

# Documentación Sassdoc

## Componentes Sass

Primero hay que instalar sassdoc dentro de la carpeta donde previamente hemos iniciado npm:

### npm install sassdoc

```
alvaro@alvaro-Lenovo-Ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$ npm install sassdoc
npm WARN deprecated core-js@2.6.12: core-js@3.4 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whins, feature detection in old core-js versions could cause a slowdown up to 100x even if nothing is polyfilled. Please, upgrade your dependencies to the actual version of core-js.
added 266 packages, and audited 505 packages in 20s
31 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
To address all issues, run:
  npm audit fix
Run `npm audit` for details.
alvaro@alvaro-Lenovo-Ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$
```

Modificamos el archivo **package.json** para añadir el script con el que ejecutaremos sassdoc:

### code package.json

```
package.json U x
home > alvaro > Escritorio > DIW_20212022 > UD4_preprocesadoresCSS > 4.2_ComponentesSASS > package.json > {} scripts > sassdoc
1  {
2    "name": "4.2_componentessass",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \"Error: no test specified\" && exit 1",
8      "scss": "node-sass -w -watch scss -o css",
9      "sassdoc": "sassdoc scss -d sassdocs"
10   },
11   "author": "",
12   "license": "ISC",
13   "dependencies": {
14     "node-sass": "^7.0.1",
15     "sassdoc": "^2.7.3"
16   }
17 }
18
```

Comprobamos los archivos **.scss**:

### ls; tree scss

```
alvaro@alvaro-Lenovo-Ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$ ls; tree scss
css gallery grid loader node_modules package.json package-lock.json scss
scss
├── gallery.scss
├── grid.scss
└── loader.scss
```

Antes de ejecutar el script que hemos creado tenemos que añadir los comentarios que necesitemos dentro de los archivos SASS.

## loader.scss

```
1  ////
2  /// Ejercicio loader.scss
3  /// @author Alvaro Membrillo
4  /// @since 1.0
5  ////
6  $duration: 10s !global;
7
8  /// contiene parámetros necesarios para centrar el texto
9  @mixin reset {
10     margin: 0;
11     padding: 0;
12     text-align: center;
13 }
14
15 /// resetea el contador de la barra de carga
16 @mixin reset-load {
17     position: absolute;
18     content: "";
19     left: 0;
20     top: 0;
21 }
22
23 /// define las dimensiones de un elemento
24 /// @param {Number} $width [100%]
25 /// @param {Number} $height
26 @mixin size($width: 100%, $height: 50px) {
27     width: $width;
28     height: $height;
29 }
30
31 /// para centrar el texto de toda la página
32 * {
33     @include reset;
34 }
35
36 .loader {
37     display: block;
38     position: relative;
39     border: 2px solid #fff;
40     @include size($height: 50px);
41 }
```

## gallery.scss

```
1  ////
2  /// Ejercicio gallery.scss
3  /// @author Alvaro Membrillo
4  /// @since 1.0
5  ////
6
7  $MAX_COLUMNS: 4 !default;
8  $RESPONSIVE_COLUMNS: (
9     1024px: 3,
10     768px: 2,
11     480px: 1,
12 );
13
14 /// función que recibe un parámetro y ajusta la distribución de las columnas del grid
15 /// @param {Number} $n
16 /// @return {Number} valor que define la distribución entre columnas
17 @function grid_columns($n) {
18     @if ($n > 0) {
19         @return ((100 / $n)) * 1%;
20     } @else {
21         @return 0;
22     }
23 }
24
25 /// función mixin que redimensiona las imágenes de la galería según el parámetro que recibe
26 /// @param {Number} $n
27 @mixin resize($n) {
28     .gallery img {
29         width: grid_columns($n);
30     }
31 }
32
33 body {
34     margin: 0 auto;
35     max-width: 1024px;
36 }
```

## grid.scss

```
grid.scss u x
grid.scss > ...
43 // función mixin para calcular el tamaño de cada elemento segun sus características
44 // @param {Number} $col primer valor, el ancho con respecto al total
45 // @param {Number} $col segundo valor, el ancho total de la fila
46 @mixin generarSize($col, $of) {
47   .col-#{$col}-of-#{$of} {
48     @if ($col == 1) {
49       width: calc((100% - (#{$of} - 1) * #{$horizontal}) / #{$of});
50     } @else if ($col == 2) {
51       width: calc(
52         #{$col} *
53         ((100% - (#{$of} - #{$col} + 1) * #{$horizontal}) / #{$of}) +
54         (#{$col} - 1) *
55         #{$horizontal}
56       );
57     } @else if ($col > 2) {
58       width: calc(
59         #{$col} *
60         ((100% - (#{$of} - #{$col} + 2) * #{$horizontal}) / #{$of}) +
61         (#{$col} - 1) *
62         #{$horizontal}
63       );
64     }
65   }
66 }
67
68 // llamamos a la función varias veces para generar todas las reglas css necesarias
69 @include generarSize(1, 2);
70 @include generarSize(1, 3);
71 @include generarSize(2, 3);
72 @include generarSize(1, 4);
73 @include generarSize(2, 4);
74 @include generarSize(3, 4);
75 }
76
```

Ahora podemos ejecutar el script con el que vamos a generar la documentación de nuestros archivos **sass** mediante **sassdoc** en el directorio **sassdocs** (que se generará tras la ejecución):

## npm run sassdoc

```
alvaro@alvaro-Lenovo-Ideapad-330-15IK8:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$ npm run sassdoc
> 4.2_componentessass@1.0.0 sassdoc
> sassdoc scss -d sassdocs
```

## tree sassdocs/

```
alvaro@alvaro-Lenovo-Ideapad-330-15IK8:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$ tree sassdocs/
sassdocs/
├── assets
│   ├── css
│   │   └── main.css
│   ├── images
│   │   ├── favicon.png
│   │   ├── logo_full_compact.svg
│   │   ├── logo_full_inline.svg
│   │   ├── logo_light_compact.svg
│   │   └── logo_light_inline.svg
│   └── js
│       ├── main.js
│       ├── main.min.js
│       ├── search.js
│       ├── sidebar.js
│       └── vendor
│           ├── fuse.min.js
│           ├── jquery.min.js
│           └── prism.min.js
└── index.html

5 directories, 14 files
```

# WEBPACK

Tenemos que crear un directorio **assets** donde almacenaremos los archivos **scss** y **js** (a su vez irán dentro de la carpeta **src**):

```
alvaro@alvaro-Lenovo-ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$ mkdir assets
alvaro@alvaro-Lenovo-ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$ cd assets/
alvaro@alvaro-Lenovo-ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS/assets$ mkdir src
alvaro@alvaro-Lenovo-ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS/assets$ cd src/
alvaro@alvaro-Lenovo-ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS/assets/src$ mkdir js
alvaro@alvaro-Lenovo-ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS/assets/src$ mkdir scss

alvaro@alvaro-Lenovo-ideapad-330-15IKB:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$ tree assets/
assets/
├── src
│   ├── js
│   └── scss
```

Movemos todos los archivos **js** y **scss** a sus directorios correspondientes (los que acabamos de crear).

Creamos un archivo llamado **styles.scss** en **assets/src/scss/**:

## code styles.scss

```
styles.scss x
home > alvaro > Escritorio > DIW_20212022 > UD4_preprocesadoresCSS > 4.2_ComponentesSASS > assets > src > scss > styles.scss
1 @import "../../node_modules/bootstrap/scss/bootstrap";
```

Y creamos otro archivo llamado **scripts.js** en **assets/src/js/**:

## code scripts.js

```
scripts.js x
home > alvaro > Escritorio > DIW_20212022 > UD4_preprocesadoresCSS > 4.2_ComponentesSASS > assets > src > js > scripts.js
1 import "../../node_modules/bootstrap/dist/js/bootstrap.bundle.min.js";
```

Ahora podemos comenzar a configurar NPM y WEBPACK en nuestro proyecto, para ello vamos a añadir los scripts necesarios dentro de nuestro fichero **package.json**.

Antes tenemos que crear el archivo **webpack.common.js** en la raíz de nuestro proyecto:

## code webpack.common.js (contiene la configuración general)

```
webpack.common.js x
home > alvaro > Escritorio > DIW_20212022 > UD4_preprocesadoresCSS > 4.2_ComponentesSASS > webpack.common.js > ...
1 const path = require("path");
2 const { CleanWebpackPlugin } = require("clean-webpack-plugin");
3 module.exports = {
4   entry: {
5     scripts: "./assets/src/js/scripts.js",
6   },
7   plugins: [new CleanWebpackPlugin()],
8   output: {
9     path: path.resolve(__dirname, "assets/dist/js/"),
10    filename: "[name].js",
11  },
12};
```

**code webpack.dev.js** (contiene la configuración del módulo de desarrollo)

```
webpack.dev.js x
home > alvaro > Escritorio > DIW_20212022 > UD4_preprocesadoresCSS > 4.2_ComponentesSASS > webpack.dev.js > ...
1  const { merge } = require("webpack-merge");
2  const common = require("./webpack.common.js");
3  module.exports = merge(common, {
4      mode: "development",
5      watch: true,
6      watchOptions: {
7          ignored: /node_modules/,
8      },
9  });
```

**code webpack.prod.js** (contiene la configuración del módulo de producción)

```
webpack.prod.js x
home > alvaro > Escritorio > DIW_20212022 > UD4_preprocesadoresCSS > 4.2_ComponentesSASS > webpack.prod.js > ...
1  const { merge } = require("webpack-merge");
2  const common = require("./webpack.common.js");
3  module.exports = merge(common, {
4      mode: "production",
5  });
```

Ahora creamos los scripts que ejecutaremos para el modo de desarrollo y el de producción (modificando el archivo **package.json**):

```
"scripts": {
  "test": "echo \"Error: no test specified\" && exit 1",
  "scss": "node-sass --watch scss -o css",
  "sassdoc": "sassdoc scss -d sassdocs",
  "build": "npm run css && webpack --config webpack.prod.js",
  "css-compile": "node-sass --include-path node_modules --output-style compressed --source-map true --source-map-content true --precision 6 assets/src/scss -o assets/dist/css/",
  "css-prefix": "postcss --replace assets/dist/css/styles.css --use autoprefixer --map",
  "css": "npm run css-compile && npm run css-prefix",
  "watch": "nodemon -e scss -x \"npm run css && webpack --config webpack.dev.js\""
},
```

Ahora ejecutamos el script npm “build” desde la carpeta raíz del proyecto:

**npm run build**

```
alvaro@alvaro-Lenovo-Ideapad-330-15IK8:~/Escritorio/DIW_20212022/UD4_preprocesadoresCSS/4.2_ComponentesSASS$ npm run build
> 4.2_componentessass@1.0.0 build
> npm run css && webpack --config webpack.prod.js

> 4.2_componentessass@1.0.0 css
> npm run css-compile && npm run css-prefix

> 4.2_componentessass@1.0.0 css-compile
> node-sass --include-path node_modules --output-style compressed --source-map true --source-map-content true --precision 6 assets/src/scss -o assets/dist/css/
An output directory must be specified when compiling a directory
```