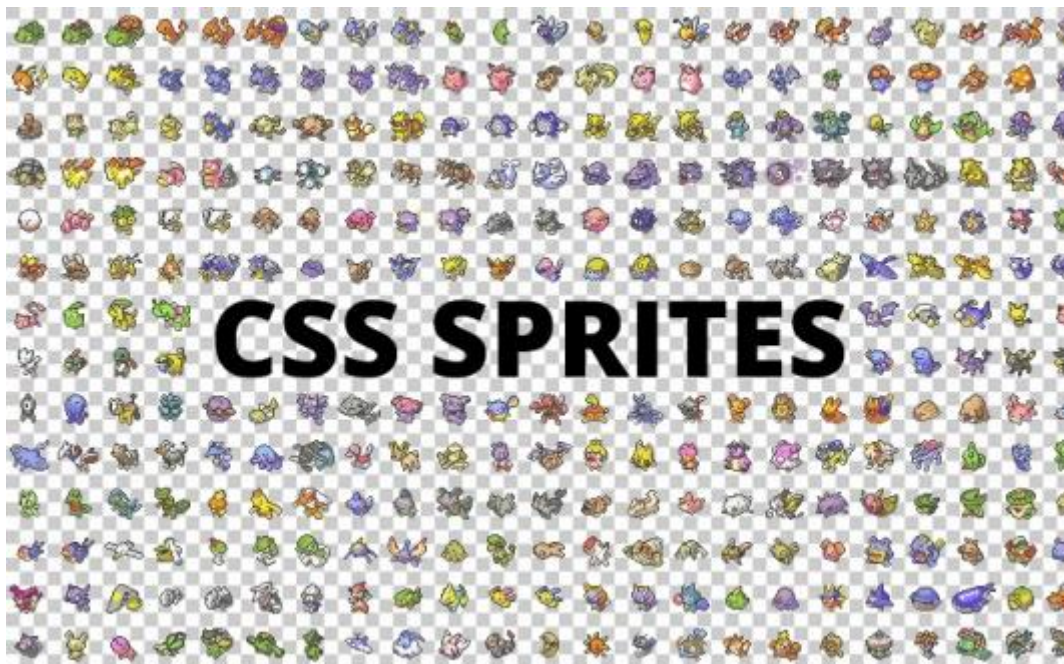


# Mejora tu velocidad de carga diseñando con CSS Sprites



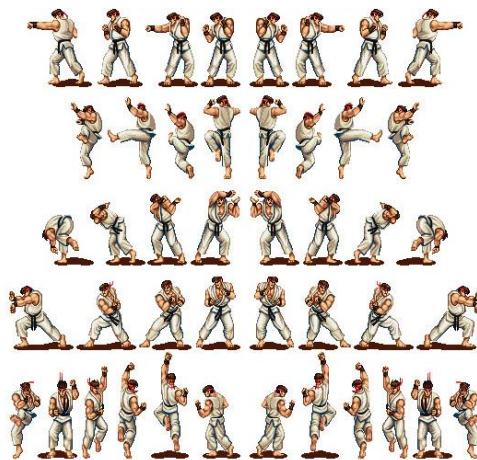
Si queremos que nuestra aplicación web tenga éxito y reciba miles de visitas cada día, además de ofrecer contenido interesante para nuestros usuarios, es fundamental que cargue de la forma más rápida posible, evitando que los visitantes tengan que esperar varios segundos hasta que la información aparezca.

Para conseguir esto, es fundamental contar con un [alojamiento web](#) que nos proporcione toda la potencia necesaria, pero además de eso, es fundamental optimizar el sitio, tanto en el apartado del diseño, como en la codificación y consultas a la base de datos.

A lo largo de este White Paper nos centraremos en una técnica de diseño llamada **CSS Sprite**, que nos **permite reducir el número de llamadas a las imágenes** que se utilizan para elaborar la parte gráfica de la web. Al reducir el número de llamadas, estaremos consiguiendo que nuestro sitio cargue más rápido.

## ¿En qué consiste CSS Sprite?

El uso de los sprites no es nada nuevo, ya que se trata de una técnica que se utilizaba hace tiempo en los videojuegos para ordenador. Consistía en una gran imagen de fondo que contenía todas las demás que se utilizaban en el videojuego. De esta forma, alterando la posición de la imagen de fondo se mostraba la imagen deseada en cada momento, lo que permitía hacer secuencias de movimientos sin tener que cargar cada vez las imágenes, o mostrar a muchos enemigos a partir de una única imagen.



Algunos sprites de Ryu para Super Street Fighter II Turbo, la imagen conjunta pesa 165 kB. Si las separamos, cada imagen de Ryu pesaría 5 kB, que multiplicado por las 44 animaciones que hay da un total de 220 kB.

Con el paso del tiempo y el desarrollo de [webs](#) cada vez más pesadas, los diseñadores tuvieron que desenterrar esta técnica para aplicarla al diseño. En este caso, lo que se hace es tener una sola imagen compuesta por todos los iconos o imágenes que formarán parte de la web, y haciendo uso de la propiedad `background-position` de CSS conseguimos visualizar cada imagen de forma individual.



Ejemplo de CSS Sprite con iconos planos y su variante en degradado

Así, si por ejemplo necesitamos usar el icono del email 20 veces (✉), con este método evitamos tener que meterlo como 20 imágenes, reduciendo el peso de la página y el tiempo de carga.

## Ventajas en el uso de CSS Sprite

Son dos las principales ventajas que ofrece esta técnica a la hora de desarrollar un portal web pero ambas buscan un fin en común, mejorar la velocidad de carga de la web.

**a) Reduce el consumo de recursos del servidor**

Al tener una única imagen, se reducen el número de peticiones HTTP. Si tuviéramos imágenes independientes, el navegador tendría que hacer una petición por cada imagen que necesitase por lo que el **servidor** debería atender todas esas peticiones. Al utilizar CSS Sprite, tenemos una única imagen por lo que el servidor sólo tendrá que atender una petición. Además, seleccionar la imagen que queremos utilizar en cada parte de la web mediante CSS es más rápido que realizar una petición al servidor para que este nos sirva esa información.

**b) Mejora la velocidad de carga de la web**

Cuantas más imágenes tengamos en una página web más despacio cargará el contenido. Si en cambio tenemos una imagen nuestra web será más liviana y cargará más rápidamente.

## Ejemplo de uso de CSS Sprite

Explicado el funcionamiento de esta técnica es hora de ponerlo en marcha y ver lo sencillo que resulta utilizarlo. En nuestro ejemplo tendremos una sección con iconos que enlazan a distintas redes sociales y a nuestro canal RSS. Al pasar el ratón por encima de uno de estos iconos, cambiaremos la imagen por otra.



En la imagen podemos ver mejor la idea. Al principio tenemos los iconos con un tono grisáceo que dan sensación de inactividad, pero al pasar el puntero del ratón se sustituirán por el mismo icono con color, indicando al usuario que se encuentra activo.

Lo primero que debemos hacer es crear nuestra imagen sprite que contengan todos los iconos que vamos a utilizar.



Imagen que en nuestro ejemplo llamamos "sprite-redessociales.png"

El siguiente paso será crear la estructura HTML que dará forma a nuestro ejemplo.

```
<div class="social">
  <a href="#" class="facebook"></a>
  <a href="#" class="twitter"></a>
  <a href="#" class="google"></a>
  <a href="#" class="youtube"></a>
  <a href="#" class="rss"></a>
</div>
```

En este caso creamos un contenedor (<div>) donde colocaremos los distintos enlaces a nuestros canales sociales. El siguiente paso será hacer uso de la propiedad “background-position” para hacer que aparezca cada icono en su sitio correspondiente. El código CSS que hace esto es el siguiente.

```
.social a.facebook {
  background: url(sprite-redessociales.png);
  background-position: -39px 0px;
}
```

Lo primero que haremos será indicarle que la imagen de fondo es aquella que contiene todos los iconos. Nosotros la hemos llamado “sprite-redessociales.png” y para indicar esto hacemos uso de la propiedad “background”.



A continuación, por medio de la propiedad “background-position” tenemos que indicar la posición del icono que queremos mostrar. Para indicar esta posición hay que medir desde la esquina superior izquierda y en dirección hacia la derecha y hacia abajo, indicando en estos casos valores negativos. En nuestro caso de la imagen de Facebook, nos hemos tenido que desplazar -39 px a la derecha y 0px hacia abajo, cero porque el inicio de la imagen del icono de Facebook está en la parte superior de la imagen.

A continuación, cuando el ratón pase por encima del icono de Facebook, deberemos cambiar la imagen por la de tono azul. Si nos fijamos, esta imagen está alineada verticalmente con la anterior, por lo que el desplazamiento horizontal será el mismo. Sólo nos quedará ajustar el vertical para que se muestre la imagen correcta.

```
.social a.facebook:hover { background-position: -39px -40px; }
```

En este caso, lo desplazamos 40px hacia abajo. Esto mismo hay que repetirlo con cada uno de los distintos iconos. Lo único será jugar con los valores para mostrar en cada sitio el icono correspondiente.



A continuación podéis ver el código del ejemplo completo.

```
<html>
  <head>
    <style type="text/css">
      .social{
        float: left;
        width: 97%;
        margin: 60px 0px 0px 80px;
      }

      .social a {
        display: inline-block;
        width: 40px;
        height: 40px;
        margin: 10px;

        /* Tiempo que dura la transición */
        -o-transition: all .8s; /* Opera */
        -moz-transition: all .8s; /* Firefox */
        -webkit-transition: all .8s; /* Chrome, Safari */
        -ms-transition: all .8s; /* Internet Explorer 9 o superior */
      }

      .social a.facebook {
        background: url(sprite-redessociales.png);
        background-position: -39px 0px;
      }
      .social a.facebook:hover { background-position: -39px -40px; }

      .social a.twitter {
        background: url(sprite-redessociales.png);
        background-position: 0px 0px;
      }
      .social a.twitter:hover { background-position: 0px -40px; }

      .social a.google {
        background: url(sprite-redessociales.png);
        background-position: -80px 0px;
      }
      .social a.google:hover { background-position: -80px -40px; }

      .social a.youtube {
        background: url(sprite-redessociales.png);
        background-position: -124px 0px;
      }
      .social a.youtube:hover { background-position: -124px -40px; }

      .social a.rss {
        background: url(sprite-redessociales.png);
        background-position: -165px 0px;
      }
      .social a.rss:hover { background-position: -165px -40px; }
    </style>
  </head>

  <body>
    <div class="social">
      <a href="#" class="facebook"></a>
      <a href="#" class="twitter"></a>
      <a href="#" class="google"></a>
      <a href="#" class="youtube"></a>
      <a href="#" class="rss"></a>
    </div>
  </body>
</html>
```

## Generadores Sprite

El problema que muchos pueden ver a esta técnica es el proceso de creación de las imágenes sprite. Esta tarea se puede hacer con algún programa editor de imágenes o bien aún más fácil por medio de herramientas online que nos podemos encontrar por la red.

Una de estas herramientas es el portal [csssprites](#), una web que nos permitirá subir las imágenes que queremos que formen parte del sprite y ella se encargará de crear el archivo único.

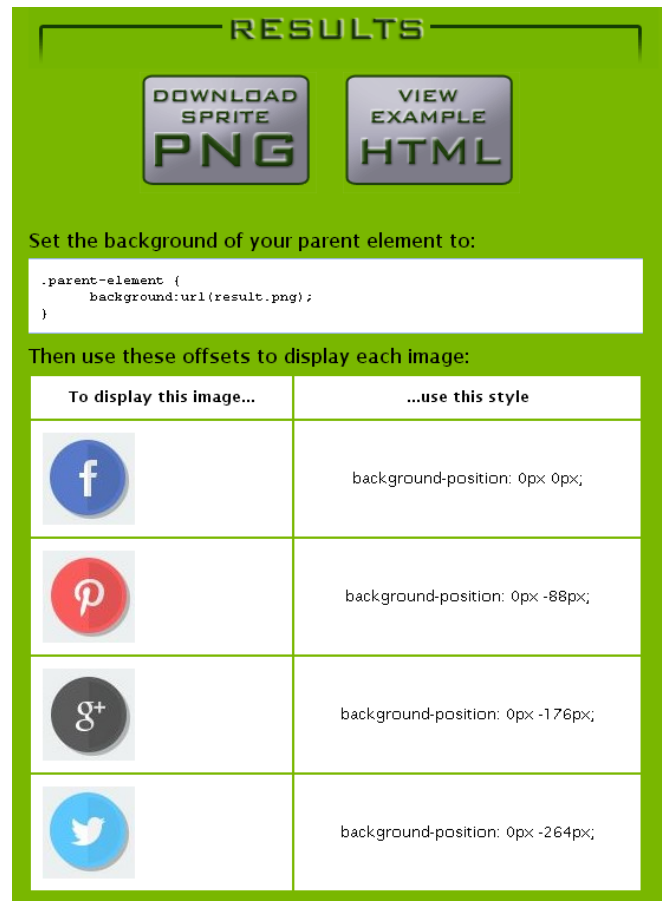


Una vez en ese sitio, deberemos indicar las imágenes que queremos subir. Por defecto permite subir tres imágenes, pero si queremos subir más lo único que tenemos que hacer es pulsar sobre el botón “**Need More?**”.



En nuestro caso hemos seleccionado las imágenes de Facebook, [Twitter](#), Pinterest y Google+. Una vez hecho esto, pulsamos en el botón “**Generate**” para que se genere nuestra única imagen que contendrá a todas las demás.









The screenshot shows a tool interface with a green border. At the top, it says "RESULTS" in a stylized font. Below that are two buttons: "DOWNLOAD SPRITE PNG" and "VIEW EXAMPLE HTML". Underneath, it says "Set the background of your parent element to:" followed by a code block:

```
.parent-element {  
  background:url(result.png);  
}
```

Then it says "Then use these offsets to display each image:" followed by a table:

To display this image...	...use this style
	background-position: 0px 0px;
	background-position: 0px -88px;
	background-position: 0px -176px;
	background-position: 0px -264px;

Además de permitir descargar la imagen en formato .png, esta herramienta nos indica de forma automática el valor que debe tener la propiedad “background-position” para cada una de las imágenes. De esta forma no tendremos que perder el tiempo en jugar con los valores de esa propiedad para mostrar cada una de las imágenes.

Como hemos podido ver a lo largo de este White Paper, el uso de la técnica CSS Sprite es algo sencillo y que nos puede ahorrar muchas peticiones al servidor, con lo que conseguimos mejorar la velocidad de la web.