

[developers.google.com](https://developers.google.com)

## Imágenes | Web | Google Developers

By *Pete LePage*

29-37 minutos



Pete is a Developer Advocate

El diseño web adaptable significa que no solo nuestros diseños pueden cambiar según las características del dispositivo: también lo puede hacer el contenido. Por ejemplo, en pantallas de alta resolución (2x), los gráficos de alta resolución garantizan nitidez. Una imagen con un ancho del 50% puede funcionar bien cuando el navegador tiene 800 px de ancho, pero en un teléfono más angosto, ocupará mucho espacio en pantalla y requerirá el mismo ancho de banda cuando se reduzca para entrar en una pantalla más pequeña.

### Dirección artística



En otros casos, es posible que se tenga que modificar la imagen con mayor agresividad: cambiar las proporciones, recortarla y hasta reemplazarla por completo. En este caso, generalmente el cambio

de imagen se denomina “dirección artística”. Consulta [responsiveimages.org/demos/](https://responsiveimages.org/demos/) para ver otros ejemplos.

## Responsive Images



Did you know that images account for more than 60% of the bytes on average needed to load a web page?

In this course you will learn how to work with images on the modern web, so that your images look great and load quickly on any device.

Along the way, you will pick up a range of skills and techniques to smoothly integrate responsive images into your development workflow. By the end of the course, you will be developing with images that adapt and respond to different viewport sizes and usage scenarios.

This is a free course offered through [Udacity](#)

[Take Course](#)

## Imágenes en lenguaje de marcado

El elemento `img` es poderoso: descarga, decodifica y representa contenido. Los navegadores modernos admiten una amplia gama de formatos de imagen. Incluir imágenes que sirvan en varios dispositivos es igual que en el escritorio y solo se requieren unos pequeños ajustes para que la experiencia sea óptima.

### TL;DR

- Utiliza tamaños relativos de imagen para evitar que sobrepasen el contenedor accidentalmente.
- Usa el elemento `picture` cuando quieras especificar distintas imágenes según las características del dispositivo (también conocido como dirección artística).
- Usa `srcset` y el descriptor `x` en el elemento `img` para proporcionarle indicios al navegador sobre cuál es la mejor imagen cuando se deba elegir entre distintas densidades.
- Si tu página solo tiene una o dos imágenes, y no se utilizan en ninguna otra parte del sitio, es buena idea usar imágenes incluidas por referencia para disminuir las solicitudes de archivo.

### Usa tamaños relativos para imágenes

Recuerda usar unidades relativas cuando se especifican los anchos de las imágenes a fin de evitar que accidentalmente excedan la ventana de visualización. Por ejemplo, con `width: 50%;`, el ancho de la imagen será el 50% del elemento contenedor (no el 50% de la ventana de visualización o el 50% del tamaño de píxeles real).

Dado que CSS permite que el contenido sobrepase el contenedor, es posible que debas usar `max-width: 100%` para evitar que las imágenes y otros contenidos excedan la capacidad. Por ejemplo:

```
img, embed, object, video {  
  max-width: 100%;  
}
```

Asegúrate de usar descriptores útiles en el atributo `alt` de los elementos `img`: brindan contexto a los lectores de pantalla y a otras tecnologías de asistencia para que tu sitio sea más accesible.

### Mejora `img` con `srcset` en dispositivos de muchos PPP

El atributo `srcset` mejora el comportamiento del elemento `img` y permite que sea más fácil proporcionar varios archivos de imagen en distintas características de dispositivos. Similar a la [función de CSS image-set](#) nativa a CSS, `srcset` permite que el navegador elija la mejor imagen según las características del dispositivo; por ejemplo, el uso de una imagen de 2x en una pantalla de 2x, o una imagen de 1x en un dispositivo de 2x (en un futuro) en una red con ancho de banda limitado.

```

```

En navegadores que no son compatibles con `srcset`, el navegador simplemente usa el archivo de imagen predeterminado especificado por el atributo `src`. Por esta razón, siempre es importante incluir una imagen de 1x que se pueda mostrar en cualquier dispositivo, independientemente de las capacidades. Cuando se admite `srcset`, la lista de imágenes o condiciones separadas por comas se analiza antes de realizar cualquier solicitud, y solo se descarga y se muestra la imagen más apropiada.

Si bien las condiciones pueden incluir desde densidad de los píxeles hasta longitud y altura, solo la densidad de los píxeles cuenta con una buena compatibilidad hoy en día. Para equilibrar el comportamiento actual con funciones futuras, continúa proporcionando simplemente la imagen de 2x en el atributo.

### Dirección artística en imágenes receptivas con `picture`





Para cambiar las imágenes según las características del dispositivo, también llamado dirección artística, usa el elemento `picture`. El elemento `picture` define una solución declarativa para proporcionar varias versiones de una imagen según distintas características, como el tamaño del dispositivo, su resolución, la orientación y otras.

Prueba interna: El elemento `picture` está llegando a los navegadores. Aunque todavía no se encuentra disponible en todos los navegadores, recomendamos usarlo debido a la sólida compatibilidad con versiones anteriores y al potencial uso del [polyfill Picturefill](#). Consulta el sitio [Responsivemaps.org](https://responsivemaps.org) para obtener más información.

Usa el elemento `picture` cuando el origen de una imagen exista en varias densidades o cuando un diseño adaptable indique que se debe usar una imagen un tanto diferente en ciertos tipos de pantallas. Similar al elemento `video`, se pueden incluir varios elementos `source`. De esta forma, es posible especificar distintos archivos de imagen a partir de consultas de medios o formatos de imagen.

```
<picture>
  <source media="(min-width: 800px)"
srcset="head.jpg, head-2x.jpg 2x">
  <source media="(min-width: 450px)"
srcset="head-small.jpg, head-small-2x.jpg 2x">
  
</picture>
```

[Pruébalo](#)

En el ejemplo anterior, si la longitud del navegador es 800 px, o más, se utiliza `head.jpg` o `head-2x.jpg` según la resolución del dispositivo. Si el navegador es entre 450 px y 800 px, se utiliza `head-small.jpg` o `head-small-2x.jpg`, también según la resolución del dispositivo. Para pantallas con una longitud menor que 450 px y para compatibilidad con versiones anteriores donde no se admite el elemento `picture`, el navegador representa el elemento `img` en su lugar y siempre hay que incluirlo.

### Imagen de tamaño relativo

Cuando no se conoce el tamaño final de la imagen, puede resultar difícil especificar un descriptor de densidad para las fuentes de imágenes. Esto es particularmente cierto para imágenes que abarcan un ancho proporcional al navegador y son fluidas, según el tamaño del navegador.

En lugar de proporcionar tamaños y densidades de imagen que sean fijos, puedes especificar el tamaño de cada imagen provista. Para hacerlo, agrega un descriptor de longitud junto al tamaño del elemento de imagen, y permite que el navegador calcule automáticamente la densidad adecuada de los píxeles y elija la mejor imagen que se deba descargar.

```

```

#### [Pruébalo](#)

En el ejemplo anterior, se representa una imagen que es la mitad de la longitud de la ventana de visualización (`sizes="50vw"`). Dependiendo del ancho del navegador y la relación de píxeles del dispositivo, se le permite al navegador elegir la imagen correcta independientemente de la longitud de la ventana del navegador. Por ejemplo, en la tabla siguiente se muestra la imagen que seleccionaría el navegador:

Ancho del navegador	Relación de píxeles del dispositivo	Imagen utilizada	Resolución adecuada
400 px	1	200 . png	1x
400 px	2	400 . png	2x
320 px	2	400 . png	2.5x
600 px	2	800 . png	2.67x
640 px	3	1000 . png	3.125x
1100 px	1	1400 . png	1.27x

#### Considera los puntos de interrupción en imágenes receptivas

Muchas veces, el tamaño de la imagen puede variar según los puntos de interrupción del diseño del sitio. Por ejemplo, en una pantalla pequeña probablemente te convenga que la imagen abarque el ancho completo de la ventana de visualización,

mientras que en pantallas más grandes solo debe tomar una proporción pequeña.

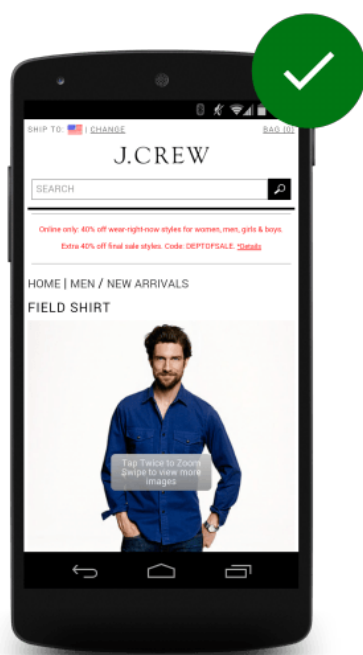
```

```

### [Pruébalo](#)

En el ejemplo anterior, el atributo `sizes` usa varias consultas de medios para especificar el tamaño de la imagen. Cuando el ancho del navegador es mayor que 600 px, la imagen es el 25% de la longitud de la ventana de visualización; cuando es entre 500 px y 600 px, la imagen es el 50% de la longitud de la ventana de visualización; y cuando es menor que 500 px, es el 100% de la longitud.

### Haz que las imágenes de producto sean expandibles



Sitio web de J. Crews con imágenes expandibles del producto.

Los clientes desean ver lo que comprarán. En sitios minoristas, los usuarios esperan poder ver primeros planos en alta resolución de los productos para poder observar mejor los detalles. [Quienes participaron del estudio](#) mostraron frustración cuando no pudieron hacerlo.

El sitio web de J. Crews es un buen ejemplo de imágenes expandibles y con capacidad táctil. Una superposición que desaparece indica que una imagen tiene capacidad táctil. Esto hace posible una imagen ampliada con buen nivel de detalle a la

vista.

## Otras técnicas de imagen

### Imágenes de compresión

La [técnica de imágenes de compresión](#) proporciona una imagen 2x muy comprimida a todos los dispositivos, independientemente de las capacidades reales del dispositivo. Según el tipo de imagen y el nivel de compresión, la calidad de la imagen no parecerá cambiar y el tamaño del archivo se reducirá de manera considerable.

[Pruébalo](#)

### Reemplazo de imágenes de JavaScript

El reemplazo de imágenes de JavaScript verifica las capacidades del dispositivo y “toma la decisión correcta”. Puedes determinar la relación de píxeles del dispositivo a través de `window.devicePixelRatio`, obtener el ancho y el alto de la pantalla e incluso posiblemente examinar conexiones de red a través de `navigator.connection` o emitiendo una solicitud falsa. Cuando hayas recolectado toda esta información, puedes decidir qué imagen cargar.

Una gran desventaja de este enfoque: el uso de JavaScript implica que retardarás la carga de imágenes al menos hasta que el analizador de lectura previa haya terminado. Esto significa que incluso las imágenes no comenzarán a descargarse hasta después de que el evento `pageLoad` se active. Además, el navegador seguramente descargará imágenes de 1x y 2x, lo cual aumentará el volumen de la página.

### Inserción de imágenes: de trama y vectoriales

Hay dos formas fundamentalmente diferentes de crear y almacenar imágenes; esto influirá en la manera de implementar imágenes adaptables.

Las **imágenes de trama**, como fotografías y otras imágenes, se representan como una cuadrícula de puntos de color individuales. Las imágenes de trama pueden provenir de una cámara o un escáner, o se pueden haber creado con el elemento `canvas` de HTML. Para almacenar imágenes de trama, se usan formatos como PNG, JPEG, y WebP.

Las **imágenes vectoriales**, como logotipos y arte lineal, se definen como un conjunto de curvas, líneas, formas, colores de relleno y degradados. Las imágenes vectoriales pueden crearse con programas como Adobe Illustrator o Inkscape, o de manera manual en código con un formato vectorial como SVG.

## SVG

SVG permite incluir gráficos vectoriales adaptables en una página web. La ventaja de los formatos de archivo vectoriales respecto de los formatos de archivo de trama es que el navegador puede representar una imagen vectorial en cualquier tamaño. Los formatos vectoriales describen la geometría de la imagen: cómo se construye a partir de líneas, curvas, colores, etc. Los formatos de trama, en cambio, solo tienen información sobre puntos individuales de color. Por lo tanto, el navegador debe calcular la forma de completar los espacios en blanco al realizar el ajuste.

A continuación, se muestran dos versiones de la misma imagen: una imagen PNG a la izquierda y un archivo SVG a la derecha. El archivo SVG se ve muy bien en cualquier tamaño, mientras que la imagen PNG junto a este comienza a verse borrosa en pantallas más grandes.



Si deseas reducir el número de solicitudes de archivo que realiza tu página, puedes codificar imágenes en línea con el formato SVG o de URI de datos. Si consultas el código fuente de esta página, verás que los siguientes logotipos se declaran incluidos por



referencia: un URI de datos y un SVG.

# HTML



# HTML



El archivo SVG ofrece [gran compatibilidad](#) en dispositivos móviles y equipos de escritorio, y las [herramientas de optimización](#) pueden reducir de manera considerable el tamaño del archivo SVG. Los dos logotipos SVG siguientes en línea parecen idénticos, pero uno tiene un tamaño aproximado de 3 KB y el otro solo ocupa 2 KB:

#### URI de datos

Los URI de datos proporcionan una manera de incluir un archivo, como una imagen, en línea configurando el `src` de un elemento `img` como una cadena codificada Base64 con el siguiente formato:

```

```

El comienzo del código para el logotipo HTML5 presentado arriba tiene el siguiente aspecto:

```

```

(La versión completa tiene más de 5000 caracteres de extensión).

Se encuentran disponibles herramientas de arrastrar y soltar, como [jpillora.com/base64-encoder](http://jpillora.com/base64-encoder), para convertir archivos ejecutables como imágenes para los URI de datos. Al igual que los archivos SVG, los URI de datos son [muy compatibles](#) con dispositivos móviles y navegadores de escritorio.

### Alineación en CSS

Los URI de datos y los SVG también se pueden incluir por referencia en CSS (tanto en dispositivos móviles como de escritorio). A continuación, se muestran dos imágenes idénticas que se implementaron como imágenes de fondo en CSS (un URI de datos y un SVG):

### Ventajas y desventajas de la inclusión por referencia

Si el código para incluir imágenes por referencia puede ser verboso, sobre todo en los URI de datos, ¿por qué lo usamos? ¡Para reducir solicitudes HTTP! Los archivos SVG y los URI de datos pueden permitir la recuperación de todos los elementos de una página web, como las imágenes y el lenguaje CSS y JavaScript, con una sola solicitud.

La desventaja:

- En dispositivos móviles, los URI de datos pueden tardar un tiempo [considerablemente mayor](#) en aparecer en dispositivos móviles cuando se comparan con imágenes de un `src` externo.
- Los URI de datos pueden aumentar considerablemente el tamaño de una solicitud HTML.
- Agregan complejidad a tu lenguaje de marcado y a tu flujo de trabajo.
- El formato de los URI de datos es considerablemente más grande que el ejecutable (hasta un 30%). Por lo tanto, no reduce el tamaño total de descarga.
- Los URI de datos no pueden almacenarse en caché. Por ello, deben descargarse para cada página en la que se usen.
- No son compatibles con IE 6 y 7 y la compatibilidad con IE 8 no es completa.

- Con HTTP/2, la reducción del número de solicitudes de recursos dejará de ser una prioridad.

Como todo lo que sea adaptable, es necesario probar cuál es la mejor opción. Usa herramientas del programador para medir el tamaño de la imagen descargada, la cantidad de solicitudes y la latencia total. A veces, los URI de datos pueden ser útiles para imágenes de trama. Por ejemplo, pueden ser útiles en una página principal que tenga una o dos fotografías que no se usen en otro lugar. Si debes insertar imágenes vectoriales, SVG es una opción mucho mejor.

## Imágenes en CSS

La propiedad `background` de CSS es una herramienta potente para agregar imágenes complejas a elementos, facilitar la incorporación de varias imágenes, repetirlas y más. Cuando se la combina con consultas de medios, esta propiedad adquiere aún más poder: permite cargar imágenes de forma condicional según la resolución de la pantalla, el tamaño de la ventana de visualización y más.

### TL;DR

- Usa la imagen que mejor se adapte a las características de la pantalla. Ten en cuenta el tamaño de la pantalla, la resolución del dispositivo y el diseño de la página.
- Cambia la propiedad `background-image` de CSS en pantallas de muchos PPP a través de consultas de medios con `min-resolution` y `-webkit-min-device-pixel-ratio`.
- Usa `srcset` para proporcionar imágenes de alta resolución además de la imagen 1x en lenguaje de marcado.
- Ten en cuenta los costos de rendimiento cuando uses técnicas de reemplazo de imágenes en JavaScript o cuando proporciones imágenes de alta resolución muy comprimidas a dispositivos de baja resolución.

### Usar solicitudes de medios para la carga condicional de imágenes o la dirección artística

Las consultas de medios no solo influyen en el diseño de la página: puedes usarlas para cargar imágenes de forma condicional o para proporcionar dirección artística según la longitud de la ventana de visualización.

Por ejemplo, en el siguiente código, solo se descarga y aplica `small.png` al div de contenido en pantallas pequeñas, mientras que en pantallas más grandes, `background-image: url(body.png)` se aplica al cuerpo y `background-image: url(large.png)` se aplica al div de contenido.

```
.example {
  height: 400px;
  background-image: url(small.png);
  background-repeat: no-repeat;
  background-size: contain;
  background-position-x: center;
}@media (min-width: 500px) {
  body {
    background-image: url(body.png);
  }
  .example {
    background-image: url(large.png);
  }
}
```

[Pruébalo](#)

### Usa image-set para proporcionar imágenes de alta resolución

La función `image-set()` en CSS mejora el comportamiento de la propiedad `background`, lo cual permite proporcionar de manera sencilla archivos de imágenes múltiples para dispositivos de diferentes características. Esto permite que el navegador elija la mejor imagen según las características del dispositivo; por ejemplo, el uso de una imagen de 2x en una pantalla de 2x, o una imagen de 1x en un dispositivo de 2x en una red con un ancho de banda limitado.

```
background-image: image-set(
  url(icon1x.jpg) 1x,
  url(icon2x.jpg) 2x
);
```

Además de cargar la imagen correcta, el navegador le cambia el tamaño de manera acorde. Es decir, el navegador presupone que las imágenes 2x tienen el doble de tamaño que las imágenes 1x. Por lo tanto, reduce el tamaño de la imagen 2x en un factor de dos para que parezca que la imagen tiene el mismo tamaño en la página.

El soporte para `image-set()` aún es nuevo, y solo es compatible con Chrome y Safari con el prefijo de proveedor `-webkit`. No olvides incluir una imagen de respaldo para cuando no se admita `image-set()`. Por ejemplo:

```
.sample {
  width: 128px;
  height: 128px;
  background-image: url(icon1x.png);
  background-image: -webkit-image-set(
    url(icon1x.png) 1x,
    url(icon2x.png) 2x
  );
}
```

```
);  
background-image: image-set(  
  url(icon1x.png) 1x,  
  url(icon2x.png) 2x  
);  
}
```

### [Pruébalo](#)

El código anterior cargará el recurso correspondiente en navegadores compatibles con `image-set`; de lo contrario, regresará al recurso 1x. La razón obvia es que pocos navegadores admiten `image-set()`, pero la mayoría sí es capaz de utilizar el recurso 1x.

### Usar consultas de medios para proporcionar imágenes de alta resolución o dirección artística

Las consultas de medios pueden crear reglas según la [relación de píxeles del dispositivo](#). De esta forma, se pueden especificar distintas imágenes para pantallas 2x en comparación con pantallas 1x.

```
@media (min-resolution: 2dppx),  
(-webkit-min-device-pixel-ratio: 2)  
{  
  /* High dpi styles & resources here */  
}
```

Chrome, Firefox y Opera admiten `(min-resolution: 2dppx)` estándar, mientras que Safari y Android requieren la sintaxis con prefijo del proveedor sin la unidad dppx. Recuerda que estos estilos solo se cargan si el dispositivo se adecua a la solicitud de medios y que debes especificar los estilos para el caso básico. Esto también añade el beneficio de garantizar la representación de un elemento si el navegador no admite consultas de medios sobre resolución.

```
.sample {  
  width: 128px;  
  height: 128px;  
  background-image: url(icon1x.png);  
}@media (min-resolution: 2dppx), /* Standard  
syntax */  
(-webkit-min-device-pixel-ratio: 2) /* Safari &  
Android Browser */  
{  
  .sample {  
    background-size: contain;  
    background-image: url(icon2x.png);  
  }  
}
```

### [Pruébalo](#)

También puedes usar la sintaxis de ancho mínimo para mostrar imágenes alternativas que dependan del tamaño de la ventana de visualización. La ventaja de esta técnica es que la imagen no se descarga si la consulta de medios no coincide. Por ejemplo, `bg.png` solo se descarga y se aplica en el `body` si el ancho del navegador es de 500 px o superior:

```
@media (min-width: 500px) {  
  body {  
    background-image: url(bg.png);  
  }  
}
```

## Usa SVG para íconos

Cuando agregues íconos a tu página, usa íconos SVG cuando sea posible o, en algunos casos, caracteres unicode.

### TL;DR

- Usa íconos SVG o unicode en lugar de imágenes de trama.

### Reemplaza íconos simples por unicode

Muchas fuentes incluyen compatibilidad con los numerosos glifos unicode, que se pueden usar en lugar de imágenes. A diferencia de las imágenes, las fuentes unicode cambian su tamaño y lucen bien independientemente de cómo aparezcan en pantalla (grandes o chicas).

Además del grupo de caracteres convencionales, unicode puede incluir símbolos de flechas (←), operadores matemáticos (√), formas geométricas (★), imágenes de control (▶), notas musicales (♫), letras griegas (Ω) y hasta piezas de ajedrez (♠).

Un carácter unicode se incluye del mismo modo que las entidades nombradas: `&#XXXX`, donde XXXX representa el número de carácter unicode. Por ejemplo:

You're a super `&#9733`;

You're a super ★

### Reemplazar íconos complejos por SVG

Si necesitas íconos más complejos, los íconos SVG son, por lo general, livianos y fáciles de usar, y su estilo se puede definir con CSS. SVG tiene varias ventajas sobre las imágenes de trama:

- Son gráficos vectoriales que se pueden ajustar a escala de manera infinita.
- CSS modifica el color, las sombras y la transparencia, y las animaciones son simples.

- Las imágenes SVG se pueden incluir por referencia directamente en el documento.
- Son semánticos.
- Proporcionan una mejor accesibilidad con los atributos adecuados.

With SVG icons, you can either add icons using inline SVG, like this checkmark:

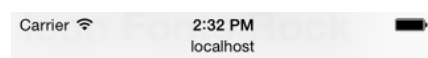
```
<svg version="1.1" xmlns="http://www.w3.org
/2000/svg"
      xmlns:xlink="http://www.w3.org
/1999/xlink"
      width="32" height="32" viewBox="0 0 32
32">
  <path d="M27 4l-15 15-7-7-5 5 12 12 20-20z"
fill="#000000"></path>
</svg>
```

or by using an image tag, like this credit card icon:



```
.
```





[Pruébalo](#)

### Usa fuentes de íconos con precaución



Font Awesome is an icon font with more than 350 different icons that can be easily customized, including size, color, shadow, etc. With Font Awesome, you can either add icons by using a unicode entity, like this HTML5 logo (🔗) or by adding special classes to an `<i>` element like the CSS3 logo (🔗).

			
&#xf083;	&#xf0c2;	&#xf0f4;	&#xf0e7;
.fa-camera-retro	.fa-cloud	.fa-coffee	.fa-bolt

			
&#xf108;	&#xf10a;	&#xf10b;	&#xf002;
.fa-desktop	.fa-tablet	.fa-mobile	.fa-search

[Ejemplo de una página que emplea FontAwesome para sus íconos de fuente.](#)

Las fuentes de íconos se usan mucho y pueden ser fáciles de usar, pero demuestran ciertas desventajas en comparación con los íconos SVG.

- Son gráficos vectoriales que se pueden ajustar a escala de manera

infinita, pero se pueden suavizar. Esto hace que los íconos no sean tan definidos como se espera.

- Ajuste de estilos limitados con CSS.
- Lograr un posicionamiento de píxeles ideal puede resultar difícil, según la altura de línea y el espaciado de letras, entre otros aspectos.
- No son semánticas y pueden ser difícil de usar en lectores de pantalla u otras tecnologías de asistencia.
- Si su ámbito no se establece correctamente, pueden terminar en archivos de gran tamaño en los que solo se usará un pequeño conjunto de los íconos disponibles.

With Font Awesome, you can either add icons by using a unicode entity, like this HTML5 logo (`<span class="awesome">&#xf13b;</span>`) or by adding special classes to an `<i>` element like the CSS3 logo (`<i class="fa fa-css3"></i>`).

[Pruébalo](#)

Hay cientos de fuentes de íconos disponibles de forma gratuita y no gratuita: [Font Awesome](#), [Pictos](#) y [Glyphicons](#) son algunos de ellos.

Asegúrate de equilibrar el peso de la solicitud HTTP adicional y el tamaño del archivo con la necesidad de íconos. Por ejemplo, si solo necesitas algunos íconos, tal vez sea mejor usar una imagen o un sprite de imagen.

## Optimiza las imágenes para mejorar el rendimiento

A menudo, las imágenes representan la mayor parte de los bytes descargados y ocupan una considerable proporción del espacio visual de la página. En consecuencia, la optimización de imágenes puede ser uno de los más eficaces medios de ahorro de bytes y de mejora de rendimiento para tu sitio web. Cuantos menos bytes deba descargar el navegador, menor será la exigencia del ancho de banda del cliente, y mayor la velocidad a la que el navegador podrá descargar y mostrar todo el contenido.

### TL;DR

- No debes elegir un formato de imagen de forma aleatoria: debes conocer los distintos formatos disponibles y usar el más adecuado.
- Para reducir el tamaño de los archivos, incluye herramientas de optimización y compresión de imágenes en tu flujo de trabajo.
- Para disminuir la cantidad de solicitudes HTTP, reemplaza las imágenes usadas frecuentemente con image sprites.



- Para mejorar el tiempo de carga inicial de la página y disminuir su peso inicial, una buena opción es cargar las imágenes cuando aparezcan en pantalla.

### Elige el formato correcto

Se deben elegir entre dos tipos de imágenes: [imágenes vectoriales](#) e [imágenes de trama](#). En las imágenes de trama, también debes elegir el formato de compresión adecuado (por ejemplo, GIF, PNG o JPG).

Las **imágenes de trama**, como fotografías y otras imágenes, se representan como una cuadrícula de puntos o píxeles individuales. Las imágenes de trama por lo general provienen de cámaras o escáneres, o pueden crearse en el navegador con el elemento canvas. A medida que aumenta el tamaño de la imagen, también aumentará el tamaño del archivo. Cuando su tamaño supera el tamaño original, las imágenes de trama se ven borrosas porque el navegador necesita adivinar cómo rellenar los píxeles faltantes.

Las **imágenes vectoriales**, como logotipos y arte lineal, se definen como un conjunto de curvas, líneas, formas y colores de relleno. Las imágenes vectoriales se crean con programas como Adobe Illustrator o Inkscape y se guardan en un formato de vectores como [SVG](#). Como las imágenes vectoriales se construyen a partir de primitivas simples, se pueden cambiar de tamaño sin perder calidad y sin realizar cambios en el tamaño del archivo.

Cuando elijas el formato adecuado, es importante tener en cuenta tanto el origen de la imagen (trama o vector) como el contenido (colores, animaciones, texto, etc.). No hay un formato que se adecue a todos los tipos de imagen; todos tienen sus ventajas y desventajas.

Para elegir el formato adecuado, comienza con estas pautas:

- Usa JPG para imágenes fotográficas.
- Usa SVG para arte vectorial y gráficos de colores sólidos, como los de logotipos y arte lineal. Si arte vectorial no está disponible, intenta con WebP o PNG.
- Usa PNG en lugar de GIF, ya que permite más colores y ofrece mejores relaciones de compresión.
- Para animaciones de mayor duración, es buena idea usar `<video>`, que proporciona una mejor calidad de imagen y le permite al usuario controlar la reproducción.

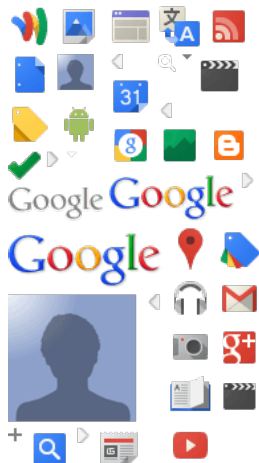
### Reducir el tamaño del archivo

Puedes disminuir considerablemente el tamaño del archivo de imagen si realizas un “procesamiento posterior” de las imágenes después de guardarlas. Existen varias herramientas de compresión

de imagen: con pérdida y sin pérdida, en línea, GUI y línea de comandos. Cuando sea posible, será mejor intentar automatizar la optimización de la imagen de modo que ocupe un lugar prioritario en tu flujo de trabajo.

Se encuentran disponibles varias herramientas que realizan mejores compresiones y sin pérdida en archivos JPG y PNG, y que no modifican la calidad de la imagen. Para JPG, prueba [jpegtran](#) o [jpegoptim](#) (disponible solamente en Linux; ejecuta con la opción `--strip-all`). Para PNG, prueba [OptiPNG](#) o [PNGOUT](#).

### Usa image sprites



Sprites con CSS es la técnica mediante la cual se combinan varias imágenes en una sola imagen de “hoja de sprites”. A continuación, puedes usar imágenes individuales si especificas la imagen de fondo de un elemento (u hoja de sprites) además de un desplazamiento para mostrar la parte adecuada.

```
.sprite-sheet {  
  background-image: url(sprite-sheet.png);  
  width: 40px;  
  height: 25px;  
}.google-logo {  
  width: 125px;  
  height: 45px;  
  background-position: -190px -170px;  
}.gmail {  
  background-position: -150px -210px;  
}.maps {  
  height: 40px;  
  background-position: -120px -165px;  
}
```

#### [Pruébalo](#)

El uso de sprites de imágenes tiene la ventaja de reducir el número de descargas necesarias para obtener varias imágenes y, al mismo tiempo, permitir el almacenamiento en caché.

## Considera la carga diferida

La carga diferida puede acelerar significativamente las cargas en páginas extensas que incluyan varias imágenes en la parte de la página que todavía no se mostró. Este método carga las imágenes según sea necesario, o cuando el contenido principal ya se ha cargado y representado. Además de mejoras en el rendimiento, el uso de la carga diferida puede generar experiencias de desplazamiento infinito.

Ten cuidado al crear páginas con desplazamiento infinito: dado que el contenido se carga a medida que se muestra, es posible que los motores de búsqueda nunca vean dicho contenido. Además, es posible que los usuarios que busquen información de pie de página nunca la encuentren porque se carga contenido nuevo continuamente.

## No utilices imágenes

A menudo, la mejor imagen no es una imagen. Cuando sea posible, usa las capacidades nativas del navegador para proporcionar la misma funcionalidad o una similar. Los navegadores generan elementos visuales que antes hubiesen requerido imágenes. Es decir, los navegadores ya no necesitan descargar archivos de imagen separados; de esta forma, se evita que aparezcan imágenes con tamaños extraños. Puedes usar unicode o fuentes especiales de íconos para representar íconos.

## Coloca texto en lenguaje de marcado y no en imágenes

Cuando sea posible, inserta texto en formato de texto y no en imágenes. Por ejemplo, si usas imágenes para los encabezados o colocas información de contacto (como números telefónicos o direcciones) directamente en imágenes, los usuarios no podrán copiar ni pegar la información, los lectores de pantalla no podrán acceder a los datos y la información no será adaptable. Como alternativa, ubica el texto en tu lenguaje de marcado y, si es necesario, usa fuentes web para alcanzar el estilo que necesitas.

## Usa CSS para reemplazar imágenes

Los navegadores modernos pueden usar características de CSS para crear estilos que antes hubiesen requerido imágenes. Por ejemplo, se pueden crear degradados complejos con la propiedad `background`, se pueden colocar sombras con `box-shadow` y se pueden agregar esquinas redondeadas con la propiedad `border-radius`.

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Pellentesque sit amet augue eu magna scelerisque porta ut ut dolor. Nullam placerat egestas nisl sed sollicitudin. Fusce placerat, ipsum ac vestibulum porta, purus dolor mollis nunc, pharetra

vehicula nulla nunc quis elit. Duis ornare fringilla dui non vehicula.  
In hac habitasse platea dictumst. Donec ipsum lectus, hendrerit  
malesuada sapien eget, venenatis tempus purus.

```
<style>
  div#noImage {
    color: white;
    border-radius: 5px;
    box-shadow: 5px 5px 4px 0
    rgba(9,130,154,0.2);
    background: linear-gradient(rgba(9, 130, 154,
1), rgba(9, 130, 154, 0.5));
  }
</style>
```

Ten en cuenta que para usar estas técnicas se necesitan ciclos de  
representación, requisito que puede ser considerable para los  
dispositivos móviles. Si los utilizas demasiado, perderás los  
beneficios obtenidos y es posible que disminuya el rendimiento.