



LENGUAJES DE MARCAS Y SISTEMAS DE GESTIÓN DE LA INFORMACIÓN

DAW1

UD1	INTRODUCCIÓN A LOS LENGUAJES DE MARCAS Y HTML5
------------	---

Duración: 6 semanas

Objetivos de la unidad:

Al finalizar esta unidad, los estudiantes serán capaces de:

1. Comprender qué es un Lenguaje de Marcas y conocer su evolución histórica.
2. Comprender los conceptos básicos de HTML5 y su importancia en el desarrollo de aplicaciones web.
3. Crear documentos HTML5 válidos y estructurados.
4. Utilizar las etiquetas y atributos más comunes de HTML5 para la creación de contenido web.
5. Aplicar estilos y formatos utilizando las nuevas características de HTML5.
6. Incorporar elementos multimedia y gráficos en una página web utilizando HTML5.
7. Comprender los principios de accesibilidad y semántica en HTML5.
8. Utilizar herramientas de depuración y validación para garantizar la calidad del código HTML5.

Contenidos:

1. Introducción a los Lenguajes de Marcas
2. Introducción a HTML5
3. Estructura básica de un documento HTML5
4. Etiquetas y atributos fundamentales de HTML5
5. Formularios HTML5
6. Estilos y formatos HTML5



7. Multimedia HTML5
8. Accesibilidad y semántica en HTML5
9. Optimización y rendimiento en HTML5
10. Depuración y validación en HTML5

1.2. Introducción a HTML5

1. Historia y evolución de HTML hacia HTML5.
2. Características principales de HTML5 y diferencias entre HTML5 y versiones anteriores.
3. Herramientas y entornos de desarrollo para HTML5.

1.2.1. Historia y evolución de HTML hacia HTML5:

HTML (HyperText Markup Language) es el lenguaje de marcado utilizado para crear la estructura y el contenido de las páginas web. Ha experimentado varias versiones a lo largo del tiempo, y HTML5 es la última y más avanzada hasta la fecha.

La evolución de HTML hacia HTML5 se ha centrado en mejorar las capacidades y funcionalidades del lenguaje para satisfacer las demandas cambiantes de la web. HTML5 se ha desarrollado en colaboración con la World Wide Web Consortium (W3C) y varias organizaciones tecnológicas para estandarizar el lenguaje y promover su adopción en todos los navegadores.

1.2.2. Características principales de HTML5 y diferencias con versiones anteriores:

HTML5 introdujo una serie de características nuevas y mejoradas en comparación con las versiones anteriores. Algunas de las características más importantes de HTML5 incluyen:

- Etiquetas semánticas: HTML5 ofrece una amplia variedad de etiquetas semánticas como `<header>`, `<nav>`, `<section>`, `<article>`, `<footer>`, etc., que permiten una estructura más clara y significativa para el contenido de una página web. Las versiones anteriores de HTML no tenían estas etiquetas específicas.
- Reproducción de multimedia: HTML5 incluye etiquetas como `<audio>` y `<video>` que permiten reproducir contenido multimedia



sin necesidad de plugins externos como Flash. Las versiones anteriores solían requerir plugins externos como Flash

- Gráficos y animaciones: Se introdujo el elemento <canvas> que proporciona una API para renderizar gráficos dinámicos y animaciones en 2D y 3D de manera nativa en el navegador. Las versiones anteriores de HTML no tenían esta funcionalidad nativa.
- Formularios mejorados: HTML5 ofrece nuevos tipos de entrada para formularios, como fechas, email, número, rango, búsqueda, entre otros. También se incluyen nuevas capacidades de validación y atributos mejorados.
- Almacenamiento local: HTML5 proporciona la API de almacenamiento local, que permite a las aplicaciones web almacenar datos en el lado del cliente, lo que brinda capacidades de almacenamiento persistente sin necesidad de cookies. Versiones anteriores de HTML no tenían esta capacidad.

1.2.3. Herramientas y entornos de desarrollo para HTML5:

Para desarrollar aplicaciones web con HTML5, hay una variedad de herramientas y entornos disponibles. Algunas de las herramientas populares son:

- Editores de texto: Puedes utilizar editores de texto simples como Sublime Text, Visual Studio Code, Atom o Notepad++ para escribir el código HTML5.
- Frameworks y librerías: Existen frameworks y librerías como Bootstrap, Foundation y Materialize CSS que facilitan el desarrollo de aplicaciones web con estilos y componentes predefinidos.
- Entornos de desarrollo integrados (IDE): Los IDEs como WebStorm, Visual Studio y Brackets ofrecen características avanzadas para el desarrollo web, como resaltado de sintaxis, depuración y administración de proyectos.
- Herramientas de depuración: Los navegadores modernos, como Google Chrome y Mozilla Firefox, vienen con herramientas de desarrollo integradas que permiten inspeccionar y depurar aplicaciones web. Ejemplos de estas herramientas son las consolas, paneles de red y herramientas de inspección de elementos.



- Frameworks JavaScript: Para complementar HTML5, puedes utilizar frameworks JavaScript como React, Angular y Vue.js para construir aplicaciones web interactivas y dinámicas.

Estas son solo algunas de las herramientas y entornos de desarrollo disponibles para trabajar con HTML5. La elección de las herramientas dependerá de tus preferencias y necesidades específicas de desarrollo.

1.3. Estructura básica de un documento HTML5

1. Sintaxis y estructura básica de un documento HTML5.
2. Etiquetas de apertura y cierre.
3. Uso del doctype y codificación de caracteres.
4. Uso de elementos contenedores en HTML5. Etiquetas semánticas en HTML5: header, nav, section, article, aside, footer.

1.3.1. Sintaxis y estructura básica de un documento HTML5:

Un documento HTML5 sigue una estructura básica que consta de las siguientes partes:

```
<!DOCTYPE html>

<html>

<head>

  <title>Título de la página</title>

</head>

<body>

  Contenido de la página

</body>
```



```
</html>
```

- **<!DOCTYPE html>**: Es el doctype y define la versión de HTML que se está utilizando, en este caso, HTML5.
- **<html>**: Es el elemento raíz y encierra todo el contenido HTML.
- **<head>**: Es la sección que contiene información sobre el documento, como el título de la página, enlaces a archivos CSS, scripts JavaScript, metadatos, etc.
- **<title>**: Define el título de la página que se mostrará en la pestaña del navegador.
- **<body>**: Contiene el contenido visible de la página, como texto, imágenes, enlaces, formularios, etc.

1.3.2. Etiquetas de apertura y cierre.

En HTML5, las etiquetas se utilizan para marcar y estructurar el contenido de la página. Cada etiqueta tiene una etiqueta de apertura `<etiqueta>` y una etiqueta de cierre `</etiqueta>`. Algunas etiquetas no necesitan una etiqueta de cierre y se llaman "etiquetas vacías" o "etiquetas auto-cerrantes".

Por ejemplo:

```
<p>Este es un párrafo.</p>
```

```

```

```
<br>
```

1.3.3. Uso del doctype y codificación de caracteres:

El doctype (`<!DOCTYPE html>`) se coloca al comienzo de un documento HTML5 y especifica la versión de HTML que se está utilizando. En el caso de HTML5, `<!DOCTYPE html>` se utiliza para indicar que el documento es un documento HTML5.



La codificación de caracteres se define en la sección `<head>` del documento utilizando la metaetiqueta `<meta charset="utf-8">`. La codificación de caracteres UTF-8 es ampliamente utilizada y permite representar una amplia gama de caracteres en diferentes idiomas.

1.3.4. Elementos contenedores en HTML5:

Los elementos contenedores se utilizan para agrupar y estructurar contenido relacionado. Algunos ejemplos de elementos contenedores comunes son:

- **`<div>`**: Se utiliza como un contenedor genérico para agrupar elementos relacionados.
- **`<header>`**: Representa el encabezado de una sección o del documento completo.
- **`<nav>`**: Representa una sección de navegación.
- **`<section>`**: Define una sección temática dentro del documento.
- **`<article>`**: Representa un contenido independiente que se puede distribuir y reutilizar de manera independiente.
- **`<aside>`**: Representa contenido relacionado pero no directamente relevante para el flujo principal del documento.
- **`<footer>`**: Representa el pie de página del documento o de una sección.

1.4. Etiquetas y atributos fundamentales de HTML5

1. Etiquetas para el texto: h1-h6, p, span, em, strong, etc.
2. Etiquetas para listas: ul, ol, li.
3. Etiquetas para enlaces: a y sus atributos.
4. Etiquetas para imágenes: img y sus atributos.
5. Atributos globales en HTML5: id, class, style, etc.



1.4.1. Etiquetas para el texto:

- **<h1> a <h6>**: Se utilizan para los encabezados de diferentes niveles, siendo **<h1>** el más importante y **<h6>** el menos importante.

```
<h1>Encabezado de nivel 1</h1>
```

```
<h2>Encabezado de nivel 2</h2>
```

- **<p>**: Define un párrafo de texto.

```
<p>Esto es un párrafo.</p>
```

- ****: Se utiliza para aplicar estilos o manipular partes específicas de un texto.

```
<p>Este es un texto de ejemplo <span style="color: red;">resaltado</span>.</p>
```

- ** y **: Se utilizan para enfatizar o resaltar el texto. **** indica énfasis y **** indica una mayor importancia o relevancia.

```
<p>Este es un texto <em>enfático</em> y  
<strong>importante</strong>.</p>
```

1.4.2. Etiquetas para listas:

- ****: Define una lista no ordenada.

```
<ul>  
  <li>Elemento 1</li>  
  <li>Elemento 2</li>  
</ul>
```

- ****: Define una lista ordenada.

```
<ol>  
  <li>Elemento 1</li>
```



```
<li>Elemento 2</li>
```

```
</ol>
```

- ****: Define un elemento de lista dentro de **** o ****.

1.4.3. Etiquetas para enlaces:

- **<a>**: Define un enlace y se utiliza junto con el atributo "href" para especificar la URL de destino.

```
<a href="https://www.ejemplo.com">Enlace</a>
```

- Atributos para **<a>**:
 - "href": Especifica la URL a la que se redirige el enlace.
 - "target": Define cómo se abrirá la URL. Por ejemplo, "_blank" para abrir en una nueva ventana o pestaña.
 - "title": Proporciona información adicional sobre el enlace cuando el usuario pasa el cursor sobre él.

1.4.4. Etiquetas para imágenes:

- ****: Se utiliza para insertar imágenes en una página web.

```

```

- Atributos para ****:
 - "src": Especifica la ruta o URL de la imagen.
 - "alt": Proporciona un texto alternativo que describe la imagen en caso de que no se pueda mostrar.

1.4.5. Atributos globales en HTML5:

- **"id"**: Se utiliza para asignar un identificador único a un elemento HTML.

```
<div id="mi-elemento"></div>
```




- **"class"**: Se utiliza para asignar una o más clases a un elemento HTML, que se pueden utilizar para aplicar estilos o seleccionar elementos con CSS o JavaScript.

```
<div class="mi-clase"></div>
```

- **"style"**: Se utiliza para aplicar estilos CSS directamente a un elemento HTML.

```
<div style="color: red; font-size: 16px;"></div>
```

1.5. Formularios en HTML5

1. Creación de formularios HTML5.
2. Elementos de formulario: input (text, checkbox, radio, etc.), textarea, select, button.
3. Atributos de validación: required, pattern, min, max, etc.
4. Uso de etiquetas fieldset y legend.

1.5.1. Creación de formularios HTML5:

Los formularios son elementos fundamentales en el desarrollo de aplicaciones web, ya que permiten la interacción y recopilación de datos por parte de los usuarios. En HTML5, se utilizan diversas etiquetas y atributos para crear formularios efectivos y funcionales.

1.5.2. Elementos de formulario:

- **<input>**: Se utiliza para crear diferentes tipos de campos de entrada, como campos de texto, casillas de verificación, botones de radio, etc.

```
<input type="text" name="nombre" placeholder="Nombre completo">
```

```
<input type="checkbox" name="opciones" value="opcion1">  
Opción 1
```

```
<input type="radio" name="opcion" value="opcion1"> Opción 1
```



- **<textarea>**: Se utiliza para crear un área de texto multi línea.

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

- **<select> y <option>**: Se utilizan para crear menús desplegables (select) con opciones (option).

```
<select name="pais">  
  <option value="1">España</option>  
  <option value="2">México</option>  
</select>
```

- **<button>**: Se utiliza para crear botones dentro de un formulario.

```
<button type="submit">Enviar</button>  
<button type="reset">Restablecer</button>
```

1.5.3. Atributos de validación:

- **"required"**: Especifica que un campo de entrada debe ser completado antes de enviar el formulario.

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

- **"pattern"**: Permite especificar una expresión regular que debe cumplir el valor de un campo de entrada.

```
<input type="text" name="codigo" pattern="[0-9]{4}" title="Ingresa  
un código de 4 dígitos">
```

- **"min" y "max"**: Establecen un rango mínimo y máximo para el valor de un campo numérico.

```
<input type="number" name="edad" min="18" max="99">
```



1.5.4. Uso de etiquetas `<fieldset>` y `<legend>`:

- **`<fieldset>`**: Se utiliza para agrupar elementos de formulario relacionados dentro de un formulario.

```
<fieldset>

<legend>Datos personales</legend>

<input type="text" name="nombre" placeholder="Nombre">

<input type="email" name="email" placeholder="Correo electrónico">

</fieldset>
```

- **`<legend>`**: Define una leyenda o título para el `<fieldset>`.

```
<fieldset>

<legend>Preferencias</legend>

<input type="checkbox" name="preferencia" value="opcion1">
Opción 1

<input type="checkbox" name="preferencia" value="opcion2">
Opción 2

</fieldset>
```

1.6. Estilos y formatos en HTML5

1. Uso de CSS para estilizar elementos HTML5.
2. Estilos internos y externos.

1.6.1. Uso de CSS para estilizar elementos HTML5:

CSS (Cascading Style Sheets) se utiliza para aplicar estilos y dar formato a los elementos HTML. Permite controlar la apariencia de los elementos, como colores, fuentes, tamaños, márgenes, etc.



1.6.2. Estilos internos y externos:

- **Estilos internos:** Se definen dentro de la etiqueta `<style>` en el encabezado del documento HTML o directamente en el atributo "style" de un elemento HTML.

```
<style>

h1 { color: blue;

}

</style>
```

- **Estilos externos (UD3):** Se definen en un archivo CSS separado y se vinculan al documento HTML utilizando la etiqueta `<link>`.

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

1.7. Multimedia en HTML5

1. Incorporación de elementos multimedia: audio y video.
2. Formatos compatibles y atributos asociados.
3. Controles de reproducción y personalización.
4. Uso de la etiqueta canvas para gráficos y animaciones.

1.7.1. Incorporación de elementos multimedia: audio y video:

HTML5 proporciona etiquetas específicas para la incorporación de elementos multimedia en una página web, como audio y video.

- **<audio>:** Se utiliza para agregar archivos de audio en diferentes formatos compatibles.

```
<audio src="mi-audio.mp3" controls></audio>
```

- **<video>:** Se utiliza para agregar archivos de video en diferentes formatos compatibles.



```
<video src="mi-video.mp4" controls></video>
```

1.7.2. Formatos compatibles y atributos asociados:

Los formatos de audio y video compatibles pueden variar según el navegador y la plataforma utilizados. Sin embargo, existen formatos comunes ampliamente compatibles, como MP3 y MP4 para audio, y MP4 y WebM para video.

Además del atributo "src" utilizado para especificar la ruta del archivo multimedia, se pueden utilizar otros atributos asociados, como:

- **"controls"**: Muestra los controles de reproducción (reproducir, pausar, detener, barra de progreso, etc.) para el elemento multimedia.
- **"autoplay"**: Inicia automáticamente la reproducción del elemento multimedia cuando se carga la página.
- **"loop"**: Reproduce el elemento multimedia en bucle.
- **"preload"**: Especifica si el navegador debe cargar el archivo multimedia cuando se carga la página ("auto", "metadata" o "none").

1.7.3. Controles de reproducción y personalización:

El atributo "controls" permite mostrar los controles de reproducción estándar para los elementos multimedia, pero también es posible personalizar y diseñar los controles según tus necesidades. Para ello, puedes utilizar JavaScript y CSS para crear tus propios controles personalizados y controlar la reproducción del elemento multimedia.

1.7.4. Uso de la etiqueta **<canvas>** para gráficos y animaciones:

La etiqueta **<canvas>** en HTML5 permite renderizar gráficos, dibujos y animaciones utilizando JavaScript. Proporciona un lienzo en el cual se pueden dibujar elementos gráficos dinámicamente.

```
<canvas id="mi-canvas"></canvas>
```

Luego, puedes utilizar JavaScript para manipular el lienzo y dibujar elementos gráficos en él.



```
const canvas = document.getElementById('mi-canvas');  
  
const ctx = canvas.getContext('2d');  
  
// Ejemplo de dibujo de un rectángulo  
  
ctx.fillStyle = 'red';  
  
ctx.fillRect(10, 10, 100, 50);
```

1.8. Accesibilidad y semántica en HTML5

1. Principios de accesibilidad web.
2. Uso de etiquetas semánticas para mejorar la estructura y significado del contenido.
3. Uso adecuado de los atributos alt y title.
4. Herramientas de evaluación y verificación de accesibilidad.

1.8.1. Principios de accesibilidad web:

La accesibilidad web se refiere a la práctica de diseñar y desarrollar sitios web de manera que sean accesibles y utilizables por todas las personas, incluyendo aquellas con discapacidades o dificultades para acceder a la información en línea. Algunos principios clave de accesibilidad web son:

- Percepción: Proporcionar contenido perceptible de diferentes maneras para que pueda ser interpretado por diferentes sentidos, como texto alternativo para imágenes, subtítulos para videos y contraste de color adecuado para facilitar la lectura.
- Operabilidad: Asegurar que los usuarios puedan operar y navegar por el sitio web utilizando diferentes dispositivos de entrada, como teclado, mouse y tecnologías de asistencia.
- Comprensión: Facilitar la comprensión del contenido mediante una estructura clara y consistente, instrucciones claras y legibilidad del texto.



- Robustez: Desarrollar sitios web que sean compatibles con diferentes tecnologías y dispositivos, para que puedan funcionar correctamente en una amplia gama de entornos y navegadores.

1.8.2. Uso de etiquetas semánticas para mejorar la estructura y significado del contenido.

Algunos elementos contenedores proporcionan un significado semántico al contenido y ayudan a los motores de búsqueda y otros agentes a comprender mejor la estructura de la página. HTML5 introdujo una serie de elementos semánticos que brindan un significado estructural más claro al contenido de una página. Estos elementos semánticos proporcionan una estructura más clara y significativa al contenido HTML5, lo que facilita la comprensión tanto para los desarrolladores como para los motores de búsqueda.

<header>: Define el encabezado de una sección o del documento completo. Puede contener logotipos, títulos, menús de navegación, etc.

```
<header>

  <h1>Título del encabezado</h1>

  <nav>

    <ul>

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

      <li><a href="#">Acerca de</a></li>

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

    </ul>

  </nav>

</header>
```

<nav>: Representa una sección de navegación y generalmente contiene enlaces a otras páginas o secciones dentro del sitio web.



```
<nav>
```

```
<ul>
```

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

```
<li><a href="#">Acerca de</a></li>
```

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

```
</ul>
```

```
</nav>
```

<section>: Define una sección temática dentro del documento y se utiliza para agrupar contenido relacionado.

```
<section>
```

```
<h2>Título de la sección</h2>
```

```
<p>Contenido de la sección</p>
```

```
</section>
```

<article>: Representa un contenido independiente y autónomo que se puede distribuir y reutilizar de manera independiente en un documento.

```
<article>
```

```
<h2>Título del artículo</h2>
```

```
<p>Contenido del artículo</p>
```

```
</article>
```

<aside>: Representa contenido relacionado pero no directamente relevante para el flujo principal del documento, como barras laterales, anuncios, enlaces relacionados, etc.

```
<aside>
```




```
<h3>Barra lateral</h3>
```

```
<ul>
```

```
<li><a href="#">Enlace relacionado 1</a></li>
```

```
<li><a href="#">Enlace relacionado 2</a></li>
```

```
<li><a href="#">Enlace relacionado 3</a></li>
```

```
</ul>
```

```
</aside>
```

<footer>: Representa el pie de página del documento o de una sección específica y generalmente contiene información de derechos de autor, enlaces de contacto, etc.

```
<footer>
```

```
<p>Derechos de autor &copy; 2023</p>
```

```
<p>Contacto: info@example.com</p>
```

```
</footer>
```

1.8.3. Uso adecuado de los atributos "alt" y "title":

Los atributos "alt" y "title" son importantes para mejorar la accesibilidad y la experiencia de usuario en un sitio web. Aquí tienes una explicación sobre su uso adecuado:

- **Atributo "alt":** El atributo "alt" se utiliza en las etiquetas `` para proporcionar un texto alternativo que describe el contenido de una imagen. Es especialmente útil para las personas con discapacidad visual que utilizan lectores de pantalla. El texto alternativo debe ser descriptivo y conciso, transmitiendo el significado o la información relevante de la imagen.

```

```



- Atributo "title": El atributo "title" se utiliza en diversas etiquetas (como `<a>`, `<input>`, etc.) para proporcionar información adicional sobre el elemento cuando el usuario pasa el cursor sobre él. Este texto se muestra en una pequeña ventana emergente o como una descripción de herramienta (tooltip). Se recomienda utilizar el atributo "title" para aclarar la funcionalidad o proporcionar información adicional cuando sea necesario.

```
<a href="#" title="Enlace a la página de inicio">Inicio</a>
```

Es importante utilizar estos atributos correctamente para garantizar una experiencia accesible y comprensible para todos los usuarios, incluidos aquellos que utilizan tecnologías de asistencia.

1.8.4. Herramientas de evaluación y verificación de accesibilidad:

Existen diversas herramientas que puedes utilizar para evaluar y verificar la accesibilidad de tu sitio web. Algunas de las herramientas populares son:

- Lighthouse: Es una herramienta de auditoría de código de Google Chrome que incluye una opción para evaluar la accesibilidad de un sitio web. Proporciona informes detallados sobre los problemas de accesibilidad y sugerencias de mejora.
- Wave: Es una extensión de navegador y una herramienta en línea que permite evaluar la accesibilidad de una página web. Identifica errores, advertencias y mejoras recomendadas, y proporciona una descripción detallada de cada problema.
- AChecker: Es una herramienta en línea que verifica la accesibilidad de una página web según las pautas de WCAG (Web Content Accessibility Guidelines). Proporciona un informe detallado y sugiere soluciones para mejorar la accesibilidad.
- Axe: Es una herramienta de prueba de accesibilidad automatizada que se puede integrar en tu flujo de trabajo de desarrollo. Proporciona informes detallados de los problemas de accesibilidad y ofrece orientación sobre cómo solucionarlos.



1.9. Optimización y rendimiento en HTML5

1. Mejores prácticas para optimizar el rendimiento de páginas web.
2. Uso de atributos async y defer para la carga de scripts.
3. Compresión y minificación de archivos HTML, CSS y JavaScript.
4. Caché de recursos estáticos y técnicas de almacenamiento local.

1.10. Depuración y validación en HTML5

1. Herramientas de depuración y consola de desarrollador.
2. Validación de código HTML5 utilizando el W3C Markup Validation Service.
3. Identificación y corrección de errores comunes.
4. Pruebas en diferentes navegadores y dispositivos.