

# Preprocesador

[https://developer.mozilla.org/en-US/docs/Glossary/CSS\\_preprocessor](https://developer.mozilla.org/en-US/docs/Glossary/CSS_preprocessor)

Un preprocesador de CSS es una herramienta que permite generar CSS a partir de la sintaxis exclusiva del preprocesador. Hay muchos preprocesadores de CSS para elegir.

La mayoría de los preprocesadores de CSS agregarán algunas funciones que no existen en CSS puro, como mixin, selector de anidamiento, selector de herencia, etc. Estas características hacen que la estructura CSS sea más legible y fácil de mantener.

Para usar un preprocesador de CSS, debe instalar un compilador de CSS en su servidor web; o use el preprocesador CSS para compilar en el entorno de desarrollo y luego cargue el archivo CSS compilado en el servidor web.

## Entorno de trabajo

- Visual Studio Code
- Navegadores: Chrome, Mozilla
- Sass
- Npm

## Extensiones VS Code

- AutoCloseTag
- AutoRenameTag
- LiveServer
- HTML CSS Support
- Intellisense for CSS Class Names
- Sass
- Sass Lint
- Scss Intellisense

# Instalación

- Como instalación independiente
- Como programa de gestión de paquetes

<https://sass-lang.com/install>

<https://github.com/sass>

## Como programa de gestión de paquetes

Como vemos hay diferentes formas de instalar SaSS. Una de las más populares es usar node.js (node-sass).

### Node:

entorno de tiempo de ejecución javascript, que nos permite ejecutar aplicaciones javascript en el servidor

### npm:

aplicación que nos permite instalar y administrar paquetes

### Preparando el entorno de trabajo:

Instalar node (si no está instalado)

<https://nodejs.org/es/download/>

Comprobar node está instalado. Desde la línea de comandos

```
node --version
```

```
node -v
```

Comprobar npm está instalado.

```
npm --version
```

```
npm -v
```

¿Cómo vamos a trabajar?


Preparamos nuestro entorno de trabajo, esto es la carpeta de nuestro proyecto

scss -> usaremos una carpeta donde escribimos estilos en SaSS.

Para realizar algunas tareas como la compilación vamos a crear el fichero package.json

### Crear package.json

```
npm init --yes //para que no haga preguntas
```



```
> css
> scss
index.html
package.json
```

## Contenido de package.json

```
{ package.json > ...
1 {
2   "name": "1_Setup",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1"
8   },
9   "keywords": [],
10  "author": "",
11  "license": "ISC"
12 }
13
```

## Uso de Sass con node

### Instalar node-sass

//instalo en la carpeta donde está el proyecto

`npm install node-sass --save-dev`

//opción -g para indicar que es una instalación global y lo podremos usar en otros proyectos

`npm i -g node-sass`

### ¿Cómo compilar?

#### En la línea de comandos.

##### Compilación manual

`node-sass -o css scss\styles.scss`

donde

`-o css ->` carpeta destino, resultado compilación

`scss\styles.scss ->` lo que quiero compilar

##### Compilación automática

`node-sass -o css scss\styles.scss -w`

`node-sass scss/styles css\styles.scss -w`

#### Con el fichero json

Lo modificamos y añadimos un script para la compilación

```
"scripts": {
  "watch": "node-sass -o css scss/styles.scss -w"
},
```

##### Para llamarlo

`npm run watch`

## Instalar SaSS desde su repositorio

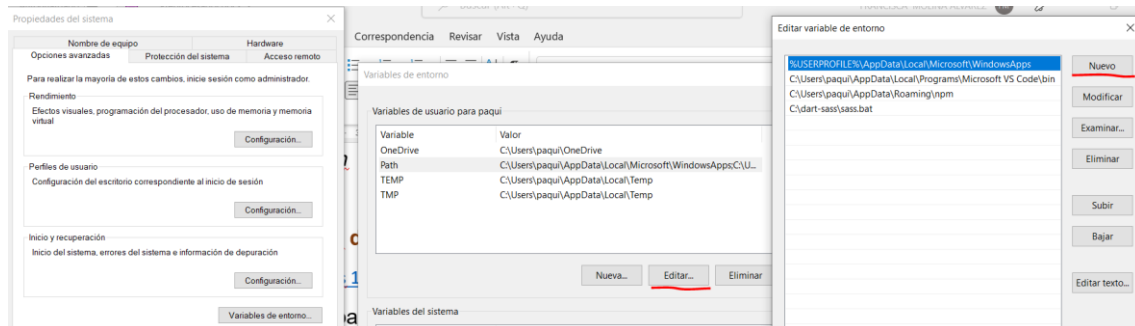
[Release Dart Sass 1.32.13 · sass/dart-sass · GitHub](#)

Descargar el paquete adecuado

Descomprimir en raíz, por ejemplo

Añadir al path de nuestro sistema operativo el ejecutable

[Path.md \(katiek2.github.io\)](#)



Compilando con SaSS y Otros tipos de compilación.

### Tipos

- Simple
- Múltiple
- Expandida
- Comprimida
- Vigilando los cambios

Simple

```
Sass scss\style.scss css\style.css
```

```
Sass scss:css ///carpeta
```

Múltiple

```
Sass file1.scss:output1.css ... filen.scss:outputn.scss
```

Expandida

```
Sass -style=expanded File.scss output.css
```

Comprimida

```
Sass -style=compressed File.scss output.css
```

Vigilando

```
Sass -watch scss\style.scss css\style.css
```

```
Sass -w scss\style.scss css\style.css
```

## Ficheros map

Se generan durante la compilación. Contienen el mapeado entre las reglas SCSS y el resultado CSS.

## Comentarios en la compilación

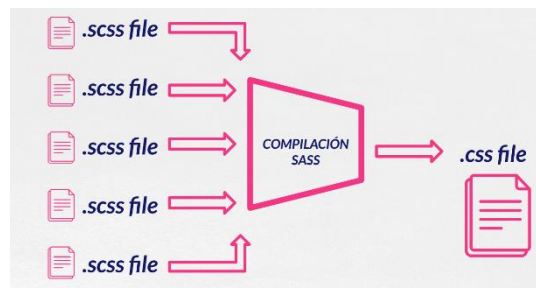
Podemos decidir si queremos que nuestros comentarios aparezcan o no en el CSS

// comentario que no aparece en CSS

/\* Comentario que se incluye menos en compressed\*/

/\*! Comentario que se incluye también en compressed\*/

## Esquema general



## Elementos básicos

- Variables
- Ámbito de las variables
- Comentarios
- Anidamiento
- Anidamiento referencia al padre
- Mixins
- Funciones
- Extend
- Placeholder

////////////////////////////////////