

## 2.5. Semántica y accesibilidad en HTML



### LENGUAJES DE PROGRAMACIÓN NIVEL 3. UNIDAD 2

#### HTML - LENGUAJE DE MARCAS PARA LA ESTRUCTURA DE CONTENIDO

## Contenido

Contenido .....	2
Introducción .....	3
Fundamentos de la Semántica HTML .....	4
Introducción a la Semántica HTML.....	4
Del Etiquetado Tradicional al Etiquetado Semántico .....	5
Etiquetas HTML Tradicionales .....	5
Etiquetas HTML Semánticas .....	6
Elementos Semánticos Básicos.....	8
Estructuración de Contenido con HTML Semántico.....	9
Organización del Contenido.....	9
Ejemplos prácticos .....	10
Accesibilidad en HTML.....	11
Principios de Accesibilidad Web .....	11
Roles ARIA y Atributos.....	12
Roles ARIA:.....	12
Atributos ARIA:.....	13
Mejores Prácticas para una Web Accesible .....	14
Uso de Formularios Accesibles .....	14
Imágenes y Multimedia .....	14
Navegación y Estructura del Sitio .....	15
Conclusión .....	15
Recursos Adicionales .....	16
Documentación y Tutoriales .....	16
Herramientas de Accesibilidad .....	16

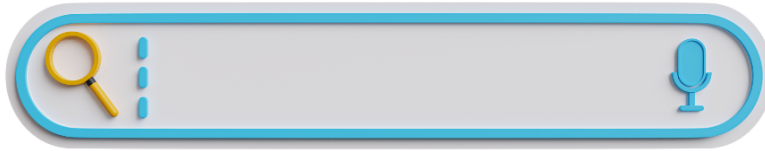
## Introducción

***Bienvenido al fascinante mundo de la semántica y accesibilidad en HTML, los pilares fundamentales que aseguran que nuestras páginas web no solo sean entendidas por todos los usuarios, sino también por las máquinas. A medida que avanzamos en la era digital, la inclusión y la claridad en la web se han vuelto imperativas.***

*La semántica en HTML va más allá del simple uso de etiquetas para formatear contenido; se trata de usar HTML para transmitir el significado y la estructura de ese contenido. Una buena semántica además mejora la accesibilidad, refuerza el SEO y facilita la mantenibilidad del código. Por otro lado, la accesibilidad se asegura de que nuestras páginas web sean utilizables por la mayor cantidad posible de personas, incluyendo aquellas con diversas discapacidades. Juntos, estos conceptos fomentan una web más inclusiva y comprensible.*

*En este módulo, exploraremos los elementos semánticos que HTML5 nos ofrece, cómo estructurar adecuadamente el contenido para que sea intuitivo tanto para los usuarios como para los motores de búsqueda, y cómo implementar prácticas de accesibilidad que garanticen que nadie quede excluido de acceder a nuestra información. Desde el correcto uso de etiquetas hasta la implementación de ARIA (Accessible Rich Internet Applications) roles y atributos, este módulo te introduce al conocimiento necesario para construir páginas web que se destaquen no solo por su estética, sino también por su inclusividad y claridad.*

## Fundamentos de la Semántica HTML



La semántica HTML no es solo una parte integral del desarrollo web moderno; es el alma que da sentido y estructura a nuestro contenido en la web.

En esta sección, exploraremos los fundamentos de la semántica en HTML y cómo su correcta aplicación no solo beneficia a los usuarios finales, sino también a los motores de búsqueda y las tecnologías de ayuda, al tiempo que facilita la mantenibilidad del código.

### Introducción a la Semántica HTML

La semántica en HTML se refiere al uso de etiquetas HTML para dar significado al contenido web, en lugar de simplemente definir su apariencia. A diferencia del HTML tradicional, cada etiqueta tiene un propósito específico y connota cierta información sobre el tipo de contenido que encierra. Por ejemplo, una etiqueta `<article>` indica que el contenido dentro de ella es un artículo independiente, que podría ser comprendido fuera del contexto del resto de la página.

La importancia de la semántica radica en su capacidad para:

- **Mejorar la Accesibilidad:** Las tecnologías de ayuda, como los lectores de pantalla, se basan en el marcado semántico para interpretar la estructura y el tipo de contenido, facilitando su navegación y comprensión por parte de usuarios con discapacidades.
- **Optimizar el SEO:** Los motores de búsqueda utilizan las etiquetas semánticas para comprender la estructura y el contenido de las páginas web, lo que ayuda a indexar la información de manera más efectiva y mejora el ranking de búsqueda.

- Facilitar la Mantenibilidad: Un código bien estructurado con marcado semántico es más fácil de leer, entender y mantener, tanto para el creador original como para otros desarrolladores que puedan trabajar en el código en el futuro.
- La habilidad para diseñar formularios eficientes y amigables es crucial para la experiencia del usuario, ya que una interfaz de formulario bien diseñada puede fomentar la interacción y mejorar la percepción general de un sitio web.

## Del Etiquetado Tradicional al Etiquetado Semántico

La evolución de HTML hacia un enfoque más semántico ha transformado la forma en que los desarrolladores estructuran el contenido de una página web. La diferencia entre las etiquetas HTML "tradicionales" y las "semánticas" radica principalmente en su propósito y significado dentro de la estructura del documento. Aquí te muestro una comparación para ilustrar estos cambios:

### Etiquetas HTML Tradicionales

Las etiquetas tradicionales, como `<div>` y `<span>`, son contenedores genéricos que no proporcionan información sobre el tipo de contenido que encierran. Su principal función es estructurar la presentación del contenido, a menudo en combinación con atributos de clase o ID para aplicar estilos CSS específicos.

```
html Copy code

<div class="header">
  <div class="navigation">
    <!-- Enlaces de navegación aquí -->
  </div>
</div>
<div class="mainContent">
  <div class="article">
    <!-- Contenido del artículo aquí -->
  </div>
</div>
<div class="footer">
  <!-- Contenido del pie de página aquí -->
</div>
```

1

En este ejemplo, las etiquetas `<div>` se utilizan para crear la estructura de la página, pero no comunican ningún significado sobre el contenido que contienen. Los nombres de las clases sugieren su propósito, pero solo son claros para los humanos que leen el código, no para las máquinas.

## Etiquetas HTML Semánticas

Las etiquetas semánticas, introducidas con HTML5, como `<header>`, `<nav>`, `<article>`, `<section>`, y `<footer>`, indican explícitamente la función y el tipo de contenido que encierran. Estas etiquetas mejoran la accesibilidad y la

---

<sup>1 1</sup> Accesibilidad:

```
<div class="header">
  <div class="navigation">
    <!-- Enlaces de navegación aquí -->
  </div>
</div>
<div class="main_Content">
  <div class="article">
    <!-- Contenido del artículo aquí -->
  </div>
</div>
<div class="footer">
  <!-- Contenido del pie de página aquí -->
</div>
```

legibilidad del código, tanto para los desarrolladores como para las tecnologías asistivas y los motores de búsqueda.

```
html Copy code

<header>
  <nav>
    <!-- Enlaces de navegación aquí -->
  </nav>
</header>
<section>
  <article>
    <!-- Contenido del artículo aquí -->
  </article>
</section>
<footer>
  <!-- Contenido del pie de página aquí -->
</footer>
```

2

Este ejemplo utiliza etiquetas semánticas que definen claramente las diferentes secciones de la página: el área de navegación (<nav> dentro de <header>), el contenido principal (<article> dentro de <section>), y el pie de página (<footer>). Cada etiqueta describe el propósito del contenido que contiene, lo que facilita su comprensión por parte de las máquinas y mejora la accesibilidad para los usuarios con discapacidades.

---

<sup>2 2</sup>Accesibilidad:

```
<header>
  <nav>
    <!-- Enlaces de navegación aquí -->
  </nav>
</header>
<section>
  <article>
    <!-- Contenido del artículo aquí -->
  </article>
</section>
<footer>
  <!-- Contenido del pie de página aquí -->
</footer>
```

## Elementos Semánticos Básicos

La transición de etiquetas HTML tradicionales a semánticas es fundamental para crear páginas web más accesibles, estructuradas y fáciles de interpretar. Mientras que las etiquetas tradicionales se centran en la presentación, las etiquetas semánticas enfatizan el significado del contenido, lo que contribuye a una mejor experiencia de usuario, accesibilidad y SEO.

En este sentido, varios elementos en HTML5 fueron introducidos o repotenciados para fomentar un uso más semántico del lenguaje:

- `<header>`: Representa un grupo de ayudas introductorias o de navegación. Puede contener cosas como títulos de sección, enlaces de navegación o logotipos.
- `<footer>`: Define el pie de página de una sección o documento, donde se suelen colocar los datos de contacto, derechos de autor y enlaces a documentos legales.
- `<nav>`: Diseñado para contener los principales bloques de navegación del sitio, como menús o índices.
- `<article>`: Encapsula un contenido independiente y autocontenido que podría distribuirse por sí solo, como un artículo de blog o una noticia.
- `<section>`: Define una sección del documento con un tema relacionado, agrupando contenido y subsecciones con un hilo temático común.
- `<aside>`: Se utiliza para contenido tangencialmente relacionado con el contenido principal, como barras laterales, publicidad o listas de enlaces.

Cada uno de estos elementos juega un papel crucial en la construcción de una estructura web lógica y accesible, permitiendo que el contenido sea presentado y comprendido de manera clara y efectiva.



## Estructuración de Contenido con HTML Semántico



La adecuada estructuración del contenido utilizando elementos semánticos en HTML no solo mejora la legibilidad y la navegación dentro de una página web, sino que también refuerza su accesibilidad y optimización para motores de búsqueda. Esta sección explora cómo podemos aplicar estos principios para organizar efectivamente el contenido.

### Organización del Contenido

Una estructura semántica clara y lógica es esencial para guiar a los usuarios a través de tu contenido de manera intuitiva. Al utilizar elementos HTML semánticos, proporcionas un contexto claro tanto para los usuarios como para los motores de búsqueda, definiendo explícitamente las diferentes partes de tu página web:

- `<header>`: Este elemento suele contener el título del sitio, el logo, y la navegación principal. Situado en la parte superior de la página, actúa como la "cara" del sitio web.
- `<nav>`: Dentro del `<header>` o como un elemento independiente, `<nav>` encierra los enlaces de navegación principal, ayudando a los usuarios a moverse por el sitio web.
- `<main>`: Contiene el contenido principal de la página, diferenciándolo claramente del contenido periférico en `<aside>` o la información en el `<footer>`.

- `<article>`: Utilizado para contenido independiente y autocontenido que podría existir por sí mismo fuera del sitio web, como un post de blog o un artículo de noticias.
- `<section>`: Agrupa contenido temáticamente relacionado dentro del `<main>`, cada `<section>` puede contener su propio `<header>`, `<footer>`, y una estructura interna.
- `<aside>`: Destinado para contenido tangencialmente relacionado con el contenido principal, como barras laterales con información adicional, perfiles o enlaces relacionados.
- `<footer>`: El pie de página de la página o sección, que suele contener información de contacto, derechos de autor, y enlaces a políticas de privacidad o términos de uso.

## Ejemplos prácticos

### Página de Blog

```
<article>
  <header>
    <h1>Título del Post</h1>
    <p>Publicado por <author>Nombre del Autor</author> el <time datetime="2021-04-06">
  </header>
  <section>
    <h2>Introducción</h2>
    <p>Contenido introductorio...</p>
  </section>
  <section>
    <h2>Contenido Principal</h2>
    <p>Detalle del contenido...</p>
  </section>
  <footer>
    <p>Comentarios...</p>
  </footer>
</article>
```

3

<sup>3</sup> Accesibilidad:

```
<article>
  <header>
    <h1>Título del Post</h1>
    <p>Publicado por <author>Nombre del Autor</author> el <time
datetime="2024-04-06
```

## Accesibilidad en HTML



La accesibilidad web es una práctica esencial que garantiza que los sitios web sean utilizables por todas las personas, incluidas aquellas con discapacidades. Al adherirse a los principios de accesibilidad, podemos crear experiencias web inclusivas que no discriminan contra usuarios con limitaciones visuales, auditivas, motoras o cognitivas.

### Principios de Accesibilidad Web

Las Pautas de Accesibilidad para el Contenido Web (WCAG) son desarrolladas por el W3C y constituyen el estándar de facto para la accesibilidad web. Estas pautas se basan en cuatro principios fundamentales, conocidos por el acrónimo POUR:

---

```

</header>
<section>
  <h2>Introducción</h2>
  <p>Contenido introductorio.
</section>
<section>
  <h2>Contenido Principal</h2>
  <p>Detalle del contenido. ...</p>
</section>
<footer>
  <p>Comentarios. ...</p>
</footer> Oo
</article>

```

- **Perceptible:** La información y los componentes de la interfaz de usuario deben presentarse de manera que todos los usuarios puedan percibirlos, independientemente de sus capacidades sensoriales. Esto incluye proporcionar alternativas textuales para el contenido no textual, como imágenes, y garantizar que la información sea visible y audible para todos.
- **Operable:** Los componentes de la interfaz y la navegación deben ser operables para todos. Esto significa que el sitio web debe ser navegable y utilizable con diversas herramientas, como teclados y lectores de pantalla, y que debe evitar diseños que puedan provocar convulsiones o reacciones físicas.
- **Comprensible:** La información y la operación de la interfaz de usuario deben ser comprensibles. Esto implica que el contenido debe ser legible y entendible, y que las páginas web deben aparecer y operar de manera predecible.
- **Robusto:** El contenido debe ser lo suficientemente robusto como para ser interpretado de manera confiable por una amplia variedad de agentes de usuario, incluidas las tecnologías asistivas. Esto significa utilizar un marcado estándar y asegurar la compatibilidad entre navegadores y dispositivos.

## Roles ARIA y Atributos

ARIA (Accessible Rich Internet Applications) es un conjunto de atributos especiales que se pueden agregar al HTML para mejorar la accesibilidad, especialmente en aplicaciones web complejas que utilizan elementos dinámicos y controles avanzados de la interfaz de usuario que no son semánticamente representados por las etiquetas HTML estándar.

### Roles ARIA:

Definen el tipo de elemento de la interfaz de usuario o la estructura de la página. Por ejemplo, `role="navigation"` indica que el elemento contiene enlaces de navegación principal dentro de la página.

```
html Copy code

<nav role="navigation">
  <!-- Enlaces de navegación aquí -->
</nav>
```

4

## Atributos ARIA:

Proporcionan propiedades adicionales a los elementos para mejorar la accesibilidad. Estos pueden incluir estados (aria-expanded, aria-selected) y propiedades (aria-label, aria-labelledby) que describen las relaciones o características de los elementos.

El uso adecuado de ARIA puede mejorar significativamente la accesibilidad de los elementos interactivos y dinámicos, pero debe utilizarse con precaución y solo cuando los elementos HTML semánticos no puedan lograr el mismo resultado.

```
html Copy code

<button aria-expanded="false" aria-controls="menu">Menú</button>
```

5

Para terminar, la accesibilidad no es un 'extra' o una consideración posterior en el desarrollo web; es una parte integral de un diseño inclusivo y ético. Al seguir los principios de WCAG y utilizar herramientas como ARIA cuando sea necesario, podemos asegurar que nuestros sitios web sean accesibles y utilizables para todos, independientemente de sus habilidades o limitaciones.

<sup>4</sup> Accesibilidad:

```
<nav role="navigation">
  <!-- Enlaces de navegación aquí -->
```

```
</nav>
```

<sup>5</sup> Accesibilidad:

```
<button aria-expanded="false" aria-controls="menu">Menú</button>
```

## Mejores Prácticas para una Web Accesible

Crear una web accesible significa garantizar que todos los usuarios, incluidos aquellos con discapacidades, puedan acceder, entender y navegar por tu sitio web sin barreras. Esta sección profundiza en las mejores prácticas para lograr una accesibilidad óptima, desde el diseño de formularios hasta la presentación de contenido multimedia y la estructuración de la navegación del sitio.

### Uso de Formularios Accesibles

Los formularios accesibles son cruciales para permitir que los usuarios interactúen con tu sitio web. Aquí hay algunos consejos para mejorar la accesibilidad de los formularios:

- **Etiquetas Claras:** Usa la etiqueta `<label>` para cada campo de entrada, asociándola correctamente con el atributo `id` del campo correspondiente para garantizar que las tecnologías asistivas puedan interpretarlos adecuadamente.
- **Instrucciones y Errores Descriptivos:** Proporciona instrucciones claras y mensajes de error descriptivos para guiar a los usuarios a través del proceso de llenado del formulario, especialmente cuando se encuentran errores de validación.
- **Elementos de Formulario Semánticos:** Utiliza elementos semánticos de HTML5 como `<fieldset>`, `<legend>`, y los tipos de entrada específicos (`<input type="email">`, `<input type="tel">`, etc.) para mejorar la estructura y la interpretación del formulario.

### Imágenes y Multimedia

El contenido visual y multimedia debe ser accesible para usuarios con discapacidades visuales o auditivas:

- **Texto Alternativo para Imágenes:** Siempre incluye un atributo `alt` descriptivo en las imágenes para proporcionar un equivalente textual que las tecnologías asistivas puedan leer.
-

```
html Copy code  
  

```

6

- Subtítulos y Transcripciones para Multimedia: Asegúrate de que todos los contenidos de audio y video incluyan subtítulos para usuarios con discapacidades auditivas y transcripciones para aquellos que no pueden acceder al contenido multimedia.

## Navegación y Estructura del Sitio

Una navegación intuitiva y una estructura de sitio bien organizada son fundamentales para la accesibilidad:

- Navegación Consistente: Mantén una estructura de navegación coherente en todo el sitio web, utilizando elementos semánticos como <nav> para los menús de navegación.
- Saltos de Contenido: Proporciona enlaces de "saltar al contenido principal" al principio de cada página para permitir a los usuarios que utilizan tecnologías asistivas omitir la navegación repetitiva.
- Encabezados Descriptivos: Usa encabezados (<h1>, <h2>, etc.) para estructurar el contenido de forma lógica y jerárquica, facilitando a los usuarios y las tecnologías de ayuda la comprensión de la estructura de la página.

## Conclusión

Hemos recorrido un camino significativo en la comprensión de la semántica y accesibilidad en HTML, aprendiendo cómo estructurar el contenido de manera que sea intuitivo, accesible y significativo tanto para los usuarios como para los motores de búsqueda. La semántica y la accesibilidad son la columna vertebral de un internet inclusivo, y al aplicar los principios que hemos explorado, estás contribuyendo a hacer de la web un lugar mejor para todos.

---

<sup>6</sup> Accesibilidad:



A medida que cerramos este tema sobre HTML, nos preparamos para embarcarnos en el siguiente viaje: el mundo de las Hojas de Estilo en Cascada (CSS). CSS es el arte de vestir tu contenido, controlando cómo se presenta visualmente en diferentes dispositivos y pantallas. Con CSS, llevarás la estructura semántica y accesible que has construido con HTML al siguiente nivel, creando experiencias visuales ricas y envolventes que cautiven a tus usuarios.

## Recursos Adicionales

Para continuar tu aprendizaje y profundizar en los temas cubiertos, aquí hay algunos recursos adicionales que pueden ser de gran ayuda:

### Documentación y Tutoriales

- MDN Web Docs (HTML): Profundiza en los elementos y atributos de HTML, con guías detalladas sobre semántica y accesibilidad. (<https://developer.mozilla.org/es/docs/Web/HTML>)
- W3C Web Accessibility Initiative: Explora las directrices y recursos ofrecidos por la WAI para mejorar la accesibilidad web. (<https://www.w3.org/WAI/resources/>)

### Herramientas de Accesibilidad

- WebAIM: Herramientas y chequeos para evaluar y mejorar la accesibilidad de tu sitio web. (<https://wave.webaim.org/>)
- AXE: Una herramienta de prueba de accesibilidad disponible como extensión de navegador que ayuda a identificar y solucionar problemas de accesibilidad en tu sitio web. (<https://www.deque.com/axe/browser-extensions/>)

