GUIA DE APRENDIZAJE HTML

**Nombre:** Deyvid Alexander Ballesteros Chávez

**3.1 Actividades de reflexión inicial:**

**1. Investiga qué es HTML y su papel en el desarrollo web.**

-El html es una combinación de hipertexto y lenguaje de marcado para diseñar páginas web, lo que permite crear textos, manipular textos y definir la estructura de la pagina mediante etiquetas para definir que cambios se quieren hacer en el texto y que se vean reflejados en la página.

**2. Reflexiona sobre la diferencia entre HTML y otros lenguajes de programación.**

-El html es un lenguaje de etiquetado para definir la estructura y diseño de páginas web , mientras que otros lenguajes de programación permiten crear cálculos, automatizar, procesar y ser básicamente el cerebro de una aplicación .

**3. Explica con tus palabras qué es una etiqueta y por qué es importante en HTML.**

-La etiqueta es un lenguaje que nos permite estructurar una pagina web, cambiar el diseño de tal modo que la computadora pueda entenderlo y también el ser humano además de permitir manipular el texto usando etiquetas que permiten al navegador identificar como mostrar el contenido.

-Es importante para diseñar y estructurar las paginas web de la forma correcta

**4. Investiga sobre la historia de HTML y cómo ha evolucionado hasta la versión**

**actual.**

**-1991:** Tim Berners-Lee, creador de la Web, publica la primera especificación llamada HTML Tags, con solo 18 etiquetas (como <p>, <a>, <img>).

**-1993:** Se lanza HTML 1.0, primera versión oficial, usada para estructurar documentos simples.

**-1995:** Aparece HTML 2.0, estandarizada por la IETF, ya con formularios y tablas.

**-1997:** Se publica HTML 3.2, incluyendo nuevas etiquetas (ej. <font>, <script>) y soporte para JavaScript.

**-1999:** Llega HTML 4.01, más estructurado y orientado a separar contenido y diseño (fomentando el uso de CSS en vez de etiquetas de estilo).

**-2000:** El W3C promueve XHTML 1.0, una versión de HTML con sintaxis más estricta basada en XML.

Aunque fue un intento de modernizar la web, resultó muy rígido y muchos desarrolladores no lo adoptaron completamente.

**-2008:** El WHATWG (grupo alternativo al W3C) comienza a desarrollar HTML5, buscando simplicidad y compatibilidad.

**-2014:** HTML5 se convierte en el estándar oficial.

* Introduce etiquetas semánticas: <header>, <footer>, <article>, <section>.
* Soporte nativo para audio, video y canvas (gráficos 2D/3D).
* API’s nuevas: geolocalización, almacenamiento local, drag & drop, etc.
* Mejora la integración con JavaScript y CSS.

-Hoy HTML es un estándar vivo: no tiene versiones cerradas como antes, sino que evoluciona continuamente gracias a la colaboración de W3C y WHATWG.

Se sigue mejorando para:

* Optimizar el rendimiento web.
* Dar soporte a aplicaciones móviles y web progresivas (PWA).
* Mejor integración con accesibilidad (WAI-ARIA).
* Soporte para realidad aumentada, WebAssembly y APIs modernas

**5. Explica la importancia de la semántica en HTML.**

-La semántica en HTML consiste en usar etiquetas que aportan significado al contenido. Esto es importante porque mejora la accesibilidad, facilita que buscadores entiendan mejor la estructura de la página, hace el código más claro y mantenible, y garantiza que el contenido sea comprensible tanto para usuarios como para navegadores y tecnologías de asistencia.

**6. Describe la diferencia entre etiquetas de bloque y etiquetas en línea.**

-Las etiquetas en bloque ocupan todo el ancho disponible creando un bloque , que comienza en una nueva línea y que no permite que otros elementos se coloque a su lado,

A diferencia de las etiquetas en línea que ocupan solo el espacio disponible para mostrar el contenido y se alinea con otros elementos en la misma línea sin forzar saltos de línea.

**7. Investiga sobre el W3C y su rol en el desarrollo de HTML.**

**-**El W3C (World Wide Web Consortium) es la organización internacional fundada en 1994 por Tim Berners-Lee, creador de la Web, con el objetivo de establecer estándares abiertos que garanticen el crecimiento y la interoperabilidad de Internet.

En el caso de HTML, el W3C ha tenido un rol clave porque se encarga de definir, regular y actualizar sus especificaciones, promoviendo buenas prácticas en accesibilidad, semántica y compatibilidad entre navegadores.

**8. Escribe un resumen sobre la relación entre HTML y CSS.**

-HTML y CSS trabajan en conjunto para construir páginas web: HTML se encarga de la estructura y el contenido (títulos, párrafos, imágenes, enlaces, formularios), mientras que CSS define la presentación y el estilo (colores, fuentes, tamaños, posiciones, diseño responsivo). Sin HTML, el sitio solo mostraría texto plano y elementos sin orden visual; sin CSS, el contenido no tendría atractivo ni usabilidad. En resumen, HTML es el esqueleto de la web y CSS es su apariencia, y juntos permiten crear páginas claras, funcionales y visualmente agradables.

**9. Investiga la accesibilidad web y cómo HTML contribuye a ella.**

-La accesibilidad web es el conjunto de prácticas y tecnologías que buscan que cualquier persona, incluidas aquellas con discapacidades visuales, auditivas, motoras o cognitivas, pueda acceder, entender y usar los contenidos y servicios de una página web sin barreras.

En este aspecto, HTML contribuye directamente a la accesibilidad gracias a:

* Etiquetas semánticas (<header>, <nav>, <main>, <footer>, <article>) que ayudan a los lectores de pantalla a interpretar mejor la estructura.
* Atributos especiales, como alt en imágenes (para describirlas a personas ciegas), label en formularios o title en enlaces.
* Buenas prácticas en formularios, usando campos correctamente asociados con <label> y roles de accesibilidad.

**10.Reflexiona sobre el impacto del uso incorrecto de etiquetas semánticas en una página web.**

El uso incorrecto de etiquetas semánticas en una página web puede tener varios impactos negativos: dificulta la accesibilidad, ya que los lectores de pantalla no interpretan bien la estructura y las personas con discapacidad visual pueden perderse en la navegación; afecta el SEO, porque los buscadores no comprenden con precisión la jerarquía y el significado del contenido, lo que reduce la visibilidad en los resultados; complica el mantenimiento del código, ya que otros desarrolladores tendrán problemas para identificar la función de cada parte de la página; y puede generar una mala experiencia de usuario, al presentar información desordenada o confusa. En resumen, usar mal las etiquetas semánticas debilita la claridad, accesibilidad y efectividad de una página web.

**3.2 Actividades de contextualización e identificación de conocimientos necesarios**

**para el aprendizaje:**

HTML (HyperText Markup Language) que se utiliza para marcar y estructurar el contenido de una página web, indicando al navegador cómo debe interpretarlo (por ejemplo, si es un título, un párrafo, una imagen, un enlace, etc.)

Sintaxis básica de una etiqueta HTML

La mayoría de las etiquetas tienen una estructura de apertura y cierre:

<nombre\_etiqueta>Contenido</nombre\_etiqueta>

Ejemplo:

<p>Este es un párrafo.</p>

<p> → etiqueta de apertura.

</p> → etiqueta de cierre (con /).

"Este es un párrafo." → contenido.

Tipos de etiquetas

1. Etiquetas con apertura y cierre

Ejemplo:

<h1>Título principal</h1>

<p>Párrafo de texto</p>

2. Etiquetas auto-contenidas (vacías)

No tienen contenido, solo atributos.

Ejemplo:

<img src="imagen.jpg" alt="Descripción de la imagen">

<br>

**2. Explica la diferencia entre etiquetas anidadas y etiquetas de cierre automático.**

Las etiquetas anidadas en HTML son aquellas que tienen apertura y cierre y pueden contener otras etiquetas dentro, siempre respetando el orden de apertura y cierre En cambio, las etiquetas de cierre automático o vacías no requieren etiqueta de cierre porque no contienen contenido dentro, como <img>, <br> o <hr>. En resumen, las anidadas estructuran y organizan el contenido, mientras que las de cierre automático insertan elementos puntuales sin texto interno.

**3. Identifica y explica los atributos más utilizados en HTML.**

1. id

Identifica de forma única a un elemento.

Útil para CSS, JavaScript o enlaces internos.

<p id="intro">Este es un párrafo con id único.</p>

2. class

Permite agrupar varios elementos bajo una misma clase para aplicar estilos o scripts.

<p class="texto-rojo">Párrafo con clase.</p>

3. src (source)

Define la ruta de un recurso, como imágenes, audio o video.

<img src="foto.jpg" alt="Descripción de la foto">

4. alt (alternative text)

Texto alternativo para imágenes, importante para accesibilidad y SEO.

<img src="logo.png" alt="Logo de la empresa">

5. href (hyperlink reference)

Especifica la dirección de destino en enlaces.

<a href="https://www.yotube.com ">Ir a youtube</a>

6. style

Permite aplicar estilos CSS en línea.

<p style="color:blue;">Texto azul</p>

7. title

Muestra un texto emergente al pasar el cursor sobre el elemento.

<abbr title="HyperText Markup Language">HTML</abbr>

8. lang

Define el idioma del contenido, útil para accesibilidad y buscadores.

<html lang="es">

**4. Analiza el código de una página web simple y describe sus etiquetas.**

Código de ejemplo

<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title>Mi primera página</title>

</head>

<body>

<header>

<h1>Bienvenido a mi página</h1>

</header>

<nav>

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

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

</nav>

<main>

<section id="inicio">

<h2>Sobre mí</h2>

<p>Hola, soy estudiante de ADSO y esta es mi primera página.</p>

<img src="foto.jpg" alt="Foto personal">

</section>

<section id="contacto">

<h2>Contacto</h2>

<p>Puedes escribirme a <a href="mailto:correo@ejemplo.com">correo@ejemplo.com</a></p>

</section>

</main>

<footer>

<p>&copy; 2025 Mi Página Web</p>

</footer>

</body>

</html>

**Descripción de las etiquetas usadas**

<!DOCTYPE html> : Indica que el documento está escrito en HTML5.

<html lang="es"> : Raíz del documento, con el atributo lang para definir el idioma (español).

<head> : Contiene metadatos de la página (información que no se muestra directamente).

<meta charset="UTF-8"> : Define la codificación de caracteres para soportar acentos y ñ.

<title> : Título que aparece en la pestaña del navegador.

<body> : Contiene todo el contenido visible de la página.

<h1> : Título principal.

<section> : Agrupa bloques de contenido relacionados (ej. "Sobre mí" o "Contacto").

<h2> : Subtítulos dentro de cada sección.

<p> : Párrafos de texto.

<img> :Inserta una imagen (src define la ruta, alt el texto alternativo).

<a> : Enlace (puede ser a otra página, una sección interna o un correo con mailto:).

**5. Identifica los elementos estructurales de una página web (header, nav, main,aside, footer).**

<header> : Encabezado principal de la página (puede incluir títulos, logotipos, menús).

<nav> : Menú de navegación con enlaces (<a>).

<main> :Contenido principal de la página.

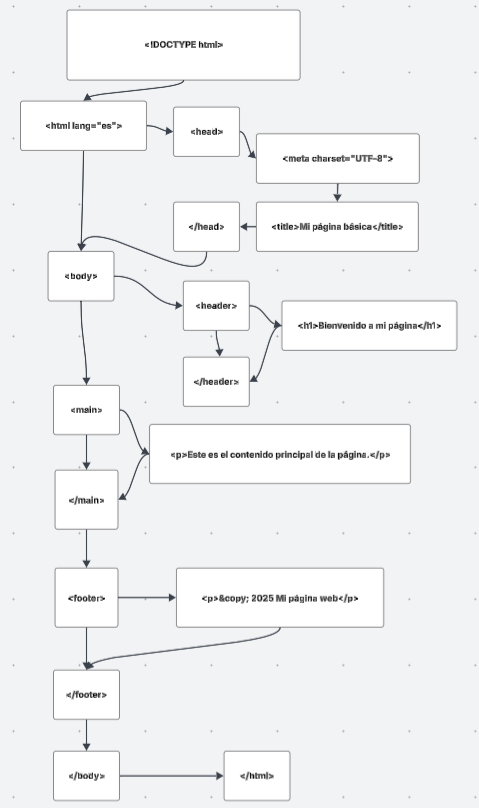
<aside>:contenido secundario lateral de la pagina.

<footer> : Pie de página (derechos de autor, información adicional).

**6. Explica la función del atributo "alt" en las imágenes.**

-sirve para proporcionar una descripción textual a una imagen que no pude ser cargada o interpretada correctamente .

**7. Crea un diagrama con la estructura básica de un documento HTML.**

****

**8. Describe las diferencias entre listas ordenadas y no ordenadas.**

-Las listas ordenadas, definidas con la etiqueta <ol>, muestran sus elementos numerados secuencialmente, ya sea con números, letras o números romanos, lo que indica un orden específico entre los elementos. Por otro lado, las listas no ordenadas, creadas con la etiqueta <ul>, presentan sus elementos con viñetas o símbolos gráficos, sin implicar ningún orden jerárquico o secuencial entre ellos.

**9. Explica la importancia de los formularios en HTML.**

-Los formularios HTML son elementos de las páginas web que nos sirven para recuperar información del usuario y enviarla al servidor para que pueda ser procesada.

**10.Investiga sobre la validación de formularios con atributos de HTML.**

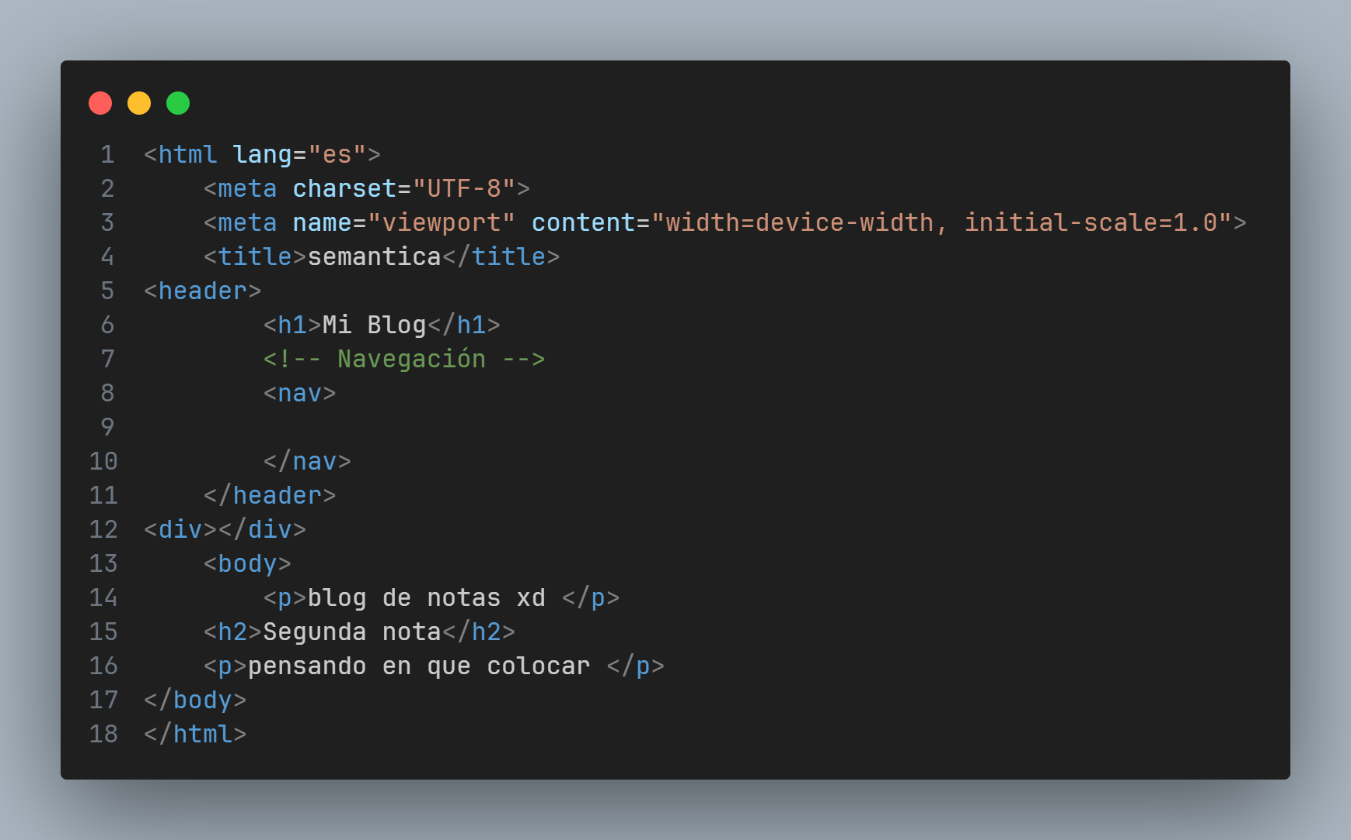
-Los atributos más utilizados incluyen required, que indica que un campo es obligatorio y no se enviará el formulario si está vacío, haciendo que el campo se considere inválido si no se rellena. Para campos de texto, los atributos minlength y maxlength permiten establecer la longitud mínima y máxima de caracteres permitidos, restringiendo la entrada del usuario

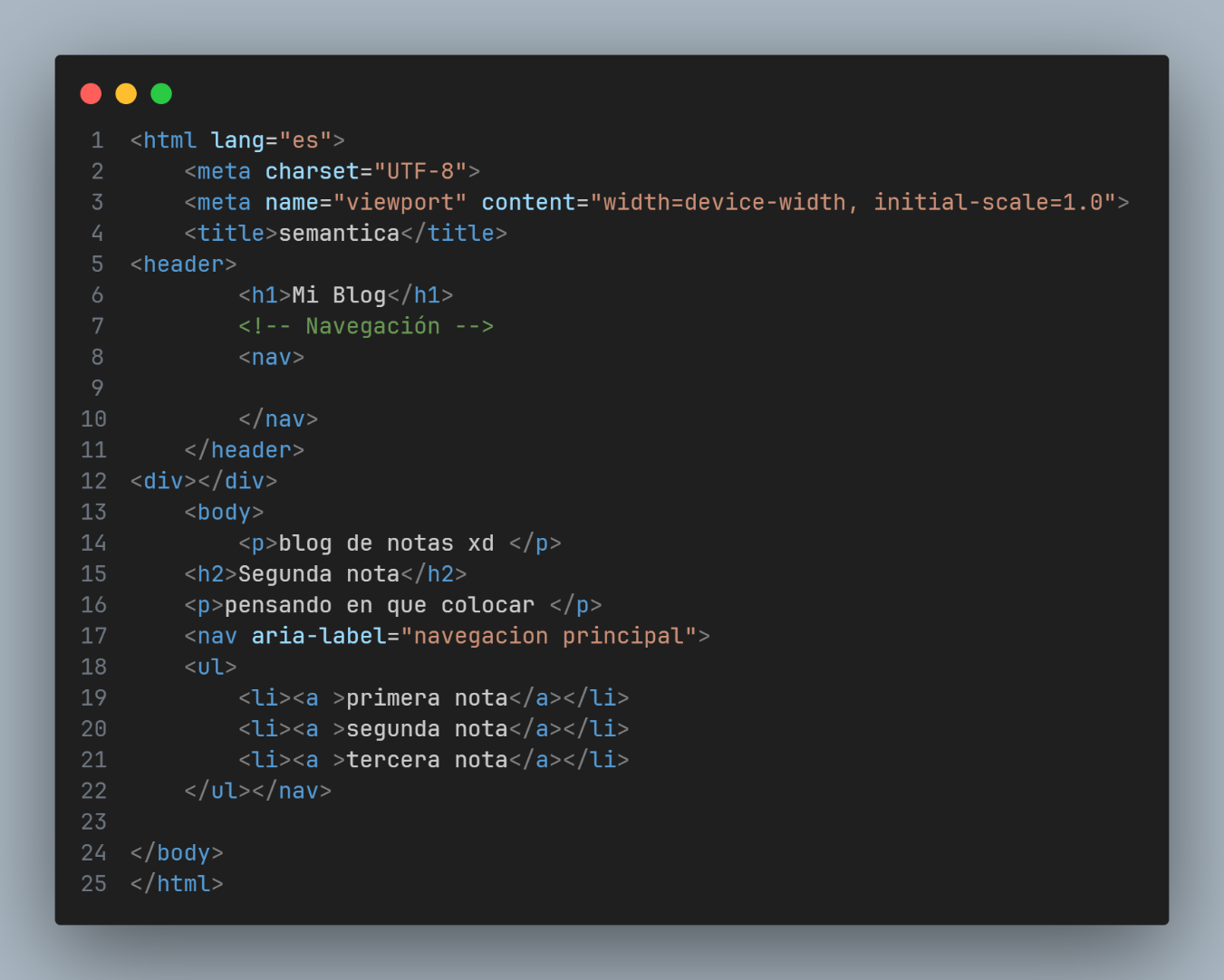
**3.3 Actividades de apropiación:**

**1. Crea una página HTML con una estructura semántica correcta.**



**2. Usa las etiquetas semánticas adecuadas para estructurar el blog de la página.**

**3. Implementa una barra de navegación secundaria utilizando "nav" y listas.**

****

**4. Diseña una sección de contacto usando y valida el formulario para que nunca sea**

**enviado con datos vacíos, si requerimos número se envie solo números, campos**

**máximos y mínimos etc.**

****

**5. Agrega imágenes a una página y usa "alt" para descripciones accesibles.**

****

**6. Crea una tabla con "table", "thead", "tbody" y "tfoot" para simular que estamos**

**listando todos los datos del formulario de contacto.**

****