

# CSS Avanzado: Diseño Responsivo

---

## Introducción al Diseño Responsivo

El diseño responsivo es una técnica que permite que nuestras páginas web se adapten y se vean bien en cualquier dispositivo, ya sea un teléfono móvil, una tablet o un ordenador de escritorio.

**¿Por qué es importante?** Hoy en día, más del 50% del tráfico web proviene de dispositivos móviles. Si nuestra web no se adapta correctamente a diferentes tamaños de pantalla, podemos perder muchos usuarios.

## 1. Media Queries

Las media queries son la herramienta principal para crear diseños responsivos. Nos permiten aplicar diferentes estilos según las características del dispositivo.

¿Cómo funcionan las media queries?

Una media query se compone de:

- La palabra clave `@media`
- Un tipo de medio (como `screen` para pantallas)
- Condiciones (como el ancho de la pantalla)
- Reglas CSS que se aplicarán cuando se cumplan las condiciones

Sintaxis básica:

```
@media screen and (max-width: 768px) {  
  /* Estilos para pantallas con ancho máximo de 768px */  
  .container {  
    width: 100%;  
  }  
}
```

Propiedades comunes para media queries:

- `width` / `max-width` / `min-width`: Ancho de la pantalla
- `height` / `max-height` / `min-height`: Altura de la pantalla
- `orientation`: Orientación (portrait o landscape)
- `aspect-ratio`: Relación de aspecto

Ejemplo práctico: Menú responsivo

```
/* Estilo para pantallas grandes */  
.menu {  
  display: flex;
```

```

    justify-content: space-around;
}

/* Estilo para pantallas pequeñas */
@media screen and (max-width: 600px) {
    .menu {
        flex-direction: column;
        align-items: center;
    }
}

```

Puntos de ruptura (breakpoints) comunes:

- **Móviles pequeños:** hasta 320px
- **Móviles:** hasta 480px
- **Tablets pequeñas:** hasta 600px
- **Tablets:** hasta 768px
- **Portátiles pequeños:** hasta 992px
- **Portátiles y escritorios:** hasta 1200px
- **Pantallas grandes:** más de 1200px

Ejercicio 1: Crear una tarjeta responsiva

HTML:

```

<div class="card">
  
  <div class="card-content">
    <h2>Título de la tarjeta</h2>
    <p>Descripción de la tarjeta con algún texto interesante.</p>
  </div>
</div>

```

CSS:

```

.card {
    border: 1px solid #ddd;
    border-radius: 8px;
    overflow: hidden;
    max-width: 800px;
    margin: 20px auto;
}

.card img {
    width: 100%;
    height: auto;
}

```

```

.card-content {
  padding: 20px;
}

/* Versión para escritorio: imagen al lado del contenido */
@media screen and (min-width: 600px) {
  .card {
    display: flex;
  }

  .card img {
    width: 40%;
  }
}

```

## 2. Unidades Relativas

Las unidades relativas son esenciales para el diseño responsivo porque se adaptan según el contexto.

Tipos de unidades relativas:

### Relativas al viewport:

- **vw**: Porcentaje del ancho del viewport (1vw = 1% del ancho)
- **vh**: Porcentaje del alto del viewport (1vh = 1% del alto)
- **vmin**: 1% de la dimensión más pequeña (ancho o alto)
- **vmax**: 1% de la dimensión más grande (ancho o alto)

### Relativas al elemento padre:

- **%**: Porcentaje relativo al elemento padre
- **em**: Relativo al tamaño de fuente del elemento padre
- **rem**: Relativo al tamaño de fuente del elemento raíz (html)

### Ejemplo de uso de unidades relativas:

```

html {
  font-size: 16px; /* Base para cálculos rem */
}

body {
  font-size: 1rem; /* 16px */
}

h1 {
  font-size: 2rem; /* 32px */
}

.hero {
  height: 50vh; /* 50% del alto de la ventana */
}

```

```
padding: 2em; /* 2 veces el tamaño de fuente actual */
}

.container {
width: 80%; /* 80% del ancho del elemento padre */
max-width: 1200px; /* Limitamos el tamaño máximo */
margin: 0 auto; /* Centramos el contenedor */
}
```

Cuándo usar cada unidad:

- **rem**: Ideal para tamaños de texto. Permite que los usuarios cambien el tamaño base de la fuente.
- **em**: Útil para espaciados relacionados con el tamaño del texto.
- **%**: Excelente para anchos y diseños relativos al contenedor.
- **vw/vh**: Perfectos para elementos que deben ser proporcionales al tamaño de la ventana.

## Ejercicio 2: Título responsivo con unidades relativas

```
h1 {
font-size: calc(1.5rem + 2vw);
margin-bottom: 0.5em;
}

p {
font-size: 1rem;
line-height: 1.5;
max-width: 70ch; /* Limita el ancho a 70 caracteres para mejor
legibilidad */
}
```

Esta regla crea un título que:

- Tiene un tamaño base de 1.5rem
- Aumenta 2vw (2% del ancho de la ventana) adicionales
- Se adapta automáticamente al tamaño de la pantalla sin necesidad de media queries

## 3. Estrategias de Diseño Adaptativo

### Mobile First vs Desktop First

#### Mobile First:

- Diseñamos primero para móviles y luego adaptamos para pantallas más grandes
- Usamos media queries con **min-width**
- Ventajas: Prioriza el contenido esencial, mejora el rendimiento en dispositivos móviles

```
/* Estilos base para móvil */
.container {
```

```

width: 100%;
padding: 10px;
}

/* Adaptaciones para tablets */
@media screen and (min-width: 768px) {
  .container {
    width: 750px;
    margin: 0 auto;
  }
}

/* Adaptaciones para escritorio */
@media screen and (min-width: 1200px) {
  .container {
    width: 1170px;
  }
}

```

### Desktop First:

- Diseñamos primero para escritorio y luego adaptamos para pantallas más pequeñas
- Usamos media queries con `max-width`
- Ventajas: Más intuitivo si ya tenemos un diseño de escritorio

```

/* Estilos base para escritorio */
.container {
  width: 1170px;
  margin: 0 auto;
}

/* Adaptaciones para tablets */
@media screen and (max-width: 1199px) {
  .container {
    width: 750px;
  }
}

/* Adaptaciones para móvil */
@media screen and (max-width: 767px) {
  .container {
    width: 100%;
    padding: 10px;
  }
}

```

## Patrones de Diseño Responsivo

### 1. Layout Fluido

Usa porcentajes para que los elementos se ajusten automáticamente.

```
.container {  
  width: 90%;  
  max-width: 1200px;  
  margin: 0 auto;  
}  
  
.column {  
  float: left;  
  width: 33.33%;  
  padding: 15px;  
}  
  
@media screen and (max-width: 600px) {  
  .column {  
    width: 100%;  
  }  
}
```

## 2. Mostrar/Ocultar Contenido

```
.menu-desktop {  
  display: block;  
}  
  
.menu-mobile {  
  display: none;  
}  
  
@media screen and (max-width: 768px) {  
  .menu-desktop {  
    display: none;  
  }  
  
  .menu-mobile {  
    display: block;  
  }  
}
```

## 3. Cambio de Disposición (Stack)

```
.features {  
  display: flex;  
}  
  
.feature {
```

```

    flex: 1;
    padding: 20px;
}

@media screen and (max-width: 600px) {
    .features {
        flex-direction: column;
    }
}

```

#### 4. Tamaño de Fuente Adaptativo

```

body {
    font-size: 16px;
}

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

p {
    font-size: calc(1rem + 0.3vw);
}

```

#### Imágenes Responsivas

##### Usando CSS:

```

img {
    max-width: 100%;
    height: auto;
}

```

##### Usando el atributo srcset de HTML:

```



```

#### Ejercicio 3: Sistema de Grid Responsivo Simple

HTML:

```
<div class="grid-container">
  <div class="grid-item">Elemento 1</div>
  <div class="grid-item">Elemento 2</div>
  <div class="grid-item">Elemento 3</div>
  <div class="grid-item">Elemento 4</div>
  <div class="grid-item">Elemento 5</div>
  <div class="grid-item">Elemento 6</div>
</div>
```

CSS:

```
.grid-container {
  display: flex;
  flex-wrap: wrap;
  gap: 20px;
  padding: 20px;
}

.grid-item {
  flex: 1 0 100%;
  background-color: #f2f2f2;
  padding: 30px;
  border-radius: 8px;
  text-align: center;
}

/* Tablets (2 columnas) */
@media screen and (min-width: 600px) {
  .grid-item {
    flex: 1 0 calc(50% - 20px);
  }
}

/* Escritorio (3 columnas) */
@media screen and (min-width: 900px) {
  .grid-item {
    flex: 1 0 calc(33.33% - 20px);
  }
}
```

## Ejemplo Práctico Completo: Página Responsiva

Vamos a construir una página simple pero completamente responsiva:

HTML:



```

<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0"
  />
    <title>Mi Página Responsiva</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <header>
      <div class="container">
        <h1>Mi Sitio Web</h1>
        <nav class="menu-desktop">
          <ul>
            <li><a href="#">Inicio</a></li>
            <li><a href="#">Servicios</a></li>
            <li><a href="#">Acerca de</a></li>
            <li><a href="#">Contacto</a></li>
          </ul>
        </nav>
        <div class="menu-mobile">
          <div class="burger">≡</div>
        </div>
      </div>
    </header>

    <section class="hero">
      <div class="container">
        <h2>Bienvenido a mi sitio</h2>
        <p>Un ejemplo de diseño web responsivo</p>
      </div>
    </section>

    <section class="features">
      <div class="container">
        <div class="feature-grid">
          <div class="feature">
            <h3>Característica 1</h3>
            <p>Descripción de la primera característica.</p>
          </div>
          <div class="feature">
            <h3>Característica 2</h3>
            <p>Descripción de la segunda característica.</p>
          </div>
          <div class="feature">
            <h3>Característica 3</h3>
            <p>Descripción de la tercera característica.</p>
          </div>
        </div>
      </div>
    </section>

```

```
<footer>
  <div class="container">
    <p>&copy; 2025 Mi Sitio Web</p>
  </div>
</footer>
</body>
</html>
```

CSS:

```
/* Estilos base */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

html {
  font-size: 16px;
}

body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  color: #333;
}

.container {
  width: 90%;
  max-width: 1200px;
  margin: 0 auto;
}

/* Encabezado */
header {
  background-color: #333;
  color: white;
  padding: 1rem 0;
}

header .container {
  display: flex;
  justify-content: space-between;
  align-items: center;
}

header h1 {
  font-size: 1.5rem;
}

/* Menú de navegación */
```

```

.menu-desktop ul {
  display: flex;
  list-style: none;
}

.menu-desktop li {
  margin-left: 1.5rem;
}

.menu-desktop a {
  color: white;
  text-decoration: none;
}

.menu-desktop a:hover {
  text-decoration: underline;
}

/* Menú móvil */
.menu-mobile {
  display: none;
}

.burger {
  font-size: 2rem;
  cursor: pointer;
}

/* Sección hero */
.hero {
  background-color: #f2f2f2;
  padding: 3rem 0;
  text-align: center;
}

.hero h2 {
  font-size: calc(1.5rem + 1vw);
  margin-bottom: 1rem;
}

.hero p {
  font-size: calc(1rem + 0.2vw);
}

/* Sección de características */
.features {
  padding: 3rem 0;
}

.feature-grid {
  display: flex;
  gap: 2rem;
}

```

```

.feature {
  flex: 1;
  padding: 1.5rem;
  background-color: #f9f9f9;
  border-radius: 8px;
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
}

.feature h3 {
  margin-bottom: 1rem;
  font-size: 1.25rem;
}

/* Footer */
footer {
  background-color: #333;
  color: white;
  padding: 1.5rem 0;
  text-align: center;
}

/* Media Queries – Enfoque Mobile First */

/* Tablets (768px y superior) */
@media screen and (max-width: 768px) {
  .feature-grid {
    flex-wrap: wrap;
  }

  .feature {
    flex: 1 0 calc(50% - 1rem);
  }
}

/* Móviles (600px y inferior) */
@media screen and (max-width: 600px) {
  .menu-desktop {
    display: none;
  }

  .menu-mobile {
    display: block;
  }

  .feature-grid {
    flex-direction: column;
  }

  .feature {
    flex: 1 0 100%;
  }

  .hero {
    padding: 2rem 0;
  }
}

```

```
}  
}
```

## Consejos para un Buen Diseño Responsivo

### 1. Siempre empieza con la etiqueta viewport:

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

2. **Prueba en múltiples dispositivos** o usa las herramientas de desarrollo del navegador.
3. **Simplifica para móviles:** En pantallas pequeñas, mantén solo lo esencial.
4. **No solo cambies tamaños:** Considera cambiar también la disposición y la organización del contenido.
5. **Usa max-width:** Limita el ancho máximo de los contenedores para mejorar la legibilidad en pantallas grandes.
6. **Cuida la tipografía:** Asegúrate de que los textos sean legibles en cualquier tamaño de pantalla.
7. **Evita elementos con ancho fijo:** Prefiere porcentajes y unidades relativas.
8. **Optimiza las imágenes:** Usa formatos modernos y tamaños adecuados para cada dispositivo.

## Ejercicio Final: Portfolio Responsivo

Ahora que entiendes los conceptos básicos del diseño responsivo, puedes crear un portfolio personal simple con:

- Encabezado con navegación que se convierte en menú hamburguesa en móviles
- Sección de presentación con imagen y texto
- Galería de proyectos que reorganiza su disposición según el tamaño de la pantalla
- Formulario de contacto adaptativo
- Pie de página con enlaces a redes sociales

¡Este proyecto te permitirá aplicar todo lo aprendido y tener una base sólida para desarrollar sitios web completamente responsivos!