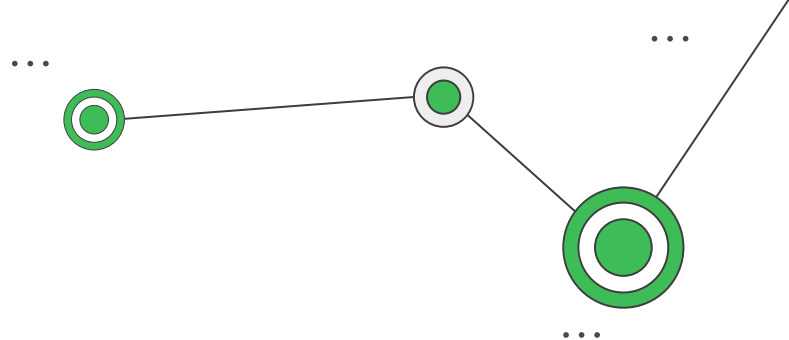
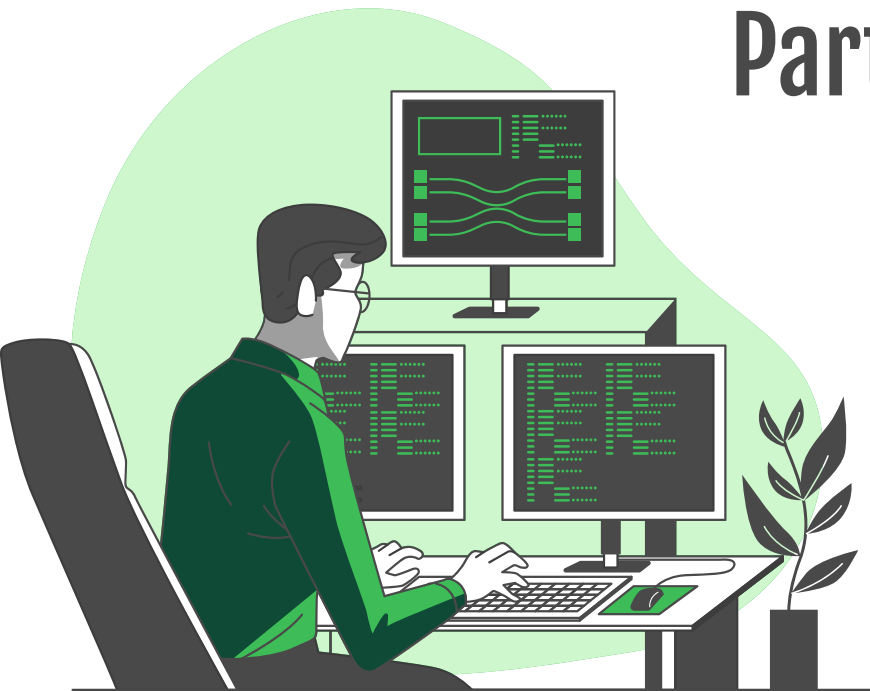




UNIDAD 01



Parte 2: Diseño de interfaces con Tailwind CSS



2º DAM - DEINT

María Millán Gamero

maria.millan@iesmartinezm.es



RESULTADO DE APRENDIZAJE

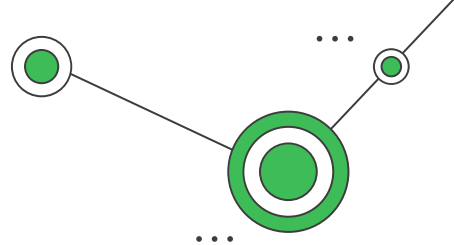
RA4. Diseña interfaces gráficas identificando y aplicando criterios de usabilidad y accesibilidad.

a	Se han identificado los principales estándares de usabilidad y accesibilidad.
b	Se ha valorado la importancia del uso de estándares para la creación de interfaces.
c	Se han creado diferentes tipos de menús cuya estructura y contenido siguen los estándares establecidos.
d	Se han distribuido las acciones en menús, barras de herramientas, botones de comando, entre otros, siguiendo un criterio coherente.
e	Se han distribuido adecuadamente los controles en la interfaz de usuario.
f	Se ha utilizado el tipo de control más apropiado en cada caso.
g	Se ha diseñado el aspecto de la interfaz de usuario (colores y fuentes entre otros
h	Se ha verificado que los mensajes generados por la aplicación son adecuados en extensión y claridad.
i	Se han realizado pruebas para evaluar la usabilidad y accesibilidad de la aplicación.

ÍNDICE

1. ¿Qué es Tailwind CSS?
2. ¿Qué es Vite?
3. Creación de un proyecto JS con Tailwind CSS
4. Colores
5. Medidas
6. Breakpoint de tamaño de pantalla
7. Flexbox
8. Grid
9. Estados
10. DarkMode
11. Primera app con Tailwind
12. Repaso JavaScript
13. Bases de TypeScript
14. Primer proyecto con TS

1. ¿QUÉ ES TAILWIND CSS?



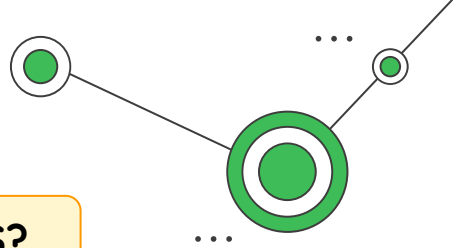
Tailwind = es un framework CSS que da prioridad a la utilidad sobre el propio estilo, pero además a diferencia de otros frameworks CSS como Bootstrap o Bulma, Tailwind no provee una serie de componentes predefinidos.

En su lugar, este framework opera en un nivel inferior y te proporciona un conjunto de clases de ayuda para estructura y estilado, de forma que, usando dichas clases, puedas crear rápidamente diseños personalizados con facilidad.



1. ¿QUÉ ES TAILWIND CSS?

¿QUÉ LO HACE INTERESANTE RESPECTO A OTROS?



- **La solución que propone es proporcionar una amplia variedad de clases CSS que tienen cada una un enfoque propio.**
- En vez de crear una clase general con numerosos atributos CSS como hace Bootstrap, Tailwind aplica diferentes clases “más pequeñas” que solo hace referencia a una propiedad CSS.
- De esta forma, cada pequeña clase utiliza una serie de modificadores sobre nuestro elemento HTML que en conjunto nos permite dar estilo a cada elemento de forma que no sea necesario ni siquiera un gran conocimiento en CSS.
- En el caso de páginas rápidas o de propósito general que deben verse bien en múltiples dispositivos el uso de Tailwind proporciona un excelente equilibrio entre compatibilidad entre navegadores, estilado, visibilidad y facilidad de uso.

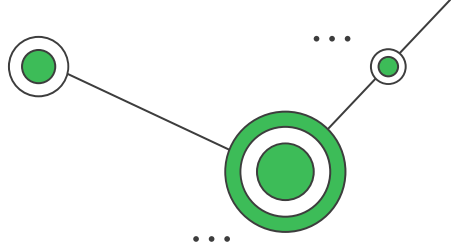
2. ¿QUÉ ES VITE?



- Vite, que en francés significa "rápido", es una herramienta de construcción y un servidor de desarrollo local de código abierto que está revolucionando la forma en que los desarrolladores crean aplicaciones web modernas.

Vite es una herramienta de desarrollo frontend moderna que aborda los puntos débiles de las herramientas de construcción tradicionales, ofreciendo una experiencia de desarrollo más rápida, eficiente y agradable. Su enfoque en la velocidad, junto con su soporte para tecnologías modernas y su facilidad de uso, lo ha convertido en una opción cada vez más popular entre los desarrolladores web.

3. CREACIÓN DEL PRIMER PROYECTO DE TAILWIND USANDO VITE



Vamos a trabajar continuamente con la web de Tailwind CSS:

- <https://tailwindcss.com/>
- <https://tailwindcss-es.com/>

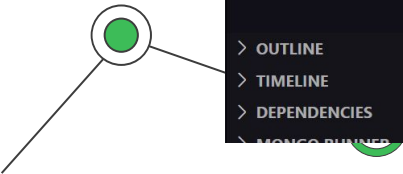
Dentro de tu carpeta del repositorio, añade una nueva carpeta que se llame:

UD01_Fundamentos

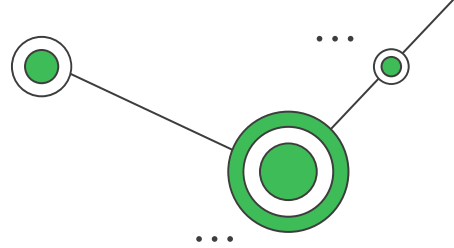
Abrimos la carpeta en VSCode y en la terminal seguiremos los pasos de la instalación que se indican en la documentación oficial:

<https://tailwindcss.com/docs/installation/using-vite>

Cuando indiquemos el tipo de proyecto será: Vanilla > JavaScript



3. CREACIÓN DEL PRIMER PROYECTO DE TAILWIND USANDO VITE



```
UD01_FUNDAMENTOS_TAILWIND
> my-first-app

PROBLEMS OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS C:\Users\milla\OneDrive - UNIR\Escritorio\DEINT_2526\UD01_Fundamentos_Tailwind> npm create vite@late
st my-first-app
> create-vite my-first-app

◇ Select a framework:
  Vanilla

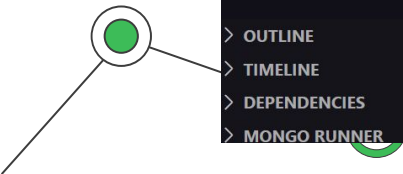
◇ Select a variant:
  JavaScript

◇ Use rolldown-vite (Experimental)?:
  No

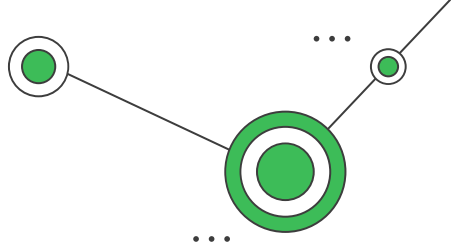
◇ Install with npm and start now?
  Yes

◇ Scaffolding project in C:\Users\milla\OneDrive - UNIR\Escritorio\DEINT_2526\UD01_Fundamentos_Tailwin
d\my-first-app...

◇ Installing dependencies with npm...
:|
```



3. CREACIÓN DEL PRIMER PROYECTO DE TAILWIND USANDO VITE



```
PS C:\Users\milla\OneDrive - UNIR\Escritorio\DEINT_2526\UD01_Fundamentos_Tailwind> npm create vite@latest my-first-app
```

→ Local: <http://localhost:5173/>

→ Network: use `--host` to expose

→ press `h + enter` to show help

```
● PS C:\Users\milla\OneDrive - UNIR\Escritorio\DEINT_2526\UD01_Fundamentos_Tailwind> cd my-first-app
```

```
● PS C:\Users\milla\OneDrive - UNIR\Escritorio\DEINT_2526\UD01_Fundamentos_Tailwind\my-first-app> npm install tailwindcss @tailwindcss/vite
```

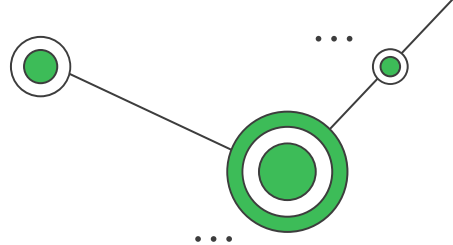
added 90 packages, and audited 105 packages in 4s

17 packages are looking for funding
run ``npm fund`` for details

found 0 vulnerabilities

```
● PS C:\Users\milla\OneDrive - UNIR\Escritorio\DEINT_2526\UD01_Fundamentos_Tailwind\my-first-app> █
```

3. CREACIÓN DEL PRIMER PROYECTO DE TAILWIND USANDO VITE



1

```
my-first-app > vite.config.ts > default
1 import { defineConfig } from 'vite'
2 import tailwindcss from '@tailwindcss/vite'
3
4 export default defineConfig({
5   plugins: [
6     tailwindcss(),
7   ],
8 })
```

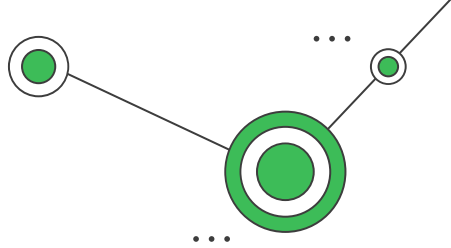
2

```
my-first-app > src > style.css > ...
1 @import "tailwindcss";
2
```

3

```
my-first-app > index.html > html > body.flex.items-center.justify-
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <link rel="icon" type="image/svg+xml" href="/vite.svg">
6     <meta name="viewport" content="width=device-width, initial-scale=1">
7     <title>Home</title>
8     <link rel="stylesheet" href="src/style.css">
9   </head>
```

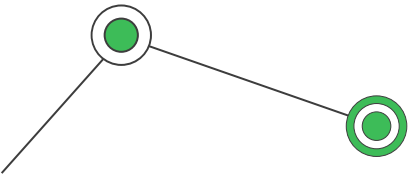
3. CREACIÓN DEL PRIMER PROYECTO DE TAILWIND USANDO VITE



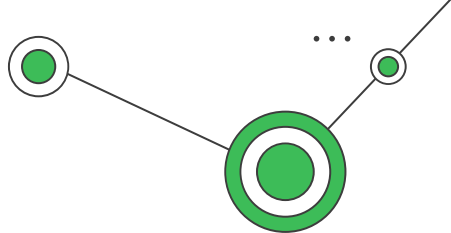
Vamos a limpiar el proyecto que nos crea por defecto, teniendo en cuenta que trabajaremos de la siguiente forma:



- ★ **Index.html** → tendremos un título con diferentes enlaces a las páginas donde iremos desarrollando las partes de fundamentos teóricos.
- ★ En **cada nuevo html**, añadiremos los ejemplos de la teoría que se va explicando.



3. CREACIÓN DEL PRIMER PROYECTO DE TAILWIND USANDO VITE



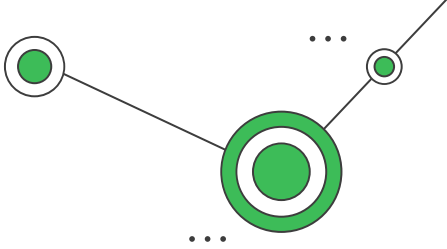
```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Home</title>
    <link rel="stylesheet" href="src/style.css">
  </head>
  <body>
    <div id="app">
      <h1>Bienvenidos a Desarrollo de interfaces</ h1>
      <a href="colores.html">Colores en Tailwind</ a>
    </div>
  </body>
</html>
```

En index.html

4. COLORES EN TAILWIND CSS

Tailwind CSS incluye una **paleta de colores predefinida** que puedes aplicar directamente a elementos HTML mediante clases utilitarias.

- ★ <https://tailwindcss.com/docs/colors>
- ★ Tailwind organiza los colores en familias y tonalidades.



	50	100	200	300	400	500	600	700	800	900	950
Rojo											
Naranja											
Ámbar											

Estructura básica de clases de color

bg-color-shade → fondo
text-color-shade → texto
border-color-shade → borde

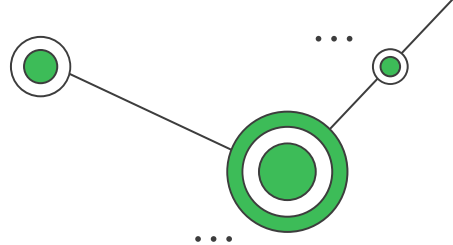
```
<div class="bg-blue-500 text-white">Texto con fondo azul</div>  
<button class="bg-green-600">Botón</button>  
<p class="text-red-400">Texto en rojo claro</p>
```

Texto con fondo azul

Botón

Texto en rojo claro

4. COLORES EN TAILWIND CSS



Podemos personalizar colores:

- Usando **@theme** en nuestro CSS añadiendo variables.

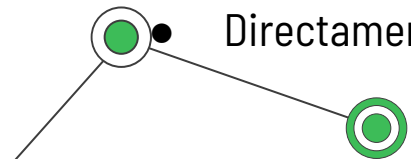
```
CSS

@import "tailwindcss";

@theme {
  --color-midnight: #121063;
  --color-tahiti: #3ab7bf;
  --color-bermuda: #78dcca;
}
```

Ahora utilidades como *bg-midnight*, *text-tahiti* y *fill-bermuda* estarán disponibles en tu proyecto además de los colores predeterminados.

- Directamente usando [] en línea, indicando el color directamente.



```
<h2 class="■ bg-orange-600 ■ text-[#ebebeb]">C
```

4. COLORES EN TAILWIND CSS



En colores.html

Colores en Tailwind CSS

1

```
<body>  
  <h2 class="bg-orange-600 text-white">Colores en Tailwind CSS</h2>  
</body>
```

2

```
<body>  
  <h2 class="bg-orange-600 text-[#ebebeb]">Colores en Tailwind CSS</h2>  
  <h1 class="bg-midnight text-tahiti">Podemos personalizar colores</h1>  
</body>
```

Colores en Tailwind CSS

Podemos personalizar colores

4. COLORES EN TAILWIND CSS



En colores.html



Vamos a crear un botón y vamos a personalizando, incluyendo su estado *hover*.

Documentación:

- <https://tailwindcss-es.com/docs/padding>
- <https://tailwindcss-es.com/docs/border-radius>
- <https://tailwindcss-es.com/docs/width>
- <https://tailwindcss-es.com/docs/hover-focus-and-other-states>

Botón

Botón

```
<button class="bg-primary text-white p-2 rounded-lg w-24 mx-auto block my-8  
hover:bg-primary-dark cursor-pointer">Botón</button>
```

```
<button class="p-2 rounded-lg w-auto mx-auto block my-8 bg-linear-to-r from-primary  
to-white">Ejemplo gradiente</button>
```

Ejemplo gradiente

4. COLORES EN TAILWIND CSS



En colores.html



Vamos a crear un botón y vamos a personalizando, incluyendo su estado *hover*.

Documentación:

- <https://tailwindcss-es.com/docs/background-image#estableciendo-paradas-de-color-del-degradado>
- <https://tailwindcss-es.com/docs/opacity>
- <https://tailwindcss-es.com/docs/border-width>

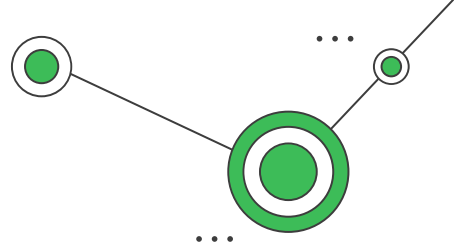
Ejemplo borde

```
<button class="p-2 rounded-lg w-auto mx-auto block my-8 bg-linear-to-bl from-primary to-white border-primary-dark border-2">Ejemplo borde</button>
```

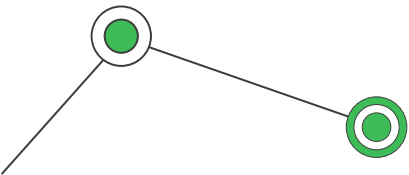
```
<button class="p-2 rounded-lg w-auto mx-auto block my-8 bg-blue-700/50 border-blue-900 border-2">Ejemplo opacidad</button>
```

Ejemplo opacidad

4. COLORES EN TAILWIND CSS



Pon en práctica lo visto y edita el `index.html` para darle un estilo personalizado a la home de nuestra primera aplicación.



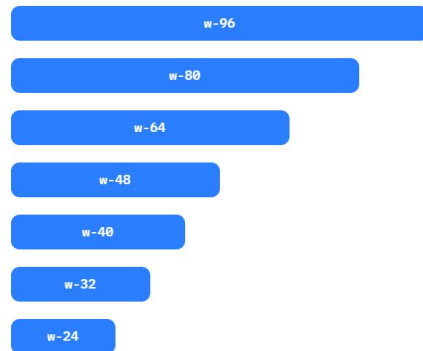
5. MEDIDAS EN TAILWIND CSS: Width

Tailwind ofrece clases utilitarias para **anchos, altos, márgenes, paddings, etc.**, usando valores predefinidos o personalizados.

★ <https://tailwindcss-es.com/docs/width>

Tailwind usa un **sistema de escala en múltiplos de 0.25rem (4px)**.

Tailwind permite usar valores entre corchetes para medidas personalizadas exactas.



```
<div class="bg-red-400 w-200">Medidas en Tailwind CSS</div>  
<div class="bg-amber-400 w-[170px]">Ajustando el ancho a nuestra medida</div>
```

Una barra horizontal compuesta de dos segmentos: uno rojo y uno amarillo. El segmento rojo contiene el texto "Medidas en Tailwind CSS" y el segmento amarillo contiene el texto "Ajustando el ancho a nuestra medida".

Medidas en Tailwind CSS
Ajustando el ancho a
nuestra medida

5. MEDIDAS EN TAILWIND CSS: Width

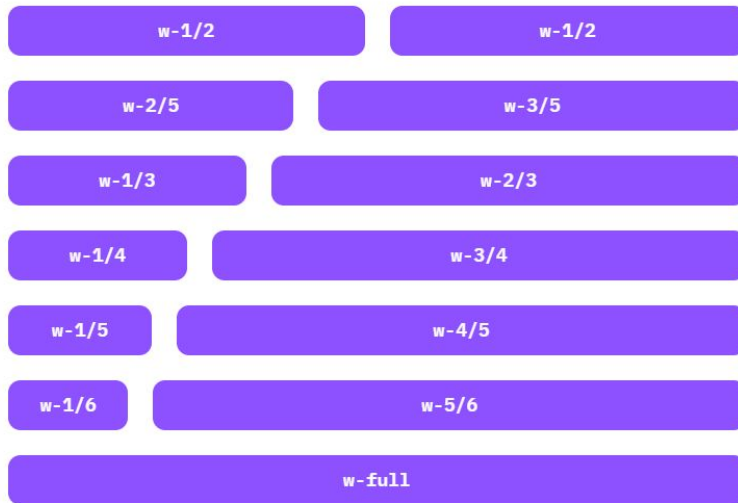
- ★ Podemos usar `w-full` o utilidades `w-<fraction>` como `w-1/2` y `w-2/5` para darle a un elemento un ancho basado en porcentaje:

Personalizar el tamaño:

Las utilidades `w-<number>` y `size-<number>` son impulsadas por la variable de tema

- **`--spacing`**, la cual puede ser personalizada en tu propio tema:

```
@theme {  
  --spacing: 1px;  
}
```



5. MEDIDAS EN TAILWIND CSS: Width



En medidas.html

```
medidas.html U style.css U X
my-first-app > src > style.css > .root
1  @import "tailwindcss";
2
3  @theme {
4    --color-midnight: #121063;
5    --color-tahiti: #3ab7bf;
6    --color-bermuda: #78dcca;
7    --color-primary: #ff0f64;
8    --color-primary-dark: #860834;
9    /* --spacing: 170px; */
10 }
11
12 :root{
13   --my-width: 200px;
14 }
15 /*
```

```
<div class="bg-red-400 w-200">Medidas en Tailwind CSS</div>
<div class="bg-amber-400 w-[170px]">Ajustando el ancho a nuestra
medida</div>
<div class="bg-blue-300 w(--my-width)">Usando la medida de mi
variable</div>
```

Medidas en Tailwind CSS

Ajustando el ancho a
nuestra medida

Usando la medida de mi
variable

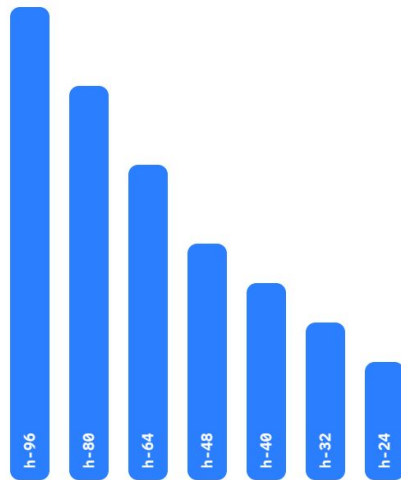
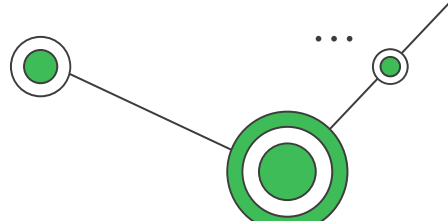
5. MEDIDAS EN TAILWIND CSS: Height

Tailwind ofrece clases utilitarias para **anchos, altos, márgenes, paddings, etc.**, usando valores predefinidos o personalizados.

★ <https://tailwindcss-es.com/docs/height>

Tailwind usa un **sistema de escala en múltiplos de 0.25rem (4px)**.

Tailwind permite usar valores entre corchetes para medidas personalizadas exactas.



Ajustando el alto de nuestro div

```
<div class="bg-lime-200 w-fit h-(--my-width)">Ajustando el alto de  
nuestro div</div>
```



H-full → se aplica siempre que el elemento tenga un padre y no sea el body

6. BREAKPOINT: DISEÑO RESPONSIVO

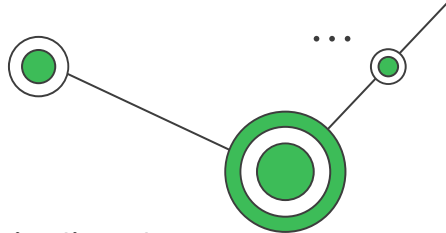
Cada clase de utilidad en Tailwind se puede aplicar condicionalmente en diferentes breakpoints, lo que hace que sea pan comido construir interfaces responsive complejas sin salir de tu HTML.

- ★ <https://tailwindcss-es.com/docs/responsive-design#using-custom-breakpoints>
- ★ Asegúrate de haber agregado la etiqueta meta viewport al <head> de tu documento.

Hay cinco breakpoints por defecto, inspirados en resoluciones de dispositivos comunes:

Prefijo del breakpoint	Ancho mínimo	CSS
`sm`	40rem (640px)	`@media (width >= 40rem) { ... }`
`md`	48rem (768px)	`@media (width >= 48rem) { ... }`
`lg`	64rem (1024px)	`@media (width >= 64rem) { ... }`
`xl`	80rem (1280px)	`@media (width >= 80rem) { ... }`
`2xl`	96rem (1536px)	`@media (width >= 96rem) { ... }`

6. BREAKPOINT: DISEÑO RESPONSIVO



Esto funciona para cada clase de utilidad en el framework, lo que significa que puedes cambiar literalmente cualquier cosa en un breakpoint dado, incluso cosas como el espaciado entre letras o los estilos del cursor.

Donde este enfoque sorprende más a la gente es que para diseñar algo para móviles, necesitas usar la versión **sin prefijo de una utilidad**, no la versión con prefijo sm:.

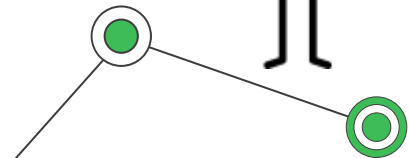
No pienses en sm: como "en pantallas pequeñas", piensa en ello como "en el breakpoint pequeño".



No uses ``sm:`` para apuntar a dispositivos móviles



Usa utilidades sin prefijo para apuntar a móviles y anúlalos en breakpoints más grandes



6. BREAKPOINT: DISEÑO RESPONSIVO



En breakpoints.html

Recuerda añadir el acceso a este html desde index.html

```
<body>
<div class="border-2 border-black p-6 w-80 mx-auto mt-10">
  <h2 class="text-xl font-bold">Diseño Responsive</h2>
  <p class="mt-4">Usar variantes de utilidad responsive para construir
interfaces de usuario adaptables.</p>
</div>
</body>
```

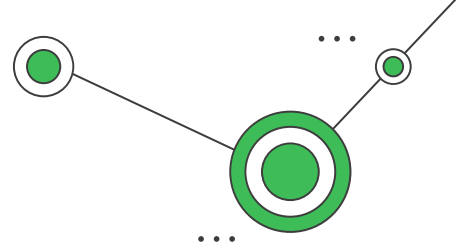
Diseño Responsive

Usar variantes de utilidad responsive para construir interfaces de usuario adaptables.

6. BREAKPOINT: DISEÑO RESPONSIVO



En breakpoints.html



```
<body class="bg-pink-300 sm:bg-lime-200 md:bg-blue-200 lg:bg-orange-300">
  <div class="border-2 border-black p-6 w-80 mx-auto mt-10">
    <h2 class="text-xl font-bold">Diseño Responsive</h2>
    <p class="mt-4">Usar variantes de utilidad responsive para construir interfaces de
    usuario adaptables.</p>
  </div>
</body>
```

Diseño Responsive

Usar variantes de utilidad responsive para construir interfaces de usuario adaptables.



Diseño Responsive

Usar variantes de utilidad responsive para construir interfaces de usuario adaptables.



Diseño Responsive

Usar variantes de utilidad responsive para construir interfaces de usuario adaptables.



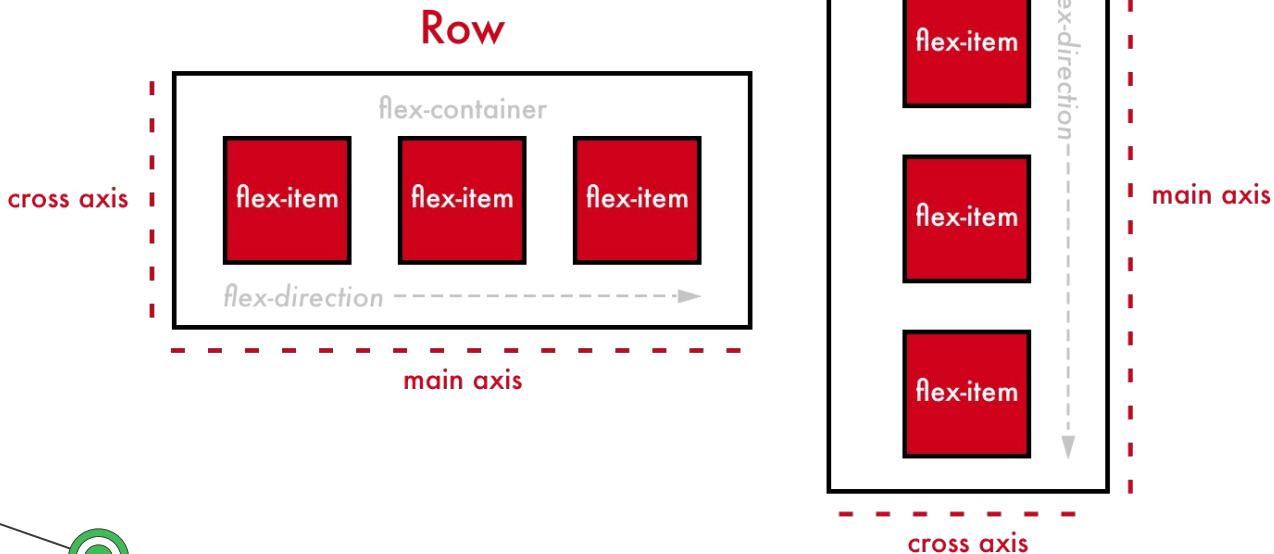
Diseño Responsive

Usar variantes de utilidad responsive para construir interfaces de usuario adaptables.

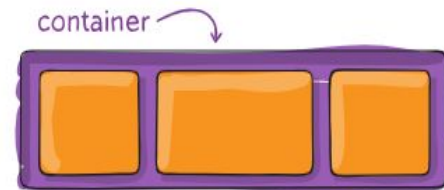
7. FLEXBOX

Chuleta Flexbox CSS

★ <https://tailwindcss-es.com/docs/flex>



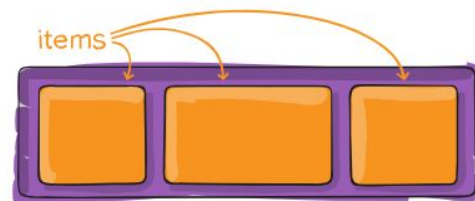
7. FLEXBOX: Propiedades del flex-container



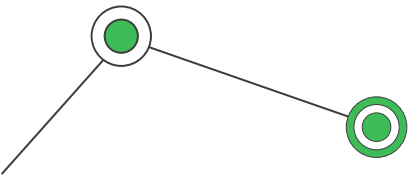
Propiedad CSS	Qué hace	Tailwind CSS
display: flex	Activa Flexbox	flex
flex-direction	Dirección de los ítems (fila/columna)	flex-row, flex-col, etc.
flex-wrap	Permite que los ítems salten de línea	flex-wrap, flex-nowrap
justify-content	Alinea ítems en eje horizontal	justify-start, center, etc.
align-items	Alinea ítems en eje vertical	items-start, center, etc.
align-content	Alinea múltiples líneas verticalmente	content-start, center, etc.
gap, row-gap, column-gap	Espaciado entre ítems	gap-*, gap-x-*, gap-y-*



7. FLEXBOX: Propiedades del flex-items



Propiedad CSS	Qué hace	Tailwind CSS
flex-grow	El ítem crece si hay espacio	grow, grow-0
flex-shrink	El ítem se encoge si no cabe	shrink, shrink-0
flex-basis	Tamaño inicial del ítem	basis-*, basis-[px]
flex	Agrupar grow, shrink y basis	flex-1, flex-auto, etc.
align-self	Alineación vertical individual	self-start, center, etc.
order	Cambia el orden visual del ítem	order-1, first, last



7. FLEXBOX



<https://tailwindcss-es.com/docs/adding-custom-styles#adding-component-classes>

Usa la **capa components** para cualquier clase más complicada que quieras agregar a tu proyecto y que aún quieras poder sobrescribir con clases de utilidad.

Tradicionalmente, estas serían clases como *card*, *btn*, *badge*, ese tipo de elementos.

```
@layer components {  
  .card {  
    background-color: var(--color-white);  
    border-radius: var(--rounded-lg);  
    padding: var(--spacing-6);  
    box-shadow: var(--shadow-xl);  
  }  
}
```

```
<div class="card rounded-none">  
  <!-- ... -->  
</div>
```

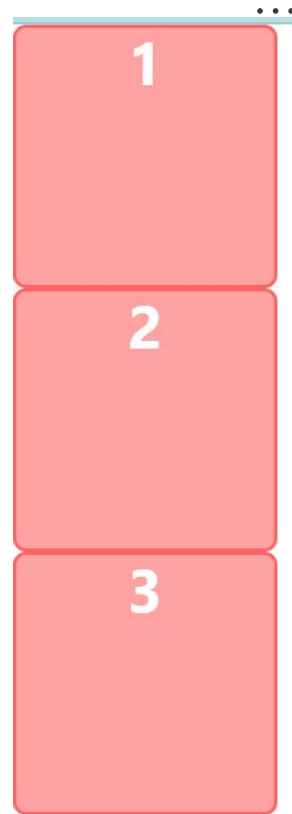
7. FLEXBOX



En flexbox.html

```
<body>
  <div>
    <div class="card">1</div>
    <div class="card">2</div>
    <div class="card">3</div>
  </div>
</body>
```

```
@layer components{
  .card{
    @apply bg-red-300 w-40 h-40 rounded-lg text-white
    flex justify-center
    font-bold border-2 border-red-400 text-4xl;
  }
}
```



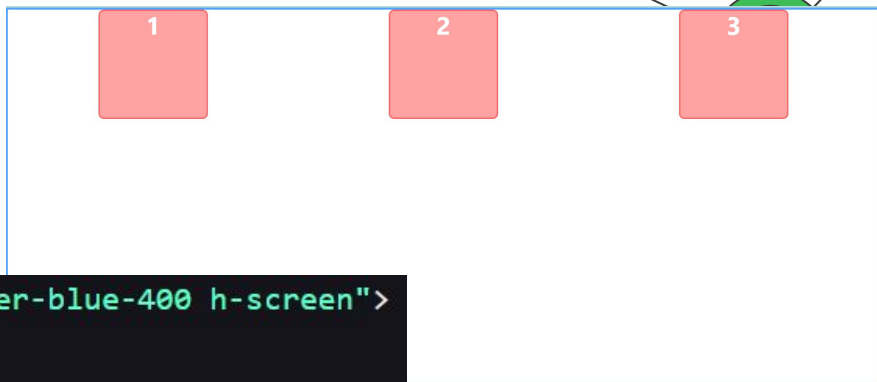
7. FLEXBOX



En flexbox.html

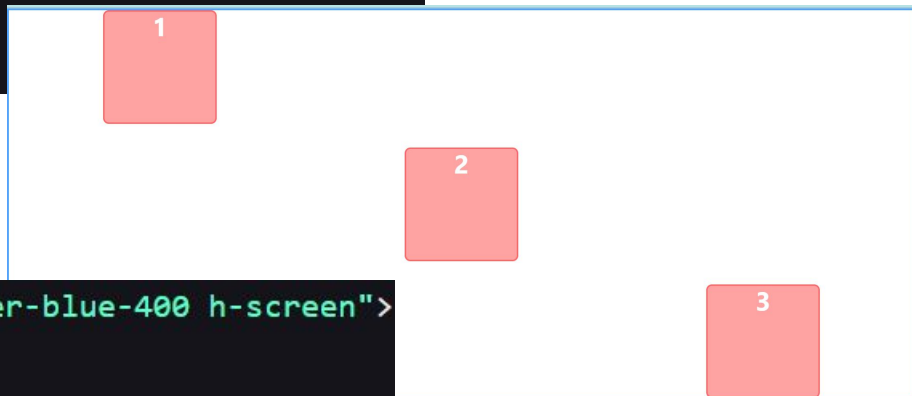
1

```
<div class="flex justify-around border-3 border-blue-400 h-screen">  
  <div class="card">1</div>  
  <div class="card">2</div>  
  <div class="card">3</div>  
</div>
```



2

```
<div class="flex justify-around border-3 border-blue-400 h-screen">  
  <div class="card">1</div>  
  <div class="card self-center">2</div>  
  <div class="card self-end">3</div>  
</div>
```



7. FLEXBOX



En flexbox.html

```
<div class="flex flex-wrap gap-4 mt-6">  
  <div class="card grow">1</div>  
  <div class="card grow">2</div>  
  <div class="card grow">3</div>  
  <div class="card grow">4</div>  
  <div class="card grow">5</div>  
  <div class="card grow">6</div>  
</div>
```

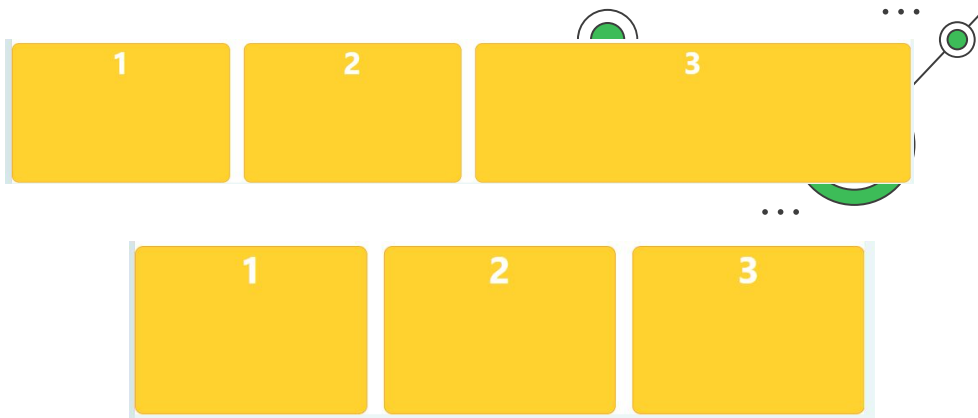
```
<div class="flex flex-wrap gap-4 mt-6">  
  <div class="card grow shrink-0">1</div>  
  <div class="card grow shrink-0">2</div>  
  <div class="card grow shrink-0">3</div>  
  <div class="card grow shrink-0">4</div>  
  <div class="card grow shrink-0">5</div>  
  <div class="card grow shrink-0">6</div>  
</div>
```



7. FLEXBOX



En flexbox.html



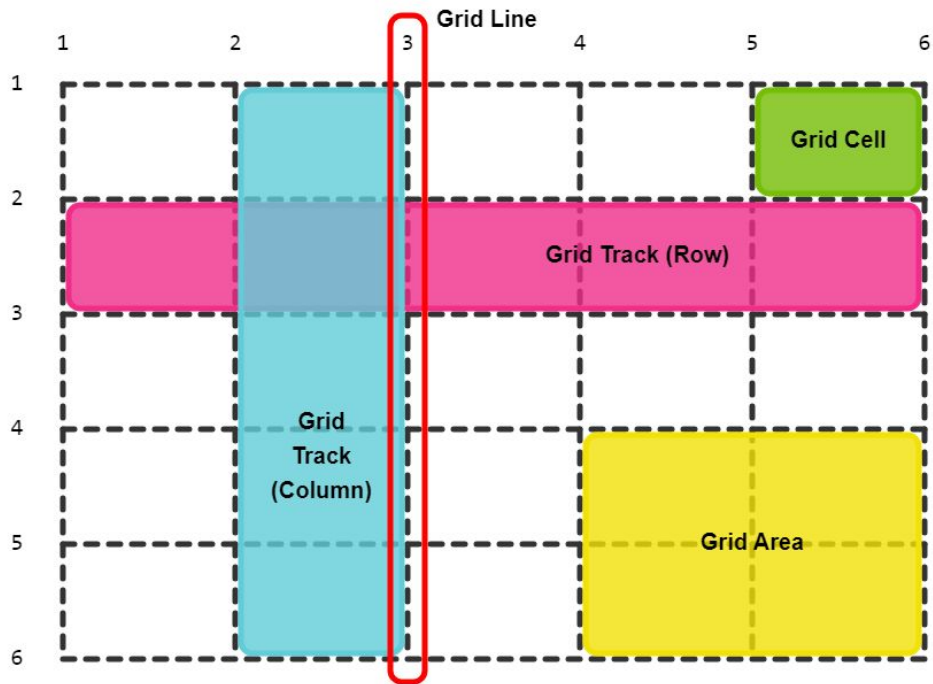
```
<div class="flex mt-6 gap-4">
  <div class="card bg-amber-300 border-1 border-amber-500 basis-1/4">1</div>
  <div class="card bg-amber-300 border-1 border-amber-500 basis-1/4">2</div>
  <div class="card bg-amber-300 border-1 border-amber-500 basis-1/2">3</div>
</div>
```

```
<div class="flex mt-6 gap-4">
  <div class="card bg-amber-300 border-1 border-amber-500 basis-1/4 sm:basis-1/3">1</div>
  <div class="card bg-amber-300 border-1 border-amber-500 basis-1/4 sm:basis-1/3">2</div>
  <div class="card bg-amber-300 border-1 border-amber-500 basis-1/2 sm:basis-1/3">3</div>
</div>
```

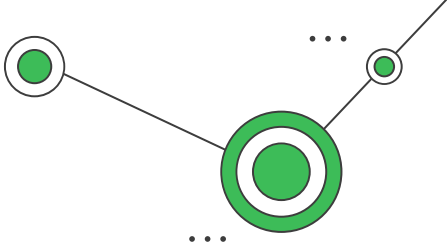
8. GRID

Chuleta Grid CSS

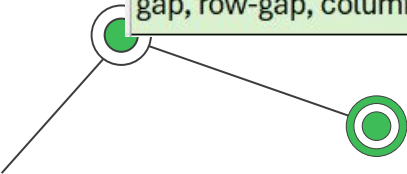
★ <https://tailwindcss-es.com/docs/display#grid>



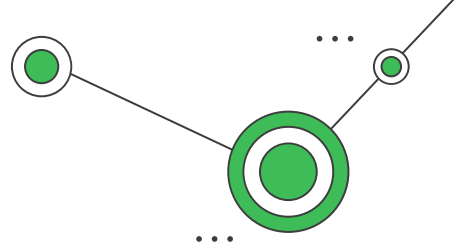
8. GRID: Propiedades del grid-container



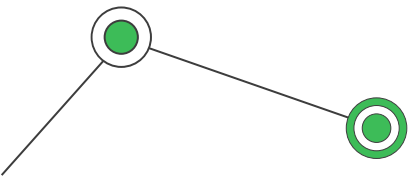
Propiedad CSS	Qué hace	Tailwind CSS
display: grid	Activa Grid	grid
grid-template-columns	Número y tamaño de columnas	grid-cols-*, grid-cols-[...]
grid-template-rows	Número y tamaño de filas	grid-rows-*, grid-rows-[...]
grid-auto-flow	Dirección automática de los ítems	grid-flow-row, grid-flow-col
justify-items	Alinea contenido en celdas (horizontal)	justify-items-start, center, etc.
align-items	Alinea contenido en celdas (vertical)	items-start, center, etc.
justify-content	Alinea todo el grid (horizontal)	justify-start, center, etc.
align-content	Alinea todo el grid (vertical)	content-start, center, etc.
gap, row-gap, column-gap	Espaciado entre filas y columnas	gap-*, gap-x-*, gap-y-*



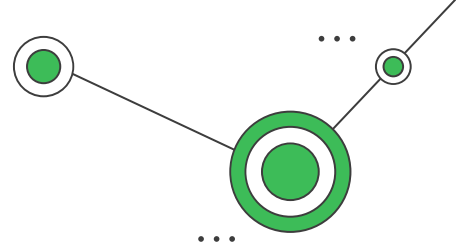
8. GRID: Propiedades del grid-cell



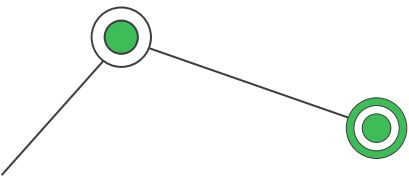
Propiedad CSS	Qué hace	Tailwind CSS
grid-column	Ocupa varias columnas o define inicio/fin	col-span-*, col-start-*
grid-row	Ocupa varias filas o define inicio/fin	row-span-*, row-start-*
justify-self	Alineación horizontal individual	justify-self-start, center, etc.
align-self	Alineación vertical individual	self-start, center, etc.



8. GRID: Propiedades del grid-cell



Propiedad CSS	Qué hace	Tailwind CSS
grid-column	Ocupa varias columnas o define inicio/fin	col-span-*, col-start-*
grid-row	Ocupa varias filas o define inicio/fin	row-span-*, row-start-*
justify-self	Alineación horizontal individual	justify-self-start, center, etc.
align-self	Alineación vertical individual	self-start, center, etc.



8. GRID

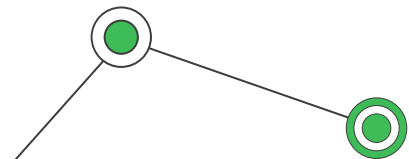
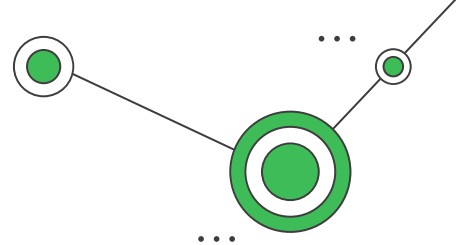


En grid.html

```
.box{  
  @apply bg-blue-300 rounded-lg text-white  
  grid place-content-center  
  font-bold border-2 border-blue-400 text-4xl;  
}
```

```
<body class="p-4">  
  <div>  
    <div class="box">1</div>  
    <div class="box">2</div>  
    <div class="box">3</div>  
    <div class="box">4</div>  
    <div class="box">5</div>  
    <div class="box">6</div>  
    <div class="box">7</div>  
    <div class="box">8</div>  
    <div class="box">9</div>  
  </div>  
</body>
```

1
2
3
4
5
6
7
8
9



8. GRID



En grid.html



```
<body class="p-4">
  <div class="grid grid-cols-4 gap-4">
    <div class="box col-span-2">1</div>
    <div class="box">2</div>
    <div class="box">3</div>
    <div class="box">4</div>
    <div class="box">5</div>
    <div class="box">6</div>
    <div class="box">7</div>
    <div class="box">8</div>
    <div class="box">9</div>
  </div>
</body>
```

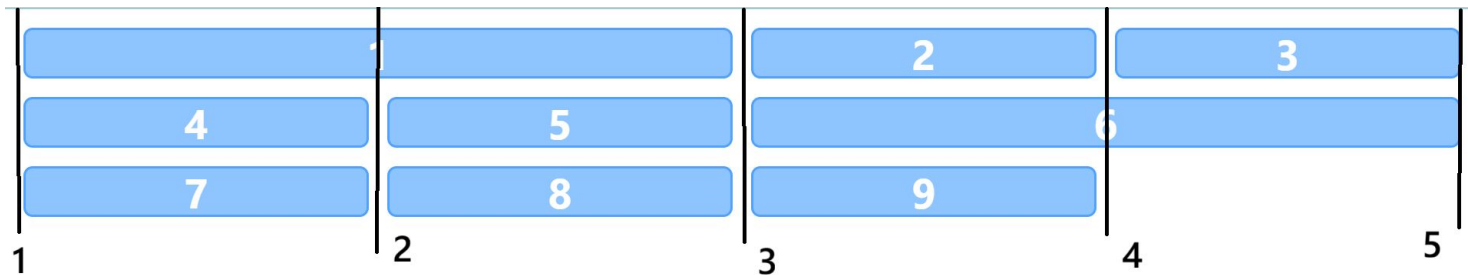
8. GRID



En grid.html



<https://tailwindcss-es.com/docs/grid-column#spanning-columns>

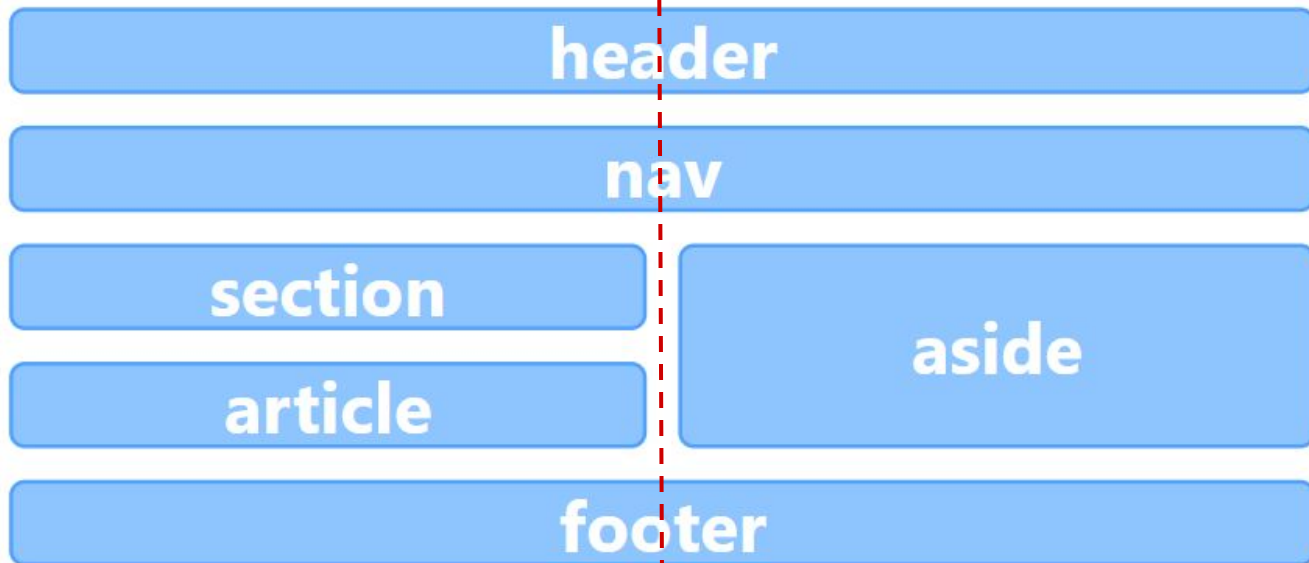


```
<div class="grid grid-cols-4 gap-4">
  <div class="box col-span-2">1</div>
  <div class="box">2</div>
  <div class="box">3</div>
  <div class="box">4</div>
  <div class="box">5</div>
  <div class="box col-[3/5]">6</div>
  <div class="box">7</div>
  <div class="box">8</div>
  <div class="box">9</div>
</div>
```

8. GRID



En grid.html



8. GRID

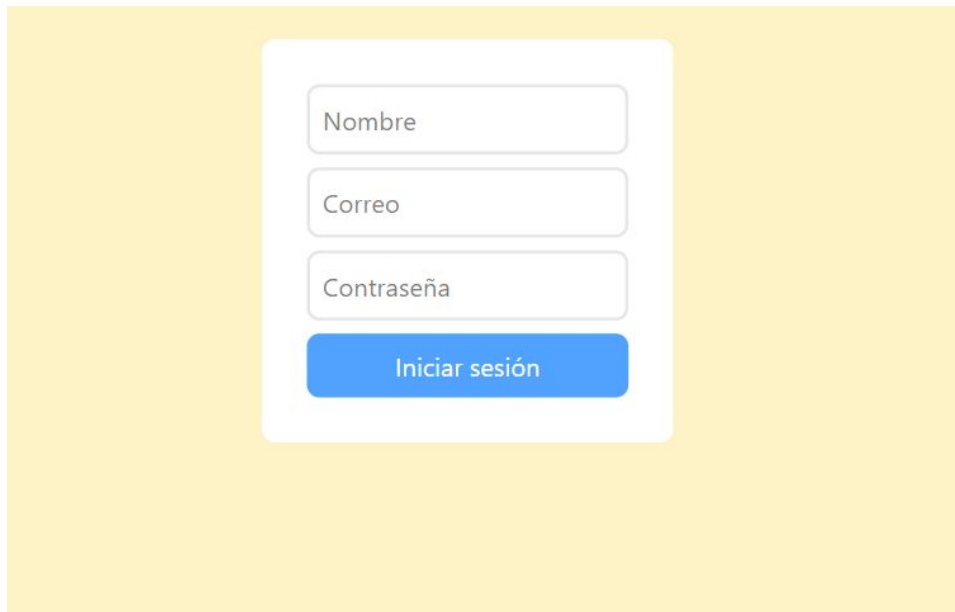


En grid.html

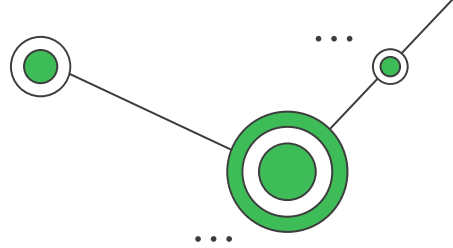
```
<div class="mt-8 grid grid-cols-2 gap-4">
  <div class="box col-span-2">header</div>
  <div class="box col-span-2">nav</div>
  <div class="box">section</div>
  <div class="box">article</div>
  <div class="box row-[3/5] col-start-2">aside</div>
  <div class="box col-span-2">footer</div>
</div>
```

9. ESTADOS EN TAILWIND

Antes de comenzar a trabajar con los estados en Tailwind, vamos a crear el siguiente formulario.

A login form mockup centered on a light yellow background. The form is a white rounded rectangle containing four stacked input fields: "Nombre", "Correo", and "Contraseña" (all in light gray text), followed by a blue button with the text "Iniciar sesión" in white.

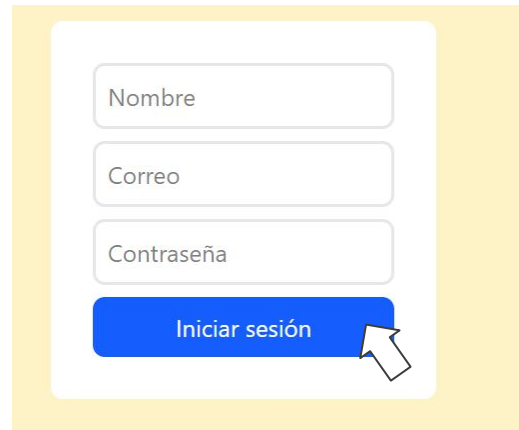
9. ESTADOS EN TAILWIND



★ <https://tailwindcss-es.com/docs/hover-focus-and-other-states>

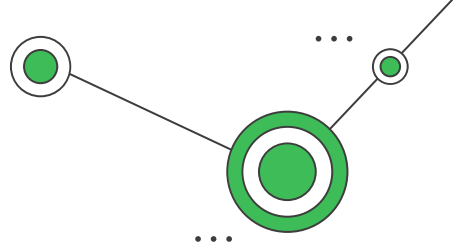
Cada clase de utilidad en Tailwind puede aplicarse condicionalmente añadiendo una variante al principio del nombre de la clase que describe la condición a la que quieres apuntar.

hover



```
<input class="border-0 p-2 m-1 rounded-lg bg-blue-400  
text-white hover:bg-blue-600 cursor-pointer" type="submit" value="Iniciar sesión">
```

9. ESTADOS EN TAILWIND



disabled

```
lg disabled:bg-red-200" t
```



focus

```
<input class="border-2 border-gray-200 p-2 m-1 rounded-lg focus:outline-none focus:ring-1 focus:ring-purple-600" type="email" placeholder="Correo"/>
```

A decorative graphic consisting of green circles of varying sizes connected by thin black lines, forming a network-like structure around the central text. Some circles have a white ring inside. Vertical ellipses are placed at several points along the lines.

UD01_Parte 2

Actividad de clase 3

Desarrolla pequeños
componentes con Tailwind CSS