

UNIDAD 2.

Desarrollo del Front end.

(El estudiante utilizará los elementos de una hoja de estilos y frameworks de hojas de estilos para la creación de un documento HTML.)

Tema I. Estructura del lenguaje de marcado de hipertexto (HTML).

Evidencia de aprendizaje Tema I -RESUMEN

Temas	Saber	Saber hacer	Resultado de Aprendizaje
Estructura del lenguaje de marcado de hipertexto (HTML)	Explicar los elementos de HTML5 para formar la estructura general de documentos: DOM (Document Object Model), etiquetas de presentación, etiquetas de estructura, metadatos, creación de formularios.	Implementar documentos HTML. Crear formularios HTML	Los estudiantes identifican los elementos para la estructura de un documento HTML.
Evidencia de aprendizaje			
A partir de un examen práctico con un tiempo definido, desarrollar una página web que contenga: el conjunto de páginas web estáticas y responsivas usando HTML, CSS, JS, Frameworks para el diseño responsivo.			

Información sobre el profesor

Profesor	Correo	Días de Clase
Bernardo Prado Díaz	Tenor_prado@yahoo.com.mx	Lunes y Martes

1. Explicación de HTML y su Estructura

HTML (HyperText Markup Language) no es un lenguaje de programación, sino un **lenguaje de marcado** que sirve para estructurar el contenido de una página web. Piensa en él como el esqueleto de un cuerpo humano. Define la jerarquía y el orden de todos los elementos: títulos, párrafos, imágenes, enlaces, etc.

Conceptos Clave:

- **DOM (Document Object Model):** Es la representación en memoria de tu documento HTML. Cuando tu navegador lee un archivo HTML, crea un árbol jerárquico de objetos, donde cada etiqueta (<p>, <h1>, etc.) es un nodo. JavaScript se usa para interactuar con este árbol, permitiendo modificar el contenido y el estilo de la página de forma dinámica.

- **Etiquetas de Estructura:** Son las que organizan el contenido de manera semántica, es decir, dándole un significado. Esto ayuda a los navegadores y a los motores de búsqueda a entender el propósito de cada sección.
 - `<header>`: Contiene la parte superior de la página (logo, barra de navegación).
 - `<main>`: El contenido principal y único de la página.
 - `<article>`: Contenido independiente y completo (ej. una entrada de blog).
 - `<section>`: Una sección temática de la página.
 - `<footer>`: La parte inferior (derechos de autor, enlaces de contacto).
 - `<div>`: Un contenedor genérico sin significado semántico, útil para agrupar elementos y darles estilo con CSS.
- **Etiquetas de Presentación:** Son las que definen el formato del texto.
 - `<h1>` a `<h6>`: Títulos y subtítulos.
 - `<p>`: Párrafos de texto.
 - `<a>`: Enlaces.
 - `<img>`: Imágenes.
 - `<strong>` y `<em>`: Texto importante y enfatizado.
- **Metadatos:** Información sobre la página que no se muestra al usuario. Se colocan en la sección `<head>`.
 - `<title>`: El título que aparece en la pestaña del navegador.
 - `<meta charset="UTF-8">`: Define la codificación de caracteres.
 - `<link rel="stylesheet" href="styles.css">`: Vincula un archivo CSS externo.
- **Formularios:** Permiten a los usuarios ingresar datos y enviarlos al servidor. Se crean con la etiqueta `<form>` y utilizan diferentes tipos de `<input>` (texto, contraseña, email, etc.) para cada campo.

2. Ejercicios Guiados con Código en Visual Studio Code

Sigue estos pasos para crear una página web completa, desde cero.

Configuración Inicial en Visual Studio Code

1. **Instalar Extensiones:** Abre Visual Studio Code, ve al menú de extensiones (icono de cubos) y busca e instala:
 - **Live Server:** Te permite ver los cambios en tu navegador en tiempo real sin tener que recargar la página.
 - **Prettier - Code formatter:** Formatea tu código automáticamente para que se vea limpio y ordenado.
2. **Crear el Archivo:** Crea una nueva carpeta para tu proyecto y, dentro de ella, un archivo llamado index.html.

Ejercicio 1: Estructura y Metadatos

Vamos a crear la estructura básica de una página.

HTML

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Mi Primera Página Web</title>
```

```
</head>
```

```
<body>
```

```
  <h1>¡Hola, Mundo!</h1>
```

```
  <h2>Bienvenidos a mi sitio web</h2>
```

```
  <p>
```

Esta es la estructura básica de una página web creada con HTML.

El título principal está definido con ``<h1>`` y este párrafo con ``<p>``.

```
  </p>
```

```
</body>
```

```
</html>
```

Para ejecutar el código: Haz clic derecho en el archivo index.html en el explorador de archivos de VS Code y selecciona **"Open with Live Server"**. Se abrirá tu navegador y podrás ver la página en vivo.

Ejercicio 2: Creación de un Formulario de Contacto

Ahora, dentro del <body>, vamos a crear un formulario simple.

HTML

```
<body>

  <h1>¡Hola, Mundo!</h1>

  <h2>Bienvenidos a mi sitio web</h2>

  <p>Esta es la estructura básica de una página web.</p>

  <section>

    <h3>Formulario de Contacto</h3>

    <form action="#" method="get">

      <label for="nombre">Nombre completo:</label><br>

      <input type="text" id="nombre" name="nombre" required><br><br>

      <label for="email">Correo electrónico:</label><br>

      <input type="email" id="email" name="email" required><br><br>

      <label for="mensaje">Tu mensaje:</label><br>

      <textarea id="mensaje" name="mensaje" rows="4" cols="50"></textarea><br><br>

      <input type="submit" value="Enviar">

    </form>

  </section>

</body>
```

- **Comentarios del Código:**

- `<form>`: La etiqueta principal que define el formulario.
- `<label>`: Etiqueta descriptiva para el campo de entrada. El atributo `for` lo vincula al `id` del `input`.
- `<input type="text">`: Un campo de entrada de texto de una sola línea.
- `<input type="email">`: Un campo especializado para correos electrónicos.
- `<textarea>`: Un campo de texto multilínea para mensajes más largos.
- `<input type="submit">`: El botón que envía el formulario.
- `required`: Un atributo para validar que el campo debe ser completado antes de enviar el formulario.

Con estos ejercicios, has creado un documento HTML con la estructura básica, etiquetas de presentación y un formulario funcional, cumpliendo con los objetivos de tu unidad de aprendizaje.

1. Explicación Completa de los Conceptos Fundamentales de HTML

El **HTML** (HyperText Markup Language) es el cimiento de cualquier página web. Imagina que estás construyendo una casa; el HTML es la estructura básica: las paredes, los techos, las puertas. No es un lenguaje de programación, sino un lenguaje de marcado que utiliza "etiquetas" para definir la estructura y el significado de tu contenido.

DOM (Document Object Model)

El **DOM** no es parte del código HTML en sí, sino una representación de tu página en la memoria del navegador. Cuando el navegador carga tu archivo HTML, lo convierte en un modelo de árbol jerárquico, donde cada etiqueta (`<p>`, `<h1>`, etc.) es un "nodo". Este modelo permite a lenguajes como **JavaScript** interactuar con la página, por ejemplo, para cambiar el texto de un párrafo o mostrar un elemento oculto.

Etiquetas de Estructura

Estas etiquetas dan **semántica** al contenido, es decir, le dan un significado. Esto es crucial para la accesibilidad y para los motores de búsqueda.

- `<!DOCTYPE html>`: Le dice al navegador que este es un documento HTML5.
- `<html>`: El contenedor principal de todo el documento.

- `<head>`: Contiene los metadatos de la página (información que no se muestra al usuario).
 - `<body>`: Contiene el contenido visible para los usuarios.
 - `<header>`: La sección de cabecera de la página (logo, navegación).
 - `<main>`: El contenido principal y único de la página.
 - `<section>`: Agrupa contenido relacionado dentro del `<main>`.
 - `<footer>`: El pie de página (información de derechos de autor, contacto).
-

Etiquetas de Presentación

Estas etiquetas se usan para dar formato al contenido. Aunque **CSS** es el método moderno para el estilo, estas etiquetas definen la presentación básica del texto.

- `<h1>` a `<h6>`: Encabezados o títulos, donde `<h1>` es el más importante.
 - `<p>`: Define un párrafo de texto.
 - `<a>`: Un enlace a otra página o recurso.
 - `<img>`: Inserta una imagen en la página.
-

Metadatos

Son datos sobre el documento. Se colocan dentro de la etiqueta `<head>`.

- `<title>`: El título que aparece en la pestaña del navegador.
 - `<meta charset="UTF-8">`: Declara la codificación de caracteres, crucial para que los acentos y la "ñ" se muestren correctamente.
 - `<meta name="viewport" content="...">`: Asegura que la página se vea bien en dispositivos móviles.
-

Creación de Formularios

Los formularios son esenciales para interactuar con el usuario. Permiten recolectar información como nombres, correos electrónicos, comentarios, etc.

- `<form>`: El contenedor de todos los elementos del formulario.

- `<input>`: Un elemento de entrada de datos. El atributo `type` define su función (por ejemplo, `type="text"`, `type="email"`, `type="password"`, `type="submit"` para el botón de envío).
 - `<label>`: Proporciona una etiqueta para un campo del formulario. El atributo `for` la asocia al `id` del `input` para mejorar la accesibilidad.
 - `<textarea>`: Un campo de entrada de texto de varias líneas, ideal para comentarios largos.
-

2. Ejercicio Guiado con Código para Visual Studio Code

A continuación, encontrarás un documento HTML completo que incluye todos los conceptos que hemos discutido. Sigue las instrucciones paso a paso para crear el archivo en Visual Studio Code.

Instrucciones de Configuración en VS Code:

1. **Instala las extensiones:** Si no las tienes, ve al menú de extensiones (el icono de cubos a la izquierda) y busca e instala "**Live Server**" y "**Prettier - Code formatter**".
2. **Crea un archivo:** Crea una nueva carpeta para tu proyecto y, dentro de ella, un archivo llamado `index.html`.
3. **Copia y pega el código:** Copia el siguiente bloque de código completo en tu archivo `index.html`.
4. **Inicia Live Server:** Haz clic derecho en cualquier parte del código en VS Code y selecciona "**Open with Live Server**". Se abrirá tu página web en el navegador, y los cambios que hagas se reflejarán automáticamente.

`<!--`

EJERCICIO GUIADO DE HTML

(Resumen del Tema I: Estructura del lenguaje de marcado de hipertexto)

Este código incluye comentarios detallados para ayudarte a entender cada parte de la estructura HTML.

Puedes ejecutarlo con la extensión "Live Server" de Visual Studio Code.

`-->`

<!--

Declaración del tipo de documento:

Informa al navegador que este es un documento HTML5. Es la primera línea y es obligatoria.

-->

<!DOCTYPE html>

<!--

Etiqueta <html>:

Es la raíz de todo el documento HTML. Todo el contenido de la página va dentro de ella.

El atributo 'lang="es"' es importante para la accesibilidad y para los motores de búsqueda, indicando que el idioma principal del documento es el español.

-->

<html lang="es">

<!--

Etiqueta <head>:

Contiene metadatos, es decir, información sobre el documento que no se muestra en la página en sí.

Piensa en esto como la "información del cerebro" de la página.

-->

<head>

<!--

Metadatos - Codificación de caracteres:

Este es uno de los metadatos más importantes. 'charset="UTF-8"' asegura que todos los caracteres

(como la 'ñ', tildes y otros símbolos) se muestren correctamente.

-->

<meta charset="UTF-8">

<!--

Metadatos - Adaptación para dispositivos:

Esto asegura que la página se vea bien y sea "responsive" en todos los dispositivos, ajustando el tamaño de la ventana (viewport) al ancho del dispositivo.

-->

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!--

Metadatos - Título de la página:

El texto dentro de <title> es lo que aparece en la pestaña del navegador.

-->

<title>Estructura de un Documento HTML</title>

</head>

<!--

Etiqueta <body>:

Contiene todo el contenido visible de la página: textos, imágenes, enlaces, etc.

-->

<body>

<!--

DOM (Document Object Model):

El navegador lee este código y crea un "árbol" de nodos, donde <body> es la raíz de todo el contenido visible.

Cada etiqueta (<h1>, <p>, etc.) es un nodo en ese árbol.

-->

<!--

Etiquetas de Estructura:

Organizan el contenido de manera semántica.

-->

<header>

<!--

Etiqueta de Presentación - Título principal:

Usa <h1> para el título principal de la página.

-->

<h1>Conceptos Clave de HTML</h1>

<nav>

<!-- Esto sería un menú de navegación, pero lo dejamos simple por ahora -->

Presentación

Formulario

</nav>

</header>

<main>

<!--

Etiqueta de Estructura - Sección temática:

`<section>` agrupa contenido relacionado, en este caso, la explicación de los conceptos.

-->

`<section id="presentacion">`

`<h2>Introducción a las Etiquetas</h2>`

`<!--`

Etiquetas de Presentación - Párrafo:

`<p>` es para un párrafo de texto.

-->

`<p>`

El lenguaje HTML utiliza etiquetas para estructurar el contenido de una página web.

La mayoría de las etiquetas tienen una apertura y un cierre, como `<code><p></code>` y `<code></p></code>`.

`</p>`

`<!--`

Etiquetas de Presentación - Enlace:

`<a>` crea un hipervínculo. El atributo 'href' especifica el destino del enlace.

-->

`<p>`

Para más información, visita la `<a href="https://developer.mozilla.org/es/docs/Web/HTML" target="_blank">`documentación de MDN`</a>`.

`</p>`

`<h3>Etiquetas de Presentación Comunes</h3>`

```
<ul>

  <li><strong>&lt;p&gt;</strong>: Párrafos</li>

  <li><strong>&lt;h1&gt;</strong> a <strong>&lt;h6&gt;</strong>: Títulos y
  subtítulos</li>

  <li><strong>&lt;a&gt;</strong>: Enlaces</li>

  <li><strong>&lt;img&gt;</strong>: Imágenes</li>

  <li><strong>&lt;strong&gt;</strong>: Texto en negritas, con énfasis
  importante</li>

</ul>
```

```
<!--

Etiqueta de Presentación - Imagen:

 inserta una imagen. 'src' es la fuente y 'alt' es el texto alternativo
para cuando la imagen no se puede cargar o para lectores de pantalla.

-->



</section>
```

```
<!--

Etiquetas de Estructura - Sección para el formulario:

Creamos una nueva sección para el formulario para mantener el código organizado.

-->

<section id="formulario">

  <h2>Formulario de Contacto</h2>

  <!--
```

Creación de Formularios:

<form> es el contenedor principal. Los atributos 'action' y 'method' determinan a dónde se envían los datos y cómo.

-->

```
<form action="#" method="get">
```

<!--

Etiqueta <label>:

Asocia un texto a un campo de entrada. El atributo 'for' debe coincidir con el 'id' del <input> para una buena práctica de accesibilidad.

-->

```
<label for="nombre">Nombre:</label><br>
```

<!--

Etiqueta <input>:

Crea un campo para la entrada de datos. El atributo 'type' define el tipo de dato. 'required' hace que el campo sea obligatorio.

-->

```
<input type="text" id="nombre" name="nombre" required><br><br>
```

```
<label for="email">Email:</label><br>
```

```
<input type="email" id="email" name="email" required><br><br>
```

```
<label for="mensaje">Mensaje:</label><br>
```

<!--

Etiqueta <textarea>:

Campo para texto de varias líneas.

-->

```
<textarea id="mensaje" name="mensaje" rows="4"
cols="50"></textarea><br><br>
```

```
<!--
```

Etiqueta <input type="submit">:

Crea un botón para enviar los datos del formulario.

```
-->
```

```
<input type="submit" value="Enviar Mensaje">
```

```
</form>
```

```
</section>
```

```
</main>
```

```
<!--
```

Etiqueta de Estructura - Pie de página:

Contiene información como derechos de autor o enlaces de contacto.

```
-->
```

```
<footer>
```

```
<p>&copy; 2025 Mi Primer Documento HTML. Todos los derechos reservados.</p>
```

```
</footer>
```

```
</body>
```

```
</html>
```

1) Elementos esenciales de HTML5 (lo mínimo que deben dominar)

a) Nivel documento

- `<!DOCTYPE html>` — activa el modo estándar (HTML5).
- `<html lang="es">` — raíz del DOM + idioma del documento.
- `<head>` — metadatos (no visibles):
 - `<meta charset="UTF-8">` — codificación.
 - `<meta name="viewport" content="width=device-width, initial-scale=1.0">` — responsivo.
 - `<title>` — título de pestaña.
 - `<meta name="description" content="...">` — SEO.
 - `<link rel="icon" href="...">`, `<link rel="stylesheet" href="...">` — favicon y CSS.
 - `<script src="..."></script>` — JS (si aplica).
- `<body>` — contenido visible.

a) Nivel documento

`<!DOCTYPE html>`

- Qué: Indicador de que el documento es HTML5.
- Cuándo: Siempre al inicio del archivo.
- Cómo: `<!DOCTYPE html>`
- Dónde: Primera línea.
- Ejemplo (base 1):

`<!DOCTYPE html>`

`<html lang="es">`

`</html>`

`<html lang="es">`

- Qué: Raíz del documento; lang define el idioma para SEO/lectores de pantalla.
- Cuándo: Siempre; ajusta lang al idioma real.

- **Cómo:** `<html lang="es"> ... </html>`
- **Dónde:** Envuelve todo.
- **Ejemplo (base 2):**

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
  <!-- aquí irán head y body -->
```

```
</html>
```

```
<head>
```

- **Qué:** Metadatos (no visibles).
- **Cuándo:** Siempre.
- **Cómo:** Contiene meta, title, link, script...
- **Dónde:** Dentro de `<html>`, antes de `<body>`.
- **Ejemplo (base 3):**

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>Sitio de Prueba</title>
```

```
  <meta name="description" content="Ejemplo educativo de HTML5">
```

```
  <link rel="icon" href="favicon.ico">
```

```
  <link rel="stylesheet" href="styles.css">
```

```
  <!-- <script src="app.js" defer></script> (si hay JS) -->
```

```
</head>
```

Notas rápidas:

- **charset="UTF-8":** acentos correctos.
- **viewport:** responsive en móviles.
- **title:** título de pestaña (único por documento).

- **description:** para buscadores.
- **link rel="icon":** ícono de pestaña.
- **link rel="stylesheet":** CSS externo.
- **script defer:** carga JS sin bloquear el render.

<body>

- **Qué:** Contenido visible.
- **Cuándo:** Siempre.
- **Cómo/Dónde:** Después de **<head>**.
- **Ejemplo (base 4 completo):**

<!DOCTYPE html>

<html lang="es">

<head> ... (metadatos anteriores) ... </head>

<body>

<!-- aquí va lo visible -->

</body>

</html>

b) Estructura semántica

- **Contenedores:** **<header>**, **<nav>**, **<main>**, **<section>**, **<article>**, **<aside>**, **<footer>**.
- **Titulares:** **<h1>...<h6>**.
- **Texto y citas:** **<p>**, ****, ****, **<blockquote>**, **<q>**, **<cite>**, **<mark>**, **<code>**, **<pre>**, **<kbd>**, **<abbr>**, **<time>**, **<small>**, **
, **<hr>.

c) Navegación y vínculos

- **** (externo o interno con **#id**), atributos: **target**, **download**, **rel**.

d) Imágenes y medios

- **** (el **alt** es obligatorio por accesibilidad).

- `<figure>`, `<figcaption>`.
- `<picture>` + `<source>` (imágenes responsivas).
- `<audio>`, `<video>`, `<track>` (subtítulos).

e) Listas y tablas

- Listas: `<ul>`, `<ol>`, `<li>`, y listas de definiciones `<dl>`, `<dt>`, `<dd>`.
- Tablas: `<table>`, `<caption>`, `<thead>`, `<tbody>`, `<tfoot>`, `<tr>`, `<th>`, `<td>`.

f) Formularios (núcleo del tema)

- Contenedor: `<form action="" method="post|get">`.
- Etiquetas: `<label for="id">`.
- Controles:
 - `<input type="text|email|password|number|date|file|checkbox|radio|range|hidden|search|tel|url|color">`
 - `<textarea>`, `<select>``<option>`, `<datalist>`, `<button>`, `<output>`, `<progress>`, `<meter>`.
- Atributos HTML de validación: `required`, `min`, `max`, `step`, `minlength`, `maxlength`, `pattern`, `placeholder`, `autocomplete`, `autofocus`, `multiple`, `checked`, `value`, `name`.
- Agrupación y accesibilidad: `<fieldset>`, `<legend>`.

g) Embeds y gráficas

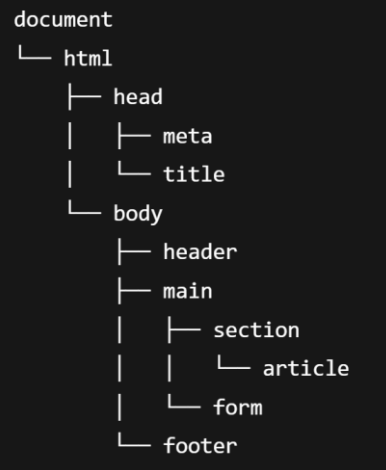
- `<iframe>`, `<embed>`, `<object>`, `<canvas>`, `<svg>`.

h) Atributos globales

- `id`, `class`, `title`, `style`, `lang`, `dir`, `tabindex`, `hidden`, `data-*`, `aria-*` (accesibilidad).

2) Mapa rápido del DOM (cómo “piensa” el navegador)

El navegador convierte el HTML en un árbol. Ejemplo:



Entender este árbol ayuda a ubicarse y a referenciar elementos (por id/class o por jerarquía).

3) Prácticas guiadas y resueltas (paso a paso)

Todas tienen comentarios por línea para que el alumno entienda qué hace cada parte.

Puedes copiarlas tal cual y ejecutar en el navegador.

Práctica 1 — Estructura mínima del documento

Objetivo: crear la plantilla base correcta.

```
<!DOCTYPE html> <!-- Activa HTML5 en modo estándar -->

<html lang="es"> <!-- Idioma del documento para SEO y lectores de pantalla -->

<head> <!-- Encabezado con metadatos -->

  <meta charset="UTF-8"> <!-- Soporta caracteres latinos/acentos -->

  <meta name="viewport" content="width=device-width, initial-scale=1.0"> <!--
Escala en móviles -->

  <title>Plantilla Base</title> <!-- Título de la pestaña -->

  <meta name="description" content="Plantilla base para HTML5"> <!-- Descripción
SEO -->

</head>

<body> <!-- Contenido visible -->

  <h1>Hola, HTML5</h1> <!-- Encabezado principal único por página -->

  <p>Documento mínimo listo para empezar.</p> <!-- Párrafo de texto -->

</body>

</html>
```

Práctica 2 — Layout semántico básico

Objetivo: estructurar la página con etiquetas semánticas.

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Estructura Semántica</title>

</head>

<body>

  <header> <!-- Cabecera del sitio -->

    <h1>Mi Sitio</h1> <!-- Título principal -->

    <nav aria-label="principal"> <!-- Navegación principal con etiqueta ARIA -->

      <a href="#inicio">Inicio</a> <!-- Enlace interno a sección -->

      <a href="#articulos">Artículos</a>

      <a href="#contacto">Contacto</a>

    </nav>

  </header>


  <main id="inicio"> <!-- Contenido principal único -->

    <section> <!-- Bloque temático -->

      <h2>Bienvenida</h2> <!-- Subtítulo -->

      <p>Este sitio demuestra estructura semántica HTML5.</p> <!-- Texto -->

    </section>


    <section id="articulos"> <!-- Otra sección temática -->
```

```
<h2>Artículos</h2>

<article> <!-- Contenido independiente -->

  <h3>Cómo estructurar HTML</h3>

  <p>Usa etiquetas semánticas para mejorar accesibilidad y SEO.</p>

  <aside> <!-- Contenido relacionado -->

    <p><strong>Tip:</strong> Un solo <code>&lt;h1&gt;</code> por página.</p>

  </aside>

</article>

</section>

</main>


<footer id="contacto"> <!-- Pie de página -->

  <address> <!-- Información de contacto -->

    Escribe a <a href="mailto:profesor@ejemplo.com">profesor@ejemplo.com</a>

  </address>

  <small>© 2025 Mi Sitio</small> <!-- Nota legal pequeña -->

</footer>

</body>

</html>
```

Práctica 3 — Texto, listas, enlaces e imagen con figura

Objetivo: trabajar texto semántico, listas y gráficos con alt.

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Contenido y Medios</title>

</head>

<body>

  <main>

    <h1>Recursos del Curso</h1> <!-- Título de la página -->

    <p>Consulta la <strong>guía</strong> y los <em>ejercicios</em> semanales.</p>
    <!-- Énfasis semántico -->

    <h2>Lista de temas</h2>

    <ol> <!-- Lista ordenada -->

      <li>HTML semántico</li>

      <li>CSS responsivo</li>

      <li>Formularios</li>

    </ol>

    <h2>Enlaces útiles</h2>

    <ul> <!-- Lista no ordenada -->

      <li><a href="https://developer.mozilla.org/es/" target="_blank"
rel="noopener">MDN Web Docs</a></li> <!-- Enlace externo -->
```

`<li><a href="#imagen">Ir a la imagen</a></li> <!-- Enlace interno -->`

`</ul>`

`<figure id="imagen"> <!-- Contenedor gráfico con título -->`

`
<!-- Siempre con alt -->`

`<figcaption>Figura 1. Imagen demostrativa para pruebas.</figcaption> <!-- Pie de
figura -->`

`</figure>`

`<p>Publicado el <time datetime="2025-08-22">22 de agosto de 2025</time>.</p>
<!-- Fecha semántica -->`

`</main>`

`</body>`

`</html>`

Práctica 4 — Tabla accesible

Objetivo: maquetar datos tabulares con encabezados y leyenda.

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Tabla Accesible</title>

</head>

<body>

  <main>

    <h1>Horario de Laboratorio</h1> <!-- Título descriptivo -->

    <table> <!-- Tabla de datos -->

      <caption>Turnos de uso por día</caption> <!-- Título de tabla -->

      <thead> <!-- Encabezado -->

        <tr> <!-- Fila -->

          <th scope="col">Día</th> <!-- Cabecera de columna -->

          <th scope="col">Mañana</th>

          <th scope="col">Tarde</th>

        </tr>

      </thead>

      <tbody> <!-- Cuerpo -->

        <tr>

          <th scope="row">Lunes</th> <!-- Cabecera de fila -->

          <td>8-12</td>

          <td>14-18</td>
```

```
</tr>

<tr>

  <th scope="row">Martes</th>

  <td>8-12</td>

  <td>14-18</td>

</tr>

</tbody>

<tfoot> <!-- Pie opcional -->

<tr>

  <td colspan="3">* Sujeto a cambios.</td> <!-- Celda que abarca 3 columnas -->

</tr>

</tfoot>

</table>

</main>

</body>

</html>
```

Práctica 5 — Formulario HTML con validación sin JS

Objetivo: crear un formulario accesible usando solo atributos HTML.

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Formulario con Validación HTML</title>

</head>

<body>

  <main>

    <h1>Registro de Usuario</h1>

    <form action="" method="post"> <!-- Formulario (acción vacía para pruebas) -->

      <fieldset> <!-- Agrupa campos relacionados -->

        <legend>Datos personales</legend> <!-- Título del grupo -->

        <label for="nombre">Nombre:</label> <!-- Etiqueta asociada al input -->

        <input id="nombre" name="nombre" type="text" required minlength="2"
placeholder="Tu nombre"> <!-- Requerido y mínimo 2 -->

        <label for="correo">Correo:</label>

        <input id="correo" name="correo" type="email" required
placeholder="usuario@dominio.com" autocomplete="email"> <!-- Validación de
email -->

        <label for="pass">Contraseña:</label>

        <input id="pass" name="pass" type="password" minlength="6"
placeholder="Mínimo 6 caracteres"> <!-- Longitud mínima -->
```

<label for="nacimiento">Fecha de nacimiento:</label>

<input id="nacimiento" name="nacimiento" type="date" min="1900-01-01" max="2025-12-31"> <!-- Rango permitido -->

</fieldset>

<fieldset>

<legend>Preferencias</legend>

<label for="rol">Rol:</label>

<select id="rol" name="rol" required> <!-- Lista desplegable -->

<option value="">Seleccione...</option> <!-- Opción nula -->

<option>Alumno</option>

<option>Profesor</option>

<option>Invitado</option>

</select>

<label>Suscripción:</label>

<label><input type="radio" name="sub" value="mensual" checked> Mensual</label> <!-- Radio (exclusivo) -->

<label><input type="radio" name="sub" value="anual"> Anual</label>

<label><input type="checkbox" name="terminos" required> Acepto términos</label> <!-- Debe marcarse -->

</fieldset>

<button type="submit">Registrar</button> <!-- Envío -->

<button type="reset">Limpiar</button> <!-- Reseteo -->

</form>

</main>

</body>

</html>

Práctica 6 — datalist, output, details/summary, progress/meter

Objetivo: usar controles nativos menos conocidos.

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Controles HTML Nativos</title>

</head>

<body>

  <main>

    <h1>Controles útiles</h1>

    <label for="lenguaje">Lenguaje favorito:</label>

    <input list="langs" id="lenguaje" name="lenguaje" placeholder="Escribe o elige">
    <!-- Caja con sugerencias -->

    <datalist id="langs"> <!-- Lista de opciones -->

      <option value="HTML">

      <option value="CSS">

      <option value="JavaScript">

      <option value="Python">

    </datalist>

    <p>Puntuación del perfil:</p>

    <progress value="60" max="100"></progress> <!-- Barra de progreso -->
```

`<p>Calidad del código:</p>`

`<meter value="0.75" min="0" max="1" low="0.4" high="0.8" optimum="0.9"></meter> <!-- Indicador semántico -->`

`<details> <!-- Desplegable nativo -->`

`<summary>Ver recomendaciones</summary>`

`<p>Usa etiquetas semánticas, valida formularios y escribe `alt` en imágenes.</p>`

`</details>`

`<form oninput="total.value = parseInt(a.value||0) + parseInt(b.value||0)"> <!--
Calcula sin JS externo -->`

`<label>Valor A <input id="a" type="number" min="0" value="5"></label>`

`<label>Valor B <input id="b" type="number" min="0" value="3"></label>`

`<output name="total">8</output> <!-- Muestra resultado -->`

`</form>`

`</main>`

`</body>`

`</html>`

4) Resultado de aprendizaje (cómo lo demuestras)

Al finalizar estas prácticas el alumno puede:

- Identificar y explicar la estructura general de un documento HTML5 (Saber).
- Implementar documentos HTML semánticos y crear formularios con validación nativa (Saber hacer).
- Entregar una página con layout, contenido, tabla accesible e inputs con validación, pasando una lista de cotejo clara (Resultado de aprendizaje).

5) Mini lista de cotejo (Tema I)

- Usa `<!DOCTYPE html>`, `<html lang>`, `<meta charset>`, `<meta viewport>`, `<title>`.
- Estructura con `<header>`, `<nav>`, `<main>`, `<section>/<article>`, `<footer>`.
- Imágenes con `alt` + `figure/figcaption` cuando aplique.
- Tablas con `caption`, `thead/tbody/tfoot`, `th` con `scope`.
- Formularios con `label` `for`, `name` en cada control y validación HTML (`required`, `type`, etc.).
- Uso razonable de elementos semánticos de texto y microcomponentes (`time`, `abbr`, `code`).
- Buenas prácticas de accesibilidad básica (`aria-label/aria-*` cuando se justifique).

a) Nivel documento

`<!DOCTYPE html>`

- **Qué:** Indicador de que el documento es **HTML5**.
- **Cuándo:** Siempre al inicio del archivo.
- **Cómo:** `<!DOCTYPE html>`
- **Dónde:** Primera línea.
- **Ejemplo (base 1):**

`<!DOCTYPE html>`

```
<html lang="es">
```

```
</html>
```

```
<html lang="es">
```

- **Qué:** Raíz del documento; lang define el idioma para SEO/lectores de pantalla.
- **Cuándo:** Siempre; ajusta lang al idioma real.
- **Cómo:** <html lang="es"> ... </html>
- **Dónde:** Envuelve todo.
- **Ejemplo (base 2):**

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<!-- aquí irán head y body -->
```

```
</html>
```

```
<head>
```

- **Qué:** Metadatos (no visibles).
- **Cuándo:** Siempre.
- **Cómo:** Contiene meta, title, link, script...
- **Dónde:** Dentro de <html>, antes de <body>.
- **Ejemplo (base 3):**

```
<head>
```

```
<meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Sitio de Prueba</title>
```

```
<meta name="description" content="Ejemplo educativo de HTML5">
```

```
<link rel="icon" href="favicon.ico">
```

```
<link rel="stylesheet" href="styles.css">
```

```
<!-- <script src="app.js" defer></script> (si hay JS) -->
```

</head>

Notas rápidas:

- charset="UTF-8": acentos correctos.
- viewport: responsive en móviles.
- title: título de pestaña (único por documento).
- description: para buscadores.
- link rel="icon": ícono de pestaña.
- link rel="stylesheet": CSS externo.
- script defer: carga JS sin bloquear el render.

<body>

- **Qué:** Contenido visible.
- **Cuándo:** Siempre.
- **Cómo/Dónde:** Después de <head>.
- **Ejemplo (base 4 completo):**

```
<!DOCTYPE html>
```

```
<html lang="es">
```

```
<head> ... (metadatos anteriores) ... </head>
```

```
<body>
```

```
<!-- aquí va lo visible -->
```

```
</body>
```

```
</html>
```

b) Estructura semántica

Contenedores semánticos

- **<header>**: cabecera de sitio/sección (logo, título, navegación).
- **<nav>**: bloque con **enlaces de navegación** principales.

- **<main>: contenido principal** (solo uno por página).
- **<section>:** agrupación temática (un bloque con título).
- **<article>:** contenido **autocontenido** (post, tarjeta, noticia).
- **<aside>:** contenido relacionado (barras laterales, avisos).
- **<footer>:** pie de página (créditos, contacto).

Ejemplo (suma a la base):

```
<body>
```

```
  <header>
```

```
    <h1>Mi Sitio</h1>
```

```
    <nav aria-label="principal">
```

```
      <a href="#inicio">Inicio</a>
```

```
      <a href="#articulos">Artículos</a>
```

```
      <a href="#contacto">Contacto</a>
```

```
    </nav>
```

```
  </header>
```

```
  <main id="inicio">
```

```
    <section>
```

```
      <h2>Bienvenida</h2>
```

```
      <p>Este sitio demuestra estructura semántica HTML5.</p>
```

```
    </section>
```

```
    <section id="articulos">
```

```
      <h2>Artículos</h2>
```

```
      <article>
```

```
        <h3>Cómo estructurar HTML</h3>
```

```
<p>Usa etiquetas semánticas para accesibilidad y SEO.</p>

<aside>Tip: un solo <code>&lt;h1&gt;</code> por página.</aside>

</article>

</section>

</main>


<footer id="contacto">

  <small>© 2025 Mi Sitio</small>

</footer>

</body>
```

Titulares h1...h6

- **Qué:** jerarquía de títulos.
- **Regla: 1 h1 por documento;** luego h2 para secciones, h3 para subsecciones, etc.

Texto y citas

- p párrafos; strong (énfasis fuerte, semántico), em (énfasis).
- blockquote cita larga (puede llevar cite="URL"), q cita corta en línea.
- cite (fuente/obra), mark resaltar, code fragmento de código, pre preserva espacios/formatos, kbd entrada de teclado.
- abbr abreviaturas con title, time datetime="YYYY-MM-DD", small notas pequeñas.
- br salto de línea **puntual** (no para espaciado), hr separación temática.

Ejemplo sumando:

```
<section>

<h2>Texto y Citas</h2>

<p>Lee la <strong>guía</strong> y las <em>instrucciones</em>.</p>

<blockquote cite="https://developer.mozilla.org/">

  "Escribe HTML semántico para mejorar la accesibilidad."
```

</blockquote>

<p>Atajo: presiona <kbd>Ctrl</kbd> + <kbd>S</kbd> para guardar.</p>

<p>Publicado el <time datetime="2025-08-22">22/08/2025</time>.</p>

</section>

c) Navegación y vínculos

- **Externo:** target="_blank" + rel="noopener noreferrer".
- **Interno:** a #id dentro de la misma página.
- **Descarga:** download.

Ejemplos:

MDN

Ir al pie de página

Descargar guía PDF

d) Imágenes y medios

img

- **Regla de oro:** alt **descriptivo**; si decorativa, alt="".

figure + figcaption

- Para agrupar imagen y pie.

<figure>

<figcaption>Figura 1. Crecimiento 2025 Q1.</figcaption>

</figure>

picture (responsivo)

```
<picture>

  <source media="(min-width: 800px)" srcset="banner-large.jpg">

  <source media="(min-width: 400px)" srcset="banner-medium.jpg">

</picture>
```

audio / video / track

```
<video controls width="480">

  <source src="demo.mp4" type="video/mp4">

  <track kind="captions" src="subtitulos.vtt" srclang="es" label="Español">

  Tu navegador no soporta video.

</video>
```

Evita autoplay sin muted; **siempre** ofrece controles y subtítulos.

e) Listas y tablas

Listas

```
<h2>Temas</h2>

<ol>

  <li>HTML semántico</li>

  <li>CSS responsivo</li>

  <li>Formularios</li>

</ol>

<dl>

  <dt>DOM</dt><dd>Árbol de nodos del documento.</dd>

  <dt>SEO</dt><dd>Optimización para motores de búsqueda.</dd>
```

</dl>

Tablas (datos, no layout)

- Usa caption, thead, tbody, tfoot.
- th con scope="col|row" mejora accesibilidad.

<table>

<caption>Horario de Laboratorio</caption>

<thead>

<tr><th scope="col">Día</th><th scope="col">Mañana</th><th scope="col">Tarde</th></tr>

</thead>

<tbody>

<tr><th scope="row">Lunes</th><td>8-12</td><td>14-18</td></tr>

<tr><th scope="row">Martes</th><td>8-12</td><td>14-18</td></tr>

</tbody>

<tfoot>

<tr><td colspan="3">* Sujeto a cambios.</td></tr>

</tfoot>

</table>

f) Formularios (núcleo)

Contenedor y envío

- form action="" method="post|get" (si no hay backend, deja action="" para pruebas).
- **Cada control debe tener name** (para enviar datos).
- **Usa label for="id"** (accesibilidad).

<form action="" method="post">

<fieldset>

<legend>Registro</legend>

<label for="nombre">Nombre</label>

<input id="nombre" name="nombre" type="text" required minlength="2"
placeholder="Tu nombre">

<label for="correo">Correo</label>

<input id="correo" name="correo" type="email" required autocomplete="email">

<label for="pass">Contraseña</label>

<input id="pass" name="pass" type="password" minlength="6" placeholder="Mínimo 6">

<label for="rol">Rol</label>

<select id="rol" name="rol" required>

<option value="">Seleccione...</option>

<option value="alumno">Alumno</option>

<option value="profesor">Profesor</option>

</select>

<p>Suscripción:</p>

<label><input type="radio" name="plan" value="mensual" checked> Mensual</label>

<label><input type="radio" name="plan" value="anual"> Anual</label>

<label><input type="checkbox" name="terminos" required> Acepto términos</label>

<!-- Controles especiales -->

<label for="color">Color favorito</label>

<input id="color" name="color" type="color">

<label for="rango">Experiencia (0-10)</label>

<input id="rango" name="experiencia" type="range" min="0" max="10" step="1">

<button type="submit">Registrar</button>

</fieldset>

</form>

Validación HTML (sin JS)

- required, min, max, step, minlength, maxlength, pattern.

<input type="text" required pattern="[A-Za-zÁÉÍÓÚÑáéíóúñ]{2,40}" title="Solo letras y espacios (2-40)">

datalist, output, progress, meter

<input list="langs" name="lenguaje" placeholder="HTML, CSS...">

<datalist

id="langs"><option>HTML</option><option>CSS</option><option>JS</option></datalist>

<form oninput="total.value = parseInt(a.value||0) + parseInt(b.value||0)">

<input id="a" type="number" value="5"> +

<input id="b" type="number" value="3"> =

<output name="total">8</output>

</form>

<progress value="60" max="100"></progress>

<meter value="0.7" min="0" max="1" low="0.4" high="0.8" optimum="0.9"></meter>

Buenas prácticas de formularios:

- **Etiqueta siempre visible** (label) → el placeholder **no** sustituye al label.
 - fieldset/legend para agrupar.
 - name en cada input (si se envía).
 - Usa tipos correctos (email, date, number...) para validación nativa y móviles.
-

g) Embeds y gráficas

iframe

- Incrusta otra página (mapa, video...). Usa **title** y permisos.

<iframe

src="https://www.youtube.com/embed/VIDEO_ID"

title="Video explicativo de HTML"

width="560" height="315"

allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture"

allowfullscreen>

</iframe>

- Para seguridad extra en orígenes no confiables: sandbox y referrerpolicy.

embed / object

- Heredadas; hoy se usan poco. Prefiere iframe.

canvas vs svg

- **Canvas:** dibujo **raster** por JS (gráficos dinámicos, juegos).
 - **SVG:** **vectorial** y **accesible** (íconos, diagramas, logos).
-

h) Atributos globales

- id: **único** por documento. Para JS/ancoras.
- class: estilos/selección múltiple.

- title: tooltip (útil con cuidado).
- style: **evitar** inline; usa CSS externo.
- lang/dir: idioma/dirección (rtl en árabe).
- tabindex: orden de tabulación (úsalo con moderación).
- hidden: oculta semánticamente.
- data-*: datos personalizados para JS.
- aria-*: accesibilidad; **no reemplaza** HTML semántico.

Ejemplo:

```
<button id="btnComprar" class="btn primario" data-producto-id="A12" aria-label="Comprar producto A12">
```

Comprar

```
</button>
```

2) Mapa rápido del DOM (y cómo interactuar)

El navegador crea un **árbol** con nodos. Puedes **seleccionarlos** y **manipular** con JS:

```
<button id="saludoBtn">Saludar</button>
```

```
<p id="salida"></p>
```

```
<script>
```

```
// Busca nodos por id / selectores CSS
```

```
const btn = document.getElementById('saludoBtn');
```

```
const out = document.querySelector('#salida');
```

```
// Maneja eventos
```

```
btn.addEventListener('click', () => {
```

```
  out.textContent = '¡Hola DOM!';
```

```
});
```

```
</script>
```

Checklist de uso y errores comunes

- ☒ `<!DOCTYPE html>`, lang, charset, viewport, title presentes.
 - ☒ **Un solo** `<main>` y **un solo** `h1`.
 - ☒ Imágenes con alt adecuado (o `alt=""` si decorativas).
 - ☒ Tablas solo para **datos**, con caption y scope.
 - ☒ Formularios con label for, name, y atributos de validación.
 - ☒ Evitar `<br>/<hr>` para “dar espacio” (usa CSS).
 - ☒ No abusar de aria-* si ya hay **HTML semántico** que cubre el caso.
-

Mini-ejercicio incremental (todo junto)

Pide a los alumnos que integren:

1. Cabecera con nav (3 enlaces),
2. main con dos section (texto+cita y lista),
3. figure con figcaption,
4. Tabla accesible,
5. Formulario con validación nativa,
6. Un svg simple,
7. Un botón con JS que actualice un p en el DOM.

Si quieres, te lo empaqueto como **plantilla base** y lo agrego a tu carpeta de materiales (HTML + CSS + JS) para que lo usen tal cual en clase.