

fundamentos en la implementación de componentes front-end, HTML, CSS, JS
GA6-220501096-AA4-EV01

Karen Dayana Diaz Mora

Angie Lorena Fajardo Nuñez

Merli Yurani Ariza Herrera

ANÁLISIS Y DESARROLLO DE SOFTWARE

Presentado: Jorge Garcia

Ficha:2977373

2025

Introducción

En el mundo del desarrollo web existen tres lenguajes fundamentales que trabajan de manera conjunta para dar vida a las páginas y aplicaciones que utilizamos diariamente: **HTML, CSS y JavaScript**.

El **HTML (HyperText Markup Language)** es el encargado de estructurar el contenido de una página web mediante etiquetas, organizando títulos, párrafos, imágenes, enlaces y demás elementos. Por su parte, el **CSS (Cascading Style Sheets)** define la apariencia visual, como los colores, fuentes, tamaños y distribución de los elementos en la pantalla, permitiendo separar el contenido de la presentación. Finalmente, **JavaScript** aporta interactividad y dinamismo, haciendo posible que los usuarios puedan interactuar con botones, menús, formularios, animaciones y otros elementos sin necesidad de recargar la página.

La combinación de estos tres lenguajes constituye la base del desarrollo web moderno, garantizando páginas bien estructuradas, atractivas visualmente e interactivas, lo que facilita la creación de experiencias digitales completas y funcionales.

Qué es HTML: es el lenguaje estándar para la creación de páginas web. Sus siglas significan HyperText Markup Language (Lenguaje de Marcado de Hipertexto). Se trata de un lenguaje de marcas, es decir, un conjunto de etiquetas que se usan para: Estructurar la información (títulos, párrafos, tablas, listas, imágenes, etc.).

- Organizar el contenido de una página para que el navegador lo interprete y lo muestre al usuario.
- Dar significado semántico al texto (por ejemplo, indicar qué es un encabezado, qué es un párrafo, qué texto es importante, etc.).

no es un lenguaje de programación, sino de marcado. Solo define la estructura y el contenido de la página, mientras que la apariencia visual se gestiona con CSS y la interacción con JavaScript.

Cuando accedemos a una página web, el navegador solicita un documento de texto al servidor, el cual está escrito en **HTML (HyperText Markup Language)**, un lenguaje de etiquetas que estructura la información para que el navegador la interprete y muestre de forma visual

Lo más importante de HTML

1. **Lenguaje base de la web:** todas las páginas web están construidas con HTML.
2. **Estructura del contenido:** permite organizar textos, imágenes, enlaces, videos, formularios y más.
3. **Es semántico:** las etiquetas no solo muestran información, también le dicen al navegador y a buscadores *qué significa* cada parte (ej. `<h1>` = título principal, `<p>` = párrafo, `` = texto importante).
4. **Separación de responsabilidades:**

- **HTML** = contenido y estructura.
- **CSS** = diseño y estilos.
- **JavaScript** = interactividad y lógica.

Etiquetas HTML:

- Son la base del lenguaje.
- Se escriben con la forma `<etiqueta>contenido</etiqueta>`.
- Cada etiqueta tiene un propósito específico (ejemplo: `<p>` para párrafos, `` para destacar texto).
- Pueden anidarse, es decir, una etiqueta dentro de otra.

Semántica en HTML5:

- HTML5 se centra en el significado del contenido y no en su apariencia.
- La presentación (colores, estilos, negritas, fuentes, etc.) debe definirse en CSS.
- Ejemplo: en lugar de usar `` (negrita, visual), se recomienda `` (importancia, semántico).

Ventajas de la semántica:

- Separa el contenido de la presentación.
- Facilita que tanto navegadores como buscadores (ej. Googlebot) comprendan mejor la información de una página.
- Mejora la accesibilidad, el SEO y la interoperabilidad de la web.

```
index.html M X styles.css U app.js U
index.html > html > head > meta
...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6 <link href="css/styles.css" rel="stylesheet" />
7 <title>Agendatech</title>
8 <link rel="icon" href="src/img/favicon.ico" type="image/x-icon" />
9 </head>
10 <body>
11 <header class="site-header">
12 <a href="/" class="site-logo">
13 
14 
15 </a>
16 <button id="menu-toggle">☰</button>
17 <nav id="menu" class="menu">
18 <ul>
19 <li><a href="#">Agendatech</a></li>
20 <li><a href="#">Productos</a></li>
21 <li><a href="#">Mi cuenta</a></li>
22 <li><a href="#"></a></li>
23 </ul>
24 </nav>
25 </header>
26
27
```

```
index.html M X styles.css U app.js U
index.html > html > head
2 <html lang="en">
21 <body>
...
39 <div class="custom-carousel-container">
40 <div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
41 <div class="carousel-inner">
42 <div class="carousel-item active">
43 
44 </div>
45 <div class="carousel-item">
46 
47 </div>
48 <div class="carousel-item">
49 
50 </div>
51 </div>
52 <button class="carousel-control-prev" type="button" data-bs-target="#carouselExampleIndicators">
53 <span class="carousel-control-prev-icon"></span>
54 </button>
55 <button class="carousel-control-next" type="button" data-bs-target="#carouselExampleIndicators">
56 <span class="carousel-control-next-icon"></span>
57 </button>
58 </div>
59 </div>
60
61 </body>
62 <script type="module" src="js/app.js"></script>
63 </html>
```

¿Qué es CSS?

CSS, conocido como Hojas de Estilo en Cascada, es un lenguaje de estilo que especifica la forma en que deben mostrarse los documentos en la web. Es crucial en la creación de sitios web ya que disocia el contenido (HTML) de su diseño y visualización (colores, tipos de letra, estructura), lo que nos facilita la elaboración de páginas web que son visualmente agradables, fáciles de mantener y que se ajustan a varios tipos de dispositivos.

¿Qué hace el CSS?

- **Define la apariencia:** Regula aspectos como el color, la fuente, los márgenes, las dimensiones de los elementos y la organización general del sitio.
- **Separar contenido y presentación:** Conserva el HTML enfocado en la estructura, mientras que el CSS se ocupa de la apariencia.
- **Facilita el mantenimiento:** Al tener los estilos en un archivo aislado o consolidado, los ajustes se implementan en varias páginas de forma efectiva.
- **Permite diseños adaptativos:** Es fundamental para desarrollar sitios web que se vean atractivos en diferentes tamaños de pantalla, abarcando desde computadoras hasta dispositivos móviles.

¿Cómo funciona CSS?

1. Composición de un documento CSS:

Un archivo CSS está formado por normas que determinan la presentación de elementos HTML específicos. Cada norma incluye un selector y un bloque de declaraciones.

- **Selector:** El selector señala a cuáles elementos HTML se le asigna la norma. Puede ser un elemento particular (como p para los párrafos, h1 para los títulos), una clase (como. destacado), o un ID (como #principal).
- **Bloque de declaración:** En el bloque de declaración se incluye una o varias declaraciones, las cuales están divididas por un punto y coma.
- **Declaración:** Cada declaración se compone de una propiedad (por ejemplo, color, tamaño de fuente, margen) y un valor correspondiente a esa propiedad (por ejemplo, red, 16px, 10px)

2. Uso de CSS en HTML:

- El navegador procesa tanto el código HTML como el CSS.
- El HTML establece la organización y la información del sitio web.
- El CSS establece cómo se verá ese contenido.
- El navegador genera un Modelo de Objetos del Documento (DOM) basado en el HTML.
- El navegador implementa las reglas de CSS en el DOM, alterando la forma en que se presentan visualmente los elementos HTML de acuerdo con lo que se ha indicado en el CSS.

3. Cascada y especificidad:

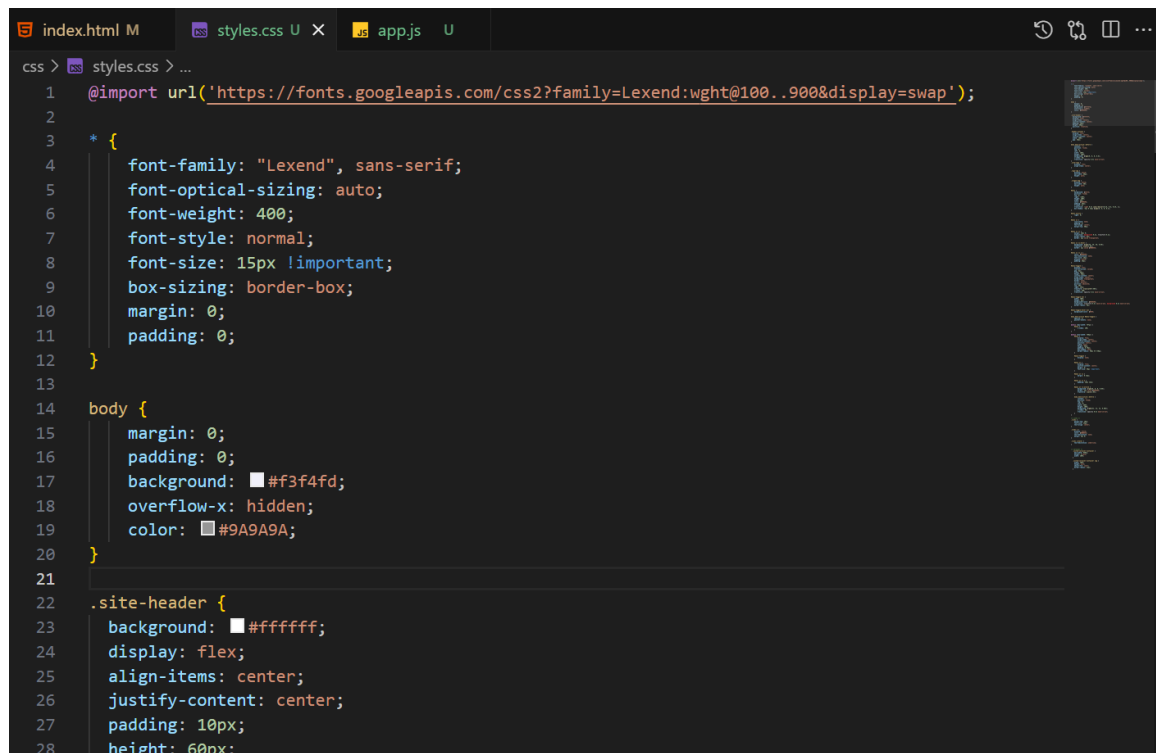
CSS opera mediante un sistema de cascada para resolver situaciones en las que varias reglas pueden afectar un mismo elemento.

- **Cascada:** Las reglas de CSS se aplican siguiendo un orden determinado, donde las reglas más concretas o con mayor especificidad son las que tienen mayor prioridad.

- **Especificidad:** La especificidad establece qué regla predomina en caso de desacuerdos. Los selectores más concretos (como los IDs) tienen una especificidad mayor que los selectores más amplios (como las clases o los elementos).
- **Herencia:** Algunas propiedades de CSS se transfieren de los elementos padre a los hijos, mientras que otras no lo hacen.

4. Ventajas de CSS:

- **División entre contenido y diseño:** CSS nos permite diferenciar el contenido (HTML) del estilo (CSS), lo que simplifica el mantenimiento y actualización del sitio web.
- **Consistencia:** CSS hace posible aplicar estilos uniformes a varios elementos, evitando tener que repetir el código en HTML.
- **Flexibilidad:** CSS nos proporciona una amplia variedad de propiedades que permiten manejar la apariencia visual de nuestro sitio web, otorgando gran libertad en el diseño.
- **Optimización:** CSS puede optimizar la velocidad de nuestro sitio web al disminuir el tamaño del código HTML y de esta forma facilitar las cargas rápidas de las páginas.

A screenshot of a web browser's developer console. The top bar shows three tabs: 'index.html M', 'styles.css U x', and 'app.js U'. The 'styles.css' tab is active, showing a list of CSS rules. The first rule is a font import for 'Lexend' from Google Fonts. The second rule is a universal selector '*' with various font and box-sizing properties. The third rule is for the 'body' element, setting margin, padding, background color to '#f3f4fd', overflow-x to 'hidden', and text color to '#9A9A9A'. The fourth rule is for '.site-header', setting background to white, display to 'flex', alignment to 'center', justification to 'center', padding to '10px', and height to '60px'.

```
css > styles.css > ...
1  @import url('https://fonts.googleapis.com/css2?family=Lexend:wght@100..900&display=swap');
2
3  * {
4      font-family: "Lexend", sans-serif;
5      font-optical-sizing: auto;
6      font-weight: 400;
7      font-style: normal;
8      font-size: 15px !important;
9      box-sizing: border-box;
10     margin: 0;
11     padding: 0;
12 }
13
14 body {
15     margin: 0;
16     padding: 0;
17     background: #f3f4fd;
18     overflow-x: hidden;
19     color: #9A9A9A;
20 }
21
22 .site-header {
23     background: #ffffff;
24     display: flex;
25     align-items: center;
26     justify-content: center;
27     padding: 10px;
28     height: 60px;
```

¿Qué es JavaScript?

JavaScript es un lenguaje de programación que se ejecuta directamente en el navegador. Su función principal es permitir que las páginas web sean interactivas, es decir, que los usuarios puedan interactuar con elementos como botones, formularios, menús dinámicos o animaciones sin necesidad de recargar la página. Se integra estrechamente con HTML, que define la estructura de la página, y CSS, que define su estilo visual.

Variables y tipos de datos

Las variables en JavaScript se utilizan para almacenar información que puede ser utilizada y modificada durante la ejecución del programa. Dependiendo de su uso, se pueden declarar como variables que cambian su valor o como constantes que permanecen inalterables. Los tipos de datos más comunes son los números, las cadenas de texto, los valores booleanos, los arreglos (listas de elementos) y los objetos, que permiten agrupar información relacionada bajo un mismo nombre.

Operadores

JavaScript dispone de diferentes operadores que permiten realizar cálculos y evaluaciones. Los operadores aritméticos se utilizan para realizar operaciones matemáticas básicas como suma, resta, multiplicación y división. Los operadores de comparación permiten evaluar condiciones entre valores, mientras que los operadores lógicos ayudan a combinar múltiples condiciones, facilitando la toma de decisiones dentro del programa.

Estructuras de control

Las estructuras de control permiten que el programa tome decisiones o repita acciones según ciertas condiciones. Las condicionales permiten ejecutar diferentes acciones dependiendo de si una condición es verdadera o falsa. Por otro lado, los bucles permiten repetir bloques de código un número determinado de veces o mientras se cumpla una condición, lo que resulta útil para procesar listas de datos o generar repeticiones automáticas.

Funciones

Las funciones son bloques de código que realizan acciones específicas y pueden ser reutilizadas en distintas partes del programa. Permiten organizar el código, evitando repetición, y pueden recibir información para procesarla y devolver un resultado. Gracias a las funciones, el código es más legible, modular y fácil de mantener.

Eventos

Los eventos permiten que la página web responda a las acciones del usuario, como hacer clic en un botón, pasar el cursor sobre un elemento, o cambiar el contenido de un formulario. Estos eventos se pueden vincular a funciones específicas que se ejecutan automáticamente cuando ocurre la acción, generando una experiencia interactiva y dinámica.

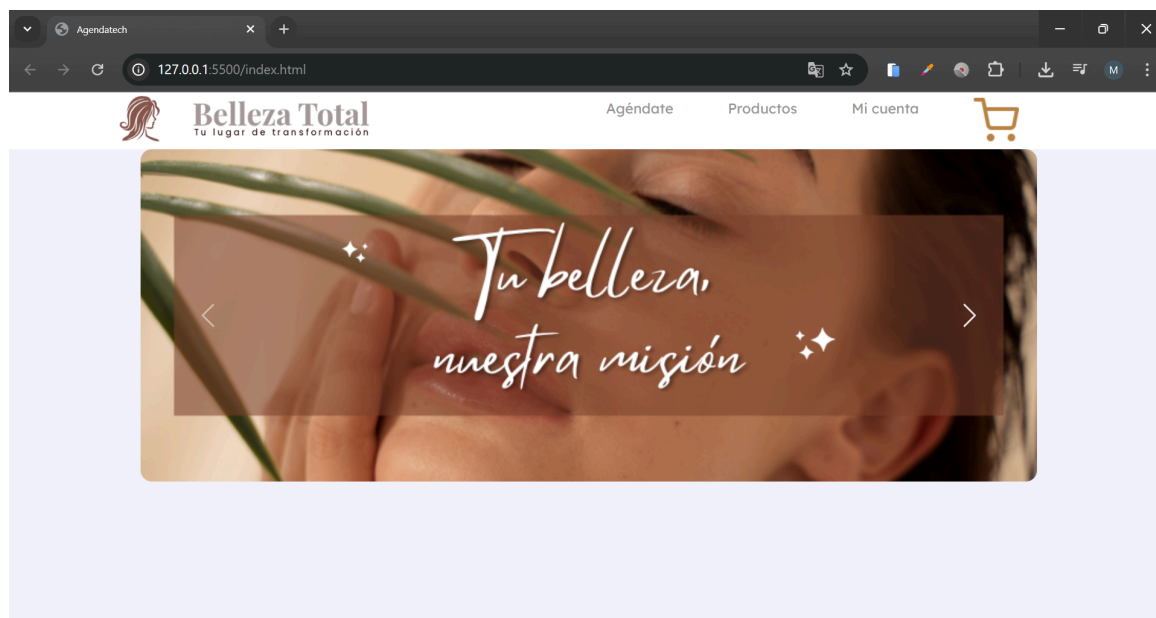
Manipulación del DOM

El DOM, o Modelo de Objetos del Documento, es la forma en que JavaScript representa la estructura de una página web. A través del DOM, es posible seleccionar elementos de la página y modificar su contenido, su estilo o su comportamiento en tiempo real, permitiendo crear páginas más dinámicas y personalizadas según la interacción del usuario.

Arrays y objetos

Los arrays son estructuras que permiten almacenar listas de elementos de manera ordenada, facilitando su acceso, modificación y recorrido. Los objetos permiten agrupar información relacionada en un solo bloque y definir propiedades y funciones asociadas, lo que hace que el código sea más organizado y cercano a la programación orientada a objetos.

```
index.html M  styles.css U  app.js U X
js > app.js > ...
1  document.addEventListener("DOMContentLoaded", function () {
2      const menuToggle = document.getElementById("menu-toggle");
3      const menu = document.getElementById("menu");
4      const body = document.body;
5
6      menuToggle.addEventListener("click", function () {
7          body.classList.toggle("menu-active");
8          menu.classList.toggle("active");
9      });
10
11     document.addEventListener("click", function (event) {
12         if (!menu.contains(event.target) && !menuToggle.contains(event.target)) {
13             body.classList.remove("menu-active");
14             menu.classList.remove("active");
15         }
16     });
17 });
```



Conclusión

La utilización de elementos frontend mediante HTML, CSS y JavaScript constituye el fundamento para desarrollar páginas web contemporáneas y operativas, donde HTML organiza la información, CSS establece la estética y la apariencia visual, y JavaScript introduce interactividad y dinamismo para ofrecer experiencias de usuario completas y cautivadoras. La combinación de estas tres tecnologías es fundamental, ya que colaboran para construir interfaces intuitivas y efectivas que destacan en el competitivo entorno digital.