



DGTIC UNAM

DIRECCIÓN GENERAL DE CÓMPUTO Y
DE TECNOLOGÍAS DE INFORMACIÓN
Y COMUNICACIÓN

Guía de implementación de Tailwindcss en proyecto Web



Edgar Vargas Zermeño
Noviembre 2022

Indice

1. ¿Qué es Tailwindcss?	2
2. Contextos de implementación	4
3. Antes de instalar	5
4. Implementación CLI en 4 pasos	6
PASO 1: Estructura	6
PASO 2: Instalación	6
PASO 3: Configuración básica	7
PASO 4: Compilación	8
5. Registro del comando de compilación	10

1. ¿Qué es Tailwindcss?

Es un marco de trabajo (**framework**) CSS basado en clases de utilidad que se aplican directamente en las etiquetas **HTML** para crear cualquier diseño de interfaz de usuario.

Ya implementado **Tailwindcss**, se elimina la apariencia por defecto asignada en los navegadores, para escribir las reglas de estilo de **Tailwindcss** y verlas reflejadas en la apariencia de nuestro proyecto.



Ver ejemplo en el [Playground de Tailwindcss](#)

Su forma de trabajo permite la creación de un vínculo estrecho entre el estilo y el contenido en un código más predecible, mantenible y escalable.

2. Contextos de implementación

Se puede utilizar básicamente en tres contextos de desarrollo:

Distribuidor de contenidos (CDN, Content Delivery Network)

La implementación con **CDN** solo sirve para hacer pruebas de maquetación y apariencia, no es recomendable usarlo en producción porque contradice el espíritu de optimización de Tailwindcss. Se omite de esta guía porque solo basta vincular su script en la cabecera del documento.

Documentación en: <https://tailwindcss.com/docs/installation/play-cdn>.

Entornos de desarrollo (Frameworks)

Tampoco se considera la implementación en entornos de desarrollo porque dependerá de la naturaleza y filosofía de cada entorno su puesta en desarrollo. Sin embargo, la implementación por línea de comandos conserva muchos factores en común, de manera que esta guía servirá para entender los conceptos antes de probarlo en un entorno de desarrollo específico.

Documentación en:

<https://tailwindcss.com/docs/installation/framework-guides>.

Línea de comandos (CLI, command line interface)

Documentación en: <https://tailwindcss.com/docs/installation>.

3. Antes de instalar

La ruta más corta para generar un proyecto con Tailwindcss es mediante el manejador de paquetes de **node (NPM, Node Package Manager)**. De forma que debe verificar que su sistema operativo cuente con **Node.js** instalado. Para revisar abra una terminal (**linux**) o línea de comandos (**Windows**) y ejecute el comando: **node -v**. La respuesta arrojará la versión de **node** y hará patente que lo tenemos instalado.

```
$ node -v  
v17.4.0
```

Ejemplo de verificación de versión de node

Ejecute después: **npm -v**. Arrojará la versión del manejador de paquetes de **Node**.

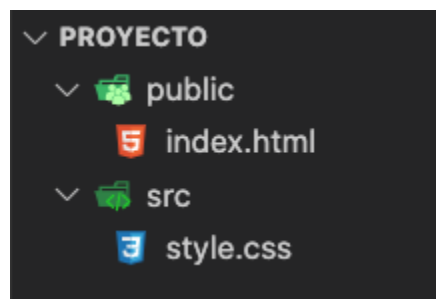
```
$ npm -v  
8.3.2
```

Si los comandos no arrojaron ningún resultado, instale **Node.js** desde el sitio oficial <https://nodejs.org/es/download/>. Al terminar ejecute los comandos descritos anteriormente.

4. Implementación CLI en 4 pasos

PASO 1: Estructura

Antes de todo necesitaremos una estructura específica dentro del directorio de trabajo, donde habrá dos carpetas con nombres sugeridos: **public** y **src**. El directorio **public** representa el contenido del directorio que usa un servidor con visibilidad pública, en él debe estar el archivo **.html** que usará los estilos y en el directorio **src** el archivo de entrada **.css**.



El archivo de entrada **style.css** debe contener las directivas de **Tailwindcss**.

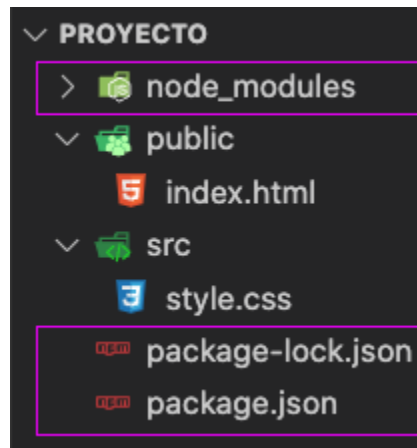
```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

PASO 2: Instalación

En la terminal o línea de comandos ubiquémonos en la carpeta del proyecto, ahí ejecutaremos el comando de instalación con el manejador de paquetes de **Node** indicando **Tailwindcss** como dependencia.

```
proyecto$ npm install -D tailwindcss
```

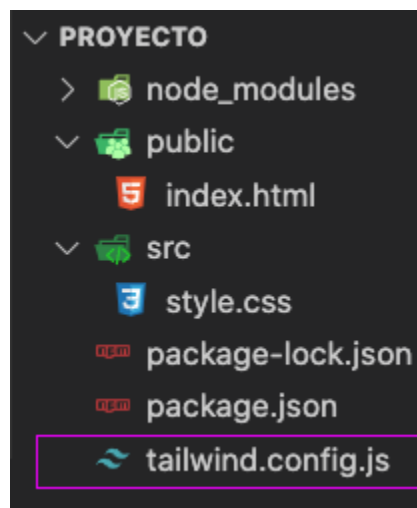
Además de leer en línea de comandos la confirmación de paquetes instalados, se generarán el directorio **node_modules** y 2 archivos **.json** nuevos.



PASO 3: Configuración básica

Para crear el archivo de configuración que define la actuación de **Tailwindcss** en el proyecto ejecutaremos un comando **NPX** de **Node**.

```
proyecto$ npx tailwindcss init
```



Una vez creado el archivo **tailwind.config.js** procederemos a indicar como valor de la propiedad **content** la ruta donde se encuentran los archivos que nos interesa rastrear para la compilación.

```

tailwind.config.js > ...
1  /** @type {import('tailwindcss').Config} */
2  module.exports = {
3    content: ["/public/**/*.html,js"],
4    theme: {
5      extend: {},
6    },
7    plugins: [],
8  }

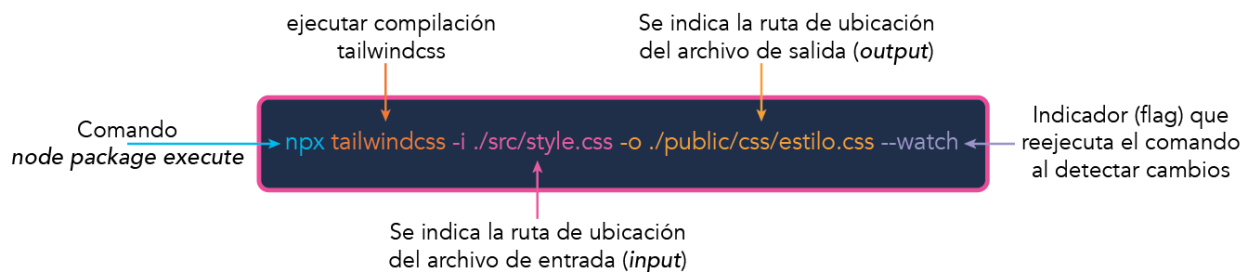
```

La ruta conformada indica que en el directorio **public**, todo archivo con extensión **.html** y **.js** que contenga clases de estilo **Tailwindcss**, será rastreado para compilar el archivo de estilos de salida que dará apariencia a las vistas de nuestro proyecto.

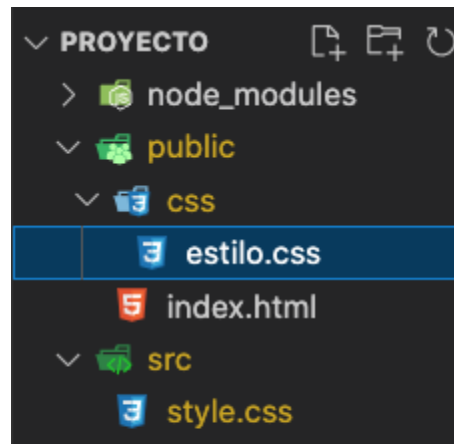
PASO 4: Compilación

Finalmente corresponde compilar el archivo **.css** de salida mediante un comando **NPM** que rastree los cambios que vamos realizando en el proyecto. Su conformación puede variar de un proyecto a otro, en este caso lo haremos corresponder con la estructura que tenemos:

```
npx tailwindcss -i ./src/styles.css -o ./public/css/estilo.css --watch
```



Al ejecutar el comando se genera el directorio **css** y el archivo de salida **estilo.css** dentro de **public**. Para detener la ejecución del comando oprima **ctrl + c**.



Para darle uso en el archivo **.html** vinculamos el archivo compilado y asignamos al menos una clase **Tailwindcss** en algún elemento de nuestro contenido.

```
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Proyecto Tailwindcss</title>
  <link rel="stylesheet" href="css/estilo.css">
</head>
```

5. Registro del comando de compilación

Podemos crear un script de **NPM** para ejecutar la compilación sin tener que escribir el comando **NPX** descrito en el PASO 4 del punto anterior. Para esto hay que editar el archivo **package.json**.



Anexamos una sección de scripts con el comando por ejecutar de la siguiente manera.

```

package.json > ...
1  {
2    "devDependencies": {
3      "tailwindcss": "^3.2.4"
4    },
5    "scripts": {
6      "watch": "npx tailwindcss -i ./src/style.css -o ./public/css/estilo.css --watch"
7    }
8  }
9

```

Ahora al ejecutar la instrucción **npm run watch** se usará el comando de compilación **NPX** según hayamos designado.