipo de codificación, y si la imagen se basa en información de un elemento externo.

```
1 <?xml versión = '1.0' encoding = 'UTF-8' standalone = 'no' ?>
```

- **DECLARACIÓN DOCTYPE**

Al igual que en HTML, podemos encontrar este tipo de declaración, que define la versión de SVG que se está utilizando. Se recomienda utilizar la 1.1.

```
1 <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"  
2 http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd>
```

- **ETIQUETA SVG**

Es de gran importancia, ya que define las declaraciones que vimos anteriormente, en base a diversos atributos.

```
1 <svg version="1.1" xmlns="http://www.w3.org/2000/svg"  
2     xmlns:xlink=http://www.w3.org/1999/xlink xml:space="preserve">  
3     <!-- Código --></svg>
```

Veamos a qué corresponde cada definición:

- **version:** contiene la versión SVG del archivo.
- **xmlns:** hace referencia al XML **namespace**, que establece un conjunto de nombres de elementos y atributos, que podremos utilizar dentro de la etiqueta **<svg>**.
- **xmlns:xlink:** hace referencia al XML *Linking Language*, que permite, al igual que XML's, agregar características relacionadas a la vinculación de los elementos.

- **xml:space:** permite definir si se debe preservar los espacios en blanco.

ELEMENTOS BÁSICOS

Es importante conocerlos, ya que éstos definen propiedades de todo archivo SVG, tales como: tamaño del lienzo y contenido a ser visualizado.

En la etiqueta SVG, se pueden establecer atributos que determinen el tamaño del lienzo o viewport SVG (**width** y **height**), además de **viewBox**, el cual define el contenido del SVG que podrá ser visualizado.

El atributo **viewBox**, permite definir los parámetros y tamaño de visión. Puedes imaginarlo como una ventana a través de la cual visualizas el contenido, es decir, que si la ventana disminuye su tamaño, verás solo una parte de éste.

INTEGRACIÓN DE SVG

Existen distintas maneras de integrar una imagen SVG a nuestros sitios web.

1. *Usando la etiqueta ****:* para hacerlo de esta manera, solo debes agregar una etiqueta **** al documento HTML, y luego agregarle el atributo **src**, el cual contendrá la dirección o **path** del archivo que queremos integrar.

```
1 
```

2. *Insertándola como imagen de fondo:* solo debemos agregar un elemento HTML, como un div, con una clase o id. Luego, debemos insertar en la hoja de estilos CSS, una imagen de fondo utilizando **background-image**, que tendrá como valor la dirección de la imagen encerrada entre los paréntesis de **url()**. No olvides establecer una altura al div y, para evitar que la imagen se repita, la propiedad **background-repeat: no-repeat**.

```
1 <div class="imagen"></div>
```

```

1 .imagen {
2     Height: 200px;
3     background-image: url(imagen.svg);
4     background-repeat: no-repeat;
5 }

```

3. Utilizando la imagen SVG de forma **inline**: otra manera de agregar una imagen SVG, es insertando directamente su código en el documento HTML. Esta tiene la ventaja de que permite realizar modificaciones en la imagen, directamente desde el editor de texto.
4. Utilizando la etiqueta **<object>**: ésta permite agregar elementos multimedia, como: audios, videos, SVG, PDF, entre otros. Se debe insertar el atributo *data*, el cual tendrá como valor la dirección del archivo, y también el atributo **type**, en el que debemos especificar el tipo de elemento multimedia, que en nuestro caso será **image/svg+xml**.

Es importante agregar un **fallback** dentro de la etiqueta **<object>**, así se evitarán problemas de compatibilidad entre navegadores. Un **fallback** en CSS, corresponde a una regla que sirva como apoyo en caso de que una propiedad no tenga compatibilidad con algún navegador.

```

1 <object data="imagen.svg" type="image/svg+xml">
2     
3 </object>

```

OPTIMIZACIÓN DE SVG

En algunos casos, al revisar el código SVG de una imagen que queramos utilizar o que hayamos creado, nos encontraremos con etiquetas vacías como, por ejemplo, las etiquetas de grupo **<g>**. Estas líneas extras de código, ensuciarían nuestros documentos si se integra la imagen de forma **inline**, por lo que es recomendable eliminarlas. Existen dos maneras de hacerlo: manual, y utilizando alguna herramienta de optimización de SVG, como es [SVG Optimiser](#).

SVG Optimiser, corresponde a una herramienta creada por Peter Collingridge, la cual permite optimizar código SVG, dejándonos seleccionar los elementos que queremos mantener o eliminar, y cómo se organizará éste.

SVG optimiser

by Peter Collingridge

[Home](#) [About](#)

Because of how Scalable Vector Graphics (SVGs) are defined, they can be rewritten without affecting the resulting image. By optimising how SVGs are written, it is often possible to reduce file size and make them easier to work with. Files created in Inkscape or Illustrator are particularly amenable to optimisation.

How to use

1. Upload a file or paste the code in the box below.
2. Choose the options. They are applied immediately so you can quickly compare results.
3. Download the file or copy the code from the text area at the bottom of the page.

1. Upload

Sorry, uploading files is temporarily broken.

Or paste SVG code here:

Paste SVG code here, then click upload

Or

Imagen 1. Primer paso en SVG optimiser.

Como se puede observar en la imagen, solo debes copiar el código SVG que quieres optimizar, y pegarlo en la casilla correspondiente. Luego, procede a seleccionar las acciones a realizar.

- **Whitespace:** acá se puede definir si queremos reducir el código, eliminando todos los espacios en blanco (**remove**), u organizarlo (**pretty**).
- **Style type:** acá se debe definir si queremos que nuestro estilo sea "óptimo" (**optimal**), es decir, una mezcla entre los estilos **inline**, con los atributos **fill** y **stroke**, y estilos con clases al interior del elemento **<style>**. También pueden seleccionarse solo los estilos al interior de **<style>** (CSS), o solo los estilos **inline** (**styleString**).

- **Truncate attribute, SVG size and style numbers:** éstos permiten reducir los decimales correspondientes a los atributos de imagen, como `d`, y al tamaño del SVG. Además de limitar la cantidad de estilos en formas importantes dentro de la imagen.
- **Remove:** estos **checkbox** nos permitirán seleccionar los elementos que queremos remover de nuestro código.

2. Optimise

Options

Whitespace: remove

Style type: optimal

Truncate attribute numbers: 1 decimal places

Truncate SVG size numbers: 0 decimal places

Truncate style numbers: 2 significant figures

☐ Remove ids
☒ Remove default attributes
☒ Remove default styles
☒ Remove non-essential styles
☒ Remove empty elements
☒ Remove redundant shapes
☒ Remove clean group
☒ Apply transforms

Analysis

Original	Optimised
<ul style="list-style-type: none"> • Filesize: 0.9 kB • Elements: 1 	<ul style="list-style-type: none"> • Filesize: 0.9 kB (98.6%) • Elements: 2

Imagen 2. Segundo paso en SVG optimiser.

Optimizar nuestro código SVG, nos da la posibilidad de disminuir significativamente el tamaño de nuestros archivos, sin afectar la organización de éste ni el resultado visual.

Por último, tendremos el código ya optimizado.

3. Download

[Download](#) optimised SVG code

Or copy code below

```
<svg xmlns="http://www.w3.org/2000/svg" aria-hidden="true" focusable="false" data-prefix="fas" data-icon="address-card" class="svg-inline--fa fa-address-card fa-w-18" role="img" viewBox="0 0 576 512"><path d="M528 32H48C21.5 32 0 53.5 0 80v352c0 26.5 21.5 48 48h480c26.5 0 48-21.5 48-48V80c0-26.5-21.5-48-48-48zm-352 96c35.3 0 64 28.7 64 64s-28.7 64-64 64-64-28.7-64-64 28.7-64 64-64zm112 236.8c0 10.6-10 19.2-22.4 19.2H86.4C74 384 64 375.4 64 364.8v-19.2c0-31.8 30.1-57.6 67.2-57.6h512c0 5.1 25.7 8 39.8 8s27.6-2.9 39.8-8h512c0 67.2 25.8 67.2 57.6v19.2zM512 312c0 4.4-3.6 8-8 8H360c-4.4 0-8-3.6-8-16c0-4.4 3.6-8 8-8h144c4.4 0 8 3.6 8 8v16zm0-64c0 4.4-3.6 8-8 8H360c-4.4 0-8-3.6-8-16c0-4.4 3.6-8 8-8h144c4.4 0 8 3.6 8 8v16zm0-64c0 4.4-3.6 8-8 8H360c-4.4 0-8-3.6-8-16c0-4.4 3.6-8 8-8h144c4.4 0 8 3.6 8 8v16zm0-64c0 4.4-3.6 8-8 8H360c-4.4 0-8-3.6-8-16c0-4.4 3.6-8 8-8h144c4.4 0 8 3.6 8 8v16z" fill="currentColor"/></svg>
```

Imagen 3. Tercer paso en SVG optimiser.