

Diseño Responsivo

Media Queries y Mobile First



2026-01



25 min.

¿Qué es Diseño Responsivo?

Diseño Responsivo

El diseño responsivo permite que un sitio web se adapte a diferentes tamaños de pantalla y dispositivos.

Beneficios:

- Mejor experiencia de usuario
- Un solo código para todos los dispositivos
- Mejor SEO
- Mantenimiento más fácil

Elementos clave:

- Media Queries
- Unidades flexibles
- Layouts fluidos
- Imágenes responsivas

Media Queries

Media Queries - Introducción

Permiten aplicar estilos basados en características del dispositivo:

```
/* Para pantallas de 768px o menos */
@media (max-width: 768px) {
    .container {
        flex-direction: column;
    }
}

/* Para pantallas de 769px o más */
@media (min-width: 769px) {
    .container {
        flex-direction: row;
    }
}
```

Media Queries - Sintaxis

```
/* Sintaxis básica */
@media media-type and (condición) {
    /* Estilos */
}

/* Ejemplos */
@media screen and (min-width: 768px) { }
@media print { }
@media (orientation: landscape) { }

/* Múltiples condiciones */
@media (min-width: 768px) and (max-width: 1024px) { }

/* OR lógico */
@media (max-width: 768px), (orientation: portrait) { }
```

Mobile First

¿Qué es Mobile First?

Mobile First es un enfoque de diseño que comienza con la versión móvil y luego escala hacia arriba.

Ventajas:

- Prioriza el contenido esencial
- Mejor rendimiento en dispositivos móviles
- Progressive enhancement
- Más fácil escalar hacia arriba que hacia abajo

Filosofía: Diseñar para las restricciones del móvil primero, luego agregar complejidad para pantallas más grandes.

Mobile First vs Desktop First

Mobile First (recomendado):

```
/* Base: Mobile */
.container {
  padding: 10px;
  flex-direction: column;
}

/* Tablet y superior */
@media (min-width: 768px) {
  .container {
    padding: 20px;
    flex-direction: row;
  }
}
```

Desktop First:

```
/* Base: Desktop */
.container {
  padding: 30px;
  flex-direction: row;
}
```

Breakpoints

Breakpoints Comunes

```
/* Mobile First approach */

/* Mobile (por defecto - 0px+) */
.container { padding: 10px; }

/* Tablet (768px+) */
@media (min-width: 768px) {
    .container { padding: 20px; }
}

/* Desktop (1024px+) */
@media (min-width: 1024px) {
    .container { padding: 30px; }
}

/* Large Desktop (1440px+) */
@media (min-width: 1440px) {
    .container { padding: 40px; }
}
```

Breakpoints Estándar

Breakpoints comúnmente usados:

- **Mobile:** 0 - 767px
- **Tablet:** 768px - 1023px
- **Desktop:** 1024px - 1439px
- **Large Desktop:** 1440px+

Importante: Los breakpoints deben basarse en tu contenido, no solo en dispositivos específicos.

```
/* Ejemplo basado en contenido */  
@media (min-width: 600px) {  
    /* El contenido se ve mejor en dos columnas */  
    .grid { grid-template-columns: 1fr 1fr; }  
}
```

Unidades Responsivas

Unidades Relativas vs Absolutas

Unidades Absolutas (evitar para diseño responsive):

```
width: 300px; /* Siempre 300px */  
font-size: 16px;
```

Unidades Relativas (preferir):

```
width: 50%; /* Relativo al parent */  
width: 50vw; /* Relativo al viewport */  
font-size: 1.5rem; /* Relativo al root */  
padding: 2em; /* Relativo al font-size del elemento */
```

Unidades Responsivas Detalladas

```
/* Porcentaje - relativo al elemento padre */
width: 50%;

/* Viewport - relativo al tamaño de la pantalla */
width: 50vw; /* 50% del ancho del viewport */
height: 100vh; /* 100% de la altura del viewport */
font-size: 5vw; /* Crece/decrece con el viewport */

/* REM - relativo al font-size del root (html) */
font-size: 1.5rem;
padding: 2rem;

/* EM - relativo al font-size del parent */
padding: 2em;
margin: 1.5em;
```

Funciones CSS Modernas

```
/* MIN - el valor más pequeño */  
width: min(500px, 100%);  
/* Nunca más de 500px, pero puede ser menos */  
  
/* MAX - el valor más grande */  
width: max(300px, 50%);  
/* Al menos 300px, pero puede ser más */  
  
/* CLAMP - valor entre mín y máx */  
width: clamp(300px, 50%, 800px);  
/* Mínimo 300px, ideal 50%, máximo 800px */  
  
font-size: clamp(1rem, 2.5vw, 2rem);  
/* Fuente responsive entre 1rem y 2rem */
```

Imágenes Responsivas

Imágenes Responsivas - CSS

```
/* Básico - imagen nunca sobresale del contenedor */
img {
    max-width: 100%;
    height: auto;
}

/* Control de ajuste con object-fit */
img {
    width: 300px;
    height: 200px;
    object-fit: cover;      /* Recorta para llenar */
    object-fit: contain;   /* Escala para caber */
    object-fit: fill;       /* Estira para llenar */
    object-position: center;
}
```

Imágenes Responsivas - HTML

```
<!-- Picture element para diferentes imágenes -->
<picture>
  <source media="(min-width: 1024px)" srcset="large.jpg">
  <source media="(min-width: 768px)" srcset="medium.jpg">
  
</picture>

<!-- Srcset para diferentes resoluciones -->

```

Contenedores Responsivos

Container con Max-Width

```
/* Contenedor que crece hasta un máximo */  
.container {  
    width: 100%;  
    max-width: 1200px;  
    margin: 0 auto;  
    padding: 0 20px;  
}  
  
/* Con media queries para diferentes tamaños */  
.container {  
    width: 100%;  
    margin: 0 auto;  
    padding: 0 15px;  
}  
  
@media (min-width: 768px) {  
    .container {  
        max-width: 720px;  
        padding: 0 20px;  
    }  
}  
  
@media (min-width: 1024px) {  
    .container {  
        max-width: 960px;
```

Grid y Flexbox Responsivos

```
/* Grid responsivo */
.grid {
  display: grid;
  grid-template-columns: 1fr; /* Mobile: 1 columna */
  gap: 20px;
}

@media (min-width: 768px) {
  .grid {
    grid-template-columns: repeat(2, 1fr); /* Tablet: 2 columnas */
  }
}

@media (min-width: 1024px) {
  .grid {
    grid-template-columns: repeat(3, 1fr); /* Desktop: 3 columnas */
  }
}

/* Auto-responsive con auto-fit */
.grid-auto {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: 20px;
}
```

Viewport Meta Tag

Viewport Meta Tag

Esencial para diseño responsive:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Sin esta etiqueta, los móviles renderizarán la página a ancho de escritorio y la escalarán.

Parámetros:

- `width=device-width` : Ancho igual al del dispositivo
- `initial-scale=1.0` : Escala inicial al 100%
- `maximum-scale=5.0` : Escala máxima permitida
- `user-scalable=yes` : Permitir zoom (recomendado)

Siempre incluir en el `<head>` de tu HTML.

Mejores Prácticas

Mejores Prácticas - Diseño Responsivo

- Mobile First approach** - Comenzar desde móvil
- Usar unidades relativas (rem, em, %, vw/vh)
- Breakpoints basados en **contenido**, no en dispositivos específicos
- Viewport meta tag siempre presente
- Imágenes responsivas con max-width: 100%
- Probar en **dispositivos reales** y herramientas de desarrollo
- Flexbox/Grid para layouts flexibles
- Touch-friendly - botones y enlaces grandes en móvil (mínimo 44x44px)

Herramientas de Desarrollo

Ejemplo Completo - Mobile First

```
/* Base: Mobile (0px+) */  
.header {  
    padding: 15px;  
    font-size: 1rem;  
}  
  
.nav {  
    flex-direction: column;  
}  
  
.grid {  
    grid-template-columns: 1fr;  
    gap: 10px;  
}  
  
/* Tablet (768px+) */  
@media (min-width: 768px) {  
    .header {  
        padding: 20px;  
        font-size: 1.2rem;  
    }  
  
.nav {  
    flex-direction: row;  
    justify-content: space-between;
```

Recursos Adicionales

Documentación:

- MDN Media Queries: https://developer.mozilla.org/es/docs/Web/CSS/Media_Queries
- MDN Responsive Design: https://developer.mozilla.org/es/docs/Learn/CSS/CSS_layout/Responsive_Design

Artículos:

- A List Apart: Responsive Web Design
- Smashing Magazine: Responsive Design

Herramientas:

- Can I Use: Compatibilidad de browsers
- Am I Responsive?: Vista previa en múltiples dispositivos

