



# **Clase 4 -** Maquetación con CSS

# LAYOUT & MAQUETACIÓN CON CSS





# POSITION & FLOAT

**Posicionamiento (position):** Con propiedades como `position: absolute`, `relative`, `fixed` o `sticky`, puedes controlar exactamente dónde colocar un elemento en la página.

**Flotación (float):** Antes se usaba mucho para hacer que los elementos "floten" a la izquierda o derecha, como en diseños de columnas, aunque hoy en día se usa menos gracias a opciones más modernas.



# FLEXBOX

Flexbox nos permite alinear elementos en un eje (horizontal o vertical). Es ideal para crear menús, filas de tarjetas, o layouts simples.

Propiedades clave:

- `display: flex;`: activa flexbox
- `flex-direction`: define la dirección (row, column)
- `justify-content`: alinea en el eje principal
- `align-items`: alinea en el eje cruzado
- `flex-grow`: cuánto puede crecer un ítem
- `flex-shrink`: cuánto puede achicarse
- `flex-basis`: tamaño base antes de crecer o encogerse
- `order`: cambia el orden visual de los ítems
- `align-self`: alinea individualmente un ítem



# GRID

Grid permite trabajar en 2 dimensiones: filas y columnas al mismo tiempo. Ideal para maquetar estructuras más complejas: dashboards, secciones completas, landing pages.

Propiedades clave:

- `display: grid;`
- `grid-template-columns`: define columnas
- `grid-template-rows`: define filas
- `gap`: separaciones internas
- `grid-area`: nombrar zonas específicas
- `place-items`: alinea horizontal y verticalmente



# TRANSICIONES

transition: propiedad duracion efecto;

```
button {  
  transition: background 0.3s ease;  
}
```



# ANIMACIONES

on **@keyframes** Definimos etapas de una animación y la aplicamos con `animation`

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



# MEDIA QUERIES

## Diseño responsive con Media Queries

Permiten aplicar estilos solo si se cumple una condición de tamaño o dispositivo

```
@media (max-width: 768px) {  
  .container {  
    flex-direction: column;  
  }  
  
  aside {  
    width: 100%;  
    border-right: none;  
    border-bottom: 1px solid #ddd;  
  }  
}
```