

Desarrollo frontend con



tailwindcss

Tailwindcss

1. Así trabajamos
2. Atomic CSS
3. Tailwindcss
4. Maquetación
5. Configuración
6. Recursos

1. Así trabajamos

SEPARATION OF CONCERNS (SoC)

El **Modelo-Vista-Controlador(MVC)** que diferencia el desarrollo en backend y frontend, trae el concepto de **separación de intereses (SoC)**, principio de diseño que enfoca el trabajo en **intereses** respecto del código para la construcción y mantenimiento de una interfaz de usuario.



SEPARATION OF CONCERNS (SoC)

Separar trae muchos beneficios pero no podemos negar que llega a ser tedioso "brincar" del **HTML** a las reglas escritas en **CSS** para sincronizar la apariencia.

Esto se vuelve más complicado cuando varios desarrolladores necesitan actualizar, modificar o depurar una aplicación de forma simultánea.



Situaciones comunes al retomar un desarrollo con CSS:

- Reglas de estilo sin nombre intuitivo.
- Se agregan reglas sin revisar las que existen.
- Sobreescritura de estilos.
- Reglas sin sentido en nuestro proyecto.
- Un cambio puede implicar el recorrido de decenas de líneas de CSS.

Esto sucede si la persona que escribió las reglas de estilo no está o estando no recuerda porque las puso.



¿Agrego más estilos?

Para un desarrollo en constante evolución significa que se han de escribir muchas reglas adicionales.

Sobre todo en equipos de desarrollo donde cada colaborador resuelve como mejor le parece.





Los archivos CSS podrían convertirse en un desastre. Los desarrolladores no se sienten cómodos al eliminar reglas de estilo porque no están seguros de qué componentes se verán afectados en el proyecto.

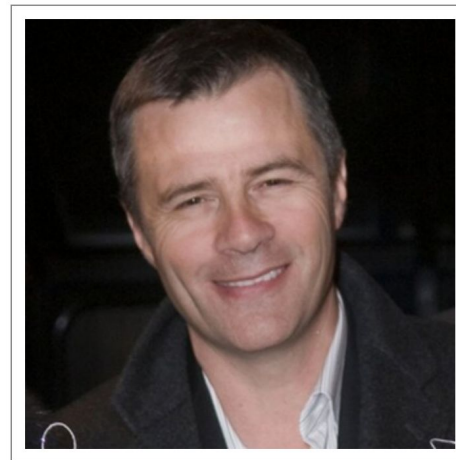


2. Atomic CSS

Atomic CSS

Atomic CSS es un término acuñado en 2013 por **Thierry Koblentz**, aparece en su artículo de *Smashing Magazine* titulado *Challenging CSS Best Practices*.

Koblentz habla de lo doloroso e intrincado que se torna manejar CSS tradicional y presenta una nueva forma de implementar diseño junto con el lenguaje de marcas, le da el adjetivo de *game-changer*.

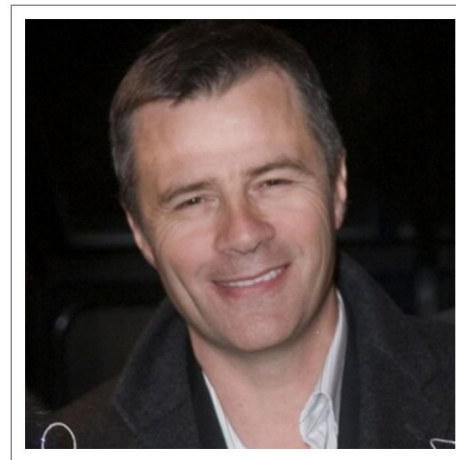


Thierry Koblentz

Utility-first

Tradicionalmente cada regla CSS representa un elemento de la interfaz de usuario, describe su apariencia.

Los frameworks bajo el principio **Utility-first** le dan la vuelta a esta idea al proporcionarnos una “caja de herramientas” con **clases de utilidad**, **clases de un solo propósito** que podemos usar para aplicarlas a los elementos HTML y diseñar componentes.



Thierry Koblenz

Utility-first

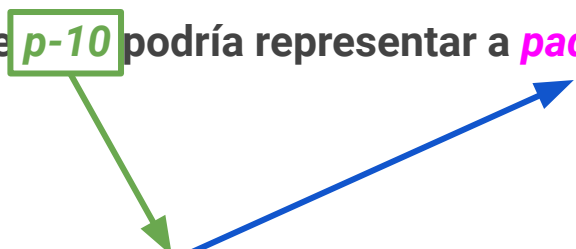
CSS convencional

```
// .css
.item{
  padding: 10px 10px 20px;
  margin: 20px 30px;
}

// .html
<div class="item"></div>
```

Donde `p-10` podría representar a *padding: 10px;*

```
<div class="p-10 pb-20 mx-20 my-30"></div>
```



Utility-first

Al aplicar estilos predefinidos a los elementos se simplifica el proceso de dar apariencia, tenemos un código más claro y se proporcionan muchas opciones de diseño adicionales que contribuyen a crear contenido personalizado.

Las clases de utilidad pueden parecer extrañas al principio, pero después de usarlas se aprecian sus beneficios.

Donde `p-10` podría representar a *padding: 10px;*

```
<div class="p-10 pb-20 mx-20 my-30"></div>
```

Utility-first - Frameworks

La idea de un enfoque basado en la utilidad para la denominación de clases ya lleva un tiempo existiendo, pero tomó impulso hasta el lanzamiento de **Tailwindcss**.



Expressive CSS



Basscss

The shed.css logo is the text "shed.css" in a light blue, lowercase, sans-serif font, set against a dark blue rectangular background.

shed.css



Windi CSS

The Tachyons logo is the word "TACHYONS" in a bold, white, uppercase, sans-serif font, centered on a blue rectangular background.

TACHYONS

CSS for designing faster than the speed of light



Atomizer

Tailwindcss

Tailwindcss ganó mucho terreno en dos años (2019 - 2021) según la encuesta anual *State of CSS 2021*



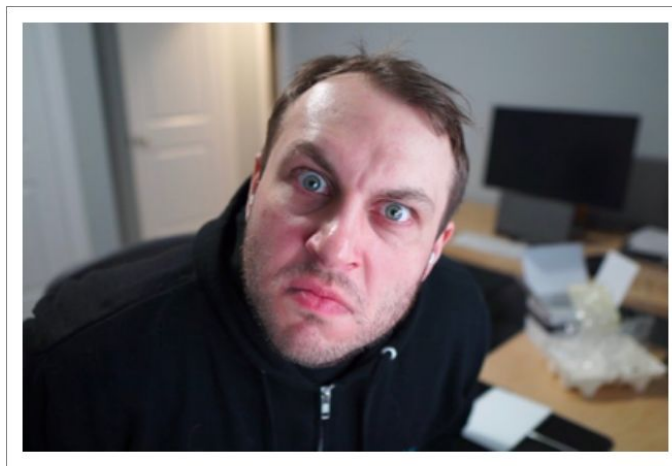
<https://2021.stateofcss.com/es-ES/technologies/css-frameworks/>

3. Tailwindcss

Tailwindcss

Tailwindcss se basa en **Utility-first** y es un proyecto CSS iniciado por Adam Wathan. El primer alfa se realizó en 2017 y la primera versión estable fue liberada en 2019.

Su nombre hace alusión a la frase naval "viento en popa", utilizada coloquialmente para referirnos a que el tiempo o circunstancias están a nuestro favor.



Adam Wathan

Tailwindcss - Características

Tailwindcss establece un conjunto de directivas que permiten la creación de un vínculo estrecho entre el estilo y el contenido. Algunas de sus características:

- Código predecible, mantenible y escalable.
- Reduce la abstracción.
- Evita la redundancia de reglas.
- Detiene el crecimiento sin control de CSS.
- Omite las reglas sin usar.
- Mejora de almacenamiento en caché
- Se favorece la compresión gzip.



Tailwindcss - Instalación

Se puede instalar en 3 contextos:

- CLI
- Framework de desarrollo
- CDN (no compilado)

Tailwindcss rastrea los archivos y directorios que indiquemos, busca los nombres de clase utilizados y genera los estilos correspondientes en un archivo CSS estático.



4. Maquetación

Maquetación responsiva con Flexbox

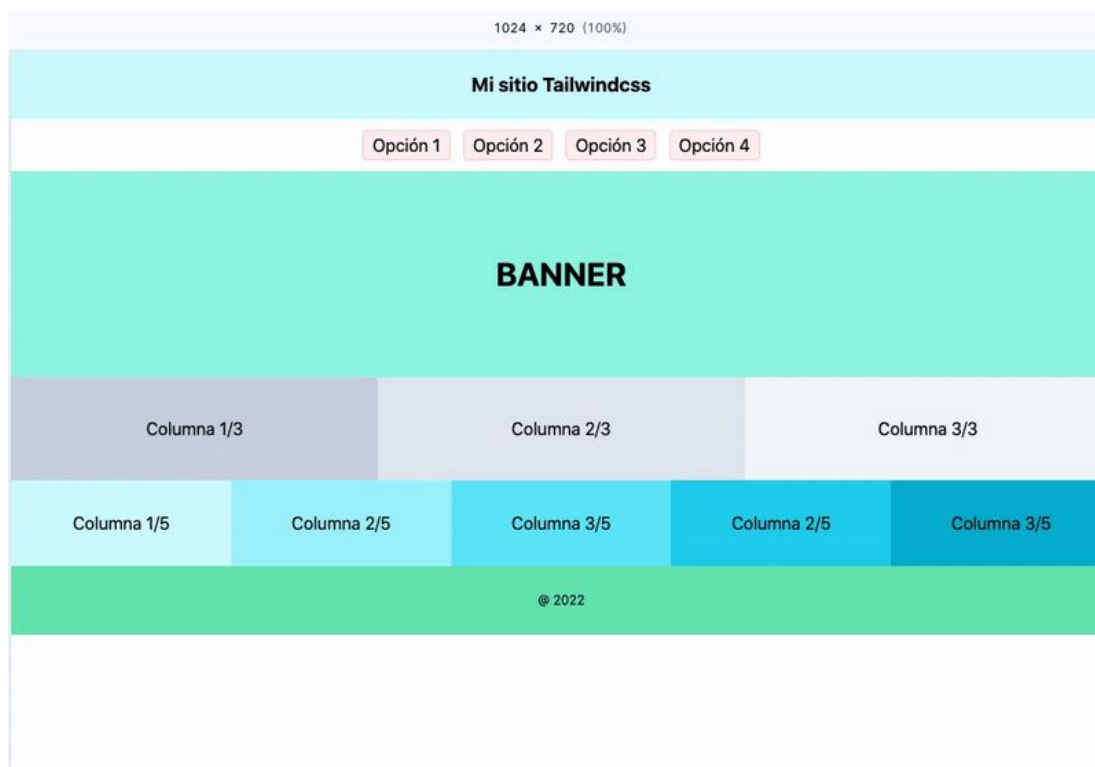
Tailwindcss implementa la maquetación mediante **Flexbox** combinado con sufijos.

md:w-1/4

El ejemplo indica un cuarto del ancho total del contenedor en pantallas medianas (**md=middle**).

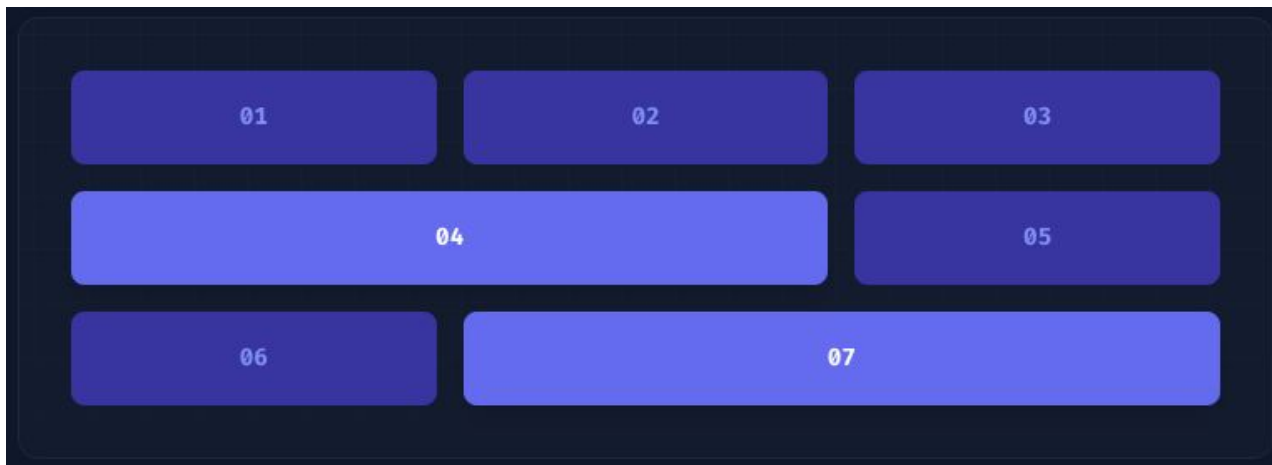
Prefijo	Ancho mínimo	@media
sm	640px	@media (min-width: 640px) { ... }
md	768px	@media (min-width: 768px) { ... }
lg	1024px	@media (min-width: 1024px) { ... }
xl	1280px	@media (min-width: 1280px) { ... }
2xl	1536px	@media (min-width: 1536px) { ... }

Maquetación responsiva con Flexbox



Maquetación responsiva con Grid Layout

Tailwind, a diferencia de otros frameworks, facilita la colocación de elementos en rejillas gracias a **Grid Layout**, basta con indicar las clases propias de **grid** en el elemento contenedor para definir columnas o filas.



5. Configuración

Directivas CSS

En una hoja de estilos se han de colocar las 3 directivas que contienen todo lo necesario para generar las clases que usaremos, con la posibilidad de utilizar *@apply*, la directiva de extracción que nos permite reutilizar y agrupar conjuntos de estilos.

```
@tailwind base;  
@tailwind components;  
@tailwind utilities;
```

Ruta de rastreo de contenido

El archivo ***tailwind.config.js*** es donde debemos indicar las rutas de contenidos por rastrear para compilar los estilos.

```
module.exports = {
  content: ["./public/**/*.{html,js}"],
  theme: {
    extend: {},
  },
  plugins: [],
}
```

Modificación de reglas

En el archivo ***tailwind.config.js*** podemos modificar y agregar parámetros que afectarán la apariencia, y también podemos hacer otras configuraciones.

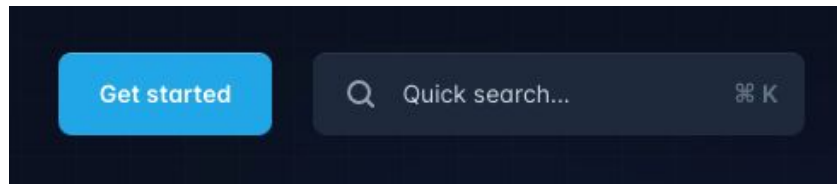
```
module.exports = {
  theme: {
    screens: {
      sm: '480px',
      md: '768px',
      lg: '976px',
      xl: '1440px',
    },
    colors: {
      'blue': '#1fb6ff',
      'green': '#13ce66',
      'yellow': '#ffc82c',
      'gray': '#8492a6',
    },
    fontFamily: {
      roboto: ['Roboto', 'sans-serif'],
      open: ['Open Sans', 'sans-serif'],
    },
    extend: {
      spacing: {
        '128': '32rem',
        '144': '36rem',
      },
      borderRadius: {
        '4xl': '2rem',
        '5xl': '2.5em',
      }
    }
  }
}
```

6. Recursos

Quick search...

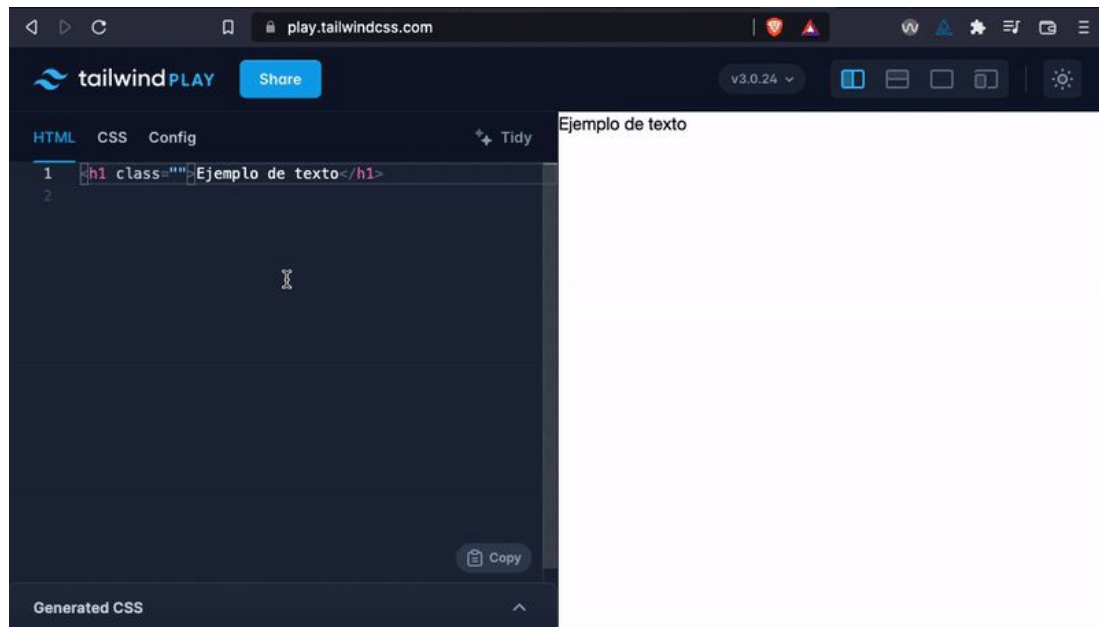
Es imposible saber de memoria todas las propiedades y encima de ello todas las clases **Tailwindcss** y sus sintaxis correspondientes. Por eso hay varios recursos que se encuentran a nuestro alcance para ensayar y verificar que nuestros procedimientos son correctos.

Podemos utilizar el buscador de la documentación oficial.



Playground

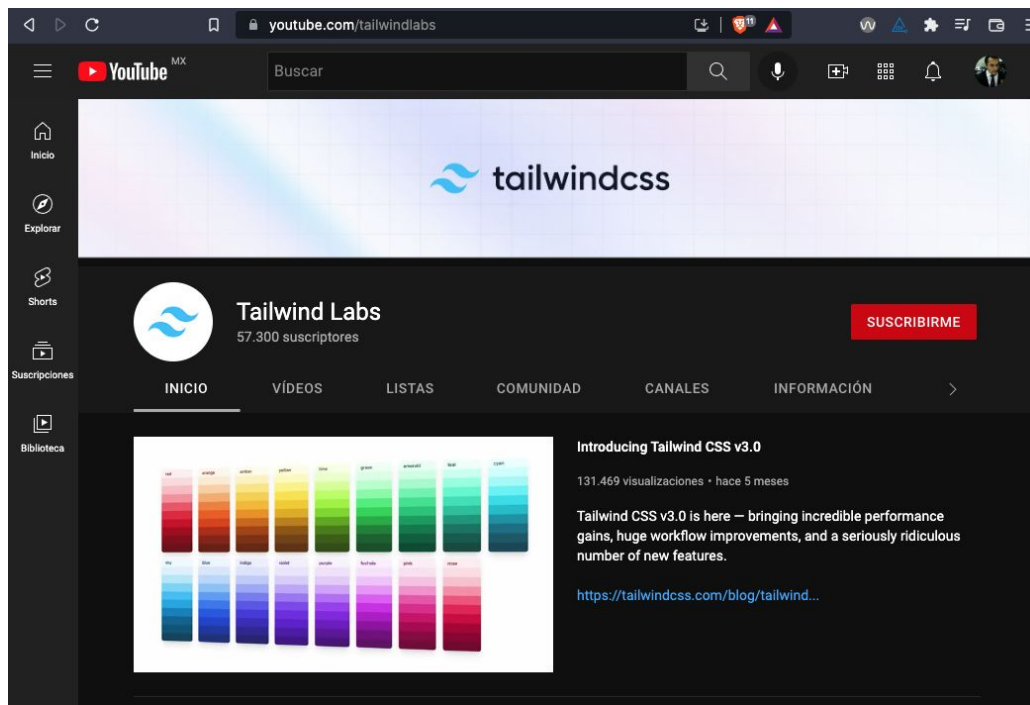
Podemos usar también el *Playground* en <https://play.tailwindcss.com/>



Screencast

Si necesitamos algo más explícito y práctico podemos recurrir al canal oficial de Youtube

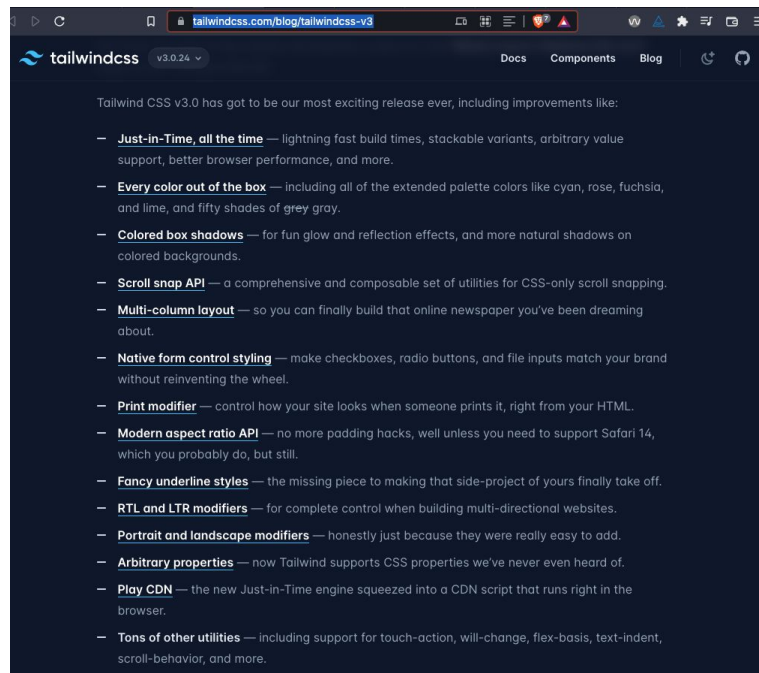
<https://www.youtube.com/tailwindlabs>



Blog

Hay un blog donde podemos buscar diversos temas

<https://tailwindcss.com/blog/tailwindcss-v3>



Otros recursos

<https://tailwindui.com/>

<https://tailwindcomponents.com/>

<https://www.tailwindtoolbox.com/>

<https://headlessui.com/>

<https://dev.to/cruip/25-places-where-you-can-get-free-tailwind-css-components-47lm>

GRACIAS

Edgar Vargas Zermeño • talos@unam.mx