

Manual Completo de HTML: De Cero a Experto

Autor: Basdonax Al **Instructor Experto:** Manus

Índice General

(Este índice será generado automáticamente y con hipervínculos al convertir a PDF. Por ahora, sirve como guía de estructura.)

- Nivel 1: Básico - Los Fundamentos de la Web**
 - 1.1 Introducción a HTML
 - 1.2 Estructura Básica de un Documento HTML
 - 1.3 Etiquetas de Encabezado (`<h1>` a `<h6>`)
 - 1.4 Párrafos y Saltos de Línea (`<p>` , `
`)
 - 1.5 Formato de Texto (`` , `` , `<i>` , `` , `<q>` , `<blockquote>`)
 - 1.6 Listas (`` , `` , `<dl>`)
 - 1.7 Resumen y Ejercicio Integrador del Nivel 1
 - Nivel 2: Intermedio - Enlaces, Imágenes y Multimedia** *(Contenido a desarrollar en la siguiente fase)*
 - Nivel 3: Avanzado - Semántica y Formularios** *(Contenido a desarrollar en la siguiente fase)*
 - Nivel 4: Experto - Accesibilidad y Buenas Prácticas** *(Contenido a desarrollar en la siguiente fase)*
 - Proyecto Integrador Final** *(Contenido a desarrollar en la siguiente fase)*
-

1. Nivel 1: Básico - Los Fundamentos de la Web

1.1 Introducción a HTML

Teoría:

HTML (HyperText Markup Language) es el **lenguaje de marcado** fundamental para la creación de páginas web. Piensa en HTML como el **esqueleto** de cualquier sitio web: define la estructura, el contenido y el orden de los elementos (títulos, párrafos, imágenes, enlaces, etc.). Sin HTML, no hay página web.

A diferencia de un lenguaje de programación, HTML no ejecuta acciones complejas; simplemente **etiqueta** y **estructura** el contenido para que el navegador web (como

Chrome, Firefox o Safari) sepa cómo mostrarlo. La versión que usaremos es **HTML5**, el estándar actual que introdujo nuevas etiquetas semánticas y funcionalidades.

Ejemplo Práctico:

Para empezar, solo necesitas un editor de texto simple (como VS Code, Sublime Text o incluso el Bloc de Notas) y un navegador. Los archivos HTML siempre terminan con la extensión `.html` o `.htm`.

HTML

```
<!-- Ejemplo 1.1: Mi primer archivo HTML -->
<!-- ¡Hola! Esto es un comentario. Los comentarios no se muestran en el
navegador. -->
<p>Este es el contenido más simple que puedo crear.</p>
```

Ejercicio 1.1:

1. Abre tu editor de texto.
2. Escribe el código del **Ejemplo 1.1**.
3. Guarda el archivo como `hola_mundo.html`.
4. Abre el archivo con tu navegador. Deberías ver el texto "Este es el contenido más simple que puedo crear."

1.2 Estructura Básica de un Documento HTML

Teoría:

Todo documento HTML bien formado debe seguir una estructura estándar para ser interpretado correctamente por los navegadores y los motores de búsqueda. Esta estructura se compone de etiquetas esenciales que definen el tipo de documento, el idioma, la configuración y el contenido visible.

| Etiqueta | Descripción |
|------------------------------------|--|
| <code><!DOCTYPE html></code> | Declaración de tipo de documento. Siempre debe ser la primera línea. Indica al navegador que el documento es HTML5. |
| <code><html></code> | Elemento raíz. Contiene absolutamente todo el contenido de la página. El atributo <code>lang="es"</code> define el idioma principal del contenido (en este caso, español). |
| <code><head></code> | Cabecera del documento. Contiene metadatos (información sobre el documento) que no son visibles para el usuario, como el título de la pestaña del navegador, la codificación de caracteres y enlaces a archivos CSS o JavaScript. |
| <code><body></code> | Cuerpo del documento. Contiene todo el contenido visible de la página: textos, imágenes, enlaces, etc. |

Ejemplo Práctico:

HTML

```

<!DOCTYPE html>
<!-- 1. Declaración de HTML5 -->
<html lang="es">
  <!-- 2. Etiqueta raíz, define el idioma como español -->
  <head>
    <!-- 3. Cabecera: Contiene metadatos -->
    <meta charset="UTF-8">
    <!-- Define la codificación de caracteres para evitar problemas con
tildes y eñes -->
    <title>Mi Primera Página Web Estructurada</title>
    <!-- El texto que aparece en la pestaña del navegador -->
  </head>
  <body>
    <!-- 4. Cuerpo: Contenido visible para el usuario -->
    <h1>¡Bienvenidos al curso de HTML!</h1>
    <p>Esta es la estructura fundamental que usaremos en todos nuestros
proyectos.</p>
  </body>
</html>

```

Ejercicio 1.2:

1. Crea un nuevo archivo llamado `mi_presentacion.html` .
2. Copia la estructura básica del **Ejemplo Práctico** anterior.
3. Modifica el contenido de la etiqueta `<title>` para que contenga tu nombre.
4. Modifica el contenido de la etiqueta `<h1>` para que sea un saludo personal.

1.3 Etiquetas de Encabezado (`<h1>` a `<h6>`)

Teoría:

Las etiquetas de encabezado (`<h1>` a `<h6>`) se utilizan para definir los títulos y subtítulos de una página. Son cruciales para la **semántica** y la **accesibilidad**, ya que establecen la jerarquía del contenido.

- `<h1>` debe usarse solo una vez por página y representa el título principal.
- `<h2>` representa las secciones principales.
- `<h3>` representa las subsecciones de `<h2>` , y así sucesivamente hasta `<h6>` .

¡Importante! Nunca uses estas etiquetas solo para hacer el texto más grande; su propósito es estructural.

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 1.3: Jerarquía de Títulos -->
<body>
  <h1>Título Principal de la Página (Solo uno)</h1>
  <p>Aquí comienza el contenido de la página.</p>

  <h2>Sección 1: Mi Experiencia</h2>
  <p>Contenido de la primera sección.</p>

  <h3>Subsección 1.1: Habilidades Técnicas</h3>
  <p>Detalle de las habilidades.</p>

  <h2>Sección 2: Proyectos</h2>
  <p>Contenido de la segunda sección.</p>
</body>
```

Ejercicio 1.3:

1. Usando el archivo `mi_presentacion.html` del ejercicio anterior, añade la siguiente estructura de títulos y subtítulos:

- `<h1>` Tu Nombre Completo
- `<h2>` Acerca de Mí
- `<h2>` Mis Intereses
- `<h3>` Interés 1
- `<h3>` Interés 2

1.4 Párrafos y Saltos de Línea (`<p>` , `
`)

Teoría:

- La etiqueta `<p>` (párrafo) es el elemento más común para agrupar bloques de texto. Los navegadores automáticamente añaden un espacio antes y después de cada párrafo.
- La etiqueta `
` (line break o salto de línea) es una etiqueta de cierre automático (no necesita `</br>`) que fuerza un salto de línea dentro de un bloque de texto, como un párrafo. **Se recomienda usar solo cuando el salto de línea es parte del contenido** (como en una dirección o un poema), y no para separar párrafos (para eso está `<p>`).

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 1.4: Párrafos y Saltos -->
<body>
  <p>
    Este es un párrafo completo. En HTML, los espacios y saltos de línea que
    escribas en el código fuente (como este) no se reflejan en la página web.
    El navegador los ignora y muestra el texto de forma continua.
  </p>

  <p>
    Esta es una dirección:

    Calle Ficticia #123

    Ciudad de la Web, País Virtual
  </p>
</body>
```

Ejercicio 1.4:

1. En tu archivo `mi_presentacion.html` , escribe un párrafo de presentación personal debajo del `<h1>` .

2. Debajo de cada `<h3>` (Interés 1 e Interés 2), escribe una descripción de dos líneas, forzando un salto de línea con `
` en el medio de cada descripción.

1.5 Formato de Texto (`` , `` , `<i>` , `` , `<q>` , `<blockquote>`)

Teoría:

HTML ofrece etiquetas para dar formato y, lo más importante, **semántica** al texto. Es fundamental entender la diferencia entre formato visual y significado semántico:

| Etiqueta | Significado Semántico | Formato Visual Común |
|---------------------------------|---|----------------------|
| <code></code> | Importancia Fuerte. Indica que el contenido es de gran importancia. | Negrita |
| <code></code> | Énfasis. Indica que el contenido debe ser enfatizado (cambio de tono al leer). | Cursiva |
| <code></code> | Estilístico. Se usa para llamar la atención sin implicar importancia. (Uso limitado en HTML5). | Negrita |
| <code><i></code> | Voz Alternativa. Se usa para términos técnicos, frases en otro idioma o pensamientos. (Uso limitado en HTML5). | Cursiva |
| <code><q></code> | Cita Corta. Para citas en línea. | Comillas |
| <code><blockquote></code> | Cita Larga. Para bloques de texto citados de otra fuente. | Sangría |

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 1.5: Formato Semántico -->
<p>
  El uso de la etiqueta <strong>strong</strong> es crucial para la
  accesibilidad,
```

```
ya que los lectores de pantalla pueden interpretarla como un cambio de voz.
</p>

<p>
  Como dijo el experto: <q>La semántica es el futuro del desarrollo web.</q>
</p>

<blockquote>
  <p>
    "El HyperText Markup Language (HTML) es el lenguaje de marcado estándar
    para documentos
    diseñados para ser mostrados en un navegador web."
  </p>
  <footer>— Definición de la W3C</footer>
</blockquote>
```

Ejercicio 1.5:

1. En tu presentación personal, usa `` para resaltar una habilidad clave (ej. **organización**).
2. Usa `` para enfatizar una frase en tu descripción (ej. *realmente disfruto* aprender).
3. Añade una cita corta de tu mentor o una frase inspiradora usando `<q>`.

1.6 Listas (`` , `` , `<dl>`)

Teoría:

Las listas son esenciales para organizar información de manera clara y concisa. HTML ofrece tres tipos principales:

1. **Lista No Ordenada (``)**: Se usa para elementos donde el orden no importa (ej. una lista de ingredientes). Los elementos se marcan con viñetas.
2. **Lista Ordenada (``)**: Se usa para elementos donde el orden es importante (ej. pasos de una receta). Los elementos se marcan con números o letras.
3. **Lista de Descripción (`<dl>`)**: Se usa para listas de términos y sus descripciones (ej. un glosario).

En los tres casos, cada elemento individual de la lista se define con la etiqueta `` (list item), excepto en la lista de descripción, que usa `<dt>` (término) y `<dd>` (descripción).

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 1.6: Tipos de Listas -->
<h3>Mis Hobbies (Lista No Ordenada)</h3>
```

```
<ul>
  <li>Leer ciencia ficción</li>
  <li>Programar pequeños proyectos</li>
  <li>Jugar videojuegos</li>
</ul>

<h3>Pasos para Crear un Archivo HTML (Lista Ordenada)</h3>
<ol>
  <li>Abrir el editor de texto.</li>
  <li>Escribir la estructura básica.</li>
  <li>Guardar con extensión `.html`.</li>
  <li>Abrir en el navegador.</li>
</ol>

<h3>Glosario Básico (Lista de Descripción)</h3>
<dl>
  <dt>HTML</dt>
  <dd>Lenguaje de marcado para la estructura de la web.</dd>
  <dt>Etiqueta</dt>
  <dd>Instrucción que le dice al navegador cómo mostrar el contenido.</dd>
</dl>
```

Ejercicio 1.6:

1. Crea una lista no ordenada (``) de tus tres comidas favoritas.
2. Crea una lista ordenada (``) con los tres pasos que seguirías para aprender una nueva habilidad.

1.7 Resumen y Ejercicio Integrador del Nivel 1

Resumen de Etiquetas Aprendidas:

| Etiqueta | Función | Nivel |
|---|--|--------|
| <code><!DOCTYPE html></code> | Declaración de tipo de documento (HTML5). | Básico |
| <code><html></code> | Contenedor raíz del documento. | Básico |
| <code><head></code> | Contiene metadatos (no visibles). | Básico |
| <code><body></code> | Contiene el contenido visible de la página. | Básico |
| <code><title></code> | Título de la pestaña del navegador. | Básico |
| <code><meta></code> | Metadatos (ej. codificación de caracteres). | Básico |
| <code><h1></code> a <code><h6></code> | Títulos y subtítulos jerárquicos. | Básico |
| <code><p></code> | Párrafo de texto. | Básico |
| <code> </code> | Salto de línea forzado. | Básico |
| <code></code> | Texto de fuerte importancia. | Básico |
| <code></code> | Texto enfatizado. | Básico |
| <code></code> , <code></code> , <code><dl></code> | Listas no ordenadas, ordenadas y de descripción. | Básico |
| <code></code> , <code><dt></code> , <code><dd></code> | Elementos de lista, término y descripción. | Básico |

Ejercicio Integrador del Nivel 1: Mi Primer Blog Post

Objetivo: Aplicar todos los conocimientos de estructura, jerarquía y formato de texto para crear el esqueleto de un artículo de blog.

Instrucciones:

1. Crea un nuevo archivo llamado `blog_post.html` con la estructura básica de HTML5 (`<!DOCTYPE html>` , `<html>` , `<head>` , `<body>`).
2. El `<title>` debe ser: "Mi Primer Artículo: Los Fundamentos de HTML".

3. Dentro del `<body>` , crea la siguiente estructura de contenido:

- Un `<h1>` con el título del artículo.
- Un párrafo `<p>` introductorio.
- Un `<h2>` para la primera sección: "La Importancia de la Estructura".
- Un párrafo `<p>` que contenga una frase clave resaltada con `` .
- Un `<h3>` para un subtema: "Etiquetas Esenciales".
- Una lista no ordenada (``) de 3 etiquetas esenciales de HTML.
- Un `<h2>` para la segunda sección: "Pasos para Empezar".
- Una lista ordenada (``) con 4 pasos para comenzar a codificar.
- Un `<blockquote>` con una cita sobre la importancia de aprender a programar.

Guía de Desarrollo (Solución Parcial):

Asegúrate de que tu código se vea limpio y bien indentado. La estructura general debe ser:

HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <!-- ... metadatos y título ... -->
</head>
<body>
  <!-- ... h1, p, h2, p con strong, h3, ul, h2, ol, blockquote ... -->
</body>
</html>
```

<!-- FIN DEL NIVEL 1 -->

2. Nivel 2: Intermedio - Enlaces, Imágenes y Multimedia

2.1 Enlaces (Hipervínculos)

Teoría:

Los **hipervínculos** son la esencia de la web, permitiendo la navegación entre diferentes páginas o recursos. La etiqueta fundamental para crear un enlace es `<a>` (anchor o ancla).

El atributo más importante de `<a>` es `href` (Hypertext REference), que especifica la URL de destino.

| Atributo | Descripción |
|---------------------|--|
| <code>href</code> | Obligatorio. Define la dirección (URL) a la que apunta el enlace. Puede ser una URL externa (<code>https://...</code>), una ruta a otro archivo local (<code>contacto.html</code>), o un ancla dentro de la misma página (<code>#seccion</code>). |
| <code>target</code> | Indica dónde se abrirá el recurso. El valor más común es <code>_blank</code> , que abre el enlace en una nueva pestaña o ventana. |
| <code>title</code> | Proporciona información adicional sobre el enlace, útil para accesibilidad y para mostrar un <i>tooltip</i> al pasar el ratón. |

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 2.1: Tipos de Enlaces -->
<body>
  <!-- Enlace externo que abre en una nueva pestaña -->
  <p>Visita el sitio de la <a href="https://www.w3.org/" target="_blank"
title="Estándares Web">W3C</a> para aprender más.</p>

  <!-- Enlace interno a otra página del sitio (debe existir el archivo) -->
  <p>Si quieres contactarnos, ve a la <a href="contacto.html">página de
contacto</a>.</p>

  <!-- Enlace a una sección dentro de la misma página (requiere un elemento
con id="seccion-final") -->
  <p>Ir a la <a href="#seccion-final">Sección Final</a>.</p>

  <!-- ... contenido de la página ... -->

  <h2 id="seccion-final">Sección Final</h2>
  <p>Aquí termina el contenido principal.</p>
</body>
```

Ejercicio 2.1:

1. Crea un archivo llamado `menu.html` .
2. Dentro del `<body>` , crea un párrafo con tres enlaces:
 - Un enlace a Google que se abra en una nueva pestaña.
 - Un enlace a tu archivo `mi_presentacion.html` (del Nivel 1).
 - Un enlace a una sección llamada "Acerca de" dentro del mismo `menu.html` (usa el atributo `id` en un `<h2>`).

2.2 Imágenes

Teoría:

La etiqueta `` se utiliza para incrustar una imagen en el documento. Es una etiqueta de **cierre automático** (no necesita ``).

| Atributo | Descripción |
|--|--|
| <code>src</code> | Obligatorio. Especifica la ruta (Source) de la imagen. Puede ser una URL o una ruta de archivo local. |
| <code>alt</code> | Obligatorio (por accesibilidad). Proporciona un texto alternativo que describe la imagen. Es leído por lectores de pantalla y se muestra si la imagen no carga. |
| <code>width</code> y <code>height</code> | Definen el ancho y alto de la imagen en píxeles. Es una buena práctica incluirlos para evitar saltos de diseño mientras la página carga. |

Rutas de Archivos:

- **Ruta Absoluta:** La URL completa de la imagen en internet (ej. `https://ejemplo.com/imagen.jpg`).
- **Ruta Relativa:** La ubicación de la imagen respecto al archivo HTML actual (ej. `imagenes/logo.png` si la imagen está en una carpeta llamada `imagenes`).

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 2.2: Inserción de Imágenes -->
<body>
  <!-- Imagen local con ruta relativa -->
```

```


<!-- Imagen externa con ruta absoluta -->


<!-- Una imagen que también es un enlace -->
<a href="https://www.w3.org/">
  
</a>
</body>
```

Ejercicio 2.2:

1. Busca una imagen en internet (o usa una de ejemplo como `https://picsum.photos/100/100`).
2. Inserta la imagen en tu archivo `mi_presentacion.html` .
3. Asegúrate de incluir un atributo `alt` descriptivo.
4. Convierte la imagen en un enlace que, al hacer clic, te lleve a tu perfil de red social favorito.

2.3 Contenedores Genéricos (`<div>` y ``)

Teoría:

`<div>` y `` son las etiquetas más genéricas y versátiles de HTML. Se utilizan para agrupar elementos y aplicarles estilos o manipularlos con JavaScript, sin añadir ningún significado semántico por sí mismos.

| Etiqueta | Tipo de Elemento | Uso Principal |
|---------------------------|-----------------------------|---|
| <code><div></code> | Bloque (Block-level) | Agrupar grandes secciones de contenido. Ocupa todo el ancho disponible y fuerza un salto de línea antes y después. |
| <code></code> | En Línea (Inline) | Agrupar pequeñas porciones de texto o elementos dentro de un bloque. Solo ocupa el espacio necesario y no fuerza saltos de línea. |

Nota: En HTML5, se recomienda usar etiquetas semánticas (como `<section>` , `<article>` , etc., que veremos en el Nivel 3) en lugar de `<div>` siempre que sea posible, para mejorar la estructura y accesibilidad.

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 2.3: Contenedores -->
<body>
  <!-- El div agrupa el encabezado y el párrafo como una unidad -->
  <div id="tarjeta-presentacion">
    <h1>Mi Nombre</h1>
    <p>Soy un desarrollador en formación.</p>
  </div>

  <p>
    Mi objetivo es aprender a usar <span class="resaltado">HTML</span>
    para estructurar el contenido y luego aplicar
    <span class="resaltado">CSS</span> para darle estilo.
  </p>
</body>
```

Ejercicio 2.3:

1. En tu `blog_post.html` , envuelve todo el contenido del artículo (desde el `<h1>` hasta el final) en un único `<div>` con el `id="articulo-principal"` .
2. Usa la etiqueta `` para resaltar la palabra "HTML5" en cualquier parte de tu texto.

2.4 Tablas

Teoría:

Las tablas se utilizan para mostrar datos tabulares (filas y columnas), como hojas de cálculo, horarios o listas de precios. **Nunca deben usarse para maquetar el diseño de una página** (para eso se usa CSS).

| Etiqueta | Descripción |
|--|--|
| <code><table></code> | Contenedor principal de la tabla. |
| <code><tr></code> | Define una fila (Table Row). |
| <code><th></code> | Define una celda de encabezado (Table Header). Se usa para títulos de columnas o filas. |
| <code><td></code> | Define una celda de datos (Table Data). |
| <code><thead></code> , <code><tbody></code> , <code><tfoot></code> | Semántica. Agrupan las filas en la cabecera, el cuerpo y el pie de la tabla, respectivamente. |

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 2.4: Tabla de Precios Semántica -->
<table border="1">
  <caption>Planes de Suscripción</caption>
  <thead>
    <tr>
      <th>Plan</th>
      <th>Características</th>
      <th>Precio</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Básico</th>
      <td>Acceso a 5 cursos</td>
      <td>$10/mes</td>
    </tr>
    <tr>
      <th>Premium</th>
      <td>Acceso ilimitado + soporte</td>
      <td>$25/mes</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="2">Total de planes</td>
      <td>2</td>
    </tr>
  </tfoot>
</table>
```

```
</tfoot>
</table>
```

Ejercicio 2.4:

1. Crea una tabla en un nuevo archivo llamada `horario.html`.
2. La tabla debe tener 3 columnas: "Día", "Materia" y "Hora".
3. Usa `<thead>` para los encabezados y `<tbody>` para al menos 3 filas de datos.
4. Añade un `<caption>` (título de la tabla) que diga "Mi Horario Semanal".

2.5 Multimedia

Teoría:

HTML5 simplificó la inserción de contenido multimedia con las etiquetas `<audio>` y `<video>`, eliminando la necesidad de *plugins* externos.

| Etiqueta | Uso | Atributos Clave |
|-----------------------------|--|---|
| <code><audio></code> | Incrusta contenido de audio. | <code>src</code> (ruta del archivo), <code>controls</code> (muestra los controles de reproducción), <code>autoplay</code> , <code>loop</code> . |
| <code><video></code> | Incrusta contenido de video. | <code>src</code> , <code>controls</code> , <code>autoplay</code> , <code>loop</code> , <code>poster</code> (imagen de previsualización), <code>width</code> , <code>height</code> . |
| <code><source></code> | Se usa dentro de <code><audio></code> o <code><video></code> para especificar múltiples fuentes de archivos (con diferentes formatos) para asegurar la compatibilidad con todos los navegadores. | |

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 2.5: Inserción de Multimedia -->
<body>
  <h3>Video de Introducción</h3>
```



```
<video width="320" height="240" controls poster="portada_video.jpg">
  <!-- Se recomienda usar múltiples formatos para compatibilidad -->
  <source src="video.mp4" type="video/mp4">
  <source src="video.webm" type="video/webm">
  <!-- Texto que se muestra si el navegador no soporta la etiqueta -->
  Tu navegador no soporta la etiqueta de video.
</video>

<h3>Música de Fondo</h3>
<audio controls loop>
  <source src="musica.mp3" type="audio/mpeg">
  Tu navegador no soporta la etiqueta de audio.
</audio>
</body>
```

Ejercicio 2.5:

1. Crea un archivo llamado `multimedia.html` .
2. Inserta un video de ejemplo (puedes usar un *placeholder* de video como `https://www.w3schools.com/html/mov_bbb.mp4` para el `src`).
3. Asegúrate de que el video tenga los atributos `controls` y `width` .
4. Debajo, inserta un elemento de audio que se reproduzca automáticamente (`autoplay`) y en bucle (`loop`).

2.6 Resumen y Ejercicio Integrador del Nivel 2

Resumen de Etiquetas Aprendidas:

| Etiqueta | Función | Nivel |
|--|--|------------|
| <code><a></code> | Crea un hipervínculo. | Intermedio |
| <code></code> | Incrusta una imagen. | Intermedio |
| <code><div></code> | Contenedor genérico de bloque. | Intermedio |
| <code></code> | Contenedor genérico en línea. | Intermedio |
| <code><table></code> | Contenedor principal de la tabla. | Intermedio |
| <code><tr></code> | Fila de la tabla. | Intermedio |
| <code><th></code> | Celda de encabezado. | Intermedio |
| <code><td></code> | Celda de datos. | Intermedio |
| <code><thead></code> , <code><tbody></code> , <code><tfoot></code> | Agrupación semántica de filas. | Intermedio |
| <code><audio></code> | Incrusta contenido de audio. | Intermedio |
| <code><video></code> | Incrusta contenido de video. | Intermedio |
| <code><source></code> | Especifica fuentes de archivos multimedia. | Intermedio |

Ejercicio Integrador del Nivel 2: Ficha de Producto

Objetivo: Crear la estructura de una ficha de producto para una tienda en línea, aplicando enlaces, imágenes, contenedores y tablas.

Instrucciones:

1. Crea un nuevo archivo llamado `ficha_producto.html` con la estructura básica.
2. Envuelve toda la ficha en un `<div>` principal con `id="ficha-producto"` .
3. Dentro, incluye:
 - Un `<h1>` con el nombre del producto.
 - Una imagen del producto (`` con `alt` y un enlace a una imagen de ejemplo).
 - Un párrafo `<p>` con una descripción.

- Una tabla (`<table>`) de "Especificaciones Técnicas" (ej. Peso, Dimensiones, Material). Usa `<thead>` y `<tbody>` .
- Un `<div>` para el precio, usando `` para resaltar el número.
- Un enlace (`<a>`) de "Añadir al Carrito".
- Un video (`<video>`) de demostración del producto.

Guía de Desarrollo (Solución Parcial):

Asegúrate de que la tabla use `<th>` para los encabezados de las filas o columnas, y que la imagen tenga un `alt` descriptivo.

HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <!-- ... -->
</head>
<body>
  <div id="ficha-producto">
    <!-- ... h1, img, p, table, div con span, a, video ... -->
  </div>
</body>
</html>
```

<!-- FIN DEL NIVEL 2 -->

3. Nivel 3: Avanzado - Semántica y Formularios

3.1 Semántica en HTML5

Teoría:

La **semántica** en HTML se refiere al significado de las etiquetas. Un código es semántico cuando las etiquetas que usas describen con precisión el tipo de contenido que contienen. Por ejemplo, usar `<h1>` para el título principal de la página es semántico, mientras que usar un `<div>` con un estilo grande no lo es.

¿Por qué es importante la semántica?

1. **Accesibilidad:** Los lectores de pantalla y otras tecnologías de asistencia se basan en la semántica para interpretar la estructura de la página a usuarios con discapacidades.
2. **SEO (Optimización para Motores de Búsqueda):** Los motores de búsqueda entienden mejor el contenido de tu página si está bien estructurado semánticamente.
3. **Mantenibilidad:** El código es más fácil de leer y mantener por otros desarrolladores.

3.2 Etiquetas Estructurales Semánticas

Teoría:

HTML5 introdujo nuevas etiquetas para reemplazar el uso excesivo de `<div>` y dar un significado claro a las diferentes secciones de una página web.

| Etiqueta | Uso Semántico |
|------------------------------|--|
| <code><header></code> | Contiene el contenido introductorio o de navegación de una sección o de toda la página (ej. logotipo, título, eslogan). |
| <code><nav></code> | Contiene enlaces de navegación principales (ej. menú principal, tabla de contenidos). |
| <code><main></code> | Contiene el contenido principal y único del documento. Solo debe haber uno por página. |
| <code><section></code> | Agrupar contenido temáticamente relacionado. Debe tener un encabezado (<code><h1></code> a <code><h6></code>). |
| <code><article></code> | Contenido independiente y auto-contenido (ej. un post de blog, un comentario, un widget). |
| <code><aside></code> | Contenido relacionado con el contenido principal, pero que puede ser considerado aparte (ej. barras laterales, publicidad, notas al margen). |
| <code><footer></code> | Contiene información de cierre de su sección o de la página (ej. derechos de autor, enlaces de contacto, información del autor). |

Ejemplo Práctico:

HTML

```

<!-- Ejemplo 3.2: Estructura Semántica de una Página -->
<body>
  <header>
    <h1>Mi Blog Personal</h1>
    <nav>
      <ul>
        <li><a href="/">Inicio</a></li>
        <li><a href="/acerca">Acerca de</a></li>
        <li><a href="/contacto">Contacto</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section>
      <h2>Últimos Artículos</h2>
      <article>
        <h3>Título del Post 1</h3>
        <p>Contenido del artículo...</p>
      </article>
      <article>
        <h3>Título del Post 2</h3>
        <p>Contenido del artículo...</p>
      </article>
    </section>

    <aside>
      <h3>Sobre el Autor</h3>
      <p>Breve biografía y enlaces sociales.</p>
    </aside>
  </main>

  <footer>
    <p>&copy; 2025 Basdonax AI. Todos los derechos reservados.</p>
  </footer>
</body>

```

Ejercicio 3.2:

1. Revisa tu archivo `blog_post.html` del Nivel 1.
2. Reemplaza los `<div>` genéricos por las etiquetas semánticas apropiadas:
 - El encabezado del blog debe ser un `<header>` que contenga el `<h1>`.
 - El menú de navegación (si lo tienes) debe estar dentro de un `<nav>`.
 - Todo el contenido principal debe estar dentro de `<main>`.

- El artículo en sí debe ser un `<article>` .
- Añade un `<footer>` con la información de derechos de autor.

3.3 Introducción a Formularios

Teoría:

Los formularios son la forma principal en que los usuarios interactúan con una página web para enviar datos (iniciar sesión, registrarse, enviar comentarios, etc.).

La etiqueta principal es `<form>` , que actúa como un contenedor para todos los campos de entrada.

| Atributo de <code><form></code> | Descripción |
|---------------------------------------|--|
| <code>action</code> | Obligatorio. Especifica la URL a la que se enviarán los datos del formulario cuando se envíe. |
| <code>method</code> | Especifica el método HTTP para enviar los datos. Los dos valores principales son: GET (los datos se añaden a la URL) y POST (los datos se envían en el cuerpo de la solicitud, más seguro para datos sensibles). |

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 3.3: Formulario Básico -->
<form action="/procesar_datos" method="post">
  <!-- Aquí irán los campos de entrada -->
  <p>Este es un formulario de ejemplo.</p>
</form>
```

Ejercicio 3.3:

1. Crea un nuevo archivo llamado `registro.html` .
2. Crea un formulario que envíe los datos a una URL simulada `/registro_usuario` usando el método `POST` .

3.4 Tipos de Input en HTML5

Teoría:

La etiqueta `<input>` es la más versátil en formularios. Su comportamiento cambia drásticamente según el atributo `type`.

| Tipo de <code>input</code> | Uso | Atributos Clave |
|----------------------------|--|---|
| <code>text</code> | Campo de texto simple (por defecto). | <code>placeholder</code> , <code>maxlength</code> |
| <code>email</code> | Campo para direcciones de correo. El navegador valida el formato. | <code>required</code> |
| <code>password</code> | Campo para contraseñas. Oculta el texto. | <code>minlength</code> |
| <code>number</code> | Campo para números. Muestra controles de incremento/decremento. | <code>min</code> , <code>max</code> , <code>step</code> |
| <code>date</code> | Selector de fecha. | |
| <code>checkbox</code> | Permite seleccionar cero o más opciones. | <code>checked</code> |
| <code>radio</code> | Permite seleccionar solo una opción de un grupo (deben tener el mismo <code>name</code>). | <code>checked</code> |
| <code>submit</code> | Botón para enviar el formulario. | <code>value</code> (texto del botón) |

Etiquetas de Apoyo:

- `<label>` : Asocia una descripción a un campo de entrada. Es vital para la accesibilidad. Se asocia usando el atributo `for` en `<label>` y el atributo `id` en `<input>`.

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 3.4: Tipos de Input -->
<form action="/login" method="post">
  <label for="usuario">Nombre de Usuario:</label>
  <input type="text" id="usuario" name="username" placeholder="Tu nombre"
  required>
```

```
<label for="clave">Contraseña:</label>
<input type="password" id="clave" name="password" minlength="8" required>
```

```
<label for="terminos">Acepto los términos:</label>
<input type="checkbox" id="terminos" name="terms" required>
```

```
<p>Método de Contacto:</p>
<input type="radio" id="email" name="contact_method" value="email" checked>
<label for="email">Correo Electrónico</label>
<input type="radio" id="telefono" name="contact_method" value="phone">
<label for="telefono">Teléfono</label>
```

```
<input type="submit" value="Iniciar Sesión">
</form>
```

Ejercicio 3.4:

1. En tu archivo `registro.html` , añade los siguientes campos al formulario:
 - Nombre completo (`text` , requerido).
 - Correo electrónico (`email` , requerido).
 - Contraseña (`password` , mínimo 8 caracteres).
 - Fecha de nacimiento (`date`).
 - Un grupo de botones de radio para seleccionar el género (Masculino, Femenino, Otro).
 - Un botón de envío (`submit`).
2. Asegúrate de usar `<label>` y asociarlos correctamente con el `id` de cada `<input>` .

3.5 Elementos de Formulario Avanzados

Teoría:

Además de `<input>` , existen otros elementos de formulario esenciales para la interacción avanzada.

| Etiqueta | Uso |
|-------------------------------|--|
| <code><textarea></code> | Campo de texto multi-línea para comentarios o mensajes largos. |
| <code><select></code> | Crea una lista desplegable (menú <i>dropdown</i>). |
| <code><option></code> | Define una opción dentro de un <code><select></code> . |
| <code><button></code> | Botón genérico. Puede ser de tipo <code>submit</code> , <code>reset</code> o <code>button</code> . |
| <code><fieldset></code> | Agrupar campos relacionados dentro de un formulario. |
| <code><legend></code> | Proporciona un título para el grupo de campos dentro de un <code><fieldset></code> . |

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 3.5: Elementos Avanzados -->
<form action="/encuesta" method="post">
  <fieldset>
    <legend>Información de Contacto</legend>
    <label for="mensaje">Tu Mensaje:</label>
    <textarea id="mensaje" name="message" rows="5" cols="40"></textarea>
  </fieldset>

  <fieldset>
    <legend>Selección de País</legend>
    <label for="pais">País:</label>
    <select id="pais" name="country">
      <option value="ar">Argentina</option>
      <option value="cl">Chile</option>
      <option value="co">Colombia</option>
      <option value="mx" selected>México</option>
    </select>
  </fieldset>

  <button type="submit">Enviar Encuesta</button>
```

```
<button type="reset">Limpiar Campos</button>
</form>
```

Ejercicio 3.5:

1. Crea un formulario de "Soporte Técnico" en un nuevo archivo `soporte.html`.
2. Usa un `<fieldset>` con `<legend>` para agrupar los campos de "Detalles del Problema".
3. Dentro del *fieldset*, incluye:
 - Un `<select>` para seleccionar el "Tipo de Problema" (ej. "Fallo de Pago", "Error de Carga", "Consulta General").
 - Un `<textarea>` para la "Descripción Detallada del Problema".
4. Añade un botón de envío (`submit`) y un botón de reinicio (`reset`).

3.6 Resumen y Ejercicio Integrador del Nivel 3

Resumen de Etiquetas Aprendidas:

| Etiqueta | Función | Nivel |
|---|--|----------|
| <code><header></code> , <code><nav></code> , <code><main></code> , <code><section></code> , <code><article></code> , <code><aside></code> , <code><footer></code> | Etiquetas estructurales semánticas de HTML5. | Avanzado |
| <code><form></code> | Contenedor principal del formulario. | Avanzado |
| <code><input></code> | Campo de entrada de datos (con varios <code>type</code>). | Avanzado |
| <code><label></code> | Etiqueta descriptiva asociada a un campo de entrada. | Avanzado |
| <code><textarea></code> | Campo de texto multi-línea. | Avanzado |
| <code><select></code> , <code><option></code> | Lista desplegable y sus opciones. | Avanzado |
| <code><button></code> | Botón genérico. | Avanzado |
| <code><fieldset></code> , <code><legend></code> | Agrupación de campos y su título. | Avanzado |

Ejercicio Integrador del Nivel 3: Esqueleto de un Periódico Digital

Objetivo: Aplicar la semántica de HTML5 para estructurar la página de inicio de un periódico digital.

Instrucciones:

1. Crea un nuevo archivo llamado `periodico.html` con la estructura básica.
2. Usa `<header>` para el encabezado del sitio (logotipo y título).
3. Usa `<nav>` para el menú principal (ej. Política, Deportes, Tecnología).
4. Usa `<main>` para el contenido principal. Dentro de `<main>` :
 - Crea una `<section>` para la "Noticia Principal" que contenga un `<article>` con un `<h1>` y un párrafo.
 - Crea otra `<section>` para "Noticias Secundarias" que contenga al menos dos `<article>` s más pequeños.
 - Usa un `<aside>` para una sección de "Publicidad" o "Lo Más Leído".
5. Añade un `<footer>` con enlaces a "Términos y Condiciones" y "Contacto".

Guía de Desarrollo (Solución Parcial):

La clave es la anidación correcta. Por ejemplo, el `<nav>` debe estar dentro del `<header>` o directamente en el `<body>` , y los `<article>` deben estar dentro de `<section>` o directamente en `<main>` .

HTML

```
<!DOCTYPE html>
<html lang="es">
<body>
  <header>...</header>
  <main>
    <section>...</section>
    <aside>...</aside>
  </main>
  <footer>...</footer>
</body>
</html>
```

<!-- FIN DEL NIVEL 3 -->

4. Nivel 4: Experto - Accesibilidad y Buenas

Prácticas

4.1 Accesibilidad Web (ARIA)

Teoría:

La **Accesibilidad Web** (Web Accessibility) significa que personas con discapacidades (visuales, auditivas, cognitivas, motoras) pueden percibir, entender, navegar e interactuar con la web. HTML, por sí mismo, es la base de la accesibilidad.

Principios Clave de Accesibilidad en HTML:

- Semántica Correcta:** Usar las etiquetas con el propósito para el que fueron diseñadas (ej. `<h1>` para un título, `<button>` para un botón).
- Atributo `alt` en Imágenes:** Siempre debe estar presente para que los lectores de pantalla describan la imagen.
- Atributo `lang`:** Especificar el idioma del documento (`<html lang="es">`) y de cualquier cambio de idioma dentro del texto.
- Asociación de `label` y `input`:** Usar `for` e `id` para que los usuarios de lectores de pantalla sepan a qué campo corresponde cada etiqueta.

ARIA (Accessible Rich Internet Applications):

ARIA es un conjunto de atributos especiales que se añaden a los elementos HTML para mejorar la accesibilidad de contenido dinámico o de elementos que no tienen un rol semántico nativo (ej. un `<div>` que se usa como botón).

| Atributo ARIA | Uso |
|-------------------------------|---|
| <code>role</code> | Define el propósito de un elemento (ej. <code>role="button"</code> , <code>role="alert"</code>). |
| <code>aria-label</code> | Proporciona una etiqueta de texto invisible para los lectores de pantalla cuando no hay una etiqueta visible. |
| <code>aria-describedby</code> | Proporciona una descripción más detallada para un elemento. |
| <code>aria-required</code> | Indica que un campo de formulario es obligatorio. |

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 4.1: Aplicando ARIA -->
<body>
  <!-- 1. Botón sin semántica nativa, pero con rol ARIA -->
  <div tabindex="0" role="button" aria-label="Cerrar ventana
emergente">X</div>

  <!-- 2. Campo de formulario con validación ARIA -->
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre" aria-required="true">

  <!-- 3. Enlace con etiqueta ARIA para mayor claridad -->
  <a href="/ayuda" aria-label="Ir a la página de ayuda y soporte">Ayuda</a>
</body>
```

Ejercicio 4.1:

1. Revisa tu formulario de registro (registro.html del Nivel 3).
2. Añade el atributo `aria-required="true"` a todos los campos obligatorios.
3. Si usaste un `<div>` para simular un botón de envío (aunque se recomienda `input type="submit"` o `<button>`), añádele `role="button"` y `tabindex="0"` para que sea navegable con teclado.

4.2 Buenas Prácticas de Codificación

Teoría:

Un código limpio y bien organizado es tan importante como un código funcional. Las buenas prácticas facilitan la lectura, el mantenimiento y la colaboración en proyectos.

| Práctica | Descripción |
|----------------------------|--|
| Indentación | Usar espacios (generalmente 2 o 4) para anidar etiquetas. Esto muestra claramente la jerarquía del documento. |
| Minúsculas | Todas las etiquetas y atributos deben estar en minúsculas (ej. <code><body></code> , no <code><BODY></code>). |
| Comentarios | Usar <code><!-- Comentario --></code> para explicar secciones complejas o dejar notas. |
| Cierre de Etiquetas | Asegurarse de que todas las etiquetas que lo requieren estén cerradas (ej. <code><div></code> con <code></div></code>). |
| Validación | Usar herramientas de validación (como el W3C Validator) para asegurar que el código cumple con el estándar HTML5. |

Ejemplo Práctico:

Mal Código (Difícil de leer):

HTML

```
<div><h1>Título</h1><p>Texto</p></div>
```

Buen Código (Fácil de leer y mantener):

HTML

```
<!-- Sección principal del contenido -->
<div id="contenido-principal">
  <h1>Título de la Sección</h1>
  <p>Este es un párrafo de texto bien indentado.</p>
</div>
```

Ejercicio 4.2:

1. Abre tu archivo `periodico.html` (del Nivel 3).
2. Revisa el código y asegúrate de que:
 - Cada etiqueta anidada esté correctamente indentada.
 - Añade un comentario al inicio de cada sección semántica (`<header>` , `<main>` , `<footer>`) explicando brevemente su contenido.

4.3 Optimización y Rendimiento

Teoría:

Un HTML optimizado se carga más rápido, lo que mejora la experiencia del usuario y el posicionamiento en buscadores.

1. Optimización del `head` :

- Colocar la etiqueta `<meta charset="UTF-8">` lo más cerca posible del inicio del `<head>`.
- Colocar los enlaces a CSS antes que los scripts de JavaScript (aunque esto es más relevante para CSS y JS, es una buena práctica de orden).

2. Imágenes con `loading="lazy"` :

- El atributo `loading="lazy"` en la etiqueta `` indica al navegador que no cargue la imagen hasta que el usuario esté cerca de verla. Esto acelera la carga inicial de la página.

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 4.3: Optimización de Carga -->
<head>
  <meta charset="UTF-8">
  <title>Página Optimizada</title>
  <!-- ... otros metadatos ... -->
</head>
<body>
  <!-- La imagen solo se cargará cuando el usuario haga scroll hasta ella -->
  

  <!-- Las imágenes importantes (como el logo) no deben tener lazy loading -->
  
</body>
```

Ejercicio 4.3:

1. Abre tu archivo `ficha_producto.html` (del Nivel 2).
2. Añade el atributo `loading="lazy"` a la etiqueta `` del producto.

4.4 Entidades HTML y Caracteres Especiales

Teoría:

Las **entidades HTML** son códigos que se utilizan para mostrar caracteres especiales que no están disponibles en un teclado estándar, o que tienen un significado especial en HTML (como `<` y `>`).

| Carácter | Nombre de la Entidad | Código Numérico | Uso |
|--------------------|-------------------------|--------------------------|--|
| <code><</code> | <code>&lt;</code> | <code>&#60;</code> | Símbolo "menor que" (para evitar que el navegador lo interprete como el inicio de una etiqueta). |
| <code>></code> | <code>&gt;</code> | <code>&#62;</code> | Símbolo "mayor que". |
| <code>&</code> | <code>&amp;</code> | <code>&#38;</code> | Símbolo "ampersand". |
| <code> </code> | <code>&nbsp;</code> | <code>&#160;</code> | Espacio de no separación (Non-breaking space). |
| <code>©</code> | <code>&copy;</code> | <code>&#169;</code> | Símbolo de Copyright. |
| <code>€</code> | <code>&euro;</code> | <code>&#8364;</code> | Símbolo de Euro. |

Ejemplo Práctico:

HTML

```
<!-- Ejemplo 4.4: Uso de Entidades -->
<footer>
  <p>&copy; 2025 - Todos los derechos reservados.</p>
  <p>Para escribir una etiqueta &lt;p> en el texto, debes usar
  &amp;lt;p&amp;gt;.</p>
</footer>
```

Ejercicio 4.4:

1. En el `<footer>` de tu archivo `periodico.html` , añade el símbolo de Copyright (`©`) y el año actual.

2. Escribe un párrafo que explique cómo se escribe la etiqueta `<h1>` en el código, usando las entidades `<` y `>` .

4.5 Resumen y Proyecto Integrador Final

Resumen de Conceptos Aprendidos:

| Concepto | Función | Nivel |
|------------------|--|---------|
| Accesibilidad | Hacer la web usable para todos. | Experto |
| ARIA | Atributos para mejorar la accesibilidad de elementos no semánticos. | Experto |
| Buenas Prácticas | Indentación, minúsculas, comentarios y validación. | Experto |
| Optimización | Uso de <code>loading="lazy"</code> y orden en el <code>head</code> . | Experto |
| Entidades HTML | Códigos para caracteres especiales. | Experto |

Proyecto Integrador Final: Página Personal Completa

Objetivo: Aplicar todos los conocimientos adquiridos (estructura, enlaces, imágenes, multimedia, semántica, formularios y accesibilidad) para construir una página web personal completa y profesional.

Instrucciones:

1. Crea un nuevo archivo llamado `portafolio.html` .
2. **Estructura Semántica:** Usa `<header>` , `<nav>` , `<main>` , `<section>` , `<article>` , `<aside>` y `<footer>` .
3. **Encabezado (`<header>`):**
 - `<h1>` con tu nombre.
 - `<nav>` con enlaces a las secciones principales (Inicio, Proyectos, Contacto).
4. **Sección Principal (`<main>`):**
 - Una `<section>` de "Acerca de Mí" con una imagen tuya (`` con `alt` y `loading="lazy"`) y una biografía en párrafos.

- Una `<section>` de "Proyectos" que contenga al menos dos `<article>` s, cada uno con un título (`<h2>`), una descripción y un enlace (`<a>`) al proyecto.
- Un `<aside>` con una lista no ordenada (``) de tus habilidades clave.

5. Formulario de Contacto:

- Una `<section>` de "Contacto" que contenga un `<form>` completo (nombre, email, mensaje en `<textarea>`).
- Asegúrate de usar `<label>` y atributos ARIA (`aria-required`).

6. Pie de Página (`<footer>`):

- Información de Copyright usando `©` .
- Un enlace a tu perfil de LinkedIn que se abra en una nueva pestaña (`target="_blank"`).

Guía de Desarrollo (Solución Final):

Este proyecto es la prueba de fuego. Revisa la jerarquía de títulos, la semántica de las etiquetas y la accesibilidad de los formularios. ¡Felicidades, has completado el manual!

HTML

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Portafolio de [Tu Nombre]</title>
</head>
<body>
  <header>
    <!-- ... nav y h1 ... -->
  </header>
  <main>
    <section id="acerca-de">
      <!-- ... img, p ... -->
    </section>
    <section id="proyectos">
      <!-- ... articles ... -->
    </section>
    <section id="contacto">
      <!-- ... form ... -->
    </section>
  </main>
  <footer>
    <!-- ... copyright y enlaces ... -->
  </footer>
```

```
</body>  
</html>
```

```
<!-- FIN DEL NIVEL 4 Y PROYECTO FINAL -->
```