

## Práctica 1. Instalación y primeros pasos

Nombre	Alejandro	Curso	2º DAW
Apellidos	Rodriguez Mena	Fecha	12/03/2023

### ¿Qué es Sass?

Para aquellos que aún no lo conocen, SASS es un lenguaje de hojas de estilo que permite escribir estilos de manera más eficiente y semántica. Es una herramienta que genera archivos CSS optimizados, y además, incorpora funcionalidades propias de los lenguajes de programación, como el uso de variables, la creación de funciones y la estructuración del código en múltiples archivos.

Aunque SASS no es una tecnología nueva (fue creado en 2007), ha ganado popularidad con el tiempo y es una herramienta indispensable para construir proyectos web de mediana a gran escala. Con su uso, se pueden desarrollar estilos más eficientes y mantener el código organizado, lo que facilita el mantenimiento a largo plazo.



Escrito en Ruby, consta de un compilador encargado de traducir nuestros ficheros en sintaxis Sass a CSS. A la hora de escribir nuestras hojas de estilo podremos hacer uso de dos sintaxis: **SCSS** (Sassy CSS) u **Original Sass**. El primero pretende asemejar la sintaxis Sass a la sintaxis CSS estándar, haciendo que cualquier fichero CSS sea sintácticamente válido a nivel de Sass.

Por otra parte, Original SASS es la sintaxis original de Sass que elimina muchos de los elementos CSS que se consideran incensarios como las llaves o los puntos y coma, obligando a estructurar nuestras reglas de cierta forma para que la semántica que aportan dichos elementos quede implícita en la estructura.

Para que nos quede más claro podemos ver un ejemplo de ambas sintaxis

<pre>/* SCSS */p {   width:   500px;   color:   #000; }</pre>	<pre>/* Original SASS */ p width: 500px color: #000</pre>
---	---

Como podemos apreciar, SCSS es idéntico a CSS, pero Original Sass simplifica la sintaxis a base de obligar a seguir una estructura más estricta.

Aunque la sintaxis original queda más limpia y se comprende mejor una vez que te acostumbras, normalmente

se recomienda hacer uso de SCSS, especialmente en los inicios pues se tiene la ventaja de que realmente es CSS estándar ampliado con las mejoras ofrecidas por SASS.

### ***Instalación de SASS***

Para poder utilizar SASS necesitamos disponer del compilador que nos permita convertir nuestros ficheros en sintaxis SCSS a CSS, y para instalar dicho compilador debemos instalar previamente Ruby. Los usuarios de Windows podrán hacerlo fácilmente a través de <http://rubyinstaller.org>, los usuarios de Mac OS X ya lo llevan instalado de serie, y los usuarios de Linux podrán instalarlo con su gestor de paquetes preferido.

Una vez instalado Ruby y su gestor de paquetes Ruby Gems, instalar Sass será tan sencillo como introducir esta línea en nuestra consola de comandos:

```
$ gem install sass
```

Con node.js es tan fácil como hacer

```
$ npm -g install sass
```

Sencillo, ¿verdad? Para comprobar que funciona, podemos coger cualquier fichero CSS que tengamos y cambiar su extensión a SCSS para luego pasárselo como parámetro al comando sass:

```
$ mv test.css test.scss  
$ sass test.scss
```

```
p {  
width: 500px;  
color: #000;  
}
```

Como hemos comentado anteriormente, la sintaxis SCSS es una ampliación de la sintaxis CSS, por lo que cualquier contenido CSS también es contenido SCSS válido.

En el ejemplo anterior, simplemente he formateado la salida porque no hemos hecho uso de ningún uso específico de Sass. Si queremos que el contenido compilado se guarde en un fichero, se lo indicaremos como segundo parámetro:

```
>sass test.scss test.css
```

Invocar este comando cada vez que modifiquemos el fichero. scss original para generar el código CSS es excesivamente tedioso. Por ese motivo, podemos hacer uso de la opción – watch del compilador, la cual monitorizará los cambios del fichero original y lo compilará automáticamente al fichero destino en caso de actualización:

```
>sass --watch test.scss:test.css
```

De esta forma, podremos modificar el fichero test.scss desde nuestro editor preferido y cada vez que guardemos el fichero se compilará automáticamente a CSS.

Como veremos en este tutorial, una de las recomendaciones a la hora de trabajar con Sass es segmentar nuestra maqueta CSS en varios ficheros, de forma que podamos hacerla lo más modular posible.

Si vamos a hacer uso de un conjunto de ficheros .scss, en vez de monitorizar fichero a fichero será preferible monitorizar el directorio que los contiene.

Normalmente se recomienda tener un directorio con los ficheros Sass y otro directorio con los ficheros compilados a CSS, para luego hacer uso de la opción – watch y así monitorizar todos los cambios que se produzcan en cualquiera de los ficheros situados en el directorio SCSS:

```
>sass --watch stylesheets/scss:stylesheets/css
```

## **Tipos de archivos en Sass Maestros y Parciales**

En Sass, existen dos tipos de archivos:

### **Maestros:**

Los maestros, son los archivos que Sass tomará para realizar el proceso de traducción o conversión de Sass a CSS. En ellos, se agregan estilos e importamos todos los parciales (partials) que necesitemos sean llevados a la hoja de estilo que se utilizará en el Sitio Web o Aplicación que estamos trabajando.

### **Parciales (Partials):**

Sass nos brinda la posibilidad de crear archivos con pequeños fragmentos de estilos, y estos son procesados sólo cuando los incluimos a los archivos maestros. Es un excelente recurso que nos permite separar y organizar nuestros estilos en pequeños fragmentos, consiguiendo una mayor legibilidad y facilitando el proceso de depuración.

Un parcial es un simple archivo de Sass nombrado al comienzo con un piso (underscore), de esta manera:

parcial.scss. El piso (underscore) le dirá a Sass que el archivo es únicamente un parcial y que no debe ser procesado por sí solo. Los parciales en Sass son usados con la directiva @import.

### **Recomendación:**

No tengan miedo a separar sus hojas de estilos en pequeños fragmentos, todo lo contrario, es importante hacerlo, a medida que el proyecto en el que estén trabajando vaya tomando forma y se haga cada vez más grande, se darán cuenta que utilizar este recurso es la tarea más maravillosa que han podido hacer. Divide y vencerás ;)

Me gusta mucho crear un archivo por componente y que mis parciales no tengan más de 150 líneas. Es una regla que sigo a diario y me ha dado muy buenos resultados.

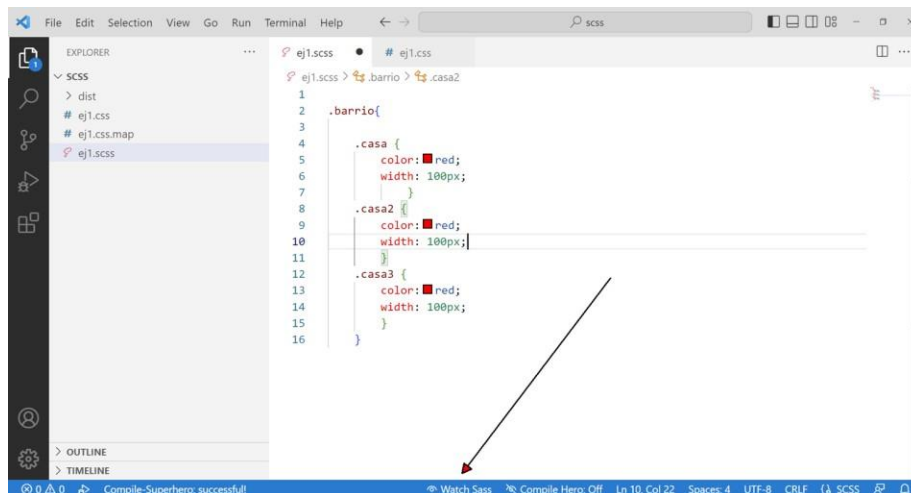
## **Herramientas Visuales**

Si te da alergia la línea de comandos, o simplemente te atraen más las herramientas de escritorio, puedes hacer uso de herramientas visuales que también realizan la compilación automática de ficheros.

Existen varias alternativas multiplataforma de pago como [CodeKit](#) o [Compass App](#). Yo os recomiendo [Scout](#), que es multiplataforma y además es gratuita.

Lo único que tendremos que crear un proyecto indicándole donde tenemos situados los fuentes y posteriormente indicarle el directorio donde alojaremos los ficheros fuente .scss y el directorio donde compilar. Una vez configurado, en la sección de «log» podremos ver la salida de las conversiones automáticas que realizará el programa.

Para compilar saas en visual Studio code debemos usar la extensión **Compile-Superhero** y pulsar en Watch sass



## Practica 1 a Realizar

1. Instala Node.js si no lo tienes
2. Instala SaaS


```
C:\Users\AlejandroRM>cd C:\Users\AlejandroRM\Desktop\Mis ejercicios\Proyectos programacion\
o de aplicaciones web\tema5\05-03.Funciones e importacion
added 17 packages, and audited 18 packages in 3s

2 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

55

Extension: Sass/Less/Stylus/Pug/Jade/Typescript/Javascript Compile Hero Pro X




## Sass/Less/Stylus/Pug/Jade/Typescript/Javascript

Eno Yao | 682,040 | ★★★★★ (46)

Easy to compile ts, tsx, scss, less, stylus, jade, pug and es6+ on save without using a build

[Disable](#) [Uninstall](#) ⚙️

This extension is enabled globally.



## Sass

v1.8.23

Syler | 1,759,375 | ★★★★★ (8)

Indented Sass syntax Highlighting, Autocomplete & Formatter

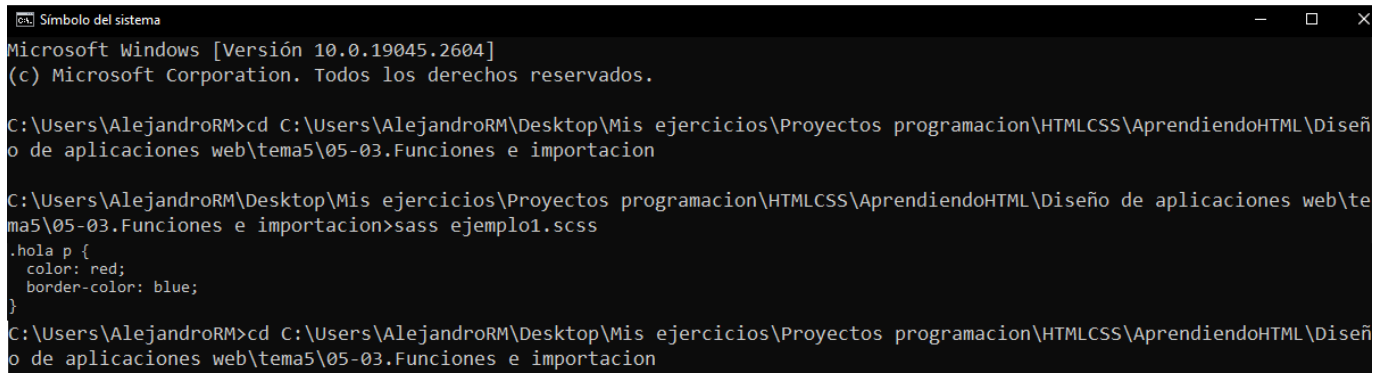
[Disable](#) [Uninstall](#) ⚙️

This extension is enabled globally.

[DETAILS](#) [FEATURE CONTRIBUTIONS](#) [CHANGELOG](#) [RUNTIME STATUS](#)



### 3. Ejecuta tu primera transpilación css con la herramienta SaaS



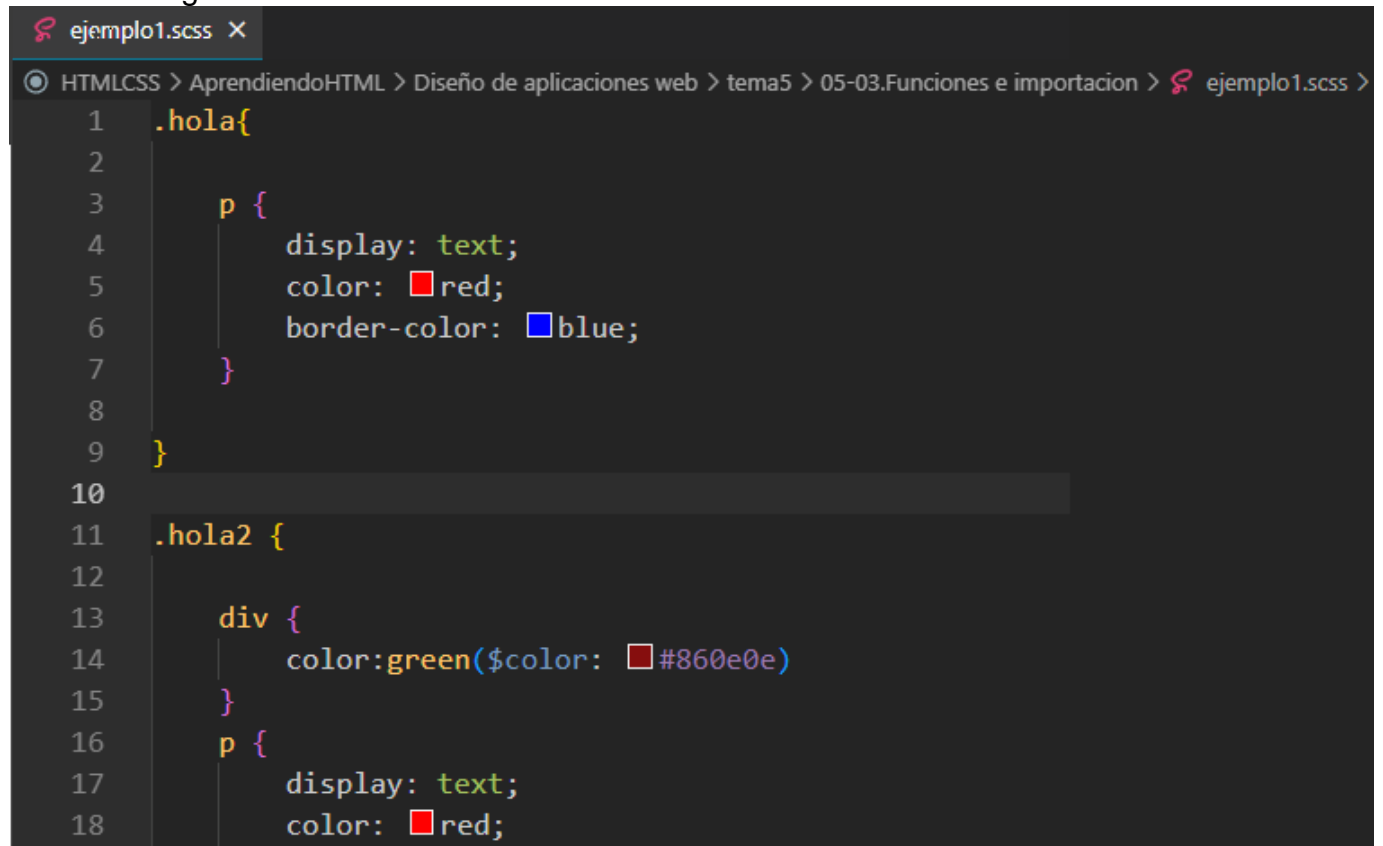
```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.2604]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\AlejandroRM>cd C:\Users\AlejandroRM\Desktop\Mis ejercicios\Proyectos programacion\HTMLCSS\AprendiendoHTML\Diseño de aplicaciones web\tema5\05-03.Funciones e importacion

C:\Users\AlejandroRM\Desktop\Mis ejercicios\Proyectos programacion\HTMLCSS\AprendiendoHTML\Diseño de aplicaciones web\tema5\05-03.Funciones e importacion>sass ejemplo1.scss
.hola p {
  color: red;
  border-color: blue;
}

C:\Users\AlejandroRM>cd C:\Users\AlejandroRM\Desktop\Mis ejercicios\Proyectos programacion\HTMLCSS\AprendiendoHTML\Diseño de aplicaciones web\tema5\05-03.Funciones e importacion
```

#### 4. Configura el Visual Studio Code



```
1 .hola{
2
3     p {
4         display: text;
5         color: red;
6         border-color: blue;
7     }
8
9 }
10
11 .hola2 {
12
13     div {
14         color: green($color: #860e0e)
15     }
16     p {
17         display: text;
18         color: red;
```

#### 5. Ejecuta tu primera transpilación a css con Visual.

