

Esta herramienta llamada Webpack es muy útil para automatizar los procesos a la hora de desarrollar este tipo de proyectos.

Se utiliza en un entorno donde este en uso el sistema npm para eso pondremos en el terminal

```
>> npm init
```

Este comando nos crea un fichero llamado package.json este documento nos encontraremos todas las dependencias necesarias para usar el proyecto, también podemos generar scripts.

```
"scripts": {  
  "build": "npm run css && webpack --config webpack.prod.js ",  
  "css-compile": "node-sass --include-path node_modules --output-style compressed --source-map true --so",  
  "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\"",  
},  
"author": "Pedro Larrubia",  
"license": "ISC",
```

Estos scripts podrán ser utilizados con el siguiente comando

```
>> npm run [el nombre del script que hayamos configurado]
```

Para utilizar webpack hemos tenido que crear los siguientes ficheros .js

```
webpack.common.js  
webpack.dev.js  
webpack.prod.js
```

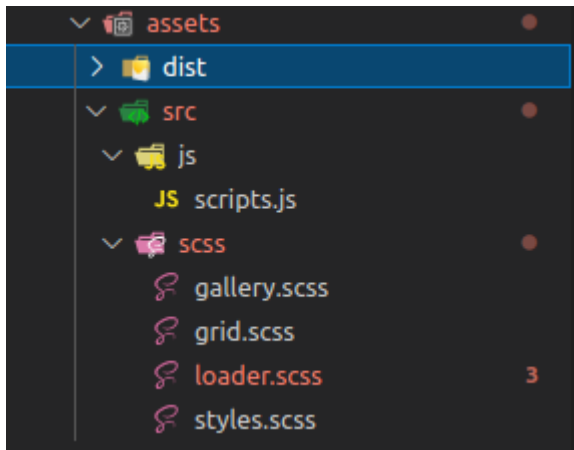
Y utilizaremos este comando

```
$ npm install bootstrap @popperjs/core autoprefixer clean-webpack-plugin node-sass  
nodemon postcss-cli webpack webpack-cli webpack-merge --save-dev
```

Es muy importante el --save-dev que este actualizara el fichero package.json con las siguientes dependencias:

```
"devDependencies": {  
  "@popperjs/core": "^2.9.2",  
  "autoprefixer": "^10.2.5",  
  "bootstrap": "^5.0.1",  
  "clean-webpack-plugin": "^4.0.0-alpha.0",  
  "node-sass": "^6.0.0",  
  "nodemon": "^2.0.7",  
  "postcss-cli": "^8.3.1",  
  "webpack": "^5.37.1",  
  "webpack-cli": "^4.7.0",  
  "webpack-merge": "^5.7.3"  
},
```

Ahora crearemos la siguiente estructura:



assets/src/scss , en esta localización tendremos los scss que después de utilizar el comando `>> npm run build` , serán compilados a css en la carpeta dist.