SASS

2. Requisitos

- HTML
- CSS (Selectores / Maquetación)
- Conceptos Básicos de Programación
- BootStrap (Idealmente)

3. Qué vamos a ver

- Sass (Preprocesador CSS).
- Gulp (Workflow de Trabajo).

ÍNDICE

El Frontend

Se enfoca en el usuario, en todo con lo que podemos interactuar y lo que vemos mientras navegamos. Así como en una primera cita, nuestra web busca causar una buena impresión y agradar al usuario, para lo cual utiliza HTML, CSS y JAVASCRIPT

Un preprocesador de CSS

Es una herramienta que nos permite escribir pseudo-código **CSS** que luego será convertido a **CSS** real. Ese pseudo-código se conforma de variables, condiciones, bucles o funciones. Podríamos decir que tenemos un lenguaje de programación que genera **CSS**

CSS

- Cambios costosos
- Desestructurado
- Redundante

PREPROCESADOR

- Variables
- Automatización
- Estructuras de control
- Optimización
- Modularidad
- Reusable

¿POR QUÉ UN PREPROCESADOR CSS ?

3. Web Component (Classic Definition)

Es la **unión de HTML y CSS** que, en unión de JavaScript si es necesario, nos proporciona una **funcionalidad definida**, y nos permite **reusar** dicha funcionalidad entre diferentes proyectos.

4. Build Tools (Task Runners)

Build Tool es una **herramienta** que nos permite **"generar" aplicaciones** a partir de un código fuente y que realiza de manera automática todas las **tareas** necesarias para ello.

En el mund Front-End, donde no se genera un ejecutable, también se llaman **Task Runners.**

https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/gulp-vs-grunt-que-diferencia-a-estos-task-runners/

1. **ENTORNO** (una propuesta…)

2. Extensiones VSCODE

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

3. Instalación NPM

NPM (Node Package manager) me va permitir instalar Gulp y las extensiones necesarias para las tareas de Gulp que necesitemos para mi workflow de desarrollo Front - End.

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

> nodejs --version / > npm --version

1. Sass

Conoceremos...

- Instalación
- Sus ventajas e inconvenientes
- Características y estructuras
- Ejercicios prácticos

3. Tema BootStrap

Crearemos un tema BootStrap

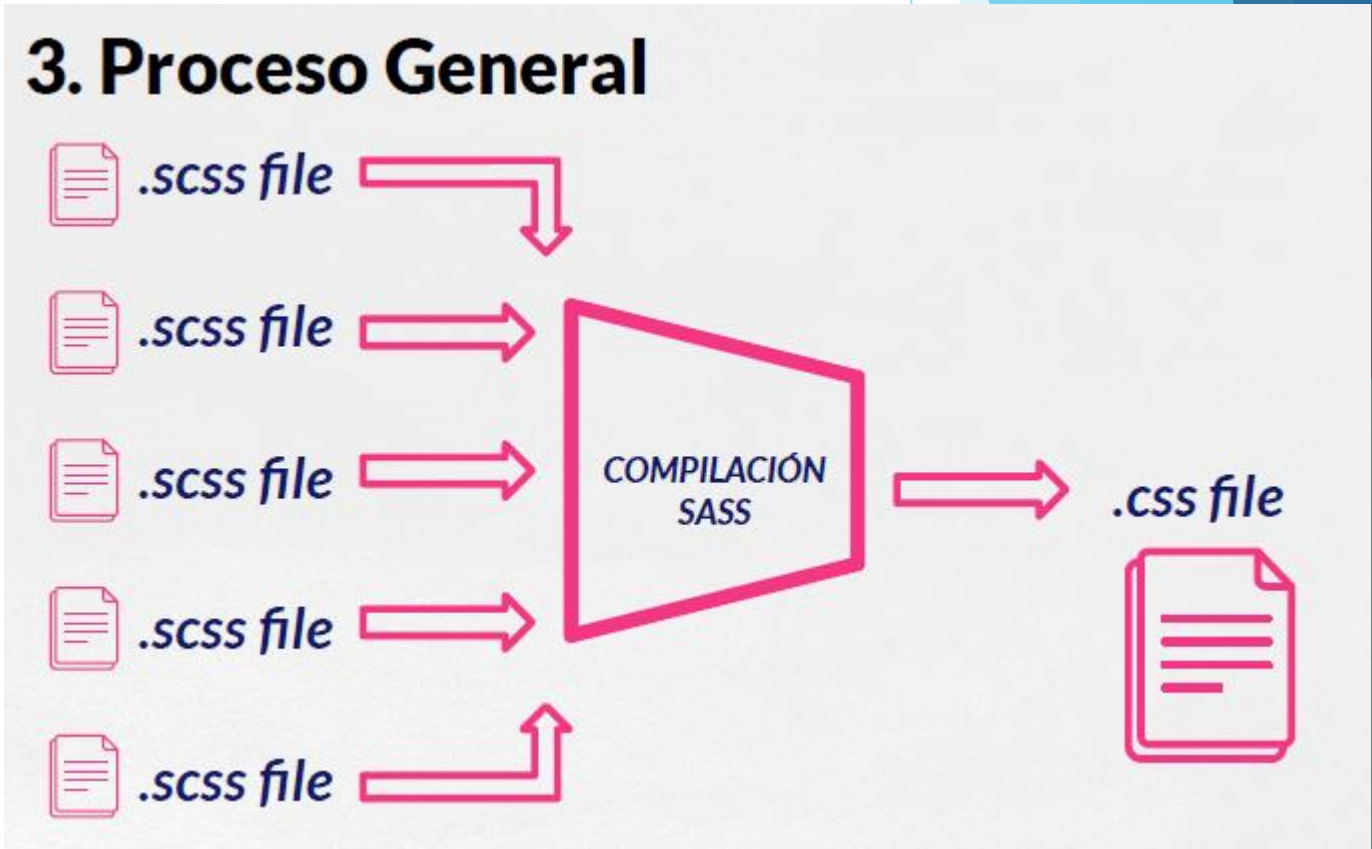
- Descargaremos BootStrap
- Conoceremos la organización de un proyecto real
- Modificaremos BootStrap para crear nuestro tema adaptado
- Crearemos un workflow para ello

Instalación desde Gestor

- **npm install -g sass** (Nodejs Instalado) – de forma global
- **choco install sass** (Windows Chocolatery)
- **brew install sass/sass/sass** (Mac Homebrew)

Para instalarlo en la carpeta de nuestro proyecto nos situamos en la misma y escribimos:

npm install sass



2. Elementos Básicos

- **Variables.**
- **Comentarios.**
- **Listas y Mapas.**
- **Interpolación.**
- **Anidamiento.**

EN RELACIÓN A LAS VARIABLES

DEFINICIÓN	USO
$nombre: expresión;	*selector_css {*
	...
$nombre: valor;	*regla_css: $nombre...*
	...
	}

VALORES POR DEFECTO

```
// Código del usuario, previo al uso (o importación)
// de mi librería.
$gris: #ddd;
$color-letra-pie: $gris;

// Código de mi librería, respeto el anterior ya
// que está previamente definido.
$gris: #eee !default;
```

ÁMBITO DE LAS VARIABLES

//Variable global fuera de todo bloque

$logo-width: 50%;

.header {

//Variable local

$header-width: 50%;

}

- Comentarios de **una sóla línea**

// ESTO ES UN COMENTARIO

- Comentarios **multilínea**

/

ESTO TAMBIÉN ES UN COMENTARIO

/

LISTAS Y MAPAS

LISTAS

```
$variable_lista: (v1, v2, v3);

$sizes: (40px, 80px, 160px);

$sizes: (
        40px,
        80px,
        160px,
);
```

MAPAS

```
$nombre_mapa: (

"clave1": valor1,

...

"claven": valorn

);

$breakpoint: (
        'pequeño' : 576px,
        'medio' : 768px,
        'grande' : 992px

);
```

Sass: sass:map (sass-lang.com)

Sass: sass:list (sass-lang.com)

EJEMPLOS

```
// Interpolación en selectores
$button-type: "error";
$btn-color : #f00;
.btn-#{$button-type}{
    background-color: $btn-color;
}

//Interpolación en el uso de funciones
$fondo : "images/fondos/default.png";
.container {
background-image: url('#{$fondo}');
}
```

(más conciso / mejor organizado)

CSS

```
nav {...}
nav li {...}
navi li a {...}
```

SASS

```
nav { ...
    li {...
        a {....}
    }
}
```

2. Tipos de Compilación

- Simple
- Múltiple (varios ficheros)
- Expandida (por defecto)
- Comprimida
- Vigilando los cambios

De esta manera se genera el fichero .css

```
sass ./scss/estilos.scss ./css/estilos.css
```

De esta manera el fichero se compila automáticamente
Con cada cambio que se haga en el ficheros sass.

```
sass --watch ./scss/estilos.scss ./css/estilos.css
```

//SIMPLE

sass file.scss output_file.scss

//MÚLTIPLE

sass file1.scss:output1.css ... fileN.scss:outputN.css

// EXPANDIDA (1 SELECTOR - 1 LÍNEA EN SALIDA - POR DEFECTO)

sass --style = expanded file.scss output_file.scss

//COMPRIMIDA (QUITA LA MAYOR CANTIDAD DE CARACTERES POSIBLES)

```
sass --style = compressed file.scss output_file.scss
```

// VIGILANDO LOS CAMBIOS Y ACTUALIZANDO FICHEROS

```
sass --watch file.scss output_file.scss
```