



# UNIVERSIDAD NACIONAL DEL CENTRO DEL PERÚ

## FACULTAD DE INGENIERÍA DE SISTEMAS



### “DESARROLLO DE APLICACIONES WEB”

#### DOCENTE:

ING. SUASNABAR TERREL JAIME

#### INTEGRANTES:

- FERNANDEZ MARCELO GERALD
- LEON GARCIA DAVID DANIEL
- MONTES VARGAS RONALD
- ROJAS MOLINA YASER
- SAMANIEGO INGA ALEX
- VASQUEZ HUAMAN JHAIR

AÑO: 2025



# INTRODUCCIÓN

CSS (Cascading Style Sheets) es un lenguaje clave en el desarrollo web que se utiliza para definir la apariencia visual de páginas HTML, como colores, fuentes y disposición de elementos. Separa el contenido de la presentación, facilitando el mantenimiento y permitiendo aplicar estilos uniformes en todo el sitio. Funciona mediante reglas compuestas por selectores y declaraciones, y aunque puede integrarse en el HTML, lo más recomendable es usar archivos CSS externos para un código más limpio y reutilizable.

El trabajo presenta los fundamentos del lenguaje CSS, explicando el uso de selectores, propiedades de texto, colores, fondos y el modelo de caja. También aborda el posicionamiento de elementos, los sistemas Flexbox y Grid, pseudo-clases, pseudo-elementos, y animaciones básicas. Ofrece una visión práctica para crear interfaces web estructuradas, estéticas y funcionales.

# OBJETIVOS

## OBJETIVO GENERAL:

Comprender los conceptos fundamentales de CSS y su aplicación en la estilización y diseño visual de páginas web, con el fin de crear interfaces modernas, estructuradas y atractivas para el usuario.

## OBJETIVOS ESPECÍFICOS:

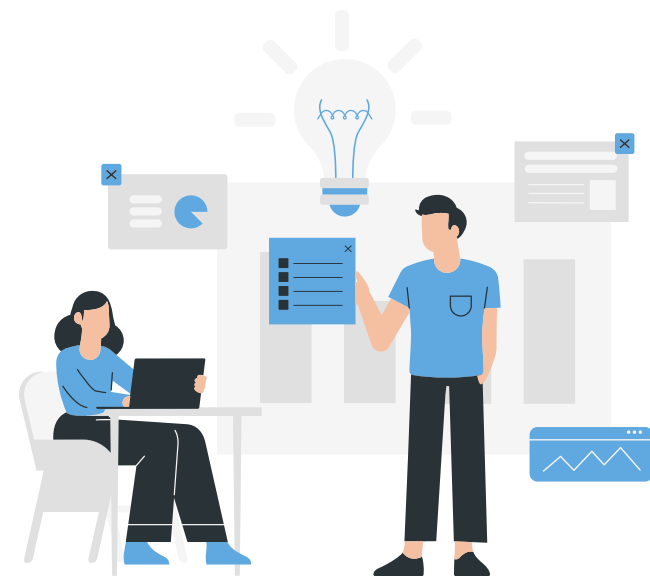
- Explicar la función de los selectores CSS y los distintos tipos existentes.
- Analizar el concepto de especificidad y su importancia en la resolución de conflictos de estilo.
- Aplicar propiedades relacionadas con el texto, fuentes, fondos y colores.
- Entender el Modelo de Caja (Box Model) como base del diseño de elementos en la web.
- Diferenciar y utilizar correctamente las unidades de medida absolutas y relativas en CSS.
- Implementar técnicas de posicionamiento de elementos usando valores como static, relative, absolute, fixed y sticky
- Diseñar distribuciones responsivas y flexibles mediante los sistemas de diseño Flexbox y Grid.
- Emplear pseudo-clases y pseudo-elementos para estilizar elementos de forma dinámica.
- Incorporar transiciones y animaciones básicas que enriquezcan la experiencia de navegación.
- Analizar el concepto de especificidad y su importancia en la resolución de conflictos de estilo.

# JUSTIFICACIÓN E IMPORTANCIA

En la era digital actual, ofrecer una experiencia web atractiva y funcional es esencial. Un sitio mal diseñado puede generar desconfianza y afectar negativamente la percepción del contenido. Por ello, el dominio de CSS básico es fundamental para mejorar la apariencia, rendimiento, accesibilidad y adaptabilidad de los sitios web en distintos dispositivos. Esta monografía busca proporcionar una base sólida en CSS, promoviendo buenas prácticas de diseño y codificación para crear sitios modernos y eficaces.



Para la elaboración de esta monografía se empleó una metodología basada en la revisión bibliográfica y documental. Se consultaron fuentes confiables como la documentación oficial del World Wide Web Consortium (W3C), libros especializados en desarrollo web y diseño con CSS, así como artículos técnicos de referencia actualizados disponibles en plataformas de desarrollo reconocidas. Además del análisis teórico, se presentan ejemplos prácticos y fragmentos de código comentado para ilustrar la aplicación de los conceptos explicados. Estos ejemplos tienen como objetivo facilitar la comprensión, especialmente para lectores que se inician en el uso de CSS. El enfoque didáctico de esta monografía permite al lector avanzar progresivamente desde los aspectos más básicos del lenguaje hasta técnicas más estructuradas, como el uso de Flexbox y Grid Layout.



## 1. Selectores y Especificidad

### Selectores

Son patrones que identifican qué elementos HTML deben ser modificados con las reglas de estilo que se definen en el código CSS. Es el primer componente de una regla CSS y funciona como una “dirección” que apunta a uno o varios elementos del documento HTML.

Selectores de tipo: se refieren directamente al nombre de una etiqueta HTML y aplican estilos a todas las instancias de esa etiqueta.

Selectores de clase: se aplican a cualquier elemento que posea el atributo class con el nombre indicado. Se utilizan precedidos por un punto (.).

Selectores de atributo: seleccionan elementos con atributos específicos.

Selectores de ID: se aplican a un único elemento con un identificador único. Se utilizan precedidos por el símbolo #.

### Especificidad

Es un mecanismo del navegador que se encarga de, en el caso de que exista ambigüedad y hayan varios bloques de estilos compitiendo uno contra otro para dar estilo a un mismo elemento, resolver dicha ambigüedad y decidir cuál es el grupo de estilos «vencedor».

Orden de prioridad (de mayor a menor):

1. Estilos en línea ( escritos directamente en el atributo style del HTML ).
2. Selectores de ID
3. Selectores de clase, atributos y pseudo-clases.
4. Selectores de tipo o etiquetas.
5. Reglas universales o herencia (peso menor)

# Ejemplo:

```
/* 1. Selectores y Especificidad */  
body {  
    font-family: 'Arial', sans-serif;  
    margin: 0;  
    padding: 0;  
}
```

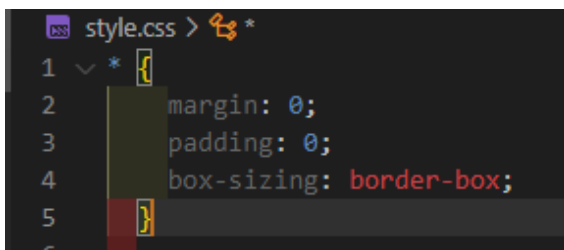
## CSS

```
p { color: blue; }  
.texto { color: green; }  
#importante { color: red; }
```

El navegador aplicará el color **rojo**, ya que el selector de ID #importante tiene mayor especificidad que los demás.

## 1.1 Selectores Universal

El selector universal en CSS es representado por el asterisco (\*). Su función principal es seleccionar todos los elementos del documento HTML, sin importar su tipo, clase, ID u otra característica. Este selector se utiliza comúnmente cuando se desea aplicar un estilo global o resetear márgenes y rellenos predeterminados del navegador.



```
style.css > *
1  * {
2    margin: 0;
3    padding: 0;
4    box-sizing: border-box;
5  }
```

En este ejemplo, todos los elementos del sitio web se verán afectados por estas reglas: se eliminan los márgenes y rellenos predeterminados, y se establece la propiedad box-sizing en border-box, lo cual facilita el control de dimensiones en el diseño web.

### Ventajas:

- Uniformiza el estilo base en todos los navegadores.
- Útil para establecer estilos iniciales comunes en toda la página.
- Facilita la implementación de diseños consistentes.

### Precauciones:

- Puede afectar el rendimiento si se usa en documentos muy grandes, ya que selecciona todos los elementos.
- No debe usarse para aplicar estilos específicos o complejos.



## 1.2 Selectores Etiqueta

El selector de etiqueta, también conocido como selector de tipo, es uno de los selectores más básicos y comunes en las hojas de estilo en cascada (CSS). Este selector permite aplicar estilos directamente a todos los elementos HTML de un mismo tipo o etiqueta, como `<p>`, `<h1>`, `<div>`, `<a>`, entre otros, sin necesidad de utilizar clases o identificadores.

```
style.css > ...  
1  p {  
2    font-size: 16px;  
3    line-height: 1.5;  
4    color: #333333;  
5  }
```

En este ejemplo, todas las etiquetas `<p>` (párrafos) del documento HTML tendrán el mismo tamaño de fuente, interlineado y color de texto. Este enfoque es ideal cuando se quiere mantener una estructura uniforme en bloques de contenido.

### Ventajas:

- Permite estilizar elementos de forma rápida y coherente.
- Reduce la necesidad de repetir clases cuando el estilo aplica a muchos elementos del mismo tipo.
- Mejora la legibilidad del código CSS.

### Precauciones:

- No debe utilizarse en exceso para personalizaciones específicas, ya que puede limitar la flexibilidad en el diseño.
- Puede ser sobrescrito por selectores más específicos como clases, identificadores o selectores compuestos.

## 1.4 Agupaciones de selectores

La agrupación de selectores permite aplicar las mismas reglas de estilo a varios elementos sin tener que repetir esas reglas para cada selector.

### Sintaxis

La agrupación se logra mediante la separación de múltiples selectores mediante comas (,). La estructura general es la siguiente:

```
selector1, selector2, selector3 {  
  propiedad: valor;  
}
```

```
h1, h2, h3 {  
  color: #333333;  
  font-family: 'Segoe UI', sans-serif;  
}
```

### Ventajas de la Agrupación

**Mantenimiento sencillo:** cualquier cambio en las propiedades afecta a todos los elementos agrupados, evitando la necesidad de modificar cada uno por separado.

**Código más limpio:** reduce la cantidad de líneas de código y mejora la claridad visual.

**Rendimiento:** permite al navegador interpretar los estilos más rápidamente debido a la simplicidad del archivo CSS.

### Aplicaciones Comunes

- Aplicar márgenes y padding uniformes a distintos tipos de contenedores (div, section, article).
- Unificar estilos de botones que comparten una lógica visual pero difieren en funcionalidad (.btn-guardar, .btn-cancelar, .btn-editar).
- Estilizar todos los elementos de formulario con bordes, tipografía y tamaños coherentes (input, select, textarea).

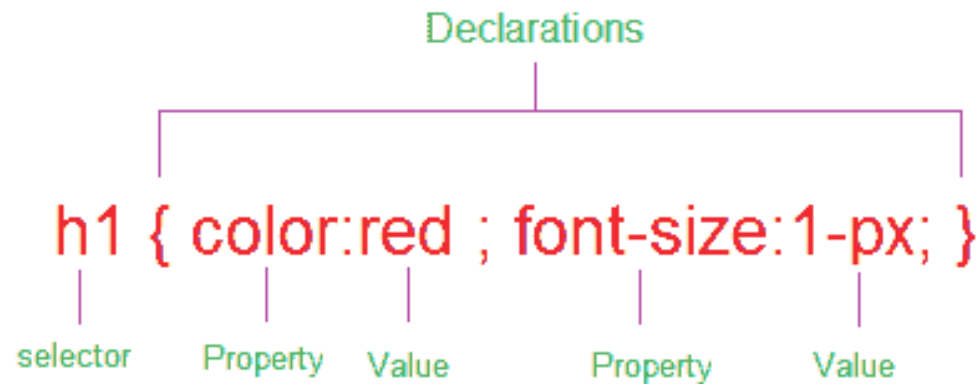
## 1.5 Selector ID

El selector ID en CSS es una herramienta que permite aplicar estilos a un único elemento HTML que tenga un identificador único (id). Es uno de los selectores más específicos y poderosos del lenguaje CSS.

### Sintaxis

CSS

```
#nombre-id {  
    propiedad: valor;  
}
```



- El selector comienza con el símbolo # seguido del nombre del id.
- Se recomienda que el id sea único dentro del documento HTML.

# Ejemplo:

```
body {
  font-family: 'Roboto', sans-serif;
  background-color: #f9f9f9;
  margin: 0;
  padding: 40px;
}

#titulo-principal {
  color: #2c3e50;
  font-size: 36px;
  text-align: center;
  margin-bottom: 20px;
  text-shadow: 2px 2px 4px rgba(0,0,0,0.1);
}


#mensaje {
  color: #a94442;
  background-color: #f2dede;
  padding: 20px;
  border-left: 6px solid #d9534f;
  border-radius: 8px;
  font-weight: bold;
  max-width: 600px;
  margin: 0 auto;
  box-shadow: 0 4px 6px rgba(0,0,0,0.1);
}
```

## Bienvenido a Mi Página

Este es un mensaje importante.


## 1.6 SELECTORES DE ETIQUETA DE CLASES

Este selector es un selector de etiqueta, que además restringe su aplicación a la condición de llevar el atributo de clase o de id que se le indica.

```
P.parrafoModificar{  
    border: 1px solid  red;  
}
```

El selector selecciona únicamente a las etiquetas p que lleven el nombre de clase parrafoModificar.

Este selector se puede aplicar con el atributo id.

```
p#parrafoModificar{  
    border: 1px solid  red;  
    font-family: 'Times New Roman', Times, serif;  
}
```

No es lo mismo el selector p.tipo1 que el selector p .tipo1, el segundo es un selector descendente

## 2. Propiedades de Texto y Fuentes

El texto es uno de los componentes más importantes de una página web, por lo tanto, su presentación debe ser clara, legible y coherente con el diseño general.

### Principales propiedades de texto y fuentes:

font-family: define la familia tipográfica.

font-size: tamaño del texto.

font-style: estilo de fuente (normal, cursiva, etc.).

font-weight: grosor del texto.

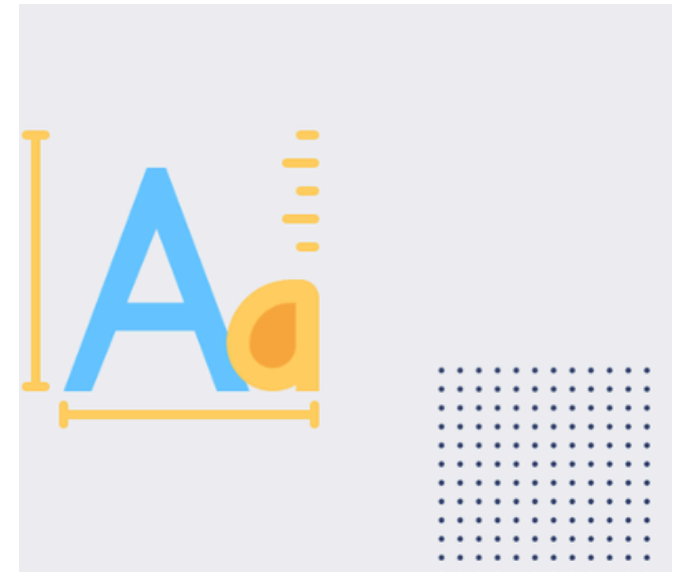
line-height: espacio entre líneas.

text-align: alineación del texto.

text-transform: transforma el texto (mayúsculas, minúsculas).

letter-spacing: espacio entre letras.

word-spacing: espacio entre palabras.



# Ejemplo:

```
/* Estilo para todo el cuerpo de la página */
body {
    font-family: 'Arial', sans-serif; /* Familia tipográfica por defecto */
    background-color: #f4f4f4; /* Color de fondo suave */
    margin: 0;
    padding: 0;
    font-size: 16px; /* Tamaño base de texto */
}

/* Estilo específico para el párrafo */
p {
    font-family: 'Helvetica', sans-serif; /* Familia tipográfica */
    font-size: 18px; /* Tamaño del texto */
    line-height: 1.6; /* Espacio entre líneas */
    text-align: justify; /* Alineación justificada */
    font-weight: 400; /* Grosor del texto */
    color: #333; /* Color de texto */
    margin: 20px; /* Márgenes */
}

h1 {
    text-align: center; /* Título centrado */
    color: #333;
}
```

## Ejemplo de Propiedades de Texto y Fuentes

El texto es uno de los componentes más importantes de una página web. Es fundamental que su presentación sea clara, legible y coherente con el diseño general del sitio. Las propiedades de texto permiten controlar la apariencia y el comportamiento del contenido textual en una página web.

## 3. Modelo de Caja (Box Model)

Todo elemento HTML se comporta como una caja rectangular compuesta por varias capas que definen cómo se posiciona y ocupa espacio en la página. Este modelo es clave para controlar el diseño y la distribución de los elementos.

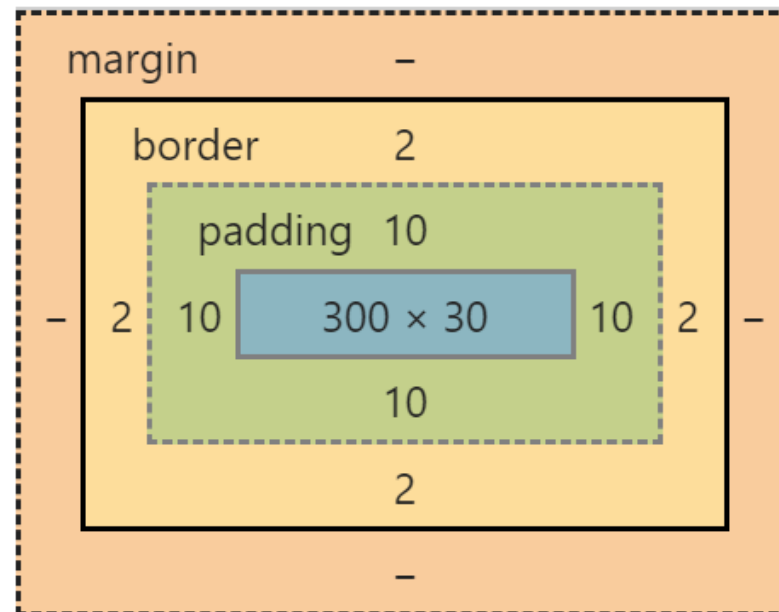
### Partes del Box Model:

Content (Contenido): el área donde se muestra el texto o los elementos hijos.

Padding: espacio entre el contenido y el borde del elemento.

Border: borde que rodea el padding (opcional).

Margin: espacio exterior entre el borde del elemento y otros elementos cercanos.





# Ejemplo:

```
/* Estilos globales */
body {
    font-family: 'Arial', sans-serif;
    background-color: #4f4f4f;
    margin: 0;
    padding: 0;
    font-size: 16px;
    text-align: center;
}

/* Estilo para la caja .box */
.box {
    width: 300px; /* Ancho de la caja */
    padding: 20px; /* Espacio interno (entre el contenido y el borde) */
    border: 5px solid black; /* Borde negro de 5px */
    margin: 30px; /* Espacio externo (entre la caja y los elementos cercanos) */
    background-color: #fff; /* Fondo blanco para la caja */
    color: #333; /* Color de texto dentro de la caja */
}

h1 {
    color: #333;
}
```

## Ejemplo del Modelo de Caja (Box Model)

Este es el contenido dentro de la caja.

## 4. Colores y Fondos

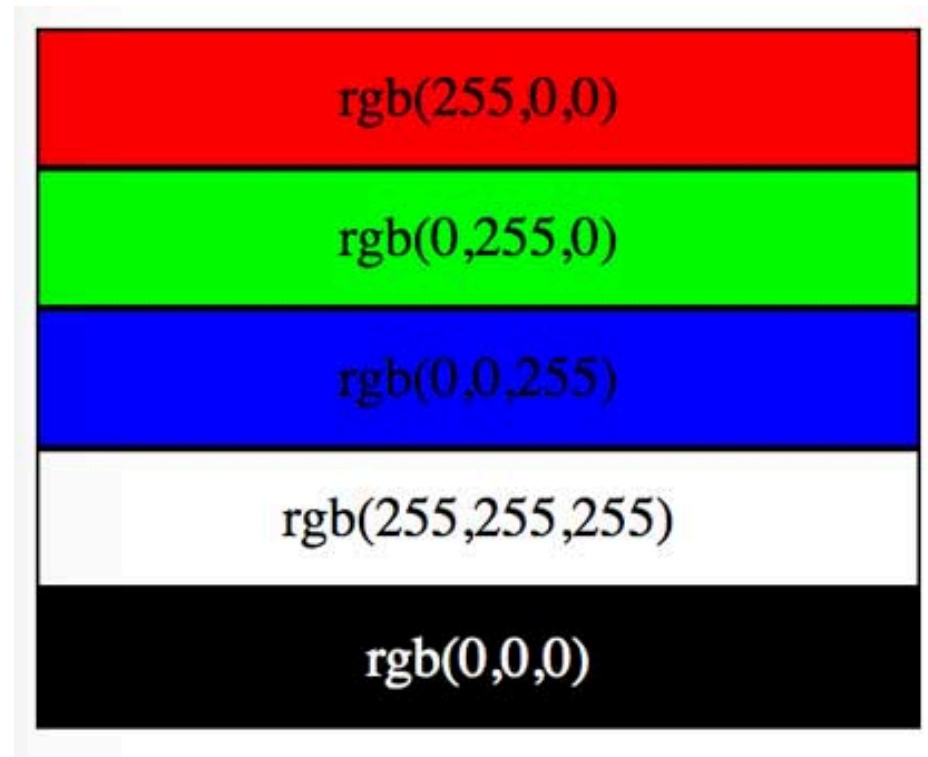
CSS permite aplicar colores de diversas formas y añadir fondos a los elementos para mejorar el diseño visual.

### Formas de definir colores

- Por nombre: red, blue, green, etc.
- Hexadecimal: #ff0000, #00ff00
- RGB: `rgb(255, 0, 0)`
- RGBA: `rgba(255, 0, 0, 0.5)` (con opacidad)

### Propiedades para fondos:

- `background-color`: color de fondo.
- `background-image`: imagen de fondo.
- `background-repeat`: repetición de la imagen.
- `background-size`: escala de la imagen.
- `background-position`: posición del fondo.



# Ejemplo:

```
/* Estilo base para los bloques */
.box {
  width: 70%; /* El ancho de los bloques será el 70% del contenedor */
  margin: 20px auto; /* Márgenes automáticos para centrar los bloques y espacio entre ellos */
  padding: 20px; /* Espacio interno dentro de los bloques */
  border: 3px solid #333; /* Borde alrededor de los bloques */
  box-sizing: border-box; /* Asegura que el borde esté incluido en el tamaño total */
  text-align: center; /* Centra el texto dentro de cada bloque */
  border-radius: 10px; /* Bordes redondeados */
}

/* Caja con fondo rojo y gradiente */
.red {
  background: linear-gradient(45deg, #ff7e5f, #feb47b); /* Gradiente de rojo a naranja */
  color: #fff; /* Texto blanco para mayor contraste */
}
```

## Ejemplo de Dos Cajas con Colores y Fondos

### Caja Roja

Este es un bloque con un fondo gradiente rojo.

### Caja Azul

Este bloque tiene un fondo azul sólido con borde blanco.

## 5. Unidades de Medida

En CSS, las unidades de medida definen tamaños, espaciados, anchos, alturas, márgenes, bordes, etc. Se dividen principalmente en dos tipos: absolutas y relativas.

### 5.1 Absolutas

Estas no dependen de ningún otro elemento, son fijas.

- px → píxeles (la más usada)
- cm → centímetros
- mm → milímetros
- in → pulgadas (1in = 2.54cm)
- pt → puntos (1pt = 1/72 de pulgada)
- pc → picas (1pc = 12pt)

### 5.2 Relativas

Estas dependen del tamaño de otro elemento o del viewport.

- em → relativo al tamaño de la fuente del elemento padre
- rem → relativo al tamaño de la fuente del root (html)
- % → porcentaje respecto al padre (por ejemplo, width: 50%)
- vw → 1% del ancho de la ventana del navegador (viewport width)
- vh → 1% de la altura de la ventana (viewport height)
- vmin → 1% del valor más pequeño entre ancho y alto del viewport
- vmax → 1% del valor más grande entre ancho y alto del viewport
- ex → altura de la letra 'x' en la fuente usada
- ch → ancho del carácter '0' en la fuente usada

## 6. Posicionamiento

La propiedad position permite cambiar el flujo normal de los elementos en la página

### Tipos de posicionamiento:

Propiedad	¿Qué hace?
<b>static</b>	Posicionamiento por defecto. No se mueve, sigue el flujo normal del HTML.
<b>relative</b>	Se posiciona relativo a donde debería estar originalmente.
<b>absolute</b>	Se posiciona absolutamente respecto al ancestro más cercano con posición relative o absolute.
<b>fixed</b>	Se posiciona fijo respecto a la ventana del navegador. No se mueve al hacer scroll.
<b>sticky</b>	Se comporta como relative hasta que alcanzas un punto del scroll, luego se vuelve fixed.

## 7. Flexbox (Diseño flexible)

Flexbox (Flexible Box Layout) es un sistema de CSS que permite distribuir y alinear elementos de forma rápida y responsiva en un contenedor.

**Objetivo:** hacer layouts (diseños) flexibles y adaptables.

### Ventajas:

- Diseños más limpios y mantenibles
- Alineación sencilla en ejes horizontal y vertical
- Ideal para menús, grids y centrado perfecto

### Conceptos Clave

Flexbox trabaja con dos ejes:

- Eje principal (main axis) → Dirección en la que se ordenan los elementos (por defecto: horizontal).
- Eje transversal (cross axis) → Dirección perpendicular al eje principal (por defecto: vertical).

### Estructura Básica

Para usar Flexbox, necesitamos:

- Un contenedor flex (padre)
- items flex (hijos)

```
.contenedor {  
  display: flex; /* ¡Activa Flexbox! */  
}
```

## 7. Flexbox (Diseño flexible)

### Propiedades del Contenedor Flex

PROPIEDAD	VALORES COMUNES	DESCRIPCIÓN
DISPLAY	flex   inline-flex	Activa el modelo Flexbox
FLEX DIRECTION	row   row-reverse   column   column-reverse	Define la dirección del eje principal
FLEX-WRAP	nowrap   wrap   wrap-reverse	Controla si los items saltan de línea
JUSTIFY CONTENT	flex-start   flex-end   center   space-between   space-around   space-evenly	Alinea items en el <b>eje principal</b>
ALIGN ITEM	stretch   flex-start   flex-end   center   baseline	Alinea items en el <b>eje transversal</b>
ALIGN CONTENT	flex-start   flex-end   center   space-between   space-around	Alinea <b>líneas múltiples</b>

## 7. Flexbox (Diseño flexible)

### Propiedades de los Items Flex (Children)

Propiedad	Valores comunes	Descripción	Ejemplo visual
order	Número (ej. 0, 1, -1)	Cambia el orden visual sin modificar HTML	order: 2; ( → → )
flex-grow	Número (ej. 0, 1, 2)	Define cómo crece el ítem si hay espacio extra	flex-grow: 1; ( → → )
flex-shrink	Número (ej. 1, 0)	Define cómo se encoge el ítem si falta espacio	flex-shrink: 0; ( ↯ → )
flex-basis	auto   px   % (ej. 100px)	Tamaño base antes de distribuir espacio	flex-basis: 200px;
align-self	auto   flex-start   flex-end   center   baseline   stretch	Alineación <b>individual</b> en el eje transversal	align-self: center; ( ↑ □ ↓ )

```

.contenedor {
  display: flex;
  flex-direction: row;    /* → → → */
  justify-content: center; /* → □ ← */
  align-items: flex-end;  /* ↓ (abajo) */
  flex-wrap: wrap;        /* ↪ salto de línea */
}

.item {
  flex: 1;                /* Todos crecen igual */
  order: 1;               /* Orden normal */
}

```




## 8. Grid Layout (Diseño de cuadrícula)


CSS Grid Layout es un sistema de diseño web bidimensional que permite organizar elementos en filas y columnas, ofreciendo una mayor flexibilidad y control sobre la posición y tamaño de los elementos en comparación con los sistemas de diseño tradicionales.





```
<> index.html X # style.css

<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=5.0">
6      <title>Document</title>
7      <Link rel="stylesheet" href="style.css">
8  </head>
9  <body>
10     <div class="container">
11         <div class="element element-1">#1</div>
12         <div class="element element-2">#2</div>
13         <div class="element element-3">#3</div>
14         <div class="element element-4">#4</div>
15     </div>
16 </body>
17 </html>
```

# DESARROLLO

```
# style.css >  .container
```

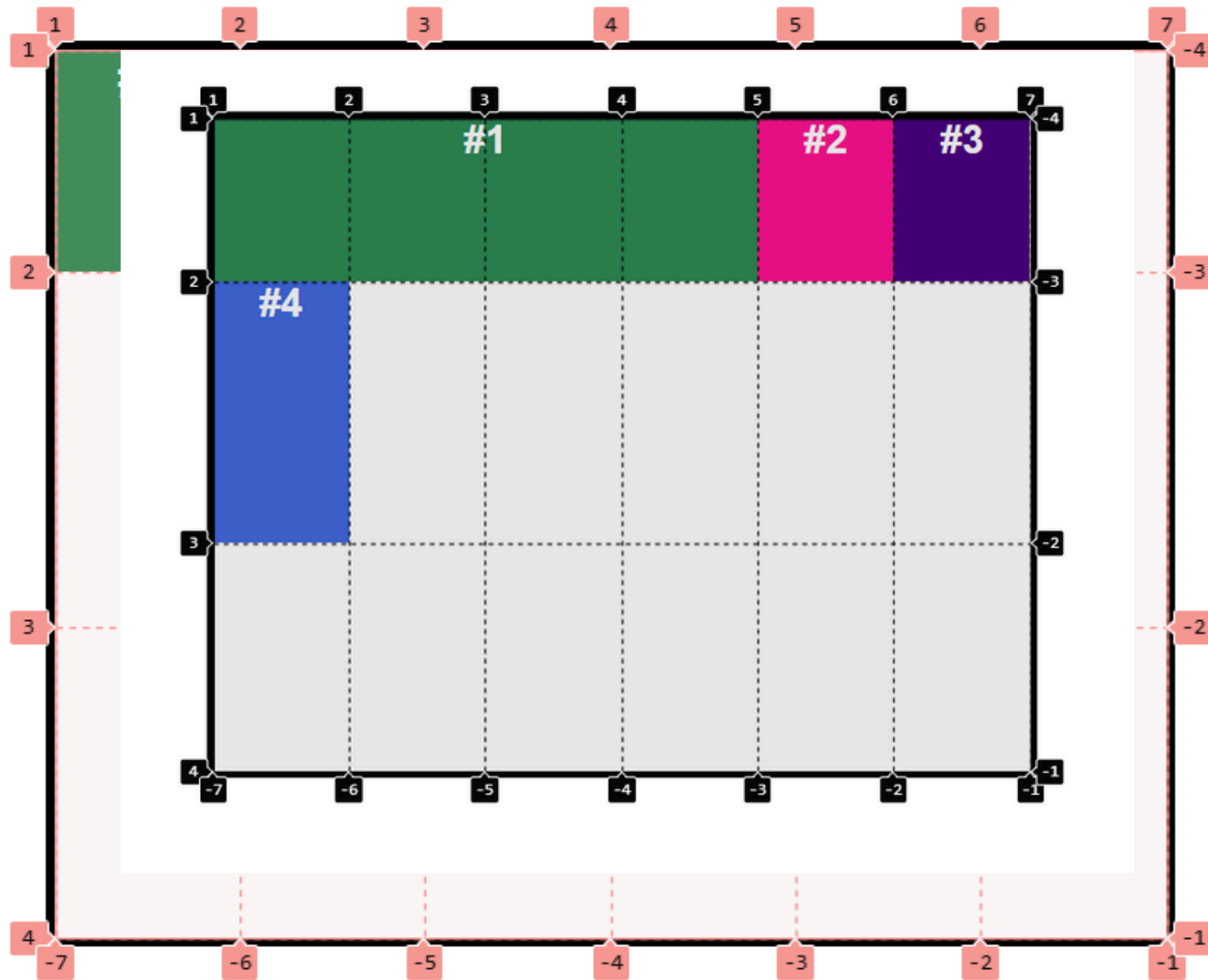
```
1  *{
2      margin: 100;
3      box-sizing: border-box;
4  }
5
6  body{
7      font-family: Arial;
8  }
9
10 .container{
11     margin: 80px auto;
12     width: 90%;
13     height: 400px;
14     max-width: 500px;
15     outline: 4px solid;
16 }
17
18 .element{
19     color:  #fff;
20     text-align: center;
21     font-size: 1.5rem;
22     font-weight: bold;
23 }
```

```
24
25 .element-1{
26     background-color:  seagreen;
27 }
28
29 .element-2{
30     background-color:  deeppink;
31 }
32
33 .element-3{
34     background-color:  indigo;
35 }
36
37 .element-4{
38     background-color:  royalblue;
39 }
40
```

#1
#2
#3
#4

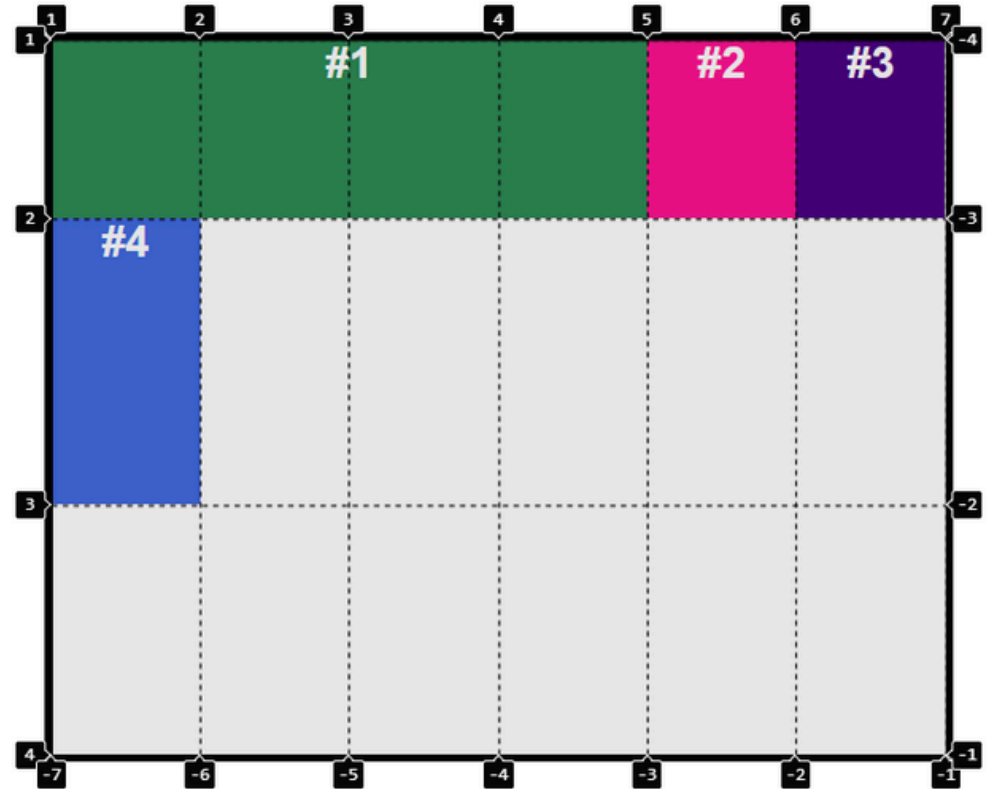
```
10  .container{
11      margin: 80px auto;
12      width: 90%;
13      height: 400px;
14      max-width: 500px;
15      outline: 4px solid;
16
17      display: grid;
18      grid-template-columns: repeat(6, 1fr);
19      grid-template-rows: 100px 10em 1fr;
20  }
```

# DESARROLLO



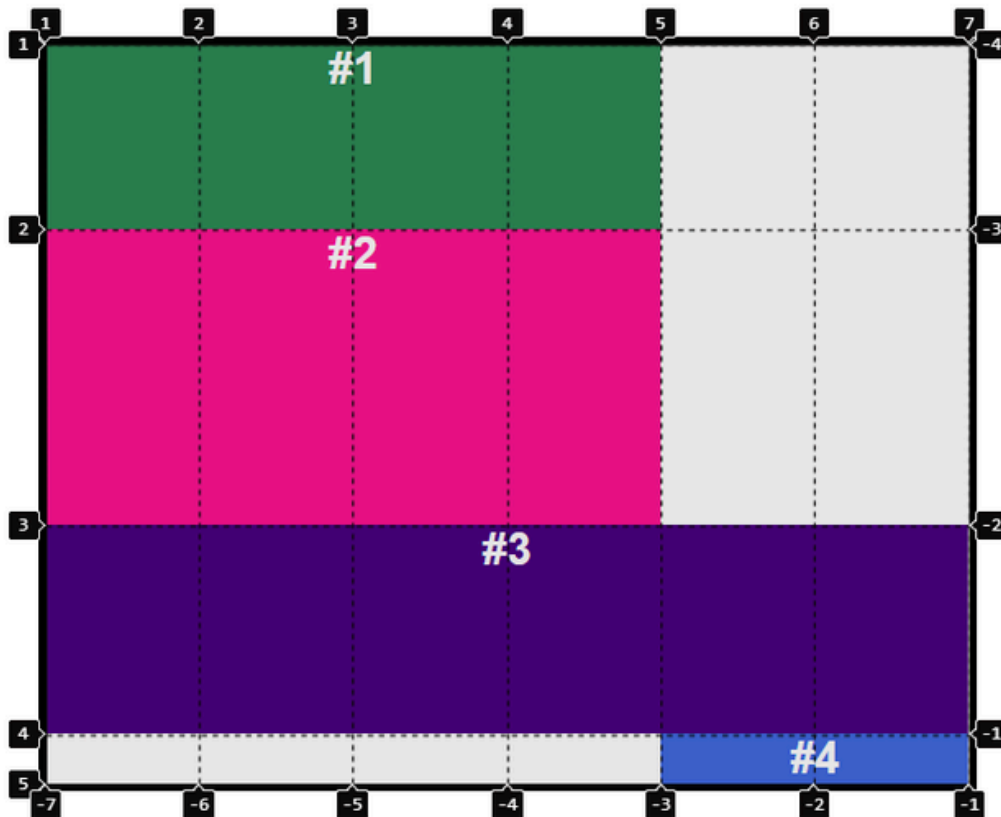
# DESARROLLO

```
29 .element-1{
30     background-color: ■ seagreen;
31     grid-column: 1/5;
32 }
33
34 .element-2{
35     background-color: ■ deeppink;
36 }
37
38 .element-3{
39     background-color: ■ indigo;
40 }
41
42 .element-4{
43     background-color: ■ royalblue;
44 }
45
```



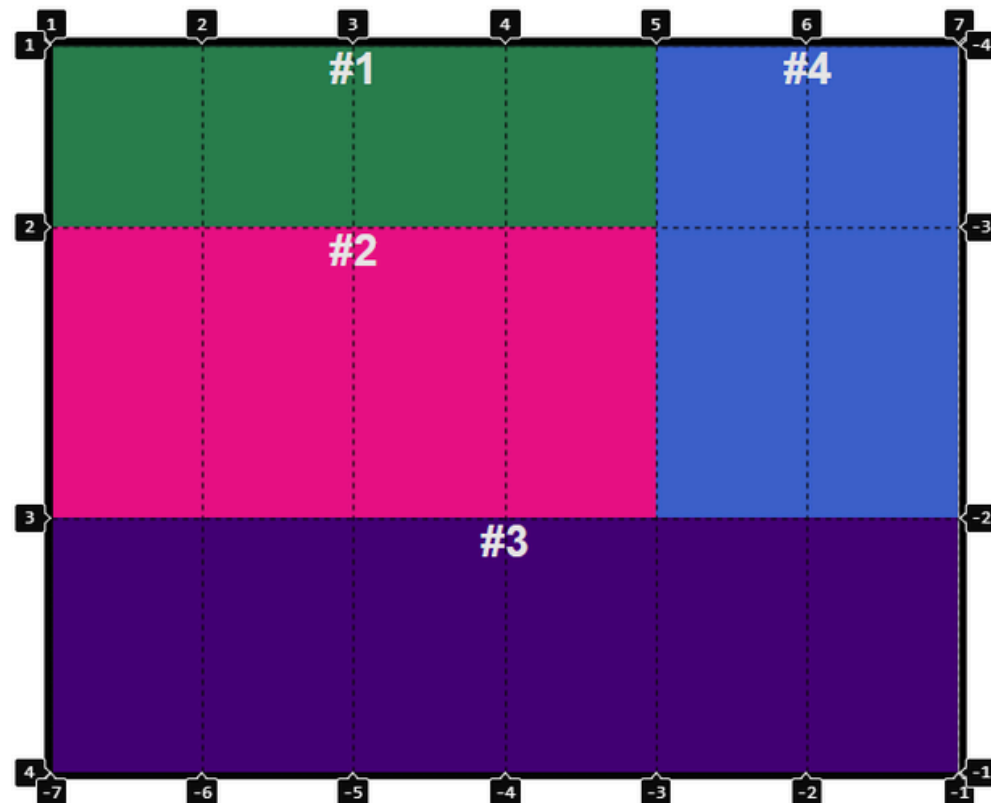
# DESARROLLO

```
29 .element-1{
30     background-color: ■ seagreen;
31     grid-column: 1/5;
32 }
33
34 .element-2{
35     background-color: ■ deeppink;
36     grid-column: 1/5;
37 }
38
39 .element-3{
40     background-color: ■ indigo;
41     grid-column: 1/7;
42 }
43
44 .element-4{
45     background-color: ■ royalblue;
46     grid-column: 5/7;
47 }
48
```



# DESARROLLO

```
29 .element-1{
30     background-color: ■ seagreen;
31     grid-column: 1/5;
32     grid-row: 1/2;
33 }
34
35 .element-2{
36     background-color: ■ deeppink;
37     grid-column: 1/5;
38     grid-row: 2/3;
39 }
40
41 .element-3{
42     background-color: ■ indigo;
43     grid-column: 1/7;
44     grid-row: 3/4;
45 }
46
47 .element-4{
48     background-color: ■ royalblue;
49     grid-column: 5/7;
50     grid-row: 1/3;
51 }
```





## 9. Pseudo-clases y Pseudo-elementos

### 9.1 Pseudo-clases

Una pseudoclase CSS es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado.

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <title>Ejemplo de Pseudoclasses</title>
6    <style>
7      a {
8        color: blue; /* Establece el color azul por defecto para los enlaces */
9        text-decoration: none; /* Quita el subrayado por defecto */
10     }
11
12     a:hover {
13       color: red; /* Cambia el color del enlace a rojo cuando el mouse pasa por encima */
14     }
15
16     a:visited {
17       color: purple; /* Cambia el color a morado si el enlace ya fue visitado */
18     }
19
20     a:active {
21       color: green; /* Cambia el color a verde mientras se hace clic en el enlace */
22     }
```

```
24     p:first-child {
25         font-weight: bold; /* Si el párrafo es el primer hijo de su contenedor, el texto será en negrita */
26     }
27
28     p:last-child {
29         font-style: italic; /* Si el párrafo es el último hijo, el texto será en cursiva */
30     }
31
32     li:nth-child(odd) {
33         background-color: #f0f0f0; /* Aplica fondo gris claro a los elementos impares de la lista */
34     }
35
36     li:nth-child(even) {
37         background-color: #d0d0d0; /* Aplica fondo gris oscuro a los elementos pares de la lista */
38     }
39 </style>
40 </head>
41 <body>
42
43     <a href="https://example.com">Visita mi página</a>
44
45     <div>
46         <p>Este es el primer párrafo.</p>
47         <p>Este es otro párrafo.</p>
48         <p>Este es el último párrafo.</p>
49     </div>
50
51     <ul>
52         <li>Elemento 1</li>
53         <li>Elemento 2</li>
54         <li>Elemento 3</li>
55         <li>Elemento 4</li>
56     </ul>
57
58 </body>
59 </html>
```

[Visita mi página](#)

**Este es el primer párrafo.**

Este es otro párrafo.

*Este es el último párrafo.*

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4

[Visita mi página](#)

**Este es el primer párrafo.**

Este es otro párrafo.

*Este es el último párrafo.*

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4

## 9.1 Pseudo-elementos

Al igual que las Pseudo-classes, los pseudo-elementos se añaden a los selectores, pero en cambio, no describen un estado especial sino que, permiten añadir estilos a una parte concreta del documento.

```
7  .cita {
8      position: relative; /* Necesario para posicionar los pseudoelementos de forma relativa */
9      padding-left: 30px; /* Agrega espacio a la izquierda para el símbolo de apertura de comillas */
10     padding-right: 30px; /* Espacio a la derecha para el símbolo de cierre de comillas */
11     font-style: italic; /* Da estilo de cursiva al texto */
12     color: #333; /* Color de texto gris oscuro */
13 }
14
15 .cita::before {
16     content: "“"; /* Inserta el símbolo de comillas de apertura */
17     position: absolute;
18     left: 0;
19     font-size: 30px; /* Tamaño grande para la comilla */
20     color: #999; /* Color gris claro */
21 }
22
23 .cita::after {
24     content: "”"; /* Inserta el símbolo de comillas de cierre */
25     position: absolute;
26     right: 0;
27     font-size: 30px; /* Tamaño grande para la comilla */
28     color: #999; /* Color gris claro */
29 }
```

# DESARROLLO

```
35     p::first-letter {
36         font-size: 200%; /* La primera letra del párrafo en tamaño grande */
37         color: darkred; /* Color rojo oscuro */
38     }
39 </style>
40 </head>
41 <body>
42
43     <div class="cita">
44         La educación es el arma más poderosa que puedes usar para cambiar el mundo.
45     </div>
46
47     <p>
48         Este es un párrafo de ejemplo para mostrar cómo se puede aplicar estilo solo
49         a la primera línea y a la primera letra usando pseudoelementos.
50     </p>
51
52 </body>
53 </html>
```

*“ La educación es el arma más poderosa que puedes usar para cambiar el mundo. ”*


**E**ste es un párrafo de ejemplo para mostrar cómo se puede aplicar estilo solo a la primera línea y a la primera letra usando pseudoelementos.

# DESARROLLO

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <title>Galería CSS Interactiva</title>
6
7    <style>
8      /* Estilos generales del cuerpo */
9      body {
10        font-family: Arial, sans-serif; /* Fuente del texto */
11        background-color: #f4f4f4; /* Color de fondo general */
12        margin: 0;
13        padding: 40px; /* Espacio alrededor del contenido */
14      }
15
16      /* Contenedor con GRID */
17      .grid-container {
18        display: grid; /* Activa Grid Layout */
19        grid-template-columns: repeat(1, minmax(250px, 1fr));
20        /* Crea columnas automáticas con mínimo 250px, máximo 1fr (flexibles) */
21        gap: 20px; /* Espacio entre las tarjetas */
22      }
23
24      /* Estilo base para cada tarjeta */
25      .card {
26        position: relative; /* Necesario para posicionar los pseudo-elementos */
27        background-color: white; /* Fondo blanco de la tarjeta */
28        border-radius: 10px; /* Bordes redondeados */
29        padding: 20px; /* Espacio interno */
30        overflow: hidden; /* Oculta contenido que se desborde (como ::before) */
31        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1); /* Sombra suave */
32        transition: transform 0.3s; /* Suaviza el efecto hover */
33      }
34
35      .card:hover {
36        transform: translateY(-10px); /* Mueve la tarjeta hacia arriba al pasar el mouse */
37      }
```

# DESARROLLO

```
39  /* ::before agrega una burbuja decorativa arriba a la derecha */
40  .card::before {
41      content: ""; /* No hay texto, solo se muestra la forma */
42      position: absolute;
43      top: -20px;
44      right: -20px;
45      width: 60px;
46      height: 60px;
47      background-color:  rgba(0, 123, 255, 0.2); /* Azul transparente */
48      border-radius: 50%; /* Lo convierte en un círculo */
49      z-index: 0; /* Coloca el círculo detrás del contenido */
50  }
51
52  /* ::after agrega un ícono decorativo (estrella por defecto) */
53  .card::after {
54      content: "★"; /* Contenido que se mostrará */
55      position: absolute;
56      bottom: 10px;
57      left: 10px;
58      font-size: 20px;
59      color:  gold; /* Color dorado */
60      opacity: 0.5; /* Semitransparente */
61  }
62
63  /* Estilo especial para las tarjetas impares (1ra, 3ra, etc.) */
64  .card:nth-child(odd) {
65      background-color:  #eaf6ff; /* Azul muy claro */
66  }
67
68  /* A las tarjetas pares se les cambia el ícono a "✓" */
69  .card:nth-child(even)::after {
70      content: "✓"; /* Cambia la estrella por un check verde */
71      color:  green;
72  }
```

```
74      /* La primera tarjeta tendrá una burbuja roja en lugar de azul */
75      .card:first-child::before {
76      |   background-color:  rgba(255, 0, 0, 0.2); /* Rojo transparente */
77      |   }
78
79      /* Estilo de los títulos dentro de las tarjetas */
80      .card h3 {
81      |   margin-top: 0;
82      |   z-index: 1; /* Encima del círculo decorativo */
83      |   position: relative; /* Necesario para que el z-index funcione */
84      |   }
85
86      /* Estilo de los párrafos en las tarjetas */
87      .card p {
88      |   z-index: 1;
89      |   position: relative;
90      |   font-size: 0.95rem; /* Tamaño de fuente ligeramente reducido */
91      |   }
92
93      </style>
94      </head>
95      <body>
```



# DESARROLLO

```
95 <body>
96
97 <!-- Contenedor de las tarjetas -->
98 <div class="grid-container">
99
100 <!-- Tarjeta 1 -->
101 <div class="card">
102 | <h3>Tarjeta 1</h3>
103 | <p>Contenido destacado con un estilo especial.</p>
104 </div>
105
106 <!-- Tarjeta 2 -->
107 <div class="card">
108 | <h3>Tarjeta 2</h3>
109 | <p>Otra tarjeta con ícono y color diferente.</p>
110 </div>
111
112 <!-- Tarjeta 3 -->
113 <div class="card">
114 | <h3>Tarjeta 3</h3>
115 | <p>Las tarjetas impares tienen fondo celeste claro.</p>
116 </div>
117
118 <!-- Tarjeta 4 -->
119 <div class="card">
120 | <h3>Tarjeta 4</h3>
121 | <p>Incluye un ícono de verificación en la esquina.</p>
122 </div>
123
124 </div>
125
126 </body>
127 </html>
```

## Tarjeta 1

Contenido destacado con un estilo especial.



## Tarjeta 2

Otra tarjeta con ícono y color diferente.



## Tarjeta 3

Las tarjetas impares tienen fondo celeste claro.



## Tarjeta 4

Incluye un ícono de verificación en la esquina.



## 10. Transiciones y Animaciones básicas

CSS también permite añadir efectos dinámicos como transiciones suaves o animaciones personalizadas

### TRANSICIONES

Los cambios de una propiedad ocurran de forma progresiva en lugar de instantánea.

- transition-property: la propiedad a animar
- transition-duration: duración del efecto
- transition-timing-function: curva de aceleración (ease, linear, etc.
- transition-delay: retardo en comenzar

```
button {  
  background-color: green;  
  transition: background-color 0.5s ease;  
}  
  
button:hover {  
  background-color: darkgreen;  
}
```

```
@keyframes aparecer {  
  from { opacity: 0; }  
  to { opacity: 1; }  
}  
  
.caja {  
  animation: aparecer 1s ease-in-out;  
}
```

### ANIMACIONES

Permiten definir una serie de etapas o fotogramas (keyframes) que describen cómo debe evolucionar una propiedad.

# CONCLUSIÓN

- La comprensión de los selectores CSS y de sus diversos tipos ha demostrado ser fundamental para aplicar estilos de forma precisa y eficiente sobre los elementos HTML. Cada tipo de selector básico, combinador, pseudo-clase o pseudo-elemento ofrece diferentes niveles de especificidad, permitiendo construir estructuras de estilo organizadas, escalables y adaptadas a las necesidades de cualquier proyecto web.
- El análisis de la especificidad permitió entender su papel crucial en la resolución de conflictos de estilo en CSS. La correcta interpretación y aplicación de las reglas de especificidad asegura que los estilos se apliquen de manera coherente, evitando ambigüedades y reduciendo la dependencia de prácticas poco recomendables.
- La adecuada utilización de las propiedades de texto, fuentes, fondos y colores refuerza la estética, la legibilidad y la comunicación visual en el diseño web. El manejo correcto de estos aspectos no solo embellece las interfaces, sino que también promueve la accesibilidad y mejora la experiencia del usuario en diversas plataformas.
- El entendimiento del Modelo de Caja (Box Model) ha sido esencial para abordar el diseño y distribución de elementos en la web. El conocimiento de cómo interactúan el contenido, el padding, el borde y el margen permite controlar el espacio y las relaciones entre elementos, asegurando composiciones visuales ordenadas y armónicas.
- La diferenciación entre unidades absolutas y relativas en CSS ha evidenciado su importancia en la creación de diseños tanto precisos como flexibles. Mientras las unidades absolutas ofrecen medidas fijas para casos específicos, las relativas permiten una mejor adaptación a diferentes dispositivos y resoluciones, favoreciendo la creación de sitios web responsivos.

# CONCLUSIÓN

- El estudio de las técnicas de posicionamiento en CSS ha permitido comprender las distintas maneras en que los elementos pueden ubicarse dentro del flujo del documento. Cada valor aporta soluciones específicas para escenarios particulares, contribuyendo a un diseño dinámico, organizado y coherente con las necesidades funcionales del proyecto web.
- La implementación de los sistemas de diseño Flexbox y Grid ha evidenciado su eficacia para construir distribuciones responsivas y flexibles. Estos métodos facilitan el diseño de interfaces adaptativas que responden adecuadamente a diferentes tamaños de pantalla, mejorando la experiencia del usuario y optimizando el rendimiento del desarrollo.
- La utilización de pseudo-clases y pseudo-elementos en CSS ha permitido enriquecer las posibilidades de estilización dinámica sin necesidad de alterar la estructura HTML. Estos recursos ofrecen alternativas eficientes para mejorar la interacción y la apariencia visual, aportando un valor añadido a la usabilidad y el atractivo de los sitios web.
- La incorporación de transiciones y animaciones básicas ha demostrado que es posible agregar dinamismo y fluidez a la navegación web, elevando la calidad estética y mejorando la interacción del usuario. Su uso adecuado contribuye a construir experiencias digitales más atractivas, modernas y profesionales.
- La profundización en el concepto de especificidad reafirma su relevancia como principio central en el manejo de estilos en CSS. Una correcta gestión de la especificidad es indispensable para mantener el orden en las hojas de estilo, asegurar el funcionamiento esperado del diseño y facilitar el mantenimiento a largo plazo del proyecto.

# REFERENCIAS

- Primeros pasos en CSS | MDN. (2025, March 27). MDN Web Docs. [https://developer.mozilla.org/es/docs/conflicting/Learn\\_web\\_development/Core/Styling\\_basics](https://developer.mozilla.org/es/docs/conflicting/Learn_web_development/Core/Styling_basics)
- W3Schools.com. (n.d.). <https://www.w3schools.com/css/>
- Coyier, C. (2024, August 12). CSS Flexbox Layout Guide | CSS-Tricks. CSS-Tricks. <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- CSS grid layout - CSS: Cascading Style Sheets | MDN. (2025, February 7). MDN Web Docs. [https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_grid\\_layout](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_grid_layout)

**GRACIAS**