

Diseño receptivo

- [Anterior \(en-US\)](#).
- [Overview: CSS layout](#)
- [Siguiente \(en-US\)](#).

En los primeros días del diseño web, las páginas se diseñaban para llenar un tamaño de pantalla en particular. Si el usuario tenía una pantalla más grande o más pequeña que la del diseñador, los resultados esperados iban desde barras de desplazamiento no deseadas hasta longitudes de línea excesivamente largas y un mal uso del espacio. A medida que estuvieron disponibles tamaños de pantalla más diversos, apareció el concepto de *diseño web responsivo* (RWD, *responsive web design*), un conjunto de prácticas que permite a las páginas web alterar su diseño y apariencia para adaptarse a diferentes anchos de pantalla, resoluciones, etc. Es una idea que cambió la forma en que diseñamos para una web multidispositivo, y en este artículo te ayudaremos a comprender las principales técnicas que necesitas saber para dominarlo.

Prerrequisitos:	Conceptos básicos de HTML (véase Introducción al HTML) y nociones de cómo funciona el CSS (véase Primeros pasos en CSS y Los elementos básicos del CSS).
Objetivo:	Comprender los conceptos fundamentales y la historia del diseño responsivo.

Diseños de sitios web históricos

En un momento de la historia, solo tenías dos opciones al diseñar un sitio web:

- Podías crear un sitio *líquido*, que se expandiría para llenar toda la ventana del navegador
- o un sitio de *ancho fijo*, que sería un tamaño fijo en píxeles.

Estos dos enfoques tendían a dar como resultado un sitio web que se veía mejor ;en la pantalla de la persona que diseñaba el sitio! El sitio líquido dio como resultado un diseño encogido en las pantallas que eran más pequeñas (como se ve a continuación) o longitudes de línea interminables en las pantallas que eran más grandes.

This layout is liquid. See what happens if you make the browser window wider or narrower.

One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney. Their names were Stephen and Joseph Montgolfier, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery.

Before that night...

Nota: Observa este [ejemplo](#) y su [código fuente](#) de un diseño líquido sencillo. Amplía o reduce la ventana del navegador y observa cómo cambia su aspecto en diferentes tamaños.

El sitio de ancho fijo se arriesgaba a una barra de desplazamiento horizontal en pantallas que eran más pequeñas que el ancho del sitio (como se ve a continuación), y a un gran espacio en blanco en los bordes del diseño en las pantallas que eran más grandes.

This layout is fixed width. See what happens if you make the browser window narrow.

Nota: Observa este [ejemplo](#) y su [código fuente](#) de un diseño sencillo con un ancho fijo.

Nuevamente, cambia el tamaño de la ventana del navegador y observa el resultado.

Nota: Las capturas de pantalla anteriores se han tomado usando el [modo de diseño responsivo](#) de las herramientas DevTools de Firefox.

A medida que la web móvil comenzó a hacerse realidad con los primeros teléfonos con funciones, las empresas que deseaban adoptar los dispositivos móviles solían crear una versión especial de su sitio web para dispositivo móvil, con una URL diferente (a menudo algo así como *m.example.com* o *example.mobi*). Esto significaba que había que desarrollar y actualizar dos versiones independientes del sitio web.

Además, estos sitios web para dispositivos móviles a menudo ofrecían una experiencia muy reducida. A medida que los dispositivos móviles se volvían más potentes y capaces de mostrar sitios web completos, esto resultaba frustrante para los usuarios de dispositivos móviles, que se veían atrapados en la versión móvil del sitio web y no podían acceder a la información que sabían que había en la versión de escritorio, que incluía todas las funciones del sitio web.

Diseño flexible antes del diseño responsivo

Se desarrollaron varios enfoques para tratar de resolver los inconvenientes de los métodos líquidos o de ancho fijo para crear sitios web. En 2004, Cameron Adams escribió una publicación titulada [Resolution dependent layout](#), que describe un método para crear un diseño que podría adaptarse a diferentes resoluciones de pantalla. Este enfoque requería JavaScript para detectar la resolución de la pantalla y cargar el CSS correcto.

Zoe Mickley Gillenwater fue determinante en [su trabajo](#) de descripción y formalización de los diversos modos en que se podían crear sitios web flexibles para intentar encontrar una situación intermedia entre llenar toda la pantalla o tener un tamaño completamente fijo.

Diseño responsivo

El término *diseño responsivo* fue acuñado por [Ethan Marcotte](#) en 2010, y describía el uso combinado de tres técnicas.

1. La primera era la idea de las redes fluidas, algo que ya exploraba Gillenwater, y que puede leerse en el artículo de Marcotte, [Fluid Grids](#) (publicado en 2009 en *A list apart*).
2. La segunda técnica era la idea de las [imágenes fluidas](#). Usando una técnica muy simple de establecer la propiedad de `max-width` al 100%, las imágenes se reducían si su columna de contención se volvía más estrecha que el tamaño intrínseco de la imagen, pero nunca se expandía. Esto permite reducir una imagen para que se ajuste a una columna de tamaño flexible, en lugar de que desborde, pero no se expande ni se pixela si la columna se ensancha más que la imagen.

3. El tercer componente clave era la consulta a los media. Las consultas a los media habilitan el tipo de cambio de diseño que Cameron Adams había explorado previamente usando JavaScript, pero usando solo CSS. En lugar de tener un único diseño para todos los tamaños de pantalla, el diseño podría cambiarse. Las barras laterales pueden reposicionarse en una pantalla más pequeña, o puede mostrarse una navegación alternativa.

Es importante comprender que el diseño web responsivo no es una tecnología independiente: es un término utilizado para describir un enfoque para el diseño web, o un conjunto de buenas prácticas utilizado para crear un diseño que puede *responder* según el dispositivo que se utiliza para ver un contenido. En la exploración original de Marcotte, esto significaba cuadrículas flexibles (mediante elementos flotantes) y consultas de media; sin embargo, en los casi 10 años desde que se escribió ese artículo, trabajar de manera responsiva se ha convertido en la norma. Los métodos de diseño CSS modernos son inherentemente responsivos, y la plataforma web dispone de herramientas integradas nuevas que facilitan el diseño de sitios web responsivos.

El resto de este artículo te indicará las diversas características de la plataforma web que puedas querer utilizar para crear un sitio responsivo.

La consulta a los media

El diseño responsivo solo pudo surgir gracias a la consulta a los media. La especificación de nivel 3 de consulta a los media se convirtió en una candidata a Recomendación en 2009, lo que significa que se consideró lista para su implementación en los navegadores. Las consultas a los media nos permiten ejecutar una serie de pruebas (por ejemplo, si la pantalla del usuario es mayor que un ancho o una resolución determinados) y aplicar CSS selectivamente para diseñar la página de manera que resulte adecuada a las necesidades del usuario.

Por ejemplo, la consulta a los media siguiente explora si la página web que se muestra lo hace como un medio de pantalla (por lo tanto, no es un documento impreso) y si la ventana tiene al menos 800 píxeles de ancho. El CSS para el selector `.container` solo se aplicará si ambas condiciones son ciertas.

```
@media screen and (min-width: 800px) {  
  .container {  
    margin: 1em 2em;  
  }  
}
```

Puedes añadir múltiples consultas a los media dentro de una hoja de estilo, y ajustar todo tu diseño o solo partes de él para que se adapte mejor a los diferentes tamaños de pantalla. Los puntos en los que se introduce una consulta a los media y se cambia el diseño se conocen como *puntos de interrupción*.

Un enfoque común cuando se usan las consultas a los media es crear un diseño sencillo de una sola columna para dispositivos de pantalla estrecha (por ejemplo, teléfonos móviles), luego implementar un diseño en columnas para pantallas más grandes cuando se sabe que hay suficiente ancho de pantalla para manejarlo. Esto se describe a menudo como diseño primero móvil.

Obtén más información sobre las consultas a los media en la documentación de MDN.

Cuadrículas flexibles

Los sitios responsivos no solo cambian su diseño entre puntos de interrupción, sino que se construyen sobre cuadrículas flexibles. Una cuadrícula flexible significa que no tienes que centrarte en todos los tamaños de dispositivo posibles y construir para ellos un diseño en píxeles perfecto. Ese enfoque sería imposible dada la gran cantidad de dispositivos de tamaños diferentes que hay, y el hecho de que, al menos en la versión de escritorio, las personas no siempre tienen la ventana de su navegador maximizada.

Al usar una cuadrícula flexible, solo necesitas añadir un punto de interrupción y cambiar el diseño en el punto en que el contenido comienza a verse mal. Por ejemplo, si las longitudes de las líneas se vuelven interminablemente largas a medida que el tamaño de la pantalla aumenta, o una caja se encoje hasta un ancho de dos palabras en cada línea a medida que el tamaño de la pantalla se reduce.

En los primeros días del diseño responsivo, nuestra única opción para el diseño de páginas web era usar elementos flotantes. Los diseños de pantalla con elementos flotantes flexibles se lograban dando a cada elemento un ancho porcentual asegurándose de que para toda la página no alcanzara más del 100%. En su trabajo original sobre cuadrículas fluidas, Marcotte detalló una fórmula para tomar un diseño de página web diseñado usando píxeles y convertirlo en porcentajes.

$$\text{target} / \text{context} = \text{result}$$

Por ejemplo, si el tamaño de nuestra columna de destino es de 60 píxeles y el contexto (o contenedor) en el que se encuentra es de 960 píxeles, dividimos 60 por 960 para obtener un valor que podemos usar en nuestro CSS, después de mover el separador de cifras decimales dos posiciones a la derecha.

```
.col {  
  width: 6.25%; /* 60 / 960 = 0.0625 */  
}
```

Este enfoque se encuentra hoy en muchos lugares de la web, y aquí está documentado en la sección de compaginación de nuestro artículo sobre métodos de compaginación heredados. Es probable que encuentres sitios web que utilizan este enfoque en su trabajo, por lo que vale la pena entenderlo, aunque no vas a construir un sitio web moderno utilizando una cuadrícula flexible basada en elementos flotantes.

El ejemplo siguiente muestra un diseño responsivo sencillo que utiliza consultas a los medios y una cuadrícula flexible. En pantallas estrechas, el diseño de página muestra las cajas en columna una encima de la otra:

This layout is responsive. See what happens if you make the browser window wider or narrow.

One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney. Their names were Stephen and Joseph Montgolfier, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery.

Before that night—a memorable night, as it was to prove—hundreds of millions of people had watched the rising smoke wreaths of their fires

En pantallas más anchas se pasa a dos columnas:

This layout is responsive. See what happens if you make the browser window wider or narrow.

One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney. Their names were Stephen and Joseph Montgolfier, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery.

Before that night—a memorable night, as it was to prove—hundreds of millions of people had watched the rising smoke-wreaths of their fires without drawing any special inspiration from the fact.”

Nota: Puedes encontrar el [ejemplo en vivo](#) y el [código fuente](#) de este ejemplo en GitHub.

Tecnologías modernas de diseño de páginas web

Los métodos modernos de diseño de páginas web, como el [diseño en columnas](#), [Flexbox](#) y [Grid](#) son responsivos por defecto. Todos estos métodos asumen que tratas de crear una cuadrícula flexible y te proporcionan los modos más fáciles de hacerlo.

Multicol

El más antiguo de estos métodos de diseño de páginas web es multicol. Cuando especificas un atributo `column-count`, esto indica en cuántas columnas deseas dividir tu contenido. El navegador entonces calcula el tamaño de estas columnas, que cambiará de acuerdo con el tamaño de la pantalla.

```
.container {  
  column-count: 3;  
}
```

Si en lugar de ello estableces el atributo `column-width`, especificas un ancho *mínimo*. El navegador crea tantas columnas de ese ancho como quepan cómodamente en el contenedor, y reparte el espacio entre todas las columnas. Por lo tanto, el número de columnas cambia según la cantidad de espacio que hay.

```
.container {  
  column-width: 10em;  
}
```


Flexbox

En el método Flexbox, los elementos flexibles se encogen y distribuyen el espacio entre los elementos según el espacio que hay en su contenedor, según su comportamiento inicial. Al cambiar los valores de `flex-grow` y `flex-shrink`, puedes indicar cómo deseas que se comporten los elementos cuando a su alrededor hay más o menos espacio.

En el ejemplo siguiente, los elementos flexibles ocupan cada uno la misma cantidad de espacio en el contenedor flexible, al utilizar la abreviatura `flex: 1` como se describe en el artículo [Flexbox: Dimensionamiento flexible de los elementos flex](#).

```
.container {  
  display: flex;  
}  
  
.item {  
  flex: 1;  
}
```

Nota: Como ejemplo, hemos reconstruido el anterior diseño de página responsivo sencillo, esta vez usando Flexbox. Puedes ver que ya no necesitamos usar valores de porcentaje extraños para calcular el tamaño de las columnas: [ejemplo](#), [código fuente](#).

Cuadrículas CSS

En el diseño de cuadrículas con CSS, la unidad `fr` permite la distribución del espacio disponible en las trazas de la cuadrícula. El ejemplo siguiente crea un contenedor de cuadrícula con tres trazas dimensionadas a `1fr`. Esto crea tres columnas, cada una de las cuales ocupa una parte del espacio que hay disponible en el contenedor. Puedes obtener más información sobre este enfoque para crear una cuadrícula en el módulo Aprender a diseñar cuadrículas en [Cuadrículas flexibles con la unidad fr](#).

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 1fr 1fr;  
}
```

Nota: La versión del diseño de página en cuadrícula es aún más simple, ya que podemos definir las columnas en `.wrapper`: [ejemplo](#), [código fuente](#).

Imágenes responsivas

El enfoque más simple para las imágenes responsivas es el que se describe en los primeros artículos de Marcotte sobre diseño responsivo. Básicamente, tomar una imagen que tenga el tamaño más grande que puedas necesitar, y reducirla. Este continúa siendo un enfoque utilizado hoy en día, y en la mayoría de las hojas de estilo encontrarás en alguna parte el CSS siguiente:

```
img {  
  max-width: 100%;  
}
```

Hay inconvenientes obvios en este enfoque. La imagen puede mostrarse mucho más pequeña que su tamaño intrínseco, lo que representa una pérdida de ancho de banda: un usuario de dispositivo móvil puede descargar una imagen que sea varias veces el tamaño de lo que ve en realidad en la ventana del navegador. Además, es posible que no desees la misma relación de aspecto de la imagen en dispositivos móviles y en ordenadores de escritorio. Por ejemplo, podría ser bueno tener una imagen cuadrada para dispositivos móviles, pero mostrar la misma escena que una imagen horizontal en el escritorio. O bien es posible que, reconociendo el tamaño más pequeño de una imagen en dispositivos móviles, desees mostrar una imagen diferente, que se entienda mejor en un tamaño de pantalla pequeño. Estas cosas no se pueden lograr simplemente reduciendo una imagen.

Las imágenes responsivas, que utilizan el elemento `<picture>` y los atributos `` `srcset` y `sizes` resuelven ambos problemas. Puedes proporcionar varios tamaños junto con «sugerencias» (metadatos que describen el tamaño de pantalla y la resolución para que la imagen sea la más adecuada), y el navegador elige la imagen que resulta más adecuada para cada dispositivo, y se asegura de que el usuario descarga un tamaño de imagen apropiado para el dispositivo que utiliza.

También puedes usar imágenes *de director artístico*, que proporcionan un recorte o una imagen completamente diferente para diferentes tamaños de pantalla.

Puedes encontrar una [guía detallada de imágenes responsivas en el artículo sobre Aprender HTML](#) en MDN.

Tipografía responsiva

Un elemento de diseño responsivo que todavía no hemos tratado en trabajos anteriores es la idea de la tipografía responsiva. Este concepto describe esencialmente el hecho de cambiar el tamaño de letra según el espacio de pantalla que reflejan las consultas a media.

En este ejemplo, queremos establecer que nuestro encabezado de nivel 1 sea 4rem, lo que significa que será cuatro veces nuestro tamaño de letra base. ¡Es un título muy grande! Solo queremos este título de encabezado gigante en los tamaños de pantalla más grandes, por lo tanto, primero creamos un título de encabezado más pequeño y luego usamos las consultas a los media para sobrescribirlo con el tamaño más grande si sabemos que el usuario tiene un tamaño de pantalla de al menos 1200px.

```
html {  
  font-size: 1em;  
}  
  
h1 {  
  font-size: 2rem;
```

```
}  
  
@media (min-width: 1200px) {  
  h1 {  
    font-size: 4rem;  
  }  
}
```

Hemos editado nuestro ejemplo anterior de cuadrícula responsiva para incluir también el tipo de respuesta utilizando el método descrito. Puedes ver cómo el título de encabezado cambia de tamaño cuando el diseño pasa a la versión de dos columnas.

En la versión para dispositivo móvil, el encabezado es más pequeño:

Watch my size!

This layout is responsive. See what happens if you make the browser window wider or narrow.

One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney. Their names were Stephen and Joseph Montgolfier, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery.

Before that night—a memorable night,

Sin embargo, en las versiones de escritorio vemos un tamaño de título de encabezado más grande:

Watch my size!

This layout is responsive. See what happens if you make the browser window wider or narrower.

One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney. Their names were Stephen and Joseph Montgolfier, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery.

Before that night—a memorable night, as it was to prove—hundreds of millions of people had watched the rising smoke-wreaths of their fires without drawing any special inspiration from the fact.”

Nota: Observa este ejemplo en: [ejemplo](#), [código fuente](#).

Como muestra este enfoque sobre la tipografía, no es necesario restringir las consultas a medios a cambiar solo el diseño de página. Se pueden usar para ajustar cualquier elemento y hacerlo más útil o atractivo según los diversos tamaños de pantalla.

El uso de unidades de ventana gráfica para tipografía responsiva

Un enfoque interesante es utilizar las unidades de ventana gráfica `vw` para habilitar la tipografía responsiva. `1vw` es igual al uno por ciento del ancho de la ventana gráfica, lo que significa que si configuras el tamaño del tipo de letra con `vw`, siempre estará en relación con el tamaño de la ventana gráfica.

```
h1 {  
  font-size: 6vw;  
}
```

El problema de hacer esto es que el usuario pierde la posibilidad de ampliar cualquier conjunto de texto configurado en unidades `vw`, porque ese texto siempre está en relación con el tamaño de la ventana gráfica. Por lo tanto, nunca hay que establecer texto utilizando solo unidades de ventana.

Hay una solución, que implica el uso de la función `calc()`. Si añades la unidad `vw` a un valor establecido con un tamaño fijo, como `em` o `rem`, el texto continúa siendo ampliable. Esencialmente, la unidad `vw` se añade sobre ese valor ampliado:

```
h1 {  
  font-size: calc(1.5rem + 3vw);  
}
```

Esto significa que necesitamos especificar el tamaño de letra para el título de encabezado una sola vez, en lugar de configurarlo para dispositivos móviles y redefinirlo en las consultas a medios. Luego, el tipo de letra aumenta gradualmente a medida que aumenta el tamaño de la ventana gráfica.

Observa un ejemplo en: [ejemplo](#), [código fuente](#).

La metaetiqueta viewport

Si observas el tipo de letra de una página HTML responsiva, en general vas a encontrar la siguiente etiqueta `<meta>` en la cabecera del documento.

```
<meta name="viewport" content="width=device-width,initial-scale=1">
```

Esta metaetiqueta informa a los navegadores de los dispositivos móviles que deben establecer el ancho de la ventana gráfica al ancho del dispositivo y escalar el documento al 100% de ese tamaño, de modo que el documento se mostrará al tamaño optimizado para esos dispositivos móviles.

¿Por qué esto es necesario? Porque los navegadores de los dispositivos móviles tienden a mentir sobre el ancho de su ventana gráfica.

Esta metaetiqueta existe porque cuando se lanzó el iPhone original y la gente comenzó a ver sitios web en una pequeña pantalla de teléfono móvil, la mayoría de los sitios web no estaban optimizados para dispositivos móviles. Por lo tanto, el navegador móvil establecía el ancho de la ventana gráfica en 960 píxeles, representaba la página con ese ancho y mostraba el resultado como una versión reducida del diseño del escritorio. Otros navegadores de dispositivos móviles (por ejemplo, en Google Android) hicieron lo mismo. Los usuarios podían acercarse y desplazarse por el sitio web para ver las partes que les interesaban, pero se veía mal. Todavía verás esto hoy en día si tienes la desgracia de encontrarte con un sitio web que no tiene un diseño de página responsivo.

El problema es que tu diseño responsivo con puntos de interrupción y consultas a media no va a funcionar según lo previsto en los navegadores de dispositivos móviles, si tienes un diseño de pantalla estrecho que se inicia con un ancho de ventana de 480px o menos, pero la ventana gráfica está configurada en 960px. E cambio, al configurar `width=device-width` anulas el ancho predeterminado `width=960px` de Apple con el ancho real del dispositivo, y tus consultas a media funcionarán según lo previsto.

Por lo tanto, *siempre* debes incluir la línea de HTML anterior en la cabecera de tus documentos.

Con la metaetiqueta `viewport` puedes usar otras configuraciones, aunque, en general vas a querer usar la línea anterior.

- `initial-scale` : Establece el zoom inicial de la página, que establecemos en 1.
- `height` : Establece una altura específica para la ventana gráfica.
- `minimum-scale` : Establece el nivel mínimo de zoom.
- `maximum-scale` : Establece el nivel máximo de zoom.
- `user-scalable` : Impide el zoom si se establece en `no` .

Deberías evitar el uso de `minimum-scale` y `maximum-scale` , y en particular establecer `user-scalable` en `no` . Hay que permitir a los usuarios hacer zoom tanto o tan poco como lo necesiten; evitarlo provoca problemas de accesibilidad.

Nota: Hay una @regla CSS establecida para reemplazar la metaetiqueta viewport : @viewport. Sin embargo, tiene poca compatibilidad con los navegadores. Se implementó en Internet Explorer y Edge, pero una vez que se lance el navegador Edge basado en Chromium, dejará de formar parte del navegador Edge.

Resumen

El diseño responsivo se refiere a un diseño página de un sitio web o una aplicación que responde al entorno en el que se visualiza. Abarca una serie de características y técnicas de CSS y HTML, y ahora es esencialmente el modo como construimos los sitios web de forma predeterminada. Piensa en los sitios web que visitas con tu dispositivo móvil; probablemente sea inusual encontrar un sitio web que tenga la versión de escritorio reducida o en que necesites desplazarse hacia los lados para encontrar las cosas. Esto se debe a que la web se ha movido a este enfoque de diseño responsivo.

Además, lograr diseños responsivos se ha vuelto mucho más fácil con la ayuda de los métodos de diseño que has aprendido en estos artículos. Si eres nuevo en el desarrollo web, hoy tienes muchas más herramientas a tu disposición que en los primeros días del diseño de página responsivo. Por lo tanto, vale la pena verificar la antigüedad de los materiales que consultas. Si bien los artículos históricos continúan siendo útiles, el uso moderno de CSS y HTML facilita mucho la creación de diseños elegantes y útiles, sin importar con qué dispositivo el visitante visita el sitio.