

Diseño adaptable personalizado

Autor: Génesis Natalya Ushca Pizarro

Fecha: 06/11/2025

Índice

| | |
|--|---|
| 1. Unidades de medida (CSS)..... | 3 |
| 2. Modelo de cajas: margin, padding, border..... | 3 |
| 3. Pseudo-clases y pseudo-elementos..... | 4 |
| 4. Posicionamiento y comportamiento de contenedores..... | 4 |
| 5. Menú horizontal tradicional..... | 5 |
| 6. Propiedades de los puntos 5–12 del tema..... | 6 |

Aunque aquí se explican solo unos cuantos ejemplos, en toda la página se aplican las propiedades CSS que se piden en la práctica.

1. Unidades de medida (CSS)

El diseño utiliza varias unidades de medida relativas para lograr un diseño adaptable:

- **vw (Viewport Width) y vh (Viewport Height):**

```
.caja-float {  
    width: 16vw; /*ancho del 16% de la ventana del navegador*/  
    height: 10vh; /*alto del 10% del viewport*/  
    background-color: #5bc5b7;  
    float: left;  
    margin: 0.5em;  
}
```

```
.overflow {  
    width: 45vw; /*ancho del 45% del viewport*/  
    height: 13vh; /*alto del 13% del viewport*/  
    border: 4px ridge #2a2dce;  
    overflow: auto;  
    padding: 0.8em;  
}
```

- **% (Porcentaje):**

```
nav li {  
    margin-right: 10%; /*margen relativo al ancho del contenedor padre*/  
}
```

- **em:**

```
header {  
    background-color: #d7ccc8;  
    color: #6d4c41;  
    padding: 1em; /*relativo al tamaño de fuente del elemento*/  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    line-height: 2;  
}
```

Estas unidades permiten que el diseño se adapte a distintos tamaños de pantalla.

2. Modelo de cajas: margin, padding, border

La página usa:

- *box-sizing: border-box;*
Todas las medidas incluyen borde y relleno → facilita el control del ancho total.
- Padding:
padding: 1em en header, section, article → crea espacio interno entre contenido y borde.
- Margin:
margin: 2em en section, *margin: 2.5em* en article → separación externa entre bloques.
- Bordes:
Se usan múltiples estilos (solid, dotted, groove, ridge) para diferenciar visualmente los ejemplos:
 - *.relative { border: 8px dotted #23375e; }*
 - *.abs { border: 3px groove #5f885b; }*

3. Pseudo-clases y pseudo-elementos

- Pseudo-clase
Es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado. Por ejemplo :**hover**:

```
nav a:hover {  
    background-color: #a1887f;  
}
```

Cambia el fondo de los enlaces de la cabecera al pasar el cursor, mejorando la interacción del usuario.

- Pseudo-elemento
Se añaden a los selectores, pero en cambio, no describen un estado especial sino que, permiten añadir estilos a una parte concreta del documento. Por ejemplo :**after**:

```
.float::after { /* Clear */  
    content: "";  
    display: table;  
    clear: both;  
}
```

Se usa comoclearfix (se explica en el punto 6)

4. Posicionamiento y comportamiento de contenedores

- **Propiedad: display**

```
.a {
    display: none; /*Oculta completamente los elementos*/
}

.b {
    display: inline; /*Elementos en Línea (sin saltos, sin tamaño definido).*/
    border: 2px solid #23375e;
}

.c {
    display: block; /*Cada párrafo ocupa toda la Línea*/
    background-color: #dd9ee5;
    width: 5vw;
    margin: 2%;
}

.d {
    /*Combina block e inline, se muestran en la misma Línea y
    se acepta width, height y márgenes verticales*/
    display: inline-block;
}
```

- **Propiedad: position**

static: valor por defecto (.static).

relative: permite desplazar elementos sin sacarlos del flujo normal (.cuadro2, .cuadro3 usan *left* para moverse).

absolute: .abs se posiciona respecto a su ancestro relativo (.rel).

fixed: .texto2 queda fijo en la pantalla (no se mueve al hacer scroll).

| | | |
|--|--|--|
| <pre>.static { position: static;</pre> | <pre>.relative { position: relative;</pre> | <pre>.rel { position: relative; line-height: 1.9; margin-top: 4%; border: 10px groove #abdd73; } .abs { position: absolute;</pre> |
| <pre>.cuadro2 { position: relative; left: 50%;</pre> | <pre>.texto2 { position: fixed; top: 20%; left: 70%;</pre> | |
| <pre>.cuadro3 { position: relative; left: 20%;</pre> | | |

- **Propiedad: float**

float: left; → saca los elementos del flujo normal y los alinea a la izquierda.

```
.caja-float {
    width: 16vw;
    height: 10vh;
    background-color: #5bc5b7;
    float: left;
    margin: 0.5em;
```

- **Propiedad: overflow**

`overflow: auto;` → añade barras de scroll si el contenido excede el contenedor.

```
.overflow {
    width: 45vw;
    height: 13vh;
    border: 4px ridge #2a2dce;
    overflow: auto;
    padding: 0.8em;
}
```

Los `.overflow-item` están en `inline-block` dentro de un contenedor estrecho → obliga al scroll horizontal.

5. Menú horizontal tradicional

```
nav ul {
    list-style-type: none; /*Elimina las viñetas*/
    display: flex; /*Alinea horizontalmente las listas*/
}

nav li {
    margin-right: 10%
}
```

Es mejor usar Flexbox porque así tenemos más control sobre los elementos.

6. Propiedades de los puntos 5–12 del tema

- Solución de problemas comunes:

- El contenedor `.float` colapsa → lo solucionamos con `clearfix`.

```
.float::after {
    content: "";
    display: table;
    clear: both;
}
```

- Espaciado no deseado con `inline-block` → corregido con `margin-right: -4px`.

```
.d {
    display: inline-block;
    width: 100px;
    height: 50px;
    background-color: #4db6ac;
    margin-right: -4px;
}
```