



Nombre: Rodrigo Mena Serna

Asignatura: Diseño de Interfaces Web

Segundo curso de Desarrollo de Aplicaciones Web

Nombre de la práctica: Documentación Inspeccionar rendimiento imágenes

Tabla de contenido

Imágenes receptivas.....	3
Imágenes en marcado	3
Resumen.....	3
Utilice tamaños relativos para las imágenes.....	3
Dirección de arte en imágenes responsive con picture	4
Imágenes de tamaño relativo	4
Elija el formato correcto.....	5
Ejercicio de ejemplo	6
Unidades de ventana gráfica.....	9
Las unidades de ventana gráfica grande, pequeña y dinámica	9
La ventana gráfica y sus unidades	9
La necesidad de nuevas unidades de ventana gráfica	10
Bibliografía	12

Imágenes receptivas

Con el diseño receptivo, no solo nuestros diseños pueden cambiar según las características del dispositivo, sino también las imágenes.

El diseño web receptivo significa que no solo nuestros diseños pueden cambiar según las características del dispositivo, sino que el contenido también puede cambiar. Por ejemplo, en pantallas de alta resolución (2x), los gráficos de alta resolución garantizan la nitidez. Una imagen que tiene un ancho del 50% puede funcionar bien cuando el navegador tiene un ancho de 800 px, pero usa demasiado espacio en un teléfono estrecho y requiere la misma sobrecarga de ancho de banda cuando se reduce para adaptarse a una pantalla más pequeña.

Imágenes en marcado

El elemento `img` es poderoso: descarga, decodifica y presenta contenido, y los navegadores modernos admiten una variedad de formatos de imagen. La inclusión de imágenes que funciona en todos los dispositivos no es diferente a las de escritorio, y solo requiere algunos ajustes menores para crear una buena experiencia.

Resumen

- Use tamaños relativos para las imágenes para evitar que se desborden accidentalmente del contenedor.
- Utilice el elemento `picture` cuando desee especificar diferentes imágenes según las características del dispositivo.
- Use `srcset` y el descriptor `x` en el elemento `img` para dar sugerencia al navegador sobre la mejor imagen para usar al elegir entre diferentes densidades.
- Si su página solo tiene una o dos imágenes y estas no se usan en ningún otro lugar de su sitio, considere usar imágenes en línea para reducir las solicitudes de archivos.

Utilice tamaños relativos para las imágenes

Recuerde usar unidades relativas cuando especifique el ancho de las imágenes para evitar que se desborden accidentalmente de la ventana gráfica. Por ejemplo, `width: 50%`; hace que el ancho de la imagen sea el 50% del elemento contenedor (no el 50% de la ventana gráfica ni el 50% del tamaño real en píxeles).

Debido a que CSS permite que el contenedor se desborde de su contenedor, es posible que deba usar `max-width: 100%` para evitar que las imágenes y otro contenido se desborden. Por ejemplo:

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

Asegúrese de proporcionar descripciones significativas a través del `alt` atributo de los elementos `img`; estos ayudan a que su sitio sea más accesible al brindar contexto a los lectores de pantalla y otras tecnologías de asistencia.

El `srcset` atributo mejora el comportamiento del elemento `img`, lo que facilita proporcionar múltiples archivos de imagen para diferentes características del dispositivo. Similar a la `image-set` función CSS de CSS, `srcset` permite que el navegador elija la mejor imagen dependiente de las características del dispositivo, por ejemplo, usando una imagen 2x en una pantalla 2x, y potencialmente en el futuro, una imagen 1x en un dispositivo 2x cuando en una red de ancho de banda limitado.

```

```

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

Dirección de arte en imágenes responsive con `picture`

Para cambiar las imágenes en función de las características del dispositivo, también conocido como dirección de arte, utilice el elemento `picture`. El elemento `picture` define una solución declarativa para proporcionar múltiples versiones de una imagen en función de diferentes características, como el tamaño del dispositivo, la resolución del dispositivo, la orientación y más.

Utilice el elemento `picture` cuando exista una fuente de imagen en múltiples densidades, o cuando un diseño receptivo dicte una imagen algo diferente en algunos tipos de pantallas. Al igual que el elemento, se pueden incluir varios elementos `video`, lo que permite especificar diferentes archivos de imagen según las consultas de medios o el formato de la imagen.`source`.

```
<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>
```

En el ejemplo anterior, si el ancho del navegador es de al menos 800 px, entonces se usa `head.jpg` o `head-2x.jpg`, según la resolución del dispositivo. Si el navegador tiene entre 450 px y 800 px, entonces se usa `head-small.jpg` o `head-small-2x.jpg`, nuevamente, dependiendo de la resolución del dispositivo. Para anchos de pantalla inferiores a 450 px y compatibilidad con versiones anteriores donde el elemento `picture` no es compatible, el navegador representa el elemento `img` en su lugar y siempre debe incluirse.

Imágenes de tamaño relativo

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

En lugar de proporcionar densidades y tamaños de imagen fijos, puede especificar el tamaño de cada imagen suministrada agregando un descriptor de ancho junto con el tamaño del elemento de la imagen, lo que permite que el navegador calcule automáticamente la densidad de píxeles efectiva y elija la mejor imagen para descargar.

```

```

El ejemplo anterior representa una imagen que tiene la mitad del ancho de la ventana gráfica (`sizes="50vw"`) y, según el ancho del navegador y la proporción de píxeles de su dispositivo, permite que el navegador elija la imagen correcta independientemente del tamaño de la ventana del navegador. Por ejemplo, la siguiente tabla muestra qué imagen elegiría el navegador:

ancho del navegador	Proporción de píxeles del dispositivo	Imagen utilizada	Resolución efectiva
400px	1	200.jpg	1x
400px	2	400.jpg	2x
320px	2	400.jpg	2.5x
600px	2	800.jpg	2.67x
640px	3	1000.jpg	3.125x
1100px	1	800.png	1.45x

Elija el formato correcto

Hay dos tipos de imágenes a considerar: imágenes vectoriales e imágenes rasterizadas. Para las imágenes rasterizadas, también debe elegir el formato de compresión correcto, por ejemplo: GIF, PNG, JPG.

Las imágenes rasterizadas, como las fotografías y otras imágenes, se representan como una cuadrícula de puntos o píxeles individuales. Las imágenes ráster generalmente provienen de una cámara o un escáner, o se pueden crear en el navegador con el elemento canvas. A medida que aumenta el tamaño de la imagen, también lo hace el tamaño del archivo. Cuando se escalan más grandes que su tamaño original, las imágenes rasterizadas se vuelven borrosas porque el navegador necesita adivinar cómo completar los píxeles que faltan.

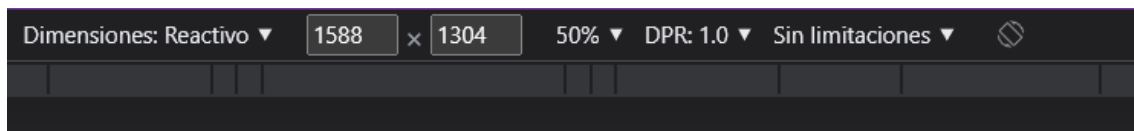
Las imágenes vectoriales, como logotipos y dibujos lineales, se definen mediante un conjunto de curvas, líneas, formas y colores de relleno. Las imágenes vectoriales se crean con programas como Adobe Illustrator o Inkscape y se guardan en un formato vectorial como SVG. Debido a que las imágenes vectoriales se crean a partir de primitivas simples, se pueden escalar sin pérdida de calidad ni cambios en el tamaño del archivo.

Comience con estas pautas al elegir el formato apropiado:

- Uso JPG para imágenes fotográficas.
- Úselo SVG para arte vectorial y gráficos de colores sólidos, como logotipos y arte lineal. Si el arte vectorial no está disponible, intente WebP o PNG.
- Use PNG en lugar de GIF, ya que permite más colores y ofrece mejores relaciones de compresión.
- Para animaciones más largas, considere usar `<video>`, que proporciona una mejor calidad de imagen y le da al usuario control sobre la reproducción.

Ejercicio de ejemplo


Lo primero que haremos será poner la proporción de píxeles del dispositivo en 1.



La imagen que se nos mostrará será head.jpg.

The `<picture>` element

Note: this sample uses the `picturefill` polyfill to provide backwards compatibility for browsers that do not support the `<picture>` element.



Viewport widths 0 to 449px:

- If `<picture>` is not supported, the fallback `` (150px×150px) is used
- 1x display: head-fb.jpg (150px×150px)
- 2x display: head-fb-2x.jpg (300px×300px)

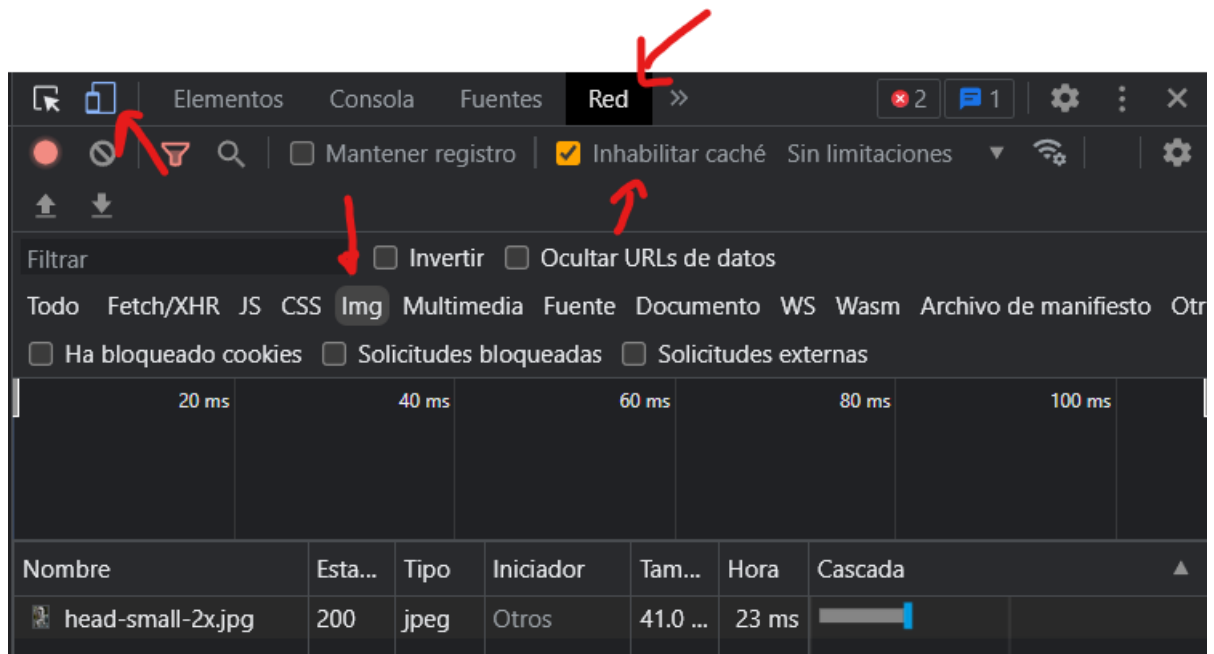
Viewport widths 450px to 799px:

- 1x display: head-small.jpg (200px×133px)
- 2x display: head-small-2x.jpg (400px×267px)

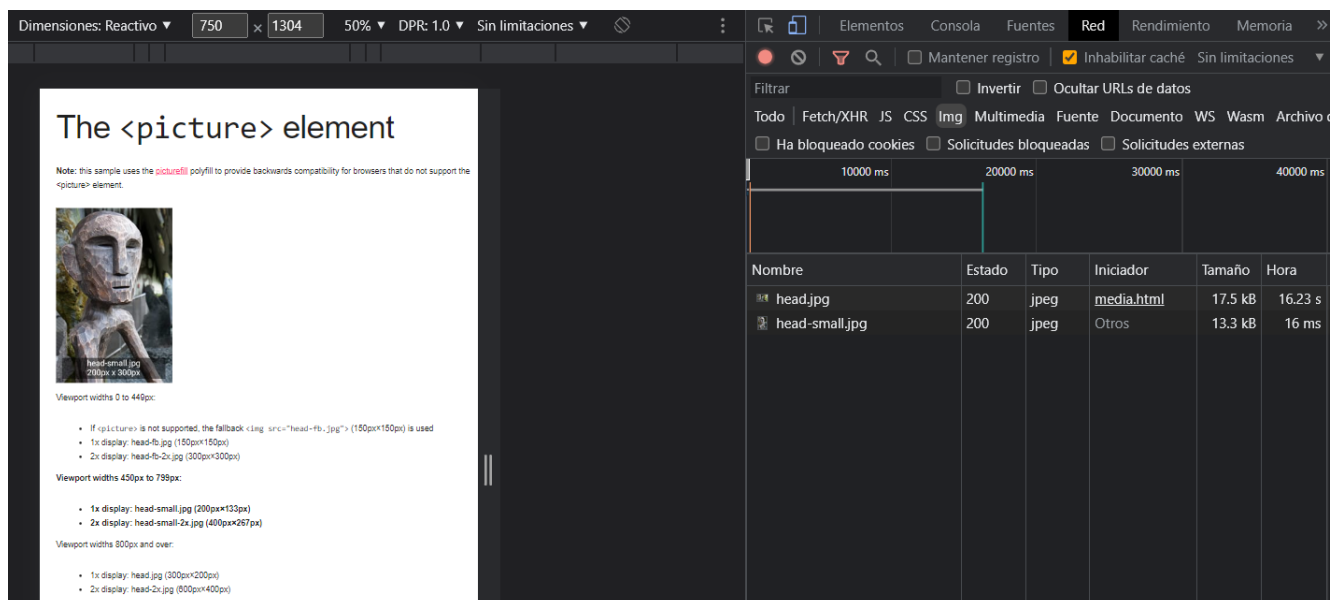
Viewport widths 800px and over:

- 1x display: head.jpg (300px×200px)
- 2x display: head-2x.jpg (600px×400px)

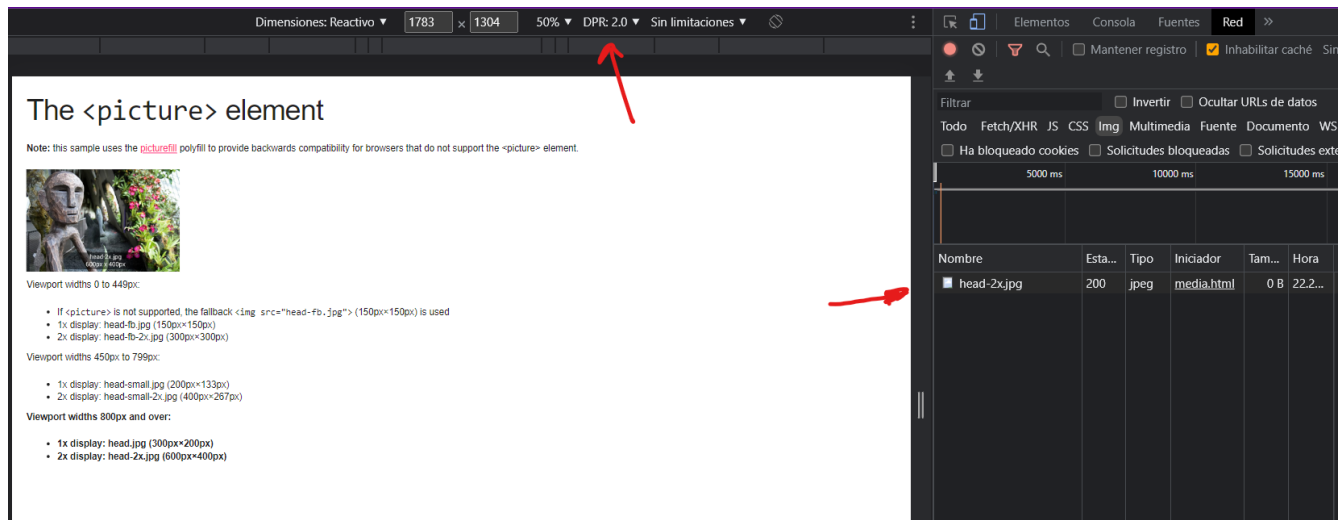
Para inspeccionar el rendimiento de una imagen abriremos la herramienta para desarrolladores en nuestro navegador. Una abierto, nos iremos al apartado “Red”, pulsaremos en activar/desactivar barra de herramientas del dispositivo, inhabilitaremos caché y por último pulsaremos en la opción “Img” para que nos muestro solo la información relacionada con las imágenes.



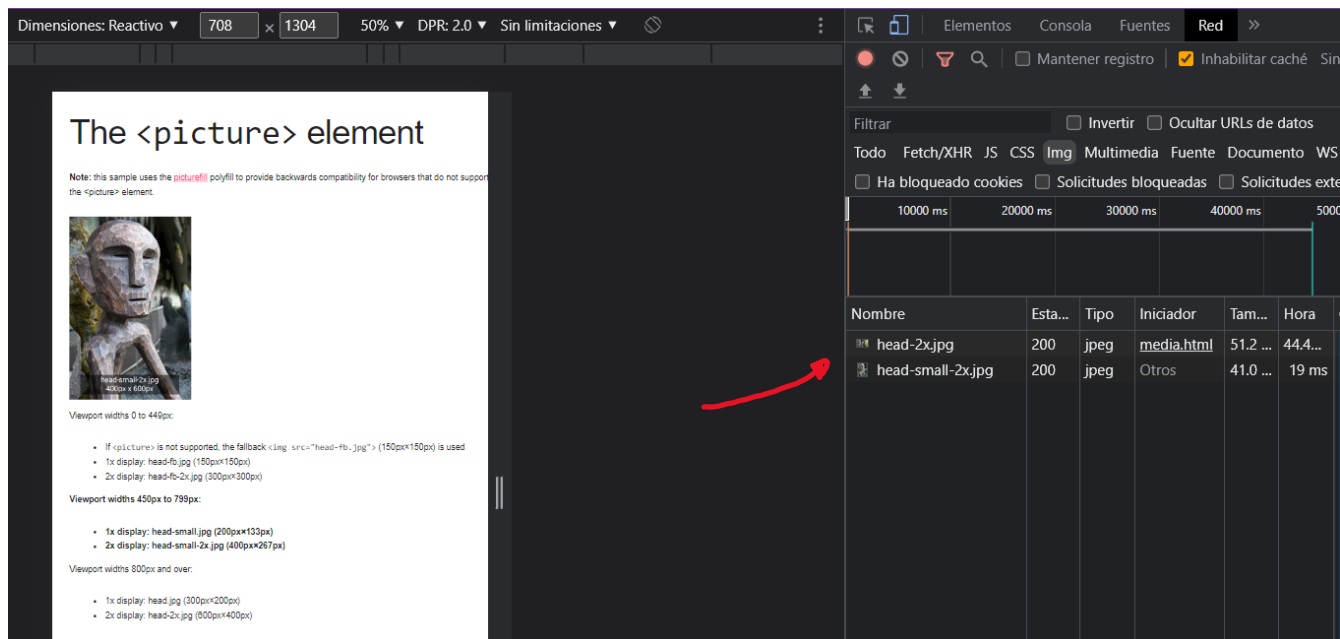
Si disminuimos la dimensión de la pantalla podremos observar como la foto ha cambiado y ahora es head-small.jpg.



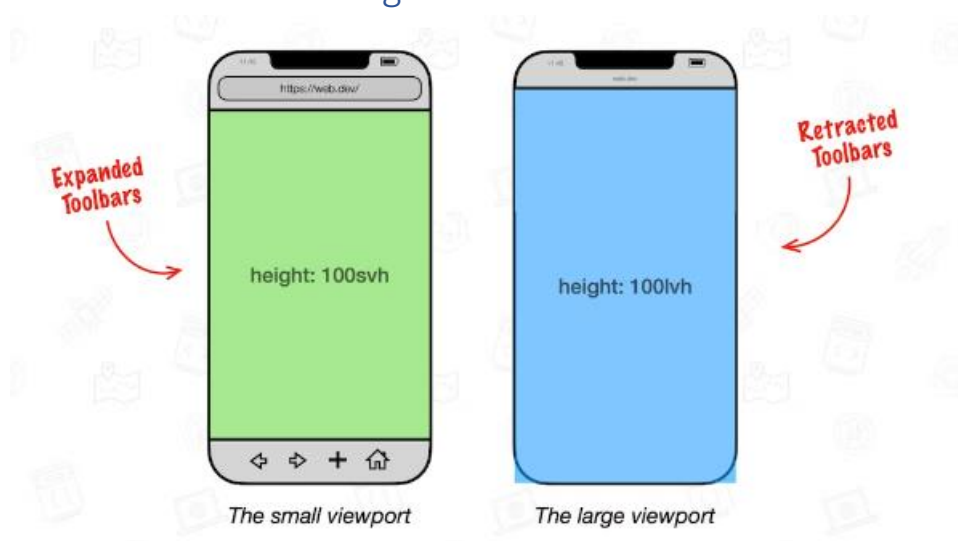
Ahora haremos lo mismo, pero cambiando la proporción de píxeles de 1 a 2. Como podemos observar, ahora se nos mostrará una imagen con nombre diferente, llamada head-2x.jpg.



Y si reducimos las dimensiones del dispositivo cambiará a head-small-2x.jpg.



Unidades de ventana gráfica



Las unidades de ventana gráfica grande, pequeña y dinámica

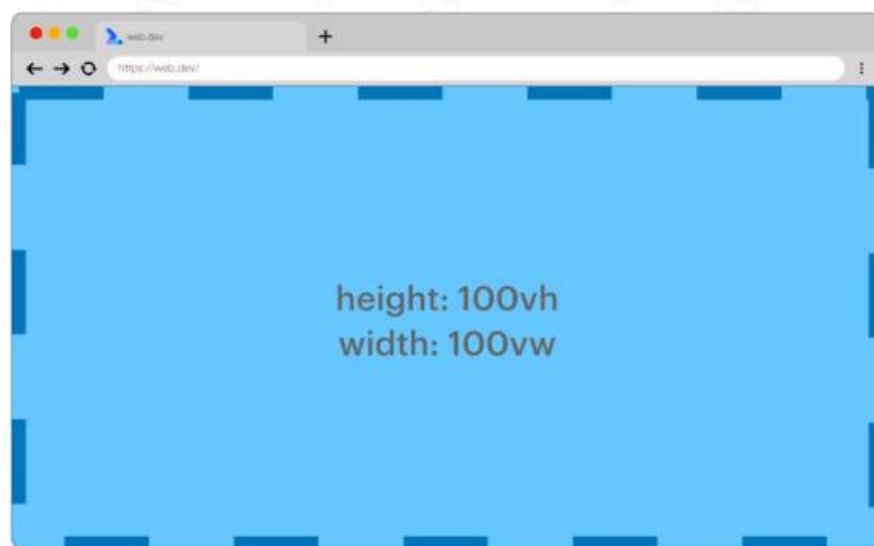
Nuevas unidades de CSS que dan cuenta de ventanas gráficas móviles con barras de herramientas dinámicas.

La ventana gráfica y sus unidades

Para dimensionar algo tan alto como la ventana gráfica, puede usar las unidades vw y .vh

- vw= 1% del ancho del tamaño de la ventana gráfica.
- vh= 1% de la altura del tamaño de la ventana gráfica.

Dale a un elemento un ancho 100vw y una altura de 100vh, y cubrirá la ventana gráfica por completo.



Las unidades vw y vh llegaron a los navegadores con estas unidades adicionales

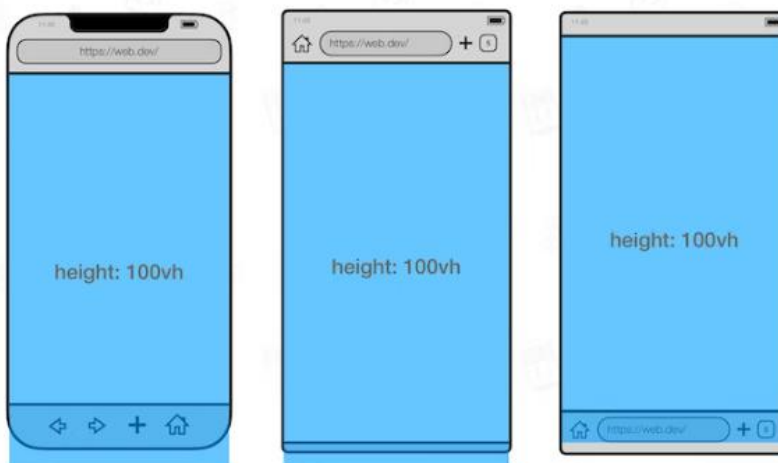
- vi= 1% del tamaño del eje en línea de la ventana gráfica.
- vb= 1% del tamaño del eje del bloque de la ventana gráfica.
- vmin= el menor de vw o vh.
- vmax= el mayor de vw o vh.

Estas unidades tienen un buen soporte de navegador.

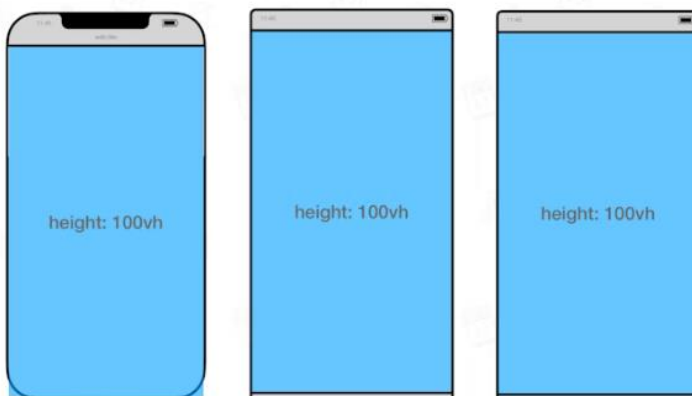
La necesidad de nuevas unidades de ventana gráfica

Si bien las unidades existentes funcionan bien en el escritorio, la historia es diferente en los dispositivos móviles. Allí, el tamaño de la ventana gráfica está influenciado por la presencia o ausencia de barras de herramientas dinámicas. Estas son interfaces de usuario como barras de direcciones y barras de pestañas.

Aunque el tamaño de la ventana gráfica puede cambiar, los tamaños vw y vh no. Como resultado, los elementos cuyo tamaño sea 100vh alto se desangrarán fuera de la ventana gráfica.

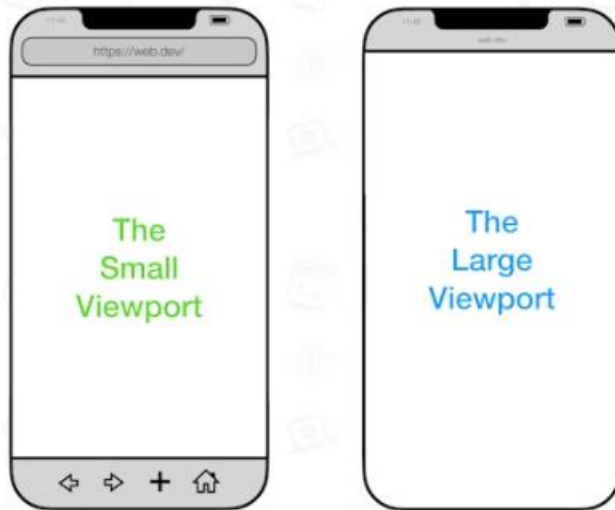


Al desplazarse hacia abajo, estas barras de herramientas dinámicas se retraerán. En este estado, los elementos dimensionados para ser 100vh altos cubrirán toda la ventana gráfica.



Para resolver este problema, los diversos estados de la ventana gráfica se han especificado en el Grupo de trabajo de CSS.

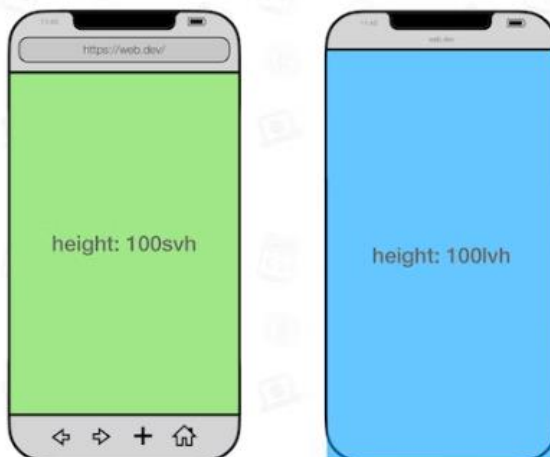
- Gran ventana gráfica: el tamaño de la ventana gráfica suponiendo que las interfaces UA que se expanden y retraen dinámicamente se retraen.
- Ventana gráfica pequeña: el tamaño de la ventana gráfica suponiendo que se expanda cualquier interfaz UA que se expanda y retraiga dinámicamente.



Las nuevas ventanas gráficas también tienen unidades asignadas:

- Las unidades que representan la ventana grande tienen el lv prefijo. Las unidades son lvw, lvh, lvi, lvb, lvmin y lvmax.
- Las unidades que representan la pequeña ventana gráfica tienen el sv prefijo. Las unidades son svw, svh, svi, svb, svmin y svmax.

Los tamaños de estas unidades de porcentaje de ventana gráfica son fijos (y, por lo tanto, estables) a menos que se cambie el tamaño de la ventana gráfica.



Además de las ventanas grandes y pequeñas, también hay una ventana dinámica que tiene una consideración dinámica de la interfaz de usuario de UA:

- Cuando se expanden las barras de herramientas dinámicas, la ventana gráfica dinámica es igual al tamaño de la ventana gráfica pequeña.
- Cuando las barras de herramientas dinámicas se retraen, la ventana gráfica dinámica tiene el mismo tamaño que la ventana grande.

Sus unidades acompañadas tienen el prefijo dv: dvw, dvh, dvi, dvb, dvmin, y dvmax. Sus tamaños están sujetos entre ellos lv* y sus sv*contrapartes.



100dvh se adapta al tamaño de ventana grande o pequeña.

Bibliografía

<https://web.dev/viewport-units/>

<https://web.dev/responsive-images/>