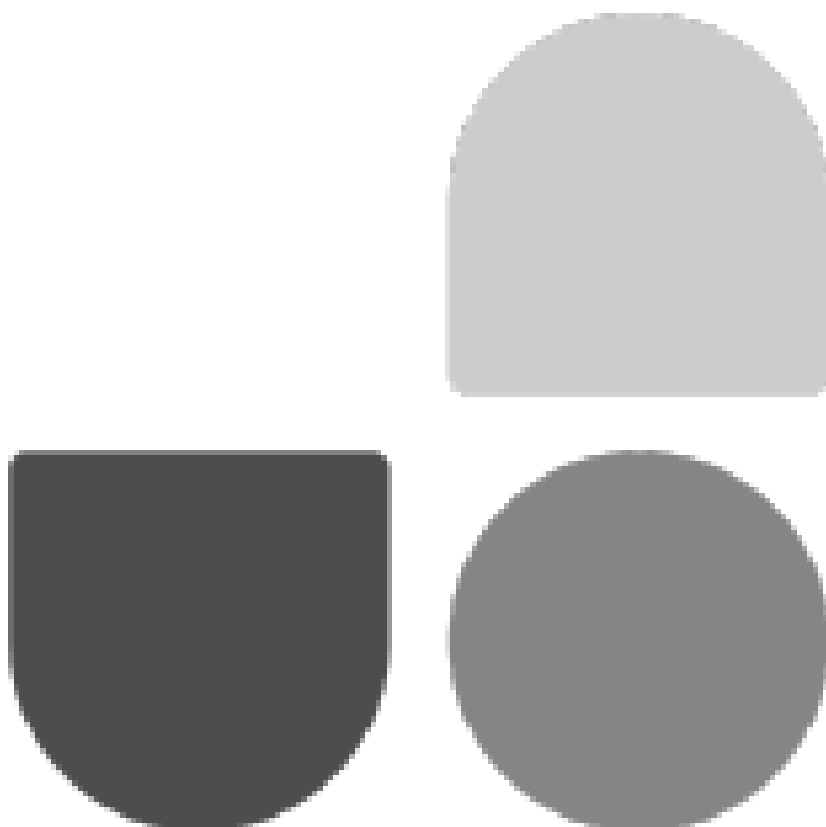


# Guía Inicial UnoCSS



Por *Manuel Amaya Orozco*

# **Índice**

- Historia y Framework (Pág. 2)
- Instalación (Pág. 3)
- Uso más común (Pág. 4)
- Ventajas e inconvenientes (Pág. 5)
- Ejemplos y sintaxis (Pág. 6)
- Conclusión (Pág. 8)

# Historia/Framework

UnoCSS es considerado el sucesor de WindiCSS, otro framework de CSS que priorizaba las utilidades, fue descontinuado en Marzo de 2023, y de ahí surgió UnoCSS.

UnoCSS hereda muchos aspectos de WindiCSS, sus atajos, modo de compilación, grupos de variantes, entre otros. Fue construido de cero por Anthony Fu, un experto que ha trabajado en otras herramientas como Vite y Vue.js, como un experimento cualquiera, teniendo en mente la demanda y flexibilidad que los usuarios suelen pedir. Fue creado para que fuera tanto intuitivo como customizable. UnoCSS también fue creado con TailwindCSS como UI inicial, pero con muchas más diferencias como un motor diferente e integración con otras herramientas y para ser más extensible.

UnoCSS no tiene un framework como tal, sino que es una biblioteca que genera clases CSS bajo demanda siguiendo una filosofía de diseño como Tailwind, pero siguiendo también su propia generación dinámica dependiendo de lo utilizado en el HTML o estilos.

El proyecto empezó a crecer en tamaño y popularidad con el tiempo, por su promesa de ser un sistema más rápido y ligero que los demás en el mercado.

A día de hoy se sigue actualizando y mejorando, tanto gracias a la comunidad como a su creador, y tiene un posible futuro prometedor.

# Instalación

Instalar UnoCSS es muy sencillo, desde una terminal dentro del proyecto en el que lo quieras instalar, solo tienes que ejecutar los siguientes comandos.

```
npm install -D unocss  
npm install -D @unocss/vite
```

Esto instalará UnoCSS en tu proyecto, si por algún motivo se trata de un proyecto sin vite, alternativamente puedes usar el siguiente comando.

```
npm install -D unocss postcss
```

Seguidamente has de acceder a tu archivo de configuración correspondiente e introducir las siguientes líneas para importar UnoCSS.

```
import { defineConfig } from 'vite'  
import UnoCSS from 'unocss/vite'
```

```
export default defineConfig({  
  plugins: [UnoCSS()],  
})
```

Si nuevamente no estás utilizando Vite, puedes añadir estas líneas alternativamente.

```
module.exports = {  
  plugins: [  
    require('unocss')(),  
    require('postcss-preset-env')(),  
  ],  
}
```

y después has de crear un archivo uno.config.js (Esto solo si no usas Vite).

```
import { defineConfig } from 'unocss'  
  
export default defineConfig({  
  // Personalización de reglas, colores, etc.  
})
```

Con esto hecho ya podrás utilizar clases de UnoCSS en cualquiera de los archivos de dentro de tu proyecto.

## **Uso más común**

El uso más común de UnoCSS es, como es de esperar, para optimizar la creación de estilos CSS para aplicaciones web. Pudiendo crear clases CSS bajo demanda, permitiendo mantener los archivos de estilo pequeños y eficientemente personalizados.

Es mayormente utilizado para proyectos grandes, ya que solo incluyen las clases que realmente se usan en el proyecto, resultando en un tamaño de archivo reducido, haciendo que sea muy útil en proyectos en el que el rendimiento es muy importante, como páginas web dinámicas o aplicaciones a gran escala. La generación dinámica de clases evita tener que predefinir todas las clases para mayor flexibilidad y personalización.

En proyectos que utilicen Vite, Vue o React, UnoCSS se integra muy bien y es considerado bastante popular.

La posibilidad de poder definir las reglas que el usuario quiera para adaptar a su proyecto lo hace muy útil a la hora de ser usado en proyectos más personalizados, pero también mantiene sus propias configuraciones predeterminadas, por lo que no es obligatorio tener que configurarlo.

# **Ventajas e Inconvenientes**

Las ventajas principales de UnoCSS son, como fueron mencionadas anteriormente, la generación dinámica de clases que solo se utilizan directamente en el proyecto, su enfoque en la generación bajo demanda, tamaño reducido de archivos, alta flexibilidad y personalización, integración con frameworks modernos como Vite, Vue y React, y una simpleza general e intuitiva.

Pero esto no significa que no tenga sus inconvenientes: si no estás acostumbrado a la generación dinámica de clases, UnoCSS requerirá una curva de aprendizaje considerable.

UnoCSS también tiene una notable falta de comunidad y documentación pública de fácil acceso, por lo que la mejor opción para aprender, documentarse o solucionar problemas sería accediendo a sus redes sociales principales como puede ser su Discord oficial.

UnoCSS también depende ligeramente de su compatibilidad con frameworks modernos como Vite, por lo que su uso con frameworks u otras herramientas más antiguas no es recomendable.

# Ejemplos y Sintaxis

Algunos ejemplos de uso de UnoCSS pueden incluir la utilización de clases utilitarias ya existentes como pueden ser las siguientes:

```
<div class="bg-primary text-white p-4"></div>
```

En este caso, bg-primary toma la configuración del color primario y establece el texto como blanco.

```
<div class="m-20 p-8"></div>
```

En este caso se establece un margin de 20px y un padding de 8px en este div.

El siguiente ejemplo demuestra cómo establecer un atajo de clase.

```
shortcuts: [  
  ['btn', 'px-4 py-2 bg-primary text-white rounded-lg'],  
]
```

Este atajo genera la clase 'btn' con sus propiedades correspondientes: padding horizontal de 4px, padding vertical de 2px, fondo de color primario, color de texto blanco y bordes redondeados. Podemos llamar a esta clase como cualquier otra:

```
<button class="btn"></button>
```

Si queremos una configuración aún más avanzada, podemos modificar algunas de las clases predeterminadas desde el archivo de configuración UnoCSS estableciendo unas reglas, como en el siguiente ejemplo:

```
rules: [  
  ['text-shadow', { textShadow: '2px 2px 4px rgba(0,0,0,0.3)' }],  
]
```

En este caso cambiamos las propiedades de text-shadow para que sea más suave, si lo utilizamos ahora podremos ver los cambios aplicados.

```
<h2 class="text-4xl text-shadow"></h2>
```

Como hemos visto, bg-primary establece un color primario, pero podemos también utilizar bg-secondary si queremos añadir otro color.

UnoCSS es de uso bastante sencillo con su generación rápida de clases, pero si se requiere un vistazo más profundo, también tiene una [guía](#) oficial a la que se puede acceder para ver más ejemplos y explicaciones más exhaustivas, además de otras páginas similares como [UnoCSS UI Example](#).



# **Conclusión**

UnoCSS es una muy buena alternativa a las otras herramientas de CSS, es de relativa facilidad de uso, permite una personalización profunda y rapidez en comparación con la competencia, aunque su falta de comunidad puede resultar en la necesidad de aprender más por cuenta propia, pero eso no quita que sea muy buena cumpliendo su función.