

# Introducción al Desarrollo web

## Curso: Desarrollo Frontend con React



# Fundamentos de CSS3: Estilos para la Web

**Unlock the visual potential  
of your web projects**

[LEARN MORE](#)

# ¿Qué aprenderemos hoy?

## Fundamentos

- Rol de CSS en desarrollo web
- Formas de integración con HTML
- Selectores y propiedades básicas

## Conceptos intermedios

- Unidades de medida
- Organización del código
- Metodologías (BEM)

## Aplicación práctica

- Ejemplos de implementación
- Buenas prácticas
- Errores comunes a evitar

# CSS: La capa de presentación de la web



## Separación de responsabilidades

- **HTML:** Estructura y contenido
- **CSS:** Presentación y diseño visual
- **JavaScript:** Comportamiento e interactividad

Esta separación permite mantener un código más limpio, reutilizable y fácil de mantener a largo plazo.

# Evolución de CSS

## CSS1(1996)

Primera especificación con propiedades básicas de fuentes, colores y márgenes

## CSS2.1(2011)

Corrección de errores y estandarización

1

2

3

4

## CSS2 (1998)

Añadió posicionamiento, z-index y tipos de medios

## CSS3 (Modular)

Módulos independientes: flexbox, grid, transiciones, animaciones, media queries...

CSS3 introdujo un enfoque modular que permite actualizaciones independientes de cada especificación, acelerando la evolución del estándar.

# ¿Cómo integrar CSS en HTML?

1

## CSS Interno (Embedded)

```
<head>
<style>
p {
  color: blue;
  font-size: 16px;
}
</style>
</head>
```

Útil para páginas únicas con estilos específicos

2

## CSS Externo (Recomendado)

```
<head>
<link rel="stylesheet"
      href="estilos.css">
</head>
```

Mejor para sitios web completos:  
facilita mantenimiento y  
reutilización

3

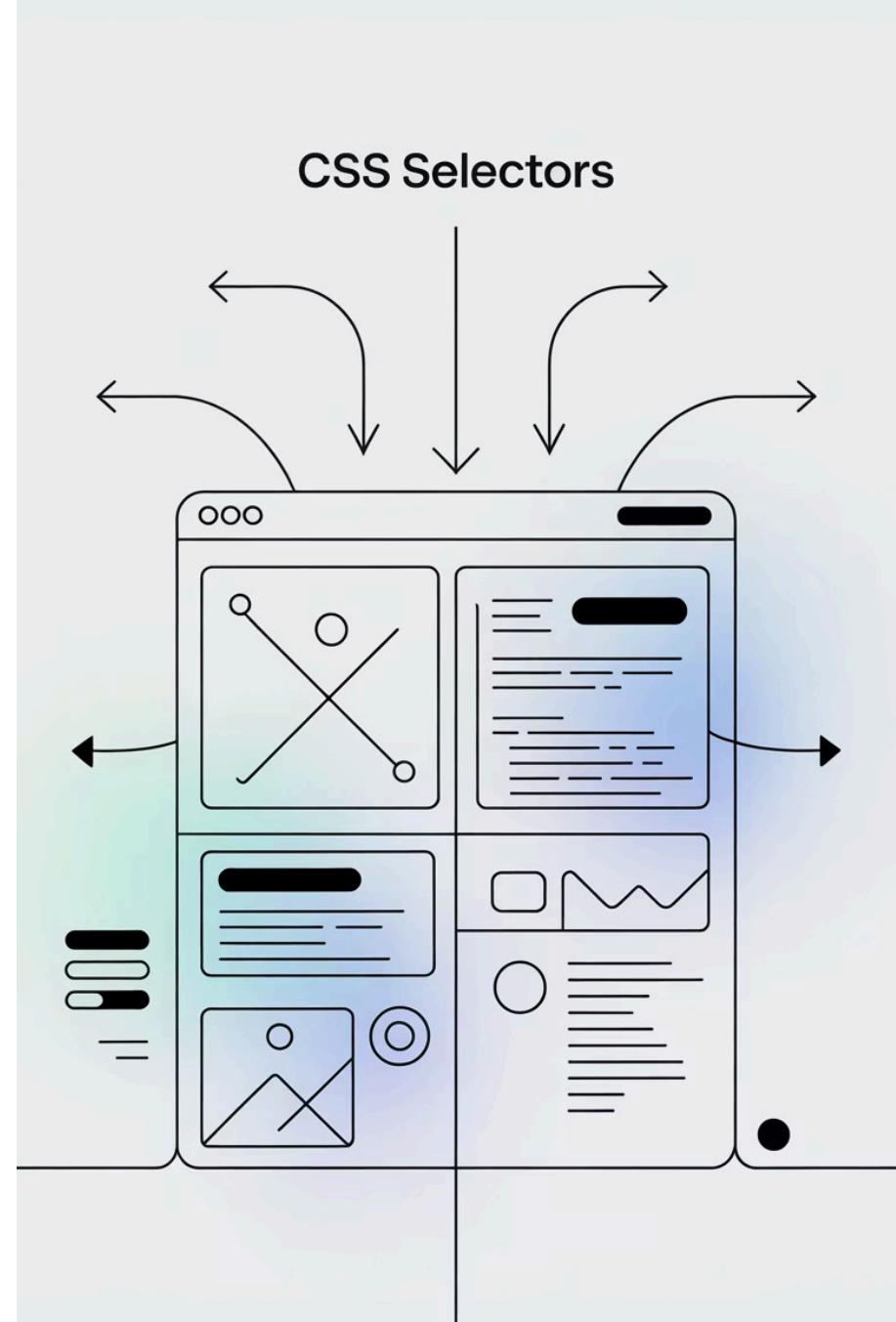
## CSS Inline

```
<p style="color: red;
          font-weight: bold;">
  Texto con estilo
</p>
```

Sólo para excepciones o estilos muy específicos

# Selectores CSS: El poder de la especificidad

Los selectores determinan a **qué elementos HTML** se aplicarán las reglas de estilo que definamos.



# Tipos de selectores básicos

1

## Selector de etiqueta

```
p {  
    color: #2589C9;  
    font-size: 16px;  
}
```

Afecta a **todos** los elementos de ese tipo

2

## Selector de clase

```
.destacado {  
    background-color: #e5f6ff;  
    padding: 10px;  
}
```

Afecta a elementos con **class="destacado"**

3

## Selector de ID

```
#cabecera {  
    height: 80px;  
    position: fixed;  
}
```

Afecta al elemento con **id="cabecera"**

# Selectores compuestos

## Selectores descendientes

```
nav a {  
    text-decoration: none;  
    color: #063E5F;  
}
```

Selecciona todos los enlaces **dentro** de elementos nav

## Selectores de hijo directo

```
ul > li {  
    margin-bottom: 5px;  
}
```

Selecciona elementos li que son **hijos directos** de ul

## Selectores agrupados

```
h1, h2, h3 {  
    font-family: 'Roboto', sans-serif;  
    color: #1E2C3B;  
}
```

Aplica el mismo estilo a **varios selectores**

## Selectores de atributo

```
input[type="submit"] {  
    background-color: #75bae6;  
    border: none;  
}
```

Selecciona elementos según sus atributos

# Cascada y especificidad



## ¿Qué estilo se aplica cuando hay conflicto?

La **especificidad** determina qué regla prevalece:

1. Estilos **!important** (usar con precaución)
2. Estilos **inline** (atributo style)
3. Selectores de **ID** (#header)
4. Selectores de **clase** (.button)
5. Selectores de **etiqueta** (p, div)

A igual especificidad, gana la regla definida más tarde

# Propiedades básicas: Texto



## Color y tipografía

```
p {  
    color: #063E5F;  
    font-family: 'Arial', sans-serif;  
    font-size: 16px;  
    font-weight: bold;  
    line-height: 1.5;  
    text-align: justify;  
}
```



## Resultado

Color y tipografía sin estilos

---

Color y tipografía con estilos

# Propiedades básicas: Fondo

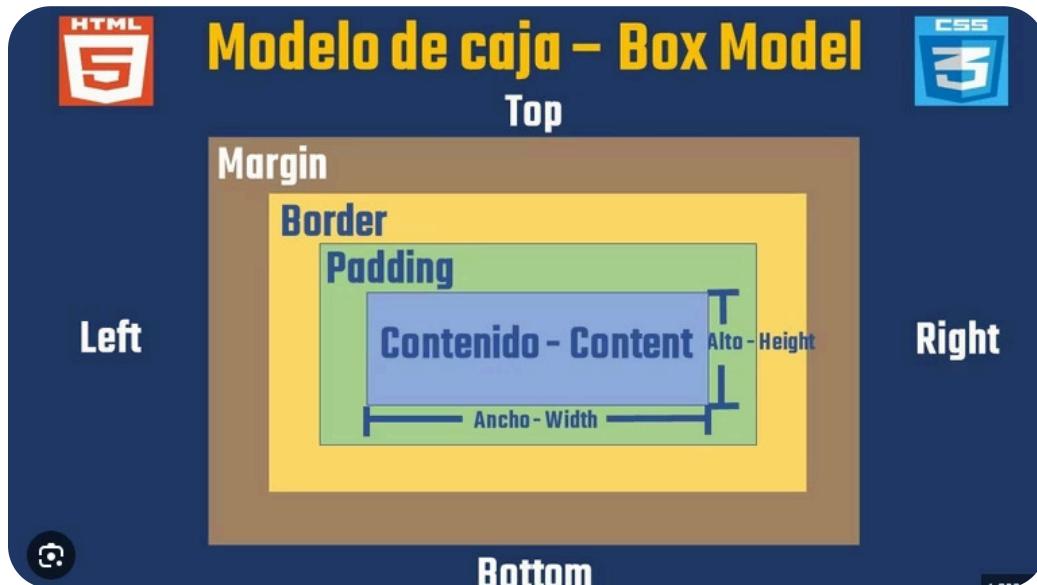
```
.tarjeta {  
background-color: #e5f6ff;  
background-image: url('patron.png');  
background-repeat: no-repeat;  
background-position: center;  
background-size: cover;  
}
```

Versión abreviada:

```
.tarjeta {  
background: #e5f6ff url('patron.png')  
no-repeat center/cover;  
}
```



# Propiedades básicas: Modelo de caja



## Componentes del modelo de caja

- **Content**: El contenido del elemento
- **Padding**: Espacio interno entre contenido y borde
- **Border**: Línea que rodea el padding y contenido
- **Margin**: Espacio externo que separa del resto de elementos

Por defecto: width y height solo incluyen el contenido (box-sizing: content-box)

# Propiedades de margen y relleno

## 1 Sintaxis individual

```
.elemento {  
    margin-top: 10px;  
    margin-right: 15px;  
    margin-bottom: 10px;  
    margin-left: 15px;  
  
    padding-top: 20px;  
    padding-right: 10px;  
    padding-bottom: 20px;  
    padding-left: 10px;  
}
```

## 2 Notación abreviada

```
.elemento {  
    /* arriba, derecha, abajo, izquierda */  
    margin: 10px 15px 10px 15px;  
  
    /* arriba/abajo, izquierda/derecha */  
    padding: 20px 10px;  
  
    /* Mismo valor en los 4 lados */  
    margin: 10px;  
}
```

**Buena práctica:** Usar la notación abreviada cuando sea posible para mantener el código conciso.

# Propiedades de bordes

```
.boton {  
    border-width: 2px;  
    border-style: solid;  
    border-color: #2589C9;  
    border-radius: 5px;  
}  
  
/* Notación abreviada */  
.boton {  
    border: 2px solid #2589C9;  
    border-radius: 5px;  
}
```

[Enviar solicitud](#)

## Estilos de borde

- solid, dashed, dotted
- double, groove, ridge
- inset, outset, none

# Box-sizing: La solución a un problema común

## Problema

```
.caja {  
    width: 200px;  
    padding: 20px;  
    border: 5px solid #75bae6;  
}  
/* Ancho total: 200 + 20*2 + 5*2 = 250px */
```

El tamaño real es **mayor** que el valor de width debido al padding y border

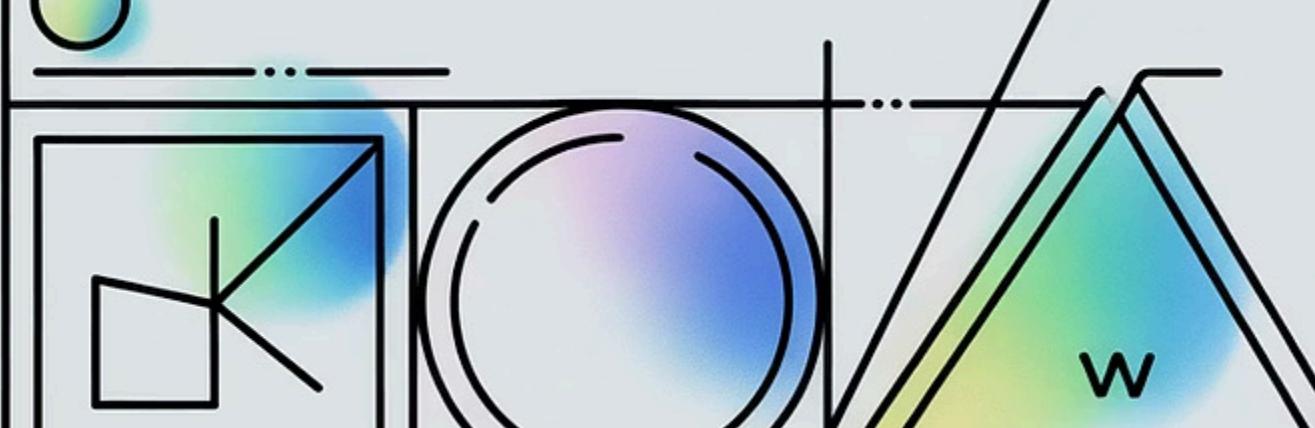
Caja 250px

## Solución

```
* {  
    box-sizing: border-box;  
}  
  
.caja {  
    width: 200px;  
    padding: 20px;  
    border: 5px solid #75bae6;  
}  
/* Ancho total: exactamente 200px */
```

Caja 200px

Con `box-sizing: border-box`, width incluye contenido, padding y border



# Unidades de medida en CSS

La elección de la unidad adecuada es crucial para el diseño responsive y la accesibilidad

# Unidades absolutas vs. relativas

## Unidades absolutas

- **px** - píxeles (la más común)
- **pt** - puntos (1/72 de pulgada)
- **cm, mm, in** - centímetros, milímetros, pulgadas

Uso recomendado: Bordes, sombras, elementos que no deben escalarse

## Unidades relativas

- **%** - porcentaje relativo al elemento padre
- **em** - relativo al tamaño de fuente del elemento
- **rem** - relativo al tamaño de fuente del elemento raíz (html)
- **vw, vh** - 1% del ancho o alto de la ventana
- **vmin, vmax** - 1% del valor mínimo o máximo entre vw y vh

Uso recomendado: Diseño responsive, accesibilidad

# Unidades relativas: Ejemplos prácticos

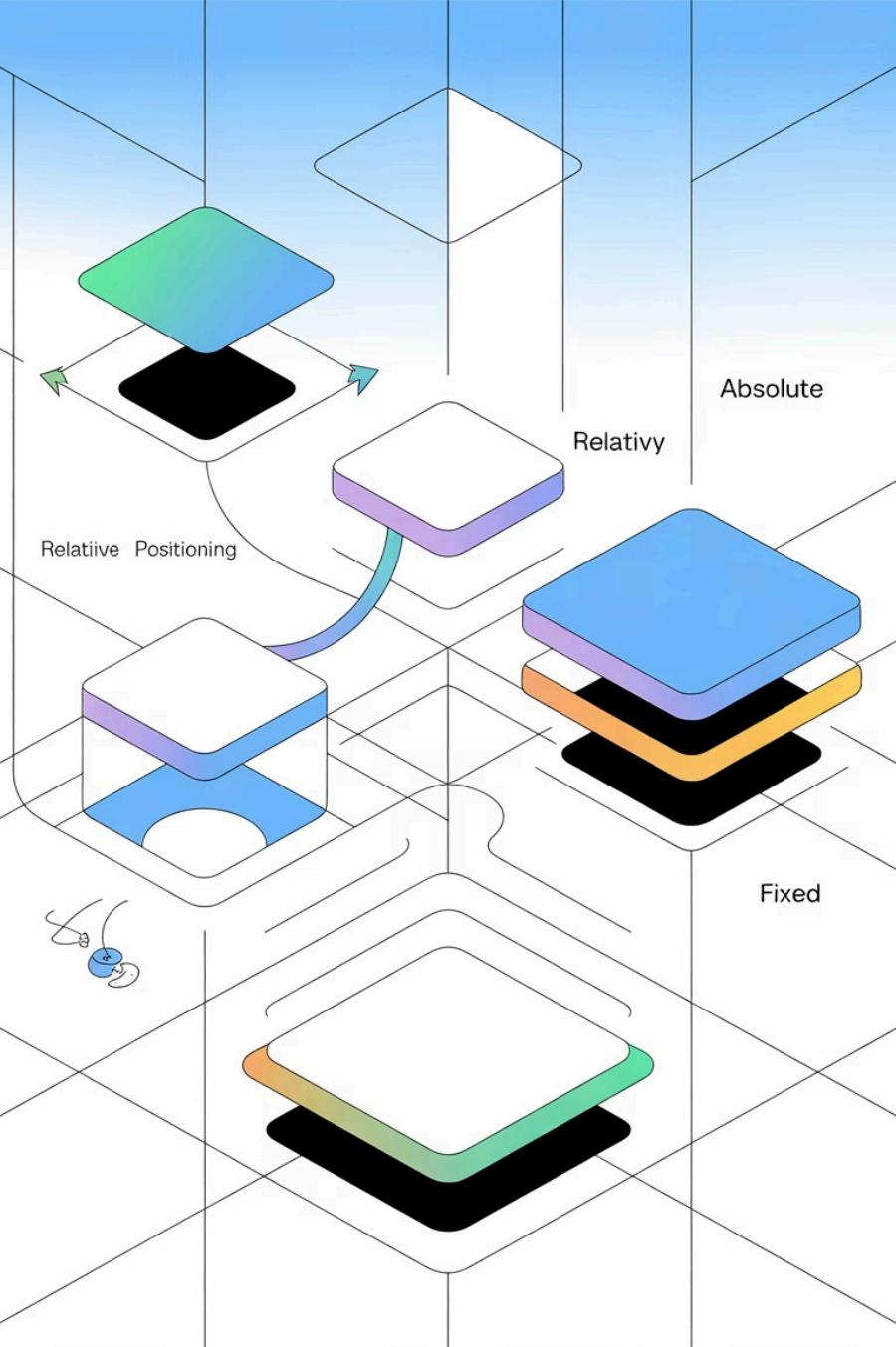
## 1 em vs rem

```
/* em se basa en el padre */  
.padre {  
    font-size: 16px; /* Tamaño base */  
}  
  
.hijo {  
    font-size: 1.5em; /* = 24px */  
    padding: 1em; /* = 24px */  
}  
  
/* rem siempre se basa en el html */  
html {  
    font-size: 16px; /* Tamaño base */  
}  
  
.hijo {  
    font-size: 1.5rem; /* = 24px */  
    padding: 1rem; /* = 16px */  
}
```

## 2 vh, vw y porcentajes

```
/* Altura completa de la ventana */  
.pantalla-completa {  
    height: 100vh;  
    width: 100vw;  
}  
  
/* Banner que ocupa la mitad del alto */  
.banner {  
    height: 50vh;  
}  
  
/* Contenedor flexible */  
.contenedor {  
    width: 80%; /* % del elemento padre */  
    max-width: 1200px;  
    margin: 0 auto;  
}
```

**Tip:** rem es ideal para manejar textos, mientras que % y vh/vw son mejores para layouts.



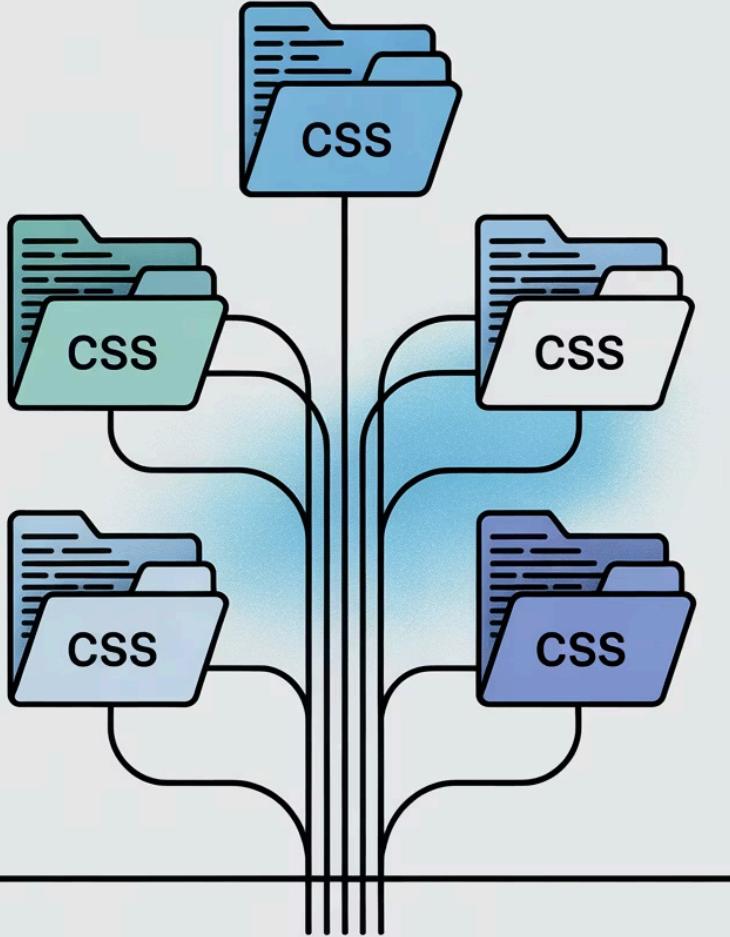
# Visualización y posicionamiento

## Propiedad display

- **block**: Ocupa todo el ancho disponible
- **inline**: Solo ocupa el espacio necesario
- **inline-block**: Híbrido (inline pero acepta dimensiones)
- **flex**: Diseño flexible unidimensional
- **grid**: Sistema de rejilla bidimensional
- **none**: Oculta el elemento (no reserva espacio)

## Propiedad position

- **static**: Posicionamiento normal (por defecto)
- **relative**: Relativo a su posición normal
- **absolute**: Relativo al ancestro posicionado más cercano
- **fixed**: Relativo a la ventana del navegador
- **sticky**: Híbrido entre relative y fixed



# Organización del código CSS

Una buena organización es clave para mantener proyectos escalables a largo plazo

# Comentarios y documentación

```
/* =====  
 SECCIÓN: ENCABEZADO  
 ===== */  
  
/* Navegación principal */  
.nav-main {  
    display: flex;  
    justify-content: space-between;  
}  
  
/* Menú desplegable - Se activa con hover  
   Requiere JavaScript para versión móvil */  
.dropdown {  
    position: relative;  
}  
  
/* TODO: Revisar comportamiento en Safari */  
.dropdown-menu {  
    display: none;  
}
```

Los comentarios claros hacen que el código sea más comprensible para ti y para otros desarrolladores que trabajen en el proyecto.

# Estructura jerárquica del CSS

1

## Reset/Normalización

```
/* Eliminar estilos por defecto */  
* {  
    margin: 0;  
    padding: 0;  
    box-sizing: border-box;  
}
```

2

## Estilos base

```
/* Tipografía y colores base */  
body {  
    font-family: 'Roboto', sans-serif;  
    color: #1E2C3B;  
    line-height: 1.5;  
}  
  
a {  
    color: #2589C9;  
    text-decoration: none;  
}
```

3

## Componentes y módulos

```
/* Botones */  
.btn {  
    padding: 8px 16px;  
    border-radius: 4px;  
}  
  
.btn-primary {  
    background-color: #75bae6;  
    color: white;  
}
```

# Otros métodos de organización



## ATOMIC CSS

Clases pequeñas y de propósito único que hacen una sola cosa.

```
.mt-10 { margin-top: 10px; }  
.text-blue { color: blue; }  
.bg-white { background: white; }
```



## SMACSS

Arquitectura escalable y modular que divide estilos en 5 categorías: Base, Layout, Módulo, Estado y Tema.



## OOCSS

CSS Orientado a Objetos que separa estructura y apariencia para maximizar la reutilización.

Independientemente de la metodología, lo importante es ser **consistente** en todo el proyecto.

# Otros métodos: Metodología BEM

## Block Element Modifier

- **Bloque:** Componente independiente (.card)
- **Elemento:** Parte de un bloque (.card\_\_title)
- **Modificador:** Variante de un bloque o elemento (.card--featured)

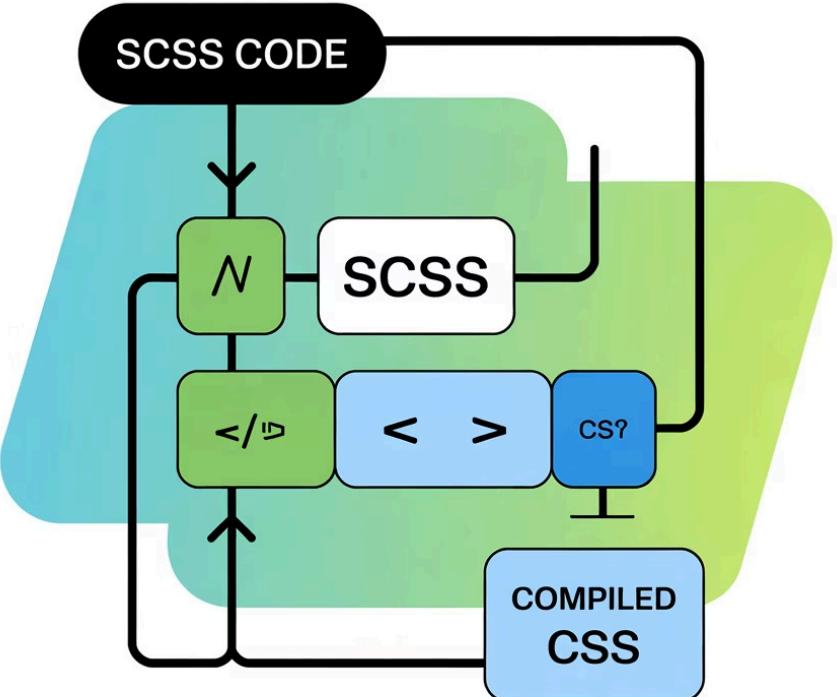
Ventajas: Estructura clara, evita colisiones de nombres y reduce la especificidad

```
<!-- HTML con BEM -->
<article class="card card--featured">
  <h2 class="card__title">Título</h2>
  <p class="card__content">Texto...</p>
  <button class="card__button card__button--primary">
    Leer más
  </button>
</article>

/* CSS con BEM */
.card { padding: 20px; }
.card--featured { border: 2px solid gold; }
.card__title { font-size: 1.5rem; }
.card__button--primary { background: blue; }
```

Más información: <https://getbem.com/>

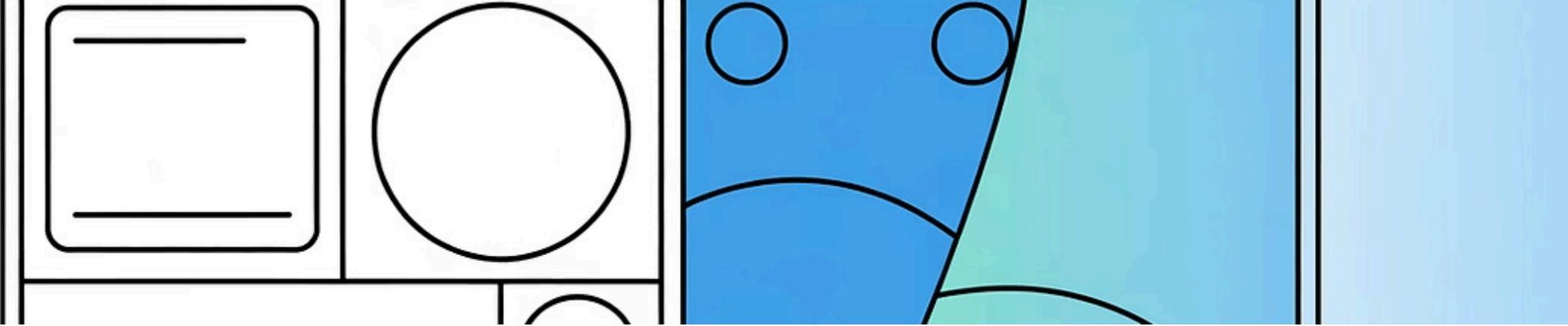
# Preprocesadores CSS



## Principales ventajas

- **Variables:** Almacenar colores, tamaños, etc.
- **Anidamiento:** Reflejar la jerarquía HTML
- **Mixins:** Reutilizar bloques de código
- **Funciones:** Operaciones y transformaciones
- **Imports:** Modularizar archivos

Opciones populares: **Sass/SCSS, Less, Stylus, PostCSS**



## Ejemplo práctico: Antes y después

Vamos a aplicar los conceptos aprendidos para transformar una página HTML básica en un diseño atractivo y profesional.

# Ejemplo: HTML base

```
<!DOCTYPE html>
<html lang="es">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Mi Portafolio</title>
<link rel="stylesheet" href="estilos.css">
</head>
<body>
<header class="cabecera">
<h1 class="cabecera_titulo">Ana Martínez</h1>
<p class="cabecera_subtitulo">Desarrolladora Web</p>
</header>

<nav class="navegacion">
<ul class="menu">
<li class="menu_item"><a href="#" class="menu_enlace">Inicio</a></li>
<li class="menu_item"><a href="#" class="menu_enlace">Proyectos</a></li>
<li class="menu_item"><a href="#" class="menu_enlace">Sobre mí</a></li>
<li class="menu_item"><a href="#" class="menu_enlace">Contacto</a></li>
</ul>
</nav>

<section class="proyectos">
<h2 class="proyectos_titulo">Mis Proyectos</h2>
<div class="tarjetas">
<article class="tarjeta">
<h3 class="tarjeta_titulo">Proyecto 1</h3>
<p class="tarjeta_descripcion">Descripción del proyecto...</p>
<a href="#" class="tarjeta_boton">Ver más</a>
</article>

<!-- Más tarjetas aquí... -->
</div>
</section>

<footer class="pie">
<p>© 2023 Ana Martínez</p>
</footer>
</body>
</html>
```

## Ana Martínez

Desarrolladora Web

- [Inicio](#)
- [Proyectos](#)
- [Sobre mí](#)
- [Contacto](#)

## Mis Proyectos

### Proyecto 1

Descripción del proyecto...

[Ver más](#)

© 2023 Ana Martínez

# Ejemplo: CSS aplicado

```
/* Reset y estilos base */
* {
margin: 0;
padding: 0;
box-sizing: border-box;
}

body {
font-family: 'Segoe UI', Tahoma, sans-serif;
line-height: 1.6;
color: #1E2C3B;
background-color: #F4FBFF;
}

/* Cabecera */
.cabecera {
background-color: #2589C9;
color: white;
padding: 3rem 2rem;
text-align: center;
}

.cabecera__titulo {
font-size: 2.5rem;
margin-bottom: 0.5rem;
}

.cabecera__subtitulo {
font-size: 1.2rem;
opacity: 0.9;
}

/* Navegación */
.navegacion {
background-color: #063E5F;
padding: 1rem 0;
}

.menu {
display: flex;
justify-content: center;
list-style: none;
}

.menu__item {
margin: 0 1rem;
}

.menu__enlace {
color: white;
text-decoration: none;
font-weight: 500;
transition: color 0.3s;
}

.menu__enlace:hover {
color: #75bae6;
}
```

# Ejemplo: CSS aplicado (continuación)

```
/* Sección de proyectos */
.proyectos {
  max-width: 1200px;
  margin: 3rem auto;
  padding: 0 2rem;
}

.proyectos__titulo {
  text-align: center;
  margin-bottom: 2rem;
  color: #063E5F;
}

.tarjetas {
  display: grid;
  grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
  gap: 2rem;
}

.tarjeta {
  background-color: white;
  border-radius: 8px;
  overflow: hidden;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
  transition: transform 0.3s, box-shadow 0.3s;
}

.tarjeta:hover {
  transform: translateY(-5px);
  box-shadow: 0 8px 16px rgba(0, 0, 0, 0.15);
}

.tarjeta__titulo {
  padding: 1.5rem 1.5rem 0;
  color: #2589C9;
}

.tarjeta__descripcion {
  padding: 1rem 1.5rem;
}

.tarjeta__boton {
  display: inline-block;
  margin: 0 1.5rem 1.5rem;
  padding: 0.5rem 1rem;
  background-color: #75bae6;
  color: white;
  text-decoration: none;
  border-radius: 4px;
  transition: background-color 0.3s;
}

.tarjeta__boton:hover {
  background-color: #2589C9;
}

/* Pie de página */
.pie {
  background-color: #1E2C3B;
  color: white;
  text-align: center;
  padding: 2rem;
  margin-top: 3rem;
}
```

# Ana Martínez

Desarrolladora Web

[Inicio](#) [Proyectos](#) [Sobre mí](#) [Contacto](#)

## Mis Proyectos

### Proyecto 1

Descripción del proyecto...

[Ver más](#)

	<cyce-eadx >
	<peseyed>
	<pcer-eddeclicde>
	<pectiad>
	<pesrcodx >
	<pceλmnax
	<yean/   >
	<yce-rcpoex   >

# Errores comunes y buenas prácticas

1

## Errores a evitar

- Abuso de **!important** y selectores de alta especificidad
- CSS **no modular** y difícil de mantener
- Uso excesivo de posicionamiento **absoluto**
- No considerar la **responsividad** desde el principio
- Duplicación de código por falta de abstracción

2

## Buenas prácticas

- Usar una **metodología** consistente (BEM, SMACSS...)
- Mantener baja **especificidad** en selectores
- Organizar CSS en **componentes reutilizables**
- Usar **variables** (CSS o preprocesador) para valores recurrentes
- Aplicar **mobile-first** para diseño responsive
- Documentar secciones importantes con comentarios

# #EDCOUNIANDES

<https://educacioncontinua.uniandes.edu.co/>

Contacto: [educacion.continua@uniandes.edu.co](mailto:educacion.continua@uniandes.edu.co)

© - Derechos Reservados: La presente obra, y en general todos sus contenidos, se encuentran protegidos por las normas internacionales y nacionales vigentes sobre propiedad Intelectual, por lo tanto su utilización parcial o total, reproducción, comunicación pública, transformación, distribución, alquiler, préstamo público e importación, total o parcial, en todo o en parte, en formato impreso o digital y en cualquier formato conocido o por conocer, se encuentran prohibidos, y solo serán lícitos en la medida en que se cuente con la autorización previa y expresa por escrito de la Universidad de los Andes.