

1. Proyecto

<https://github.com/elenaminyaeva/Ejercicio-Sass.git>

- instalar node-sass

```
[root@ubuntu:/home/usuario/proyecto_sass# npm install node-sass
npm WARN prefer global node-gyp@3.8.0 should be installed with -g

> node-sass@4.13.0 install /home/usuario/proyecto_sass/node_modules/node-sass
> node scripts/install.js

Downloading binary from https://github.com/sass/node-sass/releases/download/v4.13.0/linux-x64-57_binding.node
Download complete..] - :
Binary saved to /home/usuario/proyecto_sass/node_modules/node-sass/vendor/linux-x64-57/binding.node

> node-sass@4.13.0 postinstall /home/usuario/proyecto_sass/node_modules/node-sass
> node scripts/build.js

Binary found at /home/usuario/proyecto_sass/node_modules/node-sass/vendor/linux-x64-57/binding.node
Testing binary
Binary is fine
usuario@1.0.0 /home/usuario/proyecto_sass
├─┬ gulp-minify-css@1.2.4
│   └─┬ clean-css@3.4.28
│       └─┬ source-map@0.4.4
│           └─┬ node-sass@4.13.0 extraneous
```

- crear carpeta “sass” con .scss fichero dentro
- editar el fichero package.json

```
GNU nano 2.9.3 package.json

{
  "name": "usuario",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "scss": "node-sass --watch scss -o css"
  },
  "author": "",
  "license": "ISC",
  "devDependencies": {
    "gulp": "^4.0.2",
    "gulp-clean-css": "^4.2.0",
    "gulp-minify-css": "^1.2.4"
  }
}
```

- crear carpeta “css”
- ejecutar el comando *npm run scss*
- editar el fichero .scss

```
[root@ubuntu:/home/usuario/proyecto_sass/scss# npm run scss ]

> usuario@1.0.0 scss /home/usuario/proyecto_sass
> node-sass --watch scss -o css

=> changed: /home/usuario/proyecto_sass/scss/styles.scss
Rendering Complete, saving .css file...
Wrote CSS to /home/usuario/proyecto_sass/css/styles.css
```

resultado:

```
[root@ubuntu:/home/usuario/proyecto_sass/css# ls ]
styles.css
[root@ubuntu:/home/usuario/proyecto_sass/css# cat styles.css ]
body {
  background-color: red; }
root@ubuntu:/home/usuario/proyecto_sass/css#
```

Se debe tener en cuenta las siguientes consideraciones:

- Deben definirse módulos, utilizando al menos los parciales:
 - `_variables.scss`
 - `_mixins.scss`:
 - `_base.scss`: Estilos comunes, aplicables a todas las páginas del sitio.

```
[usuario@ubuntu:~/proyecto_sass/scss$ ls -l
total 12
-rw-r--r-- 1 root root 288 Oct 27 12:01 mixin.scss
-rw-r--r-- 1 root root 2627 Oct 26 16:18 npm-debug.log
-rw-r--r-- 1 root root 1184 Oct 27 12:28 styles.scss
```

- Además se deben definir:
 - encabezados, `h1`

index.html

```
<html>
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Mockup Landing Page</title>
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <!-- PLACE ALL OF YOUR PAGE CONTENT BELOW HERE -->
    <h1 class='header'>Tecnologías Web: Cliente</h1>
```

styles.scss

```
$color: #08088A;

h1 {
  color: $color;
}
```

- centrar contenido

styles.scss

```
$background: #b9c9fe;
$text-align: center;

td {
  padding: $padding;
  border: $border;
  background: $background;
  text-align: $text-align;
}
```

- estilos del form

mixin.scss

```
@mixin large-text {  
  font: {  
    family: Arial;  
    size: 20px;  
    weight: bold;  
  }  
  color: #380B61;  
}
```

- Anidar los estilos de los botones de form, para que se apliquen estilos específicos en el hover de ellos.

mixin.scss

```
@mixin button-bg {  
  background: $background;  
  &:hover {  
background:darken($background,8%);  
transition: all 0.3s ease;  
  }  
}
```

styles.scss

```
.class1 {  
@include button-bg;  
background: $background;  
}  
  
$background: #F5A9F2;  
  
.class2 {  
@include button-bg;  
background: $background;
```

index.html

```

<tbody>
  <tr>
    <th class="class1">
      Logo de empresa<br><span class='text'>Nombre de la empresa</span><br>Be$
    </th>
  </tr>
  <tr>
    <td class="class2">Ejemplos de productos</td>
  </tr>
  <tr>
    <td class="class3">Formulario llamada a la acción</td>
  </tr>
  <tr>
    <td class="class4"><span class='header'>Garantías</span></td>
  </tr>
</tbody>

```

- Anidar los estilos de los elementos dentro del form, para que se resalten cuando tengan el foco.

mixin.scss

```

@mixin button-bg {
  background: $background;
  &:hover {
    background: darken($background, 8%);
    transition: all 0.3s ease;
  }
  &:active {
    background: darken($background, 35%);
  }
}

```

- Utilizar el mismo estilo en los encabezados de la parte de testimonio que en la de garantías, pero con un cambio en el color de texto.

index.html

```

<!-- PLACE ALL OF YOUR PAGE CONTENT BELOW HERE -->
<h1 class='header'>Tecnologías Web: Cliente</h1>

<tr>
  <td class="class4"><span class='header'>Garantías</span></td>
</tr>

```

styles.scss

```

$font-family: times;

.header {
  font-family: $font-family;
}

```

- Utilizar algún estilo de materialize sin incluir todos los estilos. Reutiliza sólo algún componente.

mixin.scss

```
@mixin large-text {
  font: {
    family: Arial;
    size: 20px;
    weight: bold;
  }
  color: #380B61;
}
```

- Lanzar el proyecto

```
[usuario@ubuntu:~/proyecto_sass$ browser-sync start --server --files "css/*.css"]
```

```
[Browsersync] Access URLs:
```

```
-----
Local: http://localhost:3000
External: http://10.6.129.134:3000
-----
```

```
UI: http://localhost:3001
UI External: http://localhost:3001
-----
```

```
[Browsersync] Serving files from: ./
```

```
[Browsersync] Watching files...
```

```
[Browsersync] Couldn't open browser (if you are using BrowserSync in a headless
environment, you might want to set the open option to false)
```

Tecnologías Web: Cliente

Logo de empresa Nombre de la empresa Beneficios del producto
Ejemplos de productos
Formulario llamada a la acción
Garantías
Testimonios, clientes relevantes
Contacto