Logotipo

El contenido generado por IA puede ser incorrecto.

Santiago Yosa González

**Ingeniero Sistemas de Información**



Logotipo

El contenido generado por IA puede ser incorrecto.}

**Introducción a HTML5**

* ¿Qué es HTML?

Editores de código

Editor para HTML

* Etiquetas estructurales sitio HTML5:

doctype, html, head, meta, title, body

* Etiquetas semanticas de contenido:

header, nav, (div), section, footer

* Etiquetas de contenido:

h1-h6, article, p, listas, enlaces, etc.

**Unidad I**

**Maquetación de páginas Web**

**Situación**

**Analogía del Arquitecto y el Esqueleto Humano**

Imaginemos que vamos a construir un edificio o el cuerpo humano. HTML no es el color de las paredes (eso es CSS), ni es la electricidad o el ascensor (eso es JavaScript).

**HTML es la Arquitectura y el Esqueleto:**

* **Arquitectura (Diseño):** Es el plano que define dónde va la cocina ($<header>$), dónde están las habitaciones ($<section>$) y dónde está el sótano ($<footer>$). Define la estructura de la información.
* **Esqueleto Humano:** Los huesos ($<h1>$, $<p>$, $<nav>$) son los que dan forma y soporte al cuerpo. Si quitamos los músculos (CSS), el esqueleto sigue siendo funcional y nos dice qué parte del cuerpo es cada hueso.

**Pregunta Clave:** ¿Cómo nos aseguramos de que una computadora o un lector de pantalla (para personas con discapacidad visual) entienda que una sección de texto es el menú principal y no solo un párrafo grande?

**¿Qué es HTML (HyperText Markup Language)?**

* No es un lenguaje de programación, es un **lenguaje de marcado** (Markup Language).
* Su función es la **estructuración semántica** del contenido.
* Evolución: De HTML 4 a **HTML5** (el estándar actual que introdujo etiquetas semánticas y APIs avanzadas).

**El Ecosistema: Editores de Código vs. Editores WYSIWYG.**

* **Editores de Código (IDE/Editor de Texto):** VS Code, Sublime Text, Atom. Permiten control total, *syntax highlighting*, *autocompletado* (IntelliSense) y extensiones. **Es el estándar profesional.**
* **Editor para HTML:** Cualquier editor de texto plano (Bloc de notas/TextEdit) puede usarse, pero carece de herramientas profesionales.

**Anatomía del Código Base HTML5 (Estructura Mínima Viable):**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Título en la Pestaña del Navegador</title>

</head>

<body>

> </body>

</html>

**Etiquetas Estructurales del Sitio (La Arquitectura del Documento)**

**<!DOCTYPE html>**

No es una etiqueta, es una declaración. Es la primera línea y le dice al navegador que el documento es HTML5. Dato Avanzado: En versiones previas era mucho más complejo.

Fuentes:

[Developer mozilla](https://developer.mozilla.org/es/docs/Web/HTML)

[w3schools](https://www.w3schools.com/html/)

[Lenguajehtml](https://lenguajehtml.com/)

**<html lang="es">...</html>**

El elemento raíz de la página. Todo lo demás va dentro. Siempre incluye el atributo lang para especificar el idioma y ayudar a los motores de búsqueda y lectores de pantalla (SEO y Accesibilidad).

**<head>...</head>**

Contiene metadatos (datos sobre los datos). Esta información no es visible en el cuerpo de la página, pero es vital para el navegador, buscadores y redes sociales.

**<meta charset="UTF-8"><meta name="viewport" content="width=device-width, initial-scale=1.0">**

Etiqueta de metadatos autocerrada. Clave para la modernidad: charset="UTF-8" (codificación de caracteres), viewport (clave para la responsividad).

**<title>Mi Página Web Profesional</title>**  
Define el texto que aparece en la pestaña del navegador y es el título principal que usan los motores de búsqueda. Dato Avanzado: Títulos bien escritos son cruciales para el SEO (Search Engine Optimization).

**<body>... Contenido visible ...</body>**  
Contiene todo el contenido visible del sitio (texto, imágenes, videos, formularios). Es el cuerpo físico de la página.

**Etiquetas Semánticas Estructurales de Contenido (HTML5)**

Antes de HTML5, toda la estructura interna se hacía con etiquetas $<div>$ (divisiones). HTML5 introdujo etiquetas con **significado**, lo que mejora la Accesibilidad y el SEO.

* <header><h1>Mi Marca</h1><nav>...</nav></header>

Contenido introductorio o un grupo de ayudas de navegación, usualmente incluye el logo y el $<h1>$ principal. **No confundir** con el <head>.

* \<nav\>\<ul\>\<li\>\<a href="\#inicio"\>Inicio\</a\>\</li\>...\</ul\>\</nav\>

Contiene enlaces de navegación (links) importantes a otras páginas o secciones. **Dato Avanzado:** Solo para los enlaces **principales**. Los enlaces del pie de página no necesitan necesariamente un ` $<\nav>$$.

**<header>** Representa la cabecera de una página o sección. Normalmente contiene el logo, el título de la página y la navegación.

**<nav>** Representa una sección de navegación en la página. Contiene enlaces a otras partes del sitio o a páginas externas.

**<div>** Se usa para agrupar elementos y aplicar estilos con CSS. Se utiliza cuando no hay una etiqueta más específica disponible.

**<section>** Representa una sección temática dentro de una página. Puede agrupar varios elementos relacionados, como artículos o subsecciones. Se recomienda usarla cuando el contenido tiene un propósito en sí mismo.

**<footer>** Representa el pie de página de una sección o de toda la web.

**Etiquetas de Contenido (Tipografía y Multimedia)**

**<h1>Título Principal</h1><h2>Subtítulo de la Sección</h2>**

**Encabezados:** Representan la jerarquía de los títulos. **Regla de Oro (SEO/Accesibilidad):** Solo debe existir **un único** $<h1>$ por página, que es el título principal. $<h2>$ son subtítulos de $<h1>$, $<h3>$ subtítulos de $h2$, y así sucesivamente.

**<p>Este es un párrafo de texto.</p>**

**Párrafo:** Define bloques de texto. Los navegadores añaden automáticamente márgenes alrededor.

**<ul><li>Item 1</li><li>Item 2</li></ul>**

**Listas:** $<ul>$ (Lista desordenada, viñetas), <ol> (Lista ordenada, números/letras), $<li>$ (Item de la lista). **Dato Avanzado:** Las listas son la mejor manera de estructurar **cualquier conjunto de datos** (menús de navegación, pasos, *features*).

**<a href="https://google.com" target="\_blank">Google</a>**

**Enlace de Ancla (Anchor):** Crea hipervínculos. El atributo clave es href (destino del enlace). **Tipos de enlaces avanzados:** Externos (URL completa), Internos (rutas relativas), Anclas (a una sección de la misma página, Ej: <a href="#contacto">), Correo (mailto:).

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Ejemplo HTML</title>

</head>

<body>

<header>

<h1>Mi Blog de Tecnología</h1>

<nav>

<ul>

<li><a href="#">Inicio</a></li>

<li><a href="#">Artículos</a></li>

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

</ul>

</nav>

</header>

<section>

<article>

<h2>Introducción a HTML</h2>

<p>HTML es el lenguaje de marcado estándar para crear páginas web.</p>

</article>

<article>

<h2>Lista de tecnologías web</h2>

<ul>

<li>HTML</li>

<li>CSS</li>

<li>JavaScript</li>

</ul>

</article>

</section>

<footer>

<p>&copy; 2025 Mi Blog. Todos los derechos reservados.</p>

</footer>

</body>

</html>

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Ejemplo de estructura HTML</title>

</head>

<body>

<header>

<h1>Mi Sitio Web</h1>

<nav>

<ul>

<li><a href="#">Inicio</a></li>

<li><a href="#">Servicios</a></li>

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

</ul>

</nav>

</header>

<section>

<h2>Sobre Nosotros</h2>

<p>Somos un equipo apasionado por la tecnología y el desarrollo web.</p>

</section>

<div class="contenedor">

<h2>Nuestros Servicios</h2>

<p>Ofrecemos desarrollo de sitios web y aplicaciones.</p>

</div>

<footer>

<p>&copy; 2025 Mi Sitio Web. Todos los derechos reservados.</p>

</footer>

</body>

</html>

**¿Diferencias entre estructurales y semánticas?**

El marcado estructural es el que estructura el diseño de la página. Por ejemplo, una div etiqueta o span (o cualquier otra etiqueta) con CSS que la coloca en algún lugar específico de la página o que representa los elementos secundarios en una estructura determinada. Por lo tanto, el marcado estructural es como el andamiaje en el que se colocan otras etiquetas. Las etiquetas con propiedades CSS como display:flex, display:grid son estructurales. Por ejemplo, un div que representa los elementos secundarios de izquierda a derecha (en lugar de arriba a abajo por defecto) o un p elemento con el valor predeterminado display:block es una pieza estructural de marcado.

El marcado semántico es un marcado en el que el nombre de la etiqueta indica algo sobre el contenido. Por ejemplo, ul indica que los elementos secundarios son una lista desordenada, li indica un elemento de lista, p o h1 las etiquetas indican el rol jerárquico del texto en el documento, aside indica que el contenido no forma parte del contenido principal de la página, nav indica que el elemento contiene enlaces a otros lugares de la página o del sitio web.

**Algunas Etiquetas no se Cierran con </>**

<link> → Se usa para enlazar hojas de estilo CSS u otros recursos externos.

<img> → Inserta imágenes en la página web.

<br> → Inserta un salto de línea.

<hr> → Inserta una línea horizontal de separación.

<link rel="stylesheet" href="css/estilos.css"> <!-- Enlace a CSS >

<script src="js/script.js"></script> <!-- Enlace a JavaScript -->

<link rel="icon" type="image/x-icon" href="imagen">

**¿Cuál es la diferencia real entre usar <div> y <\section> o <\article>?**

La diferencia es el **significado semántico**. Un <div> es una caja vacía; no le dice nada al navegador sobre su contenido. Una <\section> le dice al navegador, a los lectores de pantalla y a los motores de búsqueda que el contenido es un **bloque temático coherente**. Usar la etiqueta correcta (semántica) mejora la accesibilidad (los lectores de pantalla pueden saltar directamente al <\main> o <\nav>) y el SEO (los buscadores entienden mejor el contexto de tu contenido).

**Si solo puedo usar un $<h1>$, ¿cómo diferencio el título principal en varias secciones de mi página?**

El <h1> representa la idea **principal de toda la página**. Las demás secciones deben usar <h2> (si son secciones principales) o <h3> (si son subsecciones temáticas). Por ejemplo:

* + <h1>Página de Servicios de la Agencia XYZ</h1> (Único en la página).
  + <section><h2>Servicios de Diseño</h2>...</section>
  + <section><h2>Portafolio</h2><article><h3>Proyecto 1: Rediseño Web</h3>...</article></section>

La jerarquía de h1 a h6 debe ser lógica y continua.

**Crear nuestro primer HTML**

Crear nuestra carpeta donde vamos a guardar nuestro trabajo:

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Luego Clic derecho a la carpeta abrir con Code

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Luego clic en File – New File o espacio del Explorer, en el icono de new file.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto. Tabla

El contenido generado por IA puede ser incorrecto.

**Importante:**

Los archivos que destinamos para HTML, les debemos agregarla extensión .html

**Buenas prácticas para nombres de archivos HTML**

**Usar solo letras, números, guiones y guiones bajos**

✅ Correcto: index.html, pagina-contacto.html, mi\_archivo.html

❌ Incorrecto: página.html (puede generar problemas por la tilde)

**Evitar espacios en los nombres**

✅ Correcto: sobre\_nosotros.html o sobre-nosotros.html

❌ Incorrecto: sobre nosotros.html (algunos servidores no lo manejan bien)

**No usar caracteres especiales ni acentos**

✅ Correcto: proyectos.html

❌ Incorrecto: proyectos#.html, mis\_páginas.html

**Mantener todo en minúsculas** (buena práctica)

✅ Correcto: productos.html

❌ Incorrecto: Productos.html o PRODUCTOS.html (algunos servidores son sensibles a mayúsculas)

**Extensión siempre .html o .htm**

✅ Correcto: index.html

❌ Incorrecto: index.HTML (evitar mayúsculas en la extensión)

**El archivo principal debe llamarse index.html**

Los navegadores buscan automáticamente index.html en una carpeta o dominio.

✅ Correcto: index.html

❌ Incorrecto: home.html (a menos que se configure en el servidor)

**Usar nombres descriptivos y claros**

✅ Correcto: contacto.html, servicios.html

❌ Incorrecto: x1.html, a.html (nombres confusos)

**Evitar nombres demasiado largos**

✅ Correcto: productos.html

❌ Incorrecto: listado-de-productos-de-la-tienda-en-linea.html

**Práctica Acompañada**

**Práctica 1: Creación de la Estructura Semántica Base del Proyecto Integrador**

**Instrucciones**

Vamos a iniciar el esqueleto de nuestra **Landing Page**. Sigue cada paso para asegurar que la arquitectura de tu documento sea robusta y semántica.

**Paso a Paso:**

1. **Configuración Inicial:**
   * Crea una carpeta llamada portafolio\_semantico.
   * Dentro, crea el archivo index.html.
   * En VS Code, escribe ! y presiona Tab para generar la estructura básica.
   * Asegúrate de que el atributo lang sea es.
   * Cambia el $<\title>$ a "Mi Portafolio - [Tu Nombre]".
2. **Creación de las Secciones Principales (Arquitectura):**
   * Dentro del $<body>$, crea las cuatro secciones principales que conformarán tu Landing Page, usando las etiquetas estructurales de HTML5.

<body>

<header>

</header>

<main>

<section id="presentacion">

</section>

<section id="habilidades">

</section>

<section id="proyectos">

</section>

</main>

<footer>

</footer>

</body>

**Relleno Semántico Básico (Contenido):**

* **Dentro de $<\header>$:**
  + Añade un $<h1>$ con tu nombre o título profesional (Ej: "Juan Pérez - Desarrollador Web").
  + Crea una etiqueta $<\nav>$ y dentro de ella una lista desordenada (<ul>) con tres ítems (<li>) que serán enlaces de ancla: **Inicio** (href="#presentacion"), **Habilidades** (href="#habilidades"), **Proyectos** (href="#proyectos").
* **Dentro de $<\section\ id="presentacion">$:**
  + Añade un $<h2>$ (Ej: "¡Hola! Soy Juan, enfocado en Front-End.")
  + Añade un párrafo (<p>) de bienvenida.

**Solución de la Práctica 1**

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

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

<title>Mi Portafolio - Juan Pérez</title>

</head>

<body>

<header>

<h1>Juan Pérez - Desarrollador Web</h1>

<nav>

<ul>

<li><a href="#presentacion">Inicio</a></li>

<li><a href="#habilidades">Habilidades</a></li>

<li><a href="#proyectos">Proyectos</a></li>

</ul>

</nav>

</header>

<main>

<section id="presentacion">

<h2>¡Hola! Soy Juan, enfocado en Front-End.</h2>

<p>Especialista en la creación de interfaces de usuario accesibles y optimizadas.</p>

</section>

<section id="habilidades">

<h2>Mis Habilidades Técnicas</h2>

</section>

<section id="proyectos">

<h2>Últimos Proyectos Destacados</h2>

</section>

</main>

<footer>

<p>&copy; 2025 Juan Pérez. Todos los derechos reservados.</p>

<p>Contacto: <a href="mailto:juan@ejemplo.com">juan@ejemplo.com</a></p>

</footer>

</body>

</html>

**Práctica 2: Refuerzo Independiente - Jerarquía y Contenido**

Ahora, enfoquemonos en la sección **"Proyectos"** (<section id="proyectos">). Tu tarea es añadir la jerarquía de títulos y el contenido de tres proyectos recientes, asegurándote de usar las etiquetas más semánticas.

**Paso a Paso a Seguir para el Estudiante:**

1. Abre la sección <section id="proyectos"> de tu código anterior.
2. Dentro, crea **tres** contenedores <article> independientes, uno para cada proyecto (Post 1, Post 2, Post 3).
3. Dentro de **cada** <article>:
   * Añade el título del proyecto usando un encabezado **apropiado** (debe ser un subtítulo del $<h2>$ de la sección, por lo tanto, usa $<h3>$).
   * Añade un párrafo (<p>) con una breve descripción del proyecto.
   * Añade una **lista desordenada** (<ul>) con las tecnologías usadas (Ej: HTML, CSS, Git).
   * Añade un enlace (<a>) que diga "Ver Proyecto" que apunte a una URL ficticia (Ej: #).

**Ejercicio Independiente - Estructura de un Artículo de Noticias**

Los estudiantes deben crear una página HTML separada (no la del portafolio) que contenga la estructura semántica de un artículo de noticia complejo.

1. Crea un nuevo archivo llamado noticia.html.
2. Asegura la estructura básica de HTML5 (<!DOCTYPE html>, <html>, etc.).
3. El contenido del $<body>$ debe ser completamente semántico:
   * Usar un $<\header>$ para el título de la noticia.
   * Usar un $<\main>$ para contener el cuerpo principal.
   * Usar un $<\article>$ para contener **todo** el texto de la noticia.
   * Dentro del $<\article>$:
     + Usar el $<h1>$ para el título principal de la noticia.
     + Usar $<h2>$ para la bajada o el subtítulo.
     + Usar <p> para los párrafos de texto.
     + Usar una $<\aside>$ para agregar un bloque de "Noticias Relacionadas" o "Sobre el Autor".
     + Usar un $<footer>$ **dentro del `$<\article>$** para incluir la fecha de publicación y los derechos del autor del artículo.

**Criterios de Logro Esperados:**

* El documento debe validar como HTML5 (sin errores de sintaxis).
* Solo debe existir **un** $<h1>$ en toda la página.
* La etiqueta $<\aside>$ debe estar fuera del cuerpo principal del artículo ($<\p>) pero dentro del $<\main>$ o dentro del $<\article>$ (dependiendo si el contenido es relevante o indirectamente relevante). **La meta es que elijan y justifiquen su ubicación.**
* El uso de $<\article>$, $<\main>$, $<\header>$ y $<footer>$ debe ser coherente con el significado semántico que se revisó en clase.

**¿Debería saber algo más?**

**Modelo de Bloque vs. Modelo en Línea (display: block vs. display: inline):**

* **Bloque:** Ocupa todo el ancho disponible y fuerza un salto de línea antes y después (Ej: <div> , <\p> , , \<\\section\> \).
* **En Línea:** Solo ocupa el ancho necesario para su contenido y permite que otros elementos lo sigan en la misma línea (Ej: <a> , <span>`, <img> ).
* **Dato Avanzado:** Esta es la base de todo layout con CSS. La etiqueta <div> es por defecto block, y <span> es por defecto inline (el comodín en línea sin semántica).

**Atributos Globales Clave:**

* **id vs. class:**
  + **id:** Identificador **único** para un solo elemento en todo el documento. Se usa para fragmentos de URL (#seccion) o JavaScript.
  + **class:** Identificador que puede ser usado por **múltiples** elementos. Se usa principalmente para aplicar estilos CSS. **Uso Avanzado:** En la práctica profesional, se prefiere usar class sobre id para CSS.

Logotipo

El contenido generado por IA puede ser incorrecto.