CURSO DE TAILWIND CSS 1

Inicializamos el proyecto inicializando el git y npm,

Luego instalamos las librerías necesarias.

**npm i tailwindcss autoprefixer postcss-cli**

ahora inicializamos tailwind con npx

**npx tailwindcss init**

Creamos el archivo postcss.config.js y realizamos la siguiente configuración

module.exports = {

    Plugin: [

            require("tailwindcss"),

            require("autoprefixer"),

    ]

}

Ahora creamos una carpeta con el nombre css y dentro un archivo con tailwind.css y agregamos las siguientes líneas para start. Revisar la documentación en https://tailwindcss.com/docs/installation

@tailwind base;

@tailwind components;

@tailwind utilities;

Una vez creado nuestro archivo html enlazamos el siguiente archivo css, el cual se creara cuando ejecutemos nuestro proyecto con tailwind

    <link *rel*="stylesheet" *href*="../public/css/style.css">

Y creamos el siguiente script

"build": "postcss css/tailwind.css -o public/css/styles.css"

En caso tal de que al ejecutar este archivo y quede vacío cambial el build por el siguiente comando.

npx tailwindcss-cli -i src/tailwind.css -o public/css/styles.css --watch

este comando si creara el archivo con todas las clases.

**Nota:**

.la siguiente línea de comando sirve para limpiar o dejar sin estilo predeterminado todo nuestro html

@tailwind *base*;

. Nos genera el código de utileria, es decir nos va a permitir usar todas las clases de tailwind

@tailwind *utilities*;

Apunte adicionales::

<h1>1.3 Directivas de Tailwind</h1>

Directiva es una instrucción que utiliza tailwind para insertar código en el archivo final de css que genera.

@tailwind base  
Esto inyecta los estilos base de Tailwind y cualquier estilo base registrado por plugins.

@tailwind components  
Esto inyecta las clases de componentes de Tailwind y cualquier clase de componente registrado por los plugins.

@tailwind utilities  
Esto inyecta las clases de utilidad de Tailwind y cualquier clase de utilidad registrada por los plugins.

@tailwind screens  
Esta directiva sirve para controlar donde Tailwind inyecta las variaciones responsivas de cada utilidad. Si se omite, Tailwind añadirá estas clases al final de tu css, por defecto

**REALIZANDO CONFIGURACIÓN ADICIONAL AL ARCHIVO DE CONFIGURACIÓN DE TAILWIND**

**Configuración**

Ya que Tailwind es un framework para construir UI a la medida, por default, se tiene un archivo *opcional* llamado tailwind.config.js en la raíz de la carpeta, donde está el package.json.

**Creando un archivo de configuración**

Para generar un archivo de configuración para Tailwind, podemos usar el *Tailwind CLI*:

npx tailwind init

**Nota**. Podemos utilizar esta herramienta cuando instalamos la dependencia via npm.  
Donde como resultado tendremos tailwind.config.js:

module.exports = {

theme: {},

variants: {},

plugins: [],

}

Cabe mencionar que cada sección, del archivo de configuración, es opcional.

**La sección *Theme***

Esta sección es donde definimos los aspectos relacionados con el diseño visual de nuestro sitio.

...

theme: {

screens: {

sm: '640px',

md: '768px',

lg: '1024px',

xl: '1280px',

},

fontFamily: {

display: ['Gilroy', 'sans-serif'],

body: ['Graphik', 'sans-serif'],

},

borderWidth: {

**default**: '1px',

'0': '0',

'2': '2px',

'4': '4px',

},

extend: {

colors: {

cyan: '#9cdbff',

},

spacing: {

'96': '24rem',

'128': '32rem',

}

}

}

...

**La sección *Variants***

Esta sección nos permite controlar el comportamiento de las utilidades core, como *responsive variants* y *pseudo-class variants*.

...

variants: {

appearance: ['responsive'],

// ...

borderColor: ['responsive', 'hover', 'focus'],

// ...

outline: ['responsive', 'focus'],

// ...

zIndex: ['responsive'],

},

...

**La sección *Plugins***

Esta sección nos permite registrar plugins de terceros con el objetivo de extender utilidades, componentes, estilos, etc.

...

plugins: [

require('tailwindcss-transforms'),

require('tailwindcss-transitions'),

require('tailwindcss-border-gradients'),

],

...

**TRABAJANDO CON LAS ETIQUETAS DE COLOR:**

Mis apuntes de esta clase:

<h1>3.1 Colores</h1>

Los elementos que pueden ser afectados por los colores son:

* Fondo.
* Texto.
* Bordes.
* Placeholder.

Taildwind por default tiene valores ya predefinidos, para conocerlos tenemos que generar un archivo de configuración con todos los valores completos:  
npx tailwindcss init tailwind.config.full.js --full

Se puede ver la paleta de colores por defecto aquí:  
<https://tailwindcss.com/docs/customizing-colors/#default-color-palette>

NOTA: podemos crear una configuración completa del archivo de tailwind con el siguiente comando

**npx tailwindcss init tailwind.config.full.js --full**

## Responsive Web Design

La web como la conocemos, hoy en día, no es una tecnología pensando en un **UX** o User Experience generando, desafortunadamente, que muchos de los sitio web no estén optimizados para los **dispotivos móviles**.

Definido por [Ethan Marcotte](https://alistapart.com/article/responsive-web-design/), es una filosofía que responde a las necediades de los usuarios y a los dispositivos que estamos usando.

## Mobile First

Como su nombre sugiere, significa que iniciaremos con el diseño de móviles y expandiendo éstas características para crear una verión en tableta o escritorio/web tradicional.

Cabe mencionar que esta filosofía no es sinónimo de limitación, por lo que tenemos que tener el mismo contenido tanto en escritorio como en móvil. Google describe las [best practices](https://developers.google.com/search/mobile-sites/mobile-first-indexing" \t "_blank) en su sitio.

## Utility first

CSS posee diferentes tipos de paradigmas para abstraer un diseño, como BEM descrito por [Tailwind](https://tailwindcss.com/docs/utility-first/" \t "_blank) en ésta sección:

<div class="chat-notification">

<div class="chat-notification-logo-wrapper">

<img class="chat-notification-logo" src="/img/logo.svg" alt="ChitChat Logo">

</div>

<div class="chat-notification-content">

<h4 class="chat-notification-title">ChitChat</h4>

<p class="chat-notification-message">You have a new message!</p>

</div>

</div>

Sin embargo, llega a ser muy complicado establecer un armonía entre desarrolladores-diseñadores si no hay una concesión de sintaxis o linter. Por ello, Tailwind establece su propio orden con el objetivo:

* No invertir tiempo en inventar nombres para las clases
* Detener el crecimiento, sin fin, de CSS
* Simplificando y asegurando los cambios al CSS

## Herramienta para desarrollo 🚀

Como desarrolladores necesitamos herramientas que nos permita exlotar nuestra creatividad al máximo por ellos exiten algunas DevTools:

* [Safari for Developers](https://developer.apple.com/safari/)
* Chrome o [Chrome for Developers](https://www.google.com/chrome/dev/)  
  Y otras más … 🤯

**Solución si no se crear el archivo css con todas las clases.**

**EN ESTA CLASE ME ENCONTRÉ CON QUE NO SE ME REFLEJABAN LOS CAMBIOS EN CUANDO AL COLOR, LES COMPARTO LOS PASOS QUE SEGUI DESPUES DE BUSCAR UNA SOLUCIÓN:**

*(EN MI CASO TUVE QUE HACER TODO DESDE CERO, PERO DEPENDE DE CADA QUIEN SI LO HACE DESDE CERO O NO)*

**1. CREE LA CARPETA DE MI PROYECTO Y DENTRO DE LA CARPETA EJECUTE LOS SIGUIENTES COMANDOS**

npm init -y

npm i tailwindcss

**2. AHORA ENTRAS A TU EDITOR DE CÓDIGO PREFERIDO Y CREAS UN DIRECTORIO:**

src/estilos.css

**3. DENTRO DEL DIRECTORIO CREADO ANTERIORMENTE PONEMOS LAS SIGUIENTES LINEAS Y GUARDAMOS:**

@tailwind base;

@tailwind components;

@tailwind utilities;

**4.COMPILAMOS PARA CREAR NUESTRO ARCHIVO DE SALIDA**

**¡OJO AQUI! YA QUE EL ARCHIVO QUE SE GENERA (output.css) DEBE TENER MÁS DE 19,000 LINEAS**  
(EN MI CASO SON LOS PASOS ENSEÑADOS EN LA CLASE, MI ARCHIVO GENERABA UNICAMENTE 418 LINEAS, LO CUAL POR LO QUE LEÍ, ES INCORRECTO)

npx tailwindcss-cli -i src/estilos.css -o public/output.css --watch

**5.CREAMOS NUETRO ARCHIVO index.html DENTRO DEL DIRECTORIO QUE SE CREO EN EL PASO ANTERIOR  
->public<- Y LLAMAMOS A LOS ESTILOS**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="output.css">

</head>

<body>

<h1 class="mt-20">Hola Tailwind</h1>

</body>

</html>

**6.AHORA EJECUTAMOS EL SIGUIENTE COMANDO**

npx tailwindcss **init**

**ESTE COMANDO NOS GENERARÁ EL ARCHIVO**

tailwind.config.js

**7.DENTRO DE ESTE ARCHIVO DEBERÁ ESTAR LO SIGUIENTE (YO LO PUSE DE PRUEBA NO ES NECESARIO)**

module.exports = {

purge: [],

theme: {

extend: {

colors: {

danger: '#ff5f40',

info: {

900: '#234e52',

800: '#285e61',

}

},

},

fontFamily: {

rale: ['Raleway'],

},

},

variants: {},

plugins: [],

}

**8. ¡NO OLVIDEN COMPILAR NUEVAMENTE EL CÓDIGO PARA ACTUALIZAR EL ARCHIVO CSS DE SALIDA!**

npx tailwindcss-cli -i src/estilos.css -o public/output.css --watch

**9. UNA VEZ COMPILADO Y ACTUALIZADO NUESTRO ARCHIVO CSS DE SALIDA, HACEMOS UNA PRUEBA PARA VERIFICAR QUE YA ESTE FUNCIONANDO TODO CORRECTO**

<body>

<h1 class="text-danger">Lorem, ipsum dolor.</h1>

<h2 class="bg-yellow-500 text-white">Lorem, ipsum dolor.</h2>

</body>

**¡Y LISTO, CON ESTOS PASOS A MI ME FUNCIONÓ, ADICIONAL A ESTO LES COMPARTO LA DOCUMENTACIÓN DE LA QUE SAQUE ESTOS PASOS POR SI GUSTAN VERIFICARLOS, SALUDOS!**

[DOCUMENTACIÓN - PASOS A SEGUIR PARA INSTALACIÓN Y PRIMEROS PASOS TAILWIND CSS](https://bluuweb.github.io/tailwindcss/03-personalizar/#ejemplos)

**COLORES**

Para los colores tenesmo las siguientes clases, recordar que las vistas son algunas.

<div *class*="w-1/5 p-4 bg-blue-800">x</div>

Donde bg-blue-800 es el color de backgroud blue con la intensidad 800 del azul.

<div *class*="border-blue-500 w-full sm:w-1/2 md:w-1/3 lg:w-1/4 xl:w-full p-4 bg-green-500 mb-1">x</div>

También podemos utilizar la propiedad border-blue-800 que cumple las mismas características del anterior un border blue con intensidad de 500.

<input *class*="placeholder-green-300" *placeholder*="johann" />

**GRID BÁSICO**

Vamos a utilizar flex para crear nuestros propios grid.

<div *class*="flex w-full gap-1">

        <div *class*="w-1/5 p-4 bg-blue-800">x</div>

        <div *class*="w-1/5 p-4 bg-blue-600">x </div>

        <div *class*="w-1/5 p-4 bg-blue-300">x </div>

        <div *class*="w-1/5 p-4 bg-blue-100">x </div>

        <div *class*="w-1/5 p-4 bg-blue-200">x </div>

    </div>

Donde flex hace referencia a display: flex, gap-1 = gap: 0.25rem , w-full = width: 100%

  <div *class*="flex flex-wrap gap-1 m-4 w-full">

        <div *class*="w-full sm:w-1/2 md:w-1/3 lg:w-1/4 xl:w-full p-4 bg-green-500 mb-1">x</div>

        <div *class*="w-full sm:w-1/2 md:w-1/3 lg:w-1/4 xl:w-1/6 p-4 bg-green-500 mb-1">x</div>

        <div *class*="w-full sm:w-1/2 md:w-1/3 lg:w-1/4 xl:w-1/6 p-4 bg-green-500 mb-1">x</div>

        <div *class*="w-full sm:w-1/2 md:w-1/3 lg:w-1/4 xl:w-1/6 p-4 bg-green-500 mb-1">x</div>

        <div *class*="w-full sm:w-1/2 md:w-1/3 lg:w-1/4 xl:w-1/6 p-4 bg-green-500 mb-1">x</div>

    </div>

**DIMENSIONES Y ESPACIOS**

En estos enlaces podemos encontrar toda la referencia para el ancho, largo, padding y margin.

**Aquí pueden ver los anchos** <https://tailwindcss.com/docs/width>

**Aquí pueden ver los padding** <https://tailwindcss.com/docs/padding>

**Aquí pueden ver los margin** <https://tailwindcss.com/docs/margin>

<div *class*="flex w-full gap-1">

        <div *class*="w-1/5 p-4 bg-blue-800">x</div>

        <div *class*="w-1/5 p-4 bg-blue-600">x </div>

        <div *class*="w-1/5 p-4 bg-blue-300">x </div>

        <div *class*="w-1/5 p-4 bg-blue-100">x </div>

        <div *class*="w-1/5 p-4 bg-blue-200">x </div>

    </div>

**CAMBIANDO LAS PROPIEDADES DE TIPOGRAFÍA**

<h1>3.3 Cambiando las propiedades de la tipografía</h1>

Se puede cambiar:

* Tipo de letra.
* Tamaño.
* Cursiva y negrita.

Se aplican clases como:  
text-3xl font-sans font-light

Se puede consultar la documentación en:  
<https://tailwindcss.com/docs/font-family/#app>  
<https://tailwindcss.com/docs/font-weight/#app>  
<https://tailwindcss.com/docs/font-size/#app>

<div>

        <h2 *class*="text-3xl font-serif"> Hello mundo cruel</h2>

        <p *class*="text-xl font-sans">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Repellat,.</p>

        <p *class*="text-xs font-mono">Lorem ipsum dolor sit, amet consectetur adipisicing elit. Repellat,</p>

        <ul *class*="text-4xl">

            <li *class*="italic">lorem  ipsum</li>

            <li *class*="font-extrabold">lorem  ipsum</li>

            <li *class*="font-light">lorem  ipsum</li>

            <li *class*="font-extralight">lorem  ipsum</li>

        </ul>

    </div>

**AJUSTANDO EL ESPACIADO ENTRE LETRAS Y LÍNEAS**

2 instrucciones que debemos aprender: **Tracking** espaciamiento entre letras **Leading** espaciamiento entre líneas

* <https://tailwindcss.com/docs/line-height>
* <https://tailwindcss.com/docs/letter-spacing>