

Metodologías para la organización y el preprocesador Sass

Metodologías para la organización y
modularización de estilos

Describir las características del proceso de diseño e implementación de un producto digital, distinguiendo el rol del desarrollador front-end y las buenas prácticas para la organización de estilos

- Unidad 1:
Metodologías para la organización y el preprocesador Sass
- Unidad 2:
El modelo de cajas y el Layout
- Unidad 3:
Utilizando Bootstrap como Framework CSS



Te encuentras aquí



¿Qué aprenderás en esta sesión?

- *Describe el rol y la importancia de las metodologías para la organización y modularización de estilos en una página web*
- *Describe el rol y la importancia de un preprocesador para el desarrollo front.*

¿Qué se te viene a la
mente si hablamos sobre
metodologías de CSS?



/* Metodologías CSS */

¿Qué es una metodología CSS?

Las metodologías CSS son un conjunto de pautas o sistemas preestablecidos que nos ayudan a escribir y organizar nuestro código para que este sea **mantenible** y **escalable**.

Algunas de estas metodologías establecen separar el código de manera modular en lugar de tener todo el código en nuestro archivo `style.css`, otras sugieren mantener un estándar en cómo nombramos las clases de nuestros elementos.



Ventajas de utilizar una metodología

Los beneficios que aporta el uso de alguna metodología CSS no es muy notorio en proyectos pequeños, pero sus ventajas se hacen notar a la hora de desarrollar proyectos más grandes, especialmente cuando tenemos una gran cantidad de archivos CSS.

Las ventajas más destacables que nos aportan las metodologías CSS son:

- Crear código CSS **intuitivo, legible y descriptivo**.
- La creación de componentes **reutilizables**.
- Nos permite tener un código **mantenible** en el tiempo.
- Crear código **escalable**.

/* Metodologías más utilizadas */

Metodologías más utilizadas

Como mencionamos anteriormente, existen diversas metodologías que podemos seguir para desarrollar nuestros proyectos, las más utilizadas son:

- BEM
- OOCSS
- SMACSS



Por ahora, veremos una breve descripción de estas 3 metodologías y las conoceremos en profundidad en las próximas sesiones.

BEM

La metodología **Bloque, Elemento y Modificador** (BEM) se basa en dividir la estructura del sitio web en:

- **Bloques:** Los cuales pueden ser las diferentes secciones del sitio como el menú de navegación, una *sección de noticias*, pie de página, entre otros.
- **Elementos:** Cada bloque contiene ciertos elementos, en el caso del bloque de *sección de noticias*, los elementos pueden ser una imagen, un título y una descripción.
- **Modificadores:** A su vez, cada elemento dispone de modificadores, los cuales se utilizan principalmente para cambiar la apariencia o estado del elemento (hover, fixed, active, disabled, color, etc.)

BEM

Sintaxis

Siguiendo el ejemplo de una sección de noticias, la sintaxis quedaría de la siguiente manera:

HTML

```
1 <body>
2   <div class="noticia">
3     <img class="noticia_imagen"></img>
4     <h5 class="noticia_titulo"></h5>
5     <p class="noticia_descripcion"></p>
6     <p class="noticia_descripcion--secondary"></p>
7   </div>
8 </body>
```

{desafío}
latam_

CSS

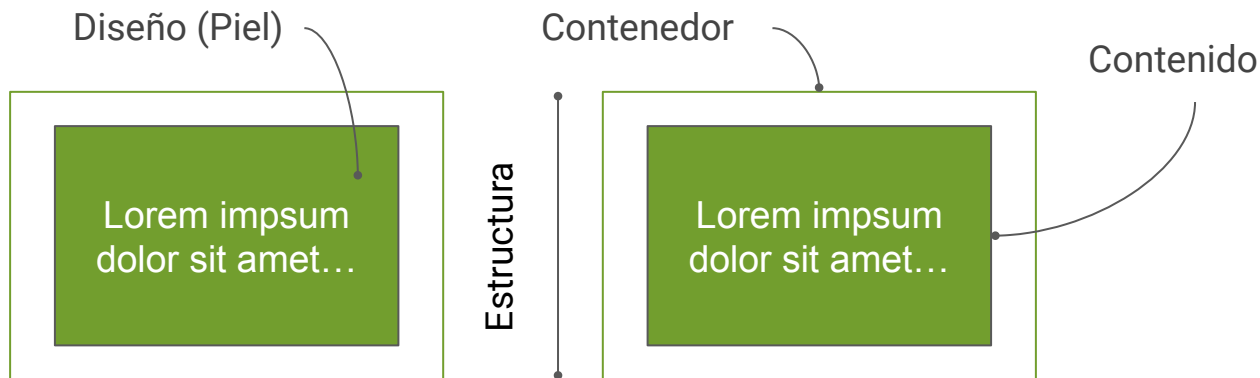
```
1 .noticia {
2   /* Estilos del bloque .noticia */
3 }
4
5 .noticia_imagen {
6   /* Estilos del elemento imagen */
7 }
8
9 .noticia_titulo {
10  /* Estilos del elemento titulo */
11 }
12
13 .noticia_descripcion--secondary{
14   color: blue;
15 }
```

OOCSS

El **CSS orientado a objetos** es una metodología para escribir CSS (y HTML) que nos permite separar el **contenedor** y el **contenido** en “objetos” CSS.

Esta metodología se basa en 2 principios:

- Separar la **estructura** de la **piel** (Características visuales).
- Separación del **contenedor** de su **contenido**.



OOCSS

Separar la estructura de la piel

Propiedades como width, height, padding, margin pertenecen a la **estructura**, y propiedades relacionadas con el color, fuente, fondo o borde se relacionan con la **piel** (skin).

CSS antes

```
1  #button {
2    width: 200px;
3    height: 50px;
4    padding: 10px;
5    border: solid 1px #ffffff;
6    background: linear-gradient(#7ad9ff, #84ffe8);
7    box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;
8  }
9
10 #box {
11   width: 400px;
12   overflow: hidden;
13   border: solid 1px #000000;
14   background: linear-gradient(#7ad9ff, #84ffe8);
15   box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;
16 }
```

{desafío}
latam_

CSS después

```
1  .button {
2    width: 200px;
3    height: 50px;
4  }
5
6  .box {
7    width: 400px;
8    overflow: hidden;
9  }
10
11 .skin {
12   border: solid 1px #000000;
13   background: linear-gradient(#7ad9ff, #84ffe8);
14   box-shadow: rgba(0, 0, 0, .5) 2px 2px 5px;
15 }
```

OOCSS

Separar el contenedor del contenido

El segundo principio promueve la separación de los contenedores de su contenido, en este ejemplo vemos que dos botones diferentes tienen la misma estructura, pero diferentes estilos.



```
1 <main id="main">
2   <button type="button" class="btn general">Botón primario</button>
3   <button type="button" class="btn warning">Botón secundario</button>
4 </main>
```

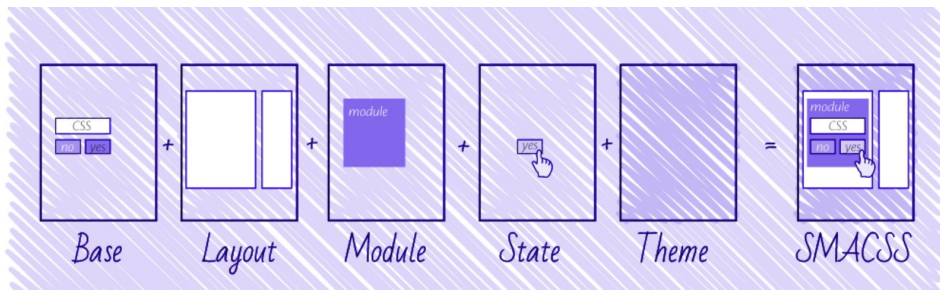
{desafío}
latam_



```
1 #main .btn {
2   width: 300px;
3   padding: 20px 10px;
4   margin: 10px;
5   font-size: 18px;
6   line-height: 1.5;
7 }
8
9 #main .general {
10  background-color: aquamarine;
11  color: black;
12 }
13
14 #main .warning {
15  background-color: azure;
16  color: blue;
17 }
```

SMACSS

Arquitectura en CSS Escalable y Modular



Fuente: [Medium](#).

Es una arquitectura que se centra en 5 categorías básicas.

- **Base:** Son los estilos por defectos de los elementos.
- **Layout:** Es la estructura general del sitio web.
- **Module:** Los elementos reutilizables de la web.
- **State:** Es el estado de cada módulo.
- **Theme:** Los colores y el estilo de los módulos.

¿Cuáles son las ventajas de trabajar con alguna de estas metodologías?



`/* Preprocesadores CSS */`

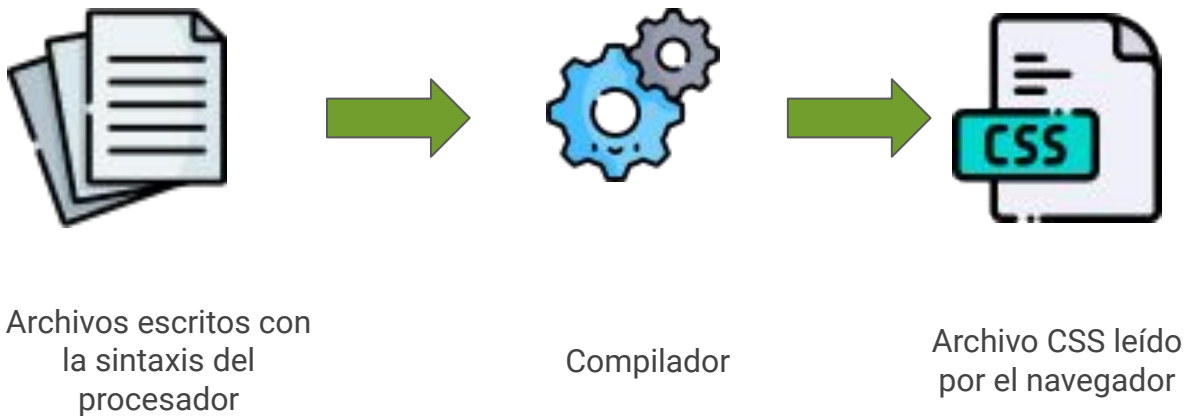
¿Qué es un preprocesador?

Los preprocesadores de CSS son una herramienta que permiten traducir (compilar) hojas de estilos escritas en una **sintaxis única de cada procesador**, y generar un archivo .CSS entendible por el navegador web.

Ofrece funciones avanzadas que no se encuentran en CSS por defecto, como:

- Variables.
- Anidación de selectores.
- Funciones.

¿Qué es un preprocesador?



Importancia del preprocesador en el desarrollo front

Principales ventajas

Los preprocesadores CSS facilitan la escritura y la mantenibilidad de las hojas de estilos, esto lo hace con funcionalidades adicionales como:

- Podemos **usar variables para aplicar el mismo color a varios elementos** sin la necesidad de repetir su código hexadecimal o RGB en cada uno de los elementos.
- Nos permite crear **funciones para reducir el código repetitivo**, permitiendo añadir varios estilos en distintos elementos, resultando en un código más limpio.
- Nos ayuda a **organizar mejor nuestro código**, separándolo en diversos archivos, convirtiendo nuestro código a nivel modular, permitiendo que nuestros proyectos sean más fáciles de mantener.

Preprocesadores más utilizados

Sass (Syntactically Awesome Style Sheets)

Sass, es considerado por muchos desarrolladores el más popular de los procesadores CSS y presenta dos sintaxis separadas:

- Sass
- SCSS



Fuente: [Sass](https://sass-lang.com/).



Durante las próximas sesiones profundizaremos en el uso de Sass

Sass

Sass y SCSS

La principal diferencia entre Sass y SCSS es que este último utiliza punto y coma (;) y corchetes ([]) tal como lo hace CSS.

SCSS

```
1  .button {  
2    background: cornflowerblue;  
3    border-radius: 5px;  
4    padding: 10px 20px;  
5  
6    &:hover {  
7      cursor: pointer;  
8    }  
9  
10   &:disabled {  
11     cursor: default;  
12     background: gray;  
13     pointer-events: none;  
14   }  
15 }
```

Sass

```
1  .button  
2    background: cornflowerblue;  
3    border-radius: 5px;  
4    padding: 10px 20px;  
5  
6    &:hover  
7      cursor: pointer;  
8  
9    &:disabled  
10     cursor: default;  
11     background: gray;  
12     pointer-events: none;
```

Preprocesadores más utilizados

Less (*Leaner Style Sheets*)



Fuente: [Less](#).

Less se parece mucho a CSS al igual que Sass, por esta razón es que ambos preprocesadores son muy sencillos de aprender y muy similares en sus funciones básicas.

Con Less podemos extraer ciertos componentes de un color que no podemos con Sass, como por ejemplo el tono, saturación, luminosidad o brillo.

Por otra parte, en Less no contamos con algunas estructuras de control como if/else, o bucles como for y while que sí las tenemos en Sass.

Less

Comparación CSS y Less

```
1  /* CSS */
2
3  #header h1 {
4      font-size: 25px;
5      font-weight: bold;
6  }
7  #header p {
8      font-size: 12px;
9  }
10 #header p a {
11     text-decoration: none;
12 }
13 #header p a: hover {
14     border-width: 1px;
15 }
```

```
1  // Less
2
3  #header {
4      h1 {
5          font-size: 25px;
6          font-weight: bold;
7      }
8      p {
9          font-size: 12px;
10         a {
11             text-decoration: none;
12             &: hover {
13                 border-width: 1px;
14             }
15         }
16     }
17 }
```



Durante el desarrollo del módulo utilizaremos Sass como procesador, pero puedes investigar sobre el uso de Less desde su [sitio oficial](#).

¿Puedes nombrar 2
beneficios que tendremos al
usar un preprocesador de
CSS?



Resumen

A

Las **metodologías CSS** son un conjunto de pautas o sistemas preestablecidos que nos ayudan a escribir y organizar nuestro código para que este sea mantenible y escalable.

B

La metodología **BEM** se basa en dividir la estructura del sitio web en Bloque, Elemento y Modificador.

C

La metodología **OOCSS** se basa en dos principios

- Separar la estructura de la piel
- Separar el contenedor de su contenido.

D

Los **preprocesadores de CSS** son una herramienta que permiten traducir (compilar) hojas de estilos escritas en una sintaxis única de cada procesador, y generar un archivo .CSS entendible por el navegador web.



Próxima sesión...

- *Describe las características principales de la metodología BEM para la organización y modularización de estilos.*

{desafío}
latam_

*Academia de
talentos digitales*

