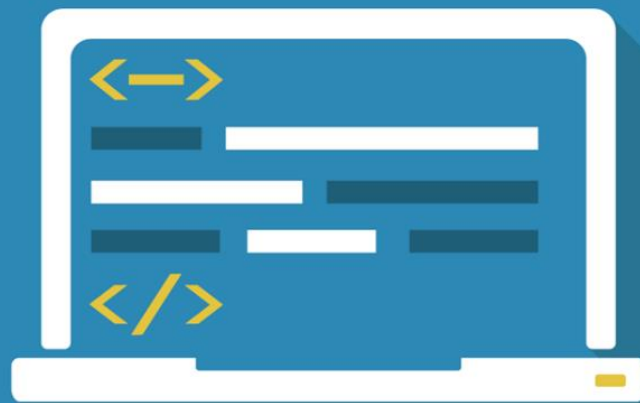


CREACIÓN DE PÁGINAS WEB



Contenido

UNIDAD 1. Lenguajes de marcado	4
LENGUAJES DE MARCAS: CLASIFICACIÓN, VERSIONES, ESTÁNDARES	4
1.1. Clasificación de los lenguajes de marcas	4
1.2. Versiones	5
1.3. Estándares	6
CARACTERÍSTICAS DE LOS LENGUAJES DE MARCADO	8
ESTRUCTURA DE UN DOCUMENTO CON LENGUAJE DE MARCADO	9
IDENTIFICACIÓN DE ETIQUETAS Y ATRIBUTOS	11
4.1. Elementos textuales	12
4.2. ESPACIADO Y SALTO DE LÍNEAS	19
4.3. ENLACES INTERNOS Y EXTERNOS	20
4.4. LISTAS ORDENADAS Y DESORDENADAS	22
4.5. TABLAS	25
ELEMENTOS ESTRUCTURALES	27
ESTRUCTURACIÓN SEMÁNTICA	30
UNIDAD 2. Elementos de multimedia y formularios	35
INSERCIÓN DE ELEMENTOS MULTIMEDIA	35
TRATAMIENTO WEB DE IMÁGENES	36
2.1. MAPA DE IMÁGENES	37
2.2. TRATAMIENTO WEB DE VÍDEOS	38
2.3. FORMATOS DE IMAGEN, AUDIO Y VÍDEO	38
ELEMENTOS INCRUSTADOS	42
HERRAMIENTAS PARA EL TRATAMIENTO DE ELEMENTOS MULTIMEDIA	44
FORMULARIOS: CARACTERÍSTICAS, MÉTODOS DE ENVÍO, PROCESAMIENTO DE LOS DATOS	46
ELEMENTOS Y ATRIBUTOS DE FORMULARIO	48
6.1. CONTROLES DE FORMULARIO	52
6.2. FORMULARIOS Y EVENTOS	56
6.3. EVENTOS COMUNES CON FORMULARIOS Y SUS CONTROLES	59
UNIDAD 3. Técnicas de accesibilidad y usabilidad	62
ACCESIBILIDAD WEB	62
1.1. ESTÁNDARES DE ACCESIBILIDAD	67
NORMATIVA EUROPEA Y ESPAÑOLA	69
HERRAMIENTAS "SOFTWARE" DE VALIDACIÓN DE LA ACCESIBILIDAD	70
USABILIDAD WEB	73
HERRAMIENTAS "SOFTWARE" DE MEDICIÓN DE USABILIDAD	74

UNIDAD 4. Herramientas de edición web y optimización de páginas web	79
FUNCIONES Y CARACTERÍSTICAS DE HERRAMIENTAS DE DESARROLLO WEB	79
1.1. Características de las herramientas de desarrollo web	79
CONFIGURACIÓN DE MÓDULOS Y MENÚS	80
2.1. Configuración de módulos	80
2.2. Configuración de menús	82
2.3. Integración	84
EDITORES DE LENGUAJE DE MARCADO	86
INSTALACIÓN HERRAMIENTAS DE PUBLICACIÓN	89
SISTEMAS GESTORES DE CONTENIDO (CMS)	91
NORMATIVA EUROPEA Y ESPAÑOLA SOBRE PROTECCIÓN DE DATOS Y LSSI-CE	94
6.1. LSSI-CE (LEY DE SERVICIOS DE LA SOCIEDAD DE LA INFORMACIÓN Y COMERCIO ELECTRÓNICO)	95
POSICIONAMIENTO ORGÁNICO	95
TÉCNICAS SEO	97
URL SEMÁNTICAS	98
PRUEBAS DE VALIDACIÓN	100
PRUEBAS DE VERIFICACIÓN	103

UNIDAD 1. Lenguajes de marcado

LENGUAJES DE MARCAS: CLASIFICACIÓN, VERSIONES, ESTÁNDARES

Un lenguaje de marcas se puede describir como un sistema que permite estructurar y codificar un documento mediante el uso de etiquetas o marcas. Estas etiquetas aportan información adicional relacionada con la organización, el formato y la presentación del contenido textual.

La idea de los lenguajes de marcas surgió en la década de los años 60, cuando IBM buscaba una solución para gestionar documentos provenientes de distintas plataformas. De esta necesidad nació el **GML** (*Generalized Markup Language*), un lenguaje diseñado para ajustar automáticamente el formato y la presentación de documentos en diferentes dispositivos, simplemente indicando el perfil deseado.

Con el tiempo, este lenguaje evolucionó y, tras un extenso proceso, fue estandarizado por la **ISO** (Organización Internacional de Normalización) en 1986. En ese momento, pasó a llamarse **SGML** (*Standard Generalized Markup Language*), convirtiéndose en el precursor de todos los lenguajes de marcas modernos.

Aunque **SGML** se caracteriza por su gran versatilidad, también es reconocido por su complejidad. A partir de él han surgido numerosos lenguajes derivados, siendo **HTML** el más conocido y utilizado. Además, existen otros lenguajes importantes como **XML**, **Latex**, **XHTML**, **Tex**, **SVG**, **MathML**, **RTF**, **RSS**, entre otros.

1.1. Clasificación de los lenguajes de marcas

Los lenguajes de marcas se pueden clasificar de diversas maneras, pero una de las categorizaciones más comunes los divide en tres grandes grupos:

- **Lenguajes de presentación:**

Estos lenguajes están diseñados para definir cómo debe formatearse el texto, es decir, especifican su apariencia. Entre los más utilizados se encuentran **RTF**, **HTML** y **TeX**, entre otros.

- **Lenguajes procedimentales:**

Aunque también se enfocan en la presentación del texto, en estos lenguajes el código es visible para quien edita el documento. Sin embargo, el programa encargado de procesar el archivo interpreta este código y muestra el contenido al usuario final con el formato adecuado. Algunos ejemplos destacados de lenguajes procedimentales son **HTML** y **TeX**.

- **Lenguajes descriptivos o semánticos:**

Este tipo de lenguajes prioriza la definición del contenido del texto en lugar de su formato. El enfoque se centra en describir el significado y la estructura del documento, dejando de lado cómo debe presentarse visualmente. Ejemplos representativos de lenguajes descriptivos incluyen **XML**, **SGML**, entre otros.

1.2. Versiones

A lo largo del tiempo, los lenguajes de marcas han evolucionado, dando lugar a diferentes versiones que han mejorado su funcionalidad, flexibilidad y compatibilidad. Algunas de las versiones más relevantes incluyen:

- **SGML (*Standard Generalized Markup Language*):**

Fue la primera versión estandarizada en 1986, marcando el inicio de los lenguajes de marcas modernos. Su complejidad permitió la creación de múltiples lenguajes derivados.

- **HTML (*HyperText Markup Language*):**

Derivado de SGML, **HTML** se convirtió en uno de los lenguajes más populares para la creación de páginas web. A lo largo de los años, se han desarrollado numerosas versiones, desde **HTML 1.0** (1993) hasta **HTML5**, lanzado oficialmente en 2014, que introdujo soporte para contenido multimedia y mayor interactividad.

- **XML (*eXtensible Markup Language*):**

Creado como una versión simplificada de SGML, **XML** se centra en la descripción de datos y se utiliza ampliamente en el intercambio de

información entre sistemas. Su flexibilidad lo ha hecho popular en aplicaciones como bases de datos y servicios web.

- **XHTML (eXtensible HyperText Markup Language):**

Surge como una fusión entre **HTML** y **XML**, combinando las capacidades de ambos para crear un lenguaje más estructurado y compatible con estándares modernos.

- **RTF (Rich Text Format):**

Aunque no es tan avanzado como SGML o HTML, **RTF** se desarrolló como un formato portátil para compartir documentos entre diferentes plataformas, manteniendo tanto el contenido como el formato básico.

- **TeX y LaTeX:**

Lenguajes utilizados principalmente para la creación de documentos técnicos y científicos. **TeX** se centra en el control del formato, mientras que **LaTeX** facilita su uso mediante plantillas y simplificaciones.

- **SVG (Scalable Vector Graphics):**

Diseñado específicamente para gráficos vectoriales, **SVG** es un lenguaje basado en XML que permite crear imágenes escalables para aplicaciones web y otros entornos digitales.

1.3. Estándares

1. Sgml (iso 8879):

El **SGML** (*Standard Generalized Markup Language*) fue estandarizado en 1986 por la **ISO** bajo la norma **ISO 8879**. Este estándar define cómo estructurar documentos de manera flexible y permite la creación de lenguajes específicos, como HTML y XML.

2. Html (*hypertext markup language*):

El lenguaje **HTML** cuenta con varios estándares establecidos por el **W3C** (*World Wide Web Consortium*) y otras organizaciones, como **WHATWG**. Estos estándares definen las versiones del lenguaje, desde HTML 4.01 hasta **HTML5**,

lanzado en 2014, que se convirtió en el estándar actual para la creación de páginas web dinámicas y multimedia.

3. Xml (*extensible markup language*):

Establecido por el **W3C**, **XML** se basa en los principios de SGML, pero se simplificó para facilitar su uso en el intercambio de datos entre diferentes sistemas. Aunque no es un estándar en el sentido de una norma ISO, se considera una recomendación ampliamente aceptada para aplicaciones web, bases de datos y servicios.

4. Xhtml (*extensible hypertext markup language*):

XHTML, desarrollado como un híbrido entre HTML y XML, fue estandarizado por el **W3C**. Introdujo reglas estrictas basadas en XML, asegurando una mayor consistencia en el código y compatibilidad con navegadores y dispositivos modernos.

5. Css (*cascading style sheets*):

Aunque no es un lenguaje de marcas, **CSS** es un estándar desarrollado por el **W3C** que complementa lenguajes como HTML y XHTML, permitiendo controlar la apariencia y diseño de los documentos. Las versiones actuales (como **CSS3**) incluyen características avanzadas como animaciones y efectos visuales.

6. Svg (*scalable vector graphics*):

SVG, un estándar desarrollado por el **W3C**, se utiliza para describir gráficos vectoriales en dos dimensiones. Basado en XML, SVG permite la creación de imágenes escalables que pueden integrarse fácilmente en aplicaciones web.

7. Mathml (*mathematical markup language*):

MathML es un estándar del **W3C** diseñado para representar notaciones matemáticas en formato estructurado y reutilizable. Se utiliza en documentos científicos y educativos, permitiendo mostrar ecuaciones matemáticas en aplicaciones web.

8. Latex y tex:

Aunque **TeX** y **LaTeX** no tienen un estándar formal, han sido adoptados como estándares de facto para la creación de documentos técnicos y científicos, especialmente en matemáticas, física e ingeniería.

9. Rtf (*rich text format*):

Desarrollado por Microsoft, **RTF** no está regido por un organismo de estandarización formal, pero sigue siendo ampliamente reconocido como un formato estándar para compartir documentos de texto enriquecido entre diferentes plataformas y aplicaciones.

CARACTERÍSTICAS DE LOS LENGUAJES DE MARCADO

Todo lenguaje de marcas debe cumplir con una serie de características esenciales que lo hacen práctico y versátil:

- **Codificación en texto plano:**

Los documentos generados están formados exclusivamente por texto sin formato, compuesto únicamente de caracteres. Esto permite utilizar diferentes sistemas de codificación, como **ASCII**, **Unicode**, entre otros.

- **Facilidad de interpretación:**

Los archivos creados con lenguajes de marcas pueden ser interpretados o leídos de manera directa. Además, su codificación es tan sencilla que puede ser realizada con cualquier editor de texto básico.

- **Portabilidad:**

Al tratarse de documentos basados en texto plano, son independientes de la plataforma, el sistema operativo o el software utilizado para crearlos. Esto asegura su compatibilidad en diversos entornos.

- **Compactibilidad:**

En un solo archivo se integra tanto el contenido textual como las marcas que lo describen o estructuran, lo que evita la necesidad de archivos adicionales.

- **Flexibilidad:**

Estos lenguajes ofrecen la posibilidad de personalización, permitiendo añadir etiquetas, atributos o elementos según las necesidades específicas del usuario o del proyecto.

ESTRUCTURA DE UN DOCUMENTO CON LENGUAJE DE MARCADO

Internet se ha convertido en el entorno virtual más diverso que la humanidad ha creado. Cada día, cientos de miles de nuevos usuarios de todo el mundo se suman a esta red global, y no hay señales de que su crecimiento vaya a detenerse. Este espacio virtual está abierto a todos, ofreciendo un lugar donde se pueden realizar negocios, comunicarse, buscar información o simplemente disfrutar explorando sus innumerables rincones.

En este capítulo, aprenderemos a construir nuestra propia página web utilizando **HTML** (*HyperText Markup Language*). Este lenguaje de marcas de hipertexto permite estructurar contenido mediante etiquetas, las cuales contienen información adicional sobre la organización y presentación del texto. HTML es interpretado por los navegadores web para representar visualmente las páginas que vemos.

La historia de **HTML** se remonta a 1991, cuando **Tim Berners-Lee**, creador de la World Wide Web, publicó el primer documento que usaba este lenguaje. En su versión inicial, HTML era bastante simple, con tan solo 22 etiquetas. Sin embargo, con el tiempo, el lenguaje ha evolucionado considerablemente. Por ejemplo, la especificación de **HTML 4.1** ya contaba con más de 90 etiquetas. Desde entonces, el desarrollo de HTML ha sido constante, adaptándose a las necesidades cambiantes de la web y ampliando sus capacidades.

Un documento en un **lenguaje de marcado** está organizado siguiendo una estructura específica que facilita tanto su interpretación por los navegadores como su correcta presentación. Esta estructura se compone de varias secciones y etiquetas clave que definen su contenido y formato.

En general, un documento HTML sigue la siguiente estructura básica:

```
<!DOCTYPE html>
<html>
  <head>
    <title> Título de la Página </title>
  </head>
  <body>
    <h1> Encabezado principal </h1>
    <p> Este es un párrafo de texto en el cuerpo del
documento. </p>
  </body>
</html>
```

- **<!DOCTYPE html>**: Es una declaración que indica al navegador que el documento está escrito en HTML5, el cual es el estándar más reciente de HTML. Esto le permite al navegador interpretar el contenido de acuerdo con las reglas de HTML5.
- **<html>**: Esta es la etiqueta de apertura del documento HTML. Todo el contenido de la página web, incluidos los elementos de la cabecera y el cuerpo, debe estar dentro de estas etiquetas.
- **<head>**: Dentro de la etiqueta **<html>**, la sección **<head>** contiene información meta sobre el documento, como el título que aparecerá en la pestaña del navegador o los enlaces a hojas de estilo externas (CSS) o scripts (JavaScript).
- **<title>**: Dentro del **<head>**, esta etiqueta define el título de la página web, el cual se muestra en la pestaña del navegador.
- **<body>**: En esta sección se encuentra el contenido visible para el usuario. Aquí es donde se colocan los elementos como textos, imágenes, enlaces, etc.
- **<h1>**: Las etiquetas de encabezado (**<h1>**, **<h2>**, **<h3>**, etc.) se utilizan para definir títulos y subtítulos. **<h1>** es el encabezado principal de la página.
- **<p>**: La etiqueta **<p>** se usa para definir párrafos de texto.

Más información

[Consulta documentación y estructura HTML](#)

Cada etiqueta tiene una función específica, lo que permite al navegador estructurar y mostrar el contenido correctamente. Al utilizar estas etiquetas y estructurarlas de manera coherente, el contenido se organiza de manera clara y fácil de entender tanto para los navegadores como para los usuarios.

IDENTIFICACIÓN DE ETIQUETAS Y ATRIBUTOS

En un documento HTML, las **etiquetas** y los **atributos** juegan un papel crucial para definir tanto el contenido como su presentación. A continuación, explicaremos cómo identificar cada uno y cómo se utilizan correctamente dentro de un documento.

Las **etiquetas** en HTML son elementos que definen partes del contenido. Por ejemplo, la etiqueta **<p>** indica un párrafo, y **<h1>** se usa para crear el encabezado principal de la página. Las etiquetas siempre están rodeadas por los símbolos **<** y **>**.

Ejemplo:

```
<p> Este es un párrafo. </p>
```

En este caso, **<p>** es la etiqueta de apertura, y **</p>** es la etiqueta de cierre. La etiqueta de cierre se diferencia de la de apertura por la barra diagonal (**/**) antes del nombre de la etiqueta. El contenido, en este caso el texto "Este es un párrafo", va entre las etiquetas de apertura y cierre.

Los **atributos** son propiedades adicionales que se añaden a las etiquetas para modificar su comportamiento o apariencia. Se colocan dentro de la etiqueta de apertura, siempre después del nombre de la etiqueta, separados por espacios. Un atributo tiene un nombre y un valor, que se asignan con el signo igual (**=**) y se colocan entre comillas.

Ejemplo:

La etiqueta **<a>** se usa para crear enlaces, y uno de sus atributos más comunes es **href**, que define la URL a la que el enlace llevará al usuario. En este caso, **href="https://www.ejemplo.com"** es el atributo de la etiqueta **<a>**, que indica la dirección web a la que se dirigirá el enlace. El texto "Visitar Ejemplo" será lo que el usuario vea y podrá hacer clic.

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

Ejemplo

Un solo elemento puede tener varios atributos. Por ejemplo, una imagen puede tener atributos como **src** (la fuente de la imagen) y **alt** (texto alternativo para la imagen en caso de que no se cargue). Aquí, **src="imagen.jpg"** indica el archivo de imagen y **alt="Descripción de la imagen"** proporciona una descripción de la imagen para los usuarios con discapacidad visual o cuando la imagen no se puede cargar.

```

```

Las **etiquetas** definen el contenido y su estructura, mientras que los **atributos** permiten personalizar y controlar cómo ese contenido se presenta o se comporta. La correcta identificación y uso de etiquetas y atributos es fundamental para crear páginas web accesibles y bien estructuradas.

4.1. Elementos textuales

Cualquier documento HTML sigue una estructura básica que permite al navegador interpretar y mostrar correctamente el contenido.

Ejemplo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

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

Esta estructura se desglosa en las siguientes partes:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"http://www.w3.org/TR/html4/strict.dtd">: Esta línea inicial define la versión de **HTML** que se está utilizando, también conocida como la **declaración del tipo de documento (DTD)**. Indica al navegador que el documento sigue los estándares de **HTML 4.01**.

<head>: Dentro de la etiqueta **<head>**, se coloca toda la información relacionada con el documento que no es visible para el usuario. Esto incluye el título de la página, las palabras clave para los motores de búsqueda, el autor del documento y otros metadatos que ayudan a definir la página.

<body>: Esta es la parte más importante del documento, ya que es donde se coloca todo el contenido visible para el usuario, como textos, imágenes, enlaces y otros elementos que se presentarán en el navegador.

4.1.1. Definición del tipo de documento (dtd)

Para garantizar que un documento HTML sea válido según los estándares del **W3C**, es necesario especificar qué versión de **HTML** se está utilizando. Es recomendable declarar siempre un **DTD** (Definición del Tipo de Documento), ya que no solo asegura la validez del documento, sino que también ayuda a separar los datos de la presentación. Además, al utilizar un **DTD**, el documento será más flexible y podrá adaptarse fácilmente a nuevas especificaciones del lenguaje.

En la especificación **HTML 4.01**, existen tres tipos de **DTD** entre los cuales se puede elegir, dependiendo de las necesidades del proyecto. A continuación, se presentan los tres tipos de **DTD**, desde el más estricto hasta el más permisivo:

- **DTD Estricto (Strict DTD):**

Este es el **DTD** más riguroso. Incluye todos los elementos y atributos que define la especificación, pero excluye aquellos relacionados con la presentación del contenido, como los atributos de estilo. El **W3C** recomienda no utilizar estos atributos de presentación, ya que su funcionalidad ha sido reemplazada por **CSS** (Hojas de Estilo en Cascada), como se verá en el siguiente capítulo.

La declaración de este **DTD** es la siguiente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- **DTD Transicional (Transitional DTD):**

Este **DTD** es más permisivo y permite el uso de los atributos de presentación que el **W3C** recomienda evitar. Incluye todos los elementos de la versión estricta, además de aquellos elementos que permiten estilos de presentación más antiguos.

La declaración de este **DTD** es la siguiente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- **DTD para Marcos (Frameset DTD):**

Este **DTD** es similar al **DTD Transicional**, pero con una diferencia importante: incluye soporte para **frames**. En lugar de usar el modelo de contenido tradicional con la etiqueta **<body>**, se emplea la etiqueta **<frameset>**, que permite dividir la página en varias secciones independientes.

La declaración de este **DTD** es la siguiente:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

4.1.2. Cabecera

La cabecera de un documento HTML se encuentra delimitada por las etiquetas **<head>** y **</head>**. Aunque la información contenida en la cabecera no es visible directamente para el usuario, juega un papel esencial en la configuración y organización de la página web. Dentro de esta sección se incluyen datos importantes como el título de la página, palabras clave para los motores de búsqueda, detalles sobre el autor, y otros metadatos que ayudan a los navegadores y motores de búsqueda a entender mejor el contenido del documento.

A continuación, se detallan algunas de las etiquetas más comunes que se pueden incluir dentro de la cabecera de un documento HTML:

<TITLE>

La etiqueta **<title>** permite especificar el título de la página web, el cual aparece en la barra de título del navegador o en la pestaña. Es fundamental asignar un título claro y representativo, ya que este será el nombre que el usuario verá cuando desee guardar la página en sus favoritos.

Ejemplo:

```
<html>  
  <head>  
    <title>Mi Página Web</title>  
  </head>  
  <body>  
  </body>
```

```
</html>
```

<META>

La etiqueta **<meta>** se utiliza para especificar metadatos, es decir, información adicional sobre el documento HTML. Esto incluye detalles como el autor de la página, tipo de contenido, y más. Los metadatos no se muestran directamente al usuario, pero son utilizados por los motores de búsqueda y otras herramientas de la web.

Los principales atributos de la etiqueta **<meta>** son:

- **name:** Define el nombre de la propiedad.
- **content:** Especifica el valor de la propiedad.
- **scheme:** Indica el esquema utilizado para interpretar el valor de la propiedad.
- **http-equiv:** Se usa para definir los encabezados HTTP y se puede emplear en lugar del atributo **name**.

Ejemplos de uso:

```
<meta name="description" content="Creación de páginas web">
```

```
<meta name="keywords" content="HTML, CSS, JavaScript, PHP">
```

```
<meta name="author" content="Juan Pérez">
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

- En el primer ejemplo, se describe el contenido de la página.
- En el segundo, se incluyen las palabras clave que los motores de búsqueda usarán para indexar el contenido.
- El tercer ejemplo proporciona información sobre el autor del sitio.
- El cuarto especifica la codificación de caracteres del documento.

<LINK>

La etiqueta **<link>** se utiliza para crear relaciones entre el documento y otros recursos. Es comúnmente usada para vincular hojas de estilo externas, archivos de scripts o incluso para organizar el contenido dentro de una serie de

documentos. Es importante destacar que los enlaces creados con **<link>** no se utilizan para mostrar contenido en la página.

Ejemplo:

```
<head>
  <title>Tema 2</title>
  <link rel="index" href="../indice.html" />
  <link rel="prev" href="tema1.html" />
  <link rel="next" href="tema3.html" />
  <link rel="alternate" media="print" href="tema2-
imp.html" />
  <link rel="alternate" hreflang="en" href="tema2-
en.html" />
</head>
```

- En las líneas **4** y **5**, se indica que el documento forma parte de una serie y que los documentos previos y siguientes son **tema1.html** y **tema3.html**, respectivamente. El navegador puede usar esta información para enlazar automáticamente con los botones "anterior" y "siguiente" de la barra de herramientas, si es compatible.
- En la línea **6**, se proporciona una versión del documento optimizada para impresión, como puede ser una versión sin menús o con un formato especial para imprimir.
- La línea **7** indica que existe una traducción al inglés de la página, que puede ser accesible a través del enlace **tema2-en.html**.

<BASE>

La etiqueta **<base>** establece una URL base para resolver todas las direcciones relativas dentro del documento. Además, permite definir el marco de destino para los enlaces creados en el documento. Para ello, se asigna una URL al atributo **href** y el destino de los enlaces al atributo **target**.

Ejemplo:

```
<base href="http://www.ejemplo.com/" target="_blank">
```

En este caso, todas las direcciones relativas en el documento se resolverán a partir de **http://www.ejemplo.com/**, y los enlaces se abrirán en una nueva ventana o pestaña, debido al atributo **target="_blank"**.

La cabecera de un documento HTML juega un papel fundamental en la organización y optimización de la página para los navegadores, motores de búsqueda y otras herramientas web. Utilizar las etiquetas adecuadas dentro de la cabecera permite mejorar la accesibilidad, el posicionamiento y la correcta presentación del contenido.

4.1.3. Cuerpo de un documento

El cuerpo de un documento HTML es la sección donde se encuentra todo el contenido visible de la página web. Esta parte del documento está delimitada por las etiquetas **<body>** y **</body>**, que se colocan justo después de la etiqueta **</head>**.

Dentro de esta sección es donde se coloca todo lo que el usuario verá en su navegador, como textos, imágenes, enlaces, tablas, formularios, y otros elementos que conforman la página web.

Aunque es posible definir algunas propiedades generales para el diseño, como el color de fondo o los márgenes, es recomendable evitar el uso de atributos obsoletos en la etiqueta **<body>**. Estos atributos han sido reemplazados por las **hojas de estilo CSS** (que veremos en el siguiente capítulo), que permiten un control más flexible y eficiente del diseño y la presentación de la página.

4.2. ESPACIADO Y SALTO DE LÍNEAS

En HTML, es importante controlar el espaciado y los saltos de línea para asegurar que el contenido se presente de manera ordenada y legible. Para lograr este objetivo, existen varias etiquetas y propiedades que nos permiten ajustar el espacio entre elementos y gestionar los saltos de línea.

**
** - Salto de Línea

La etiqueta **
** es utilizada para insertar un salto de línea en el texto. Al colocar esta etiqueta, el contenido que sigue aparecerá en una nueva línea. Es especialmente útil cuando se necesita un salto de línea dentro de párrafos o texto continuo sin crear un nuevo bloque de texto.

Ejemplo:

```
<p>Este es un párrafo con un salto de línea.<br>Este es el texto que aparecerá en la siguiente línea.</p>
```

4.2.1. Espaciado entre elementos

Para ajustar el espacio entre los elementos del documento, como párrafos o imágenes, podemos usar atributos de margen y padding en las hojas de estilo CSS. Sin embargo, en HTML también se puede controlar el espaciado mediante algunas etiquetas.

- **<p>**: Esta etiqueta define un párrafo, y por defecto, los navegadores dejan un espacio de separación entre cada párrafo.
- **<pre>**: Se utiliza para mostrar texto preformateado, respetando los espacios y saltos de línea.

Ejemplo

```
<p>Este es un párrafo con espaciado estándar entre párrafos.</p>
```

```
<p>Otro párrafo con un espacio adicional antes del  
texto.</p>
```

4.3. ENLACES INTERNOS Y EXTERNOS

Los enlaces en HTML son fundamentales para la navegación web. Nos permiten conectar diferentes partes de una misma página o redirigir a los usuarios a otras páginas de la web. Existen dos tipos principales de enlaces: **internos** y **externos**.

Para saber más: [Consulta esta guía de uso](#)

4.3.1. Enlaces Internos

Los enlaces internos son aquellos que dirigen al usuario a otra sección dentro de la misma página web. Estos enlaces se utilizan para facilitar la navegación dentro de un sitio, especialmente en páginas largas con mucho contenido. Se crean utilizando la etiqueta **<a>** y se hace referencia a un **id** en el mismo documento.

Para crear un enlace interno, es necesario seguir dos pasos:

1. Asignar un identificador único a la sección de destino utilizando el atributo **id**.
2. Crear el enlace utilizando la etiqueta **<a>**, apuntando al **id** de la sección de destino.

Ejemplo

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de Enlaces Internos</title>
</head>
<body>

<h2><a href="#seccion1">Ir a la Sección 1</a></h2>
<h2><a href="#seccion2">Ir a la Sección 2</a></h2>

<p id="seccion1">Esta es la sección 1 del documento.</p>
<p id="seccion2">Esta es la sección 2 del documento.</p>

</body>
</html>
```

En el ejemplo, el primer enlace llevará al usuario a la **Sección 1** y el segundo lo llevará a la **Sección 2**, ambos sin necesidad de recargar la página.

4.3.2. Enlaces externos

Los enlaces externos son aquellos que dirigen al usuario hacia una página web diferente, que puede estar en otro dominio. Estos enlaces permiten acceder a contenido fuera del sitio web actual, como otros sitios relacionados, redes sociales, artículos o recursos adicionales.

Para crear un enlace externo, basta con proporcionar la URL completa del destino dentro del atributo **href** de la etiqueta **<a>**.

Ejemplo

```
<a href="https://www.example.com" target="_blank">Visitar  
Example.com</a>
```

En este ejemplo, el enlace llevará al usuario al sitio web **example.com**. El atributo **target="_blank"** hace que el enlace se abra en una nueva pestaña del navegador, lo que es útil cuando se enlaza a sitios externos y se desea que el usuario no pierda la página actual.

Consideraciones Importantes

- **Enlaces internos:** Son útiles para mejorar la experiencia del usuario dentro de un sitio web, permitiendo una navegación más ágil sin recargar la página.
- **Enlaces externos:** Deben usarse cuando se necesita dirigir al usuario a recursos fuera del sitio web. Es recomendable usar **target="_blank"** para abrir los enlaces externos en una nueva pestaña y evitar que el usuario abandone tu página.

4.4. LISTAS ORDENADAS Y DESORDENADAS

Las listas en HTML son útiles para presentar información de manera estructurada y clara. Existen dos tipos principales de listas: **listas ordenadas** y **listas desordenadas**. Cada tipo de lista tiene su propia función y presentación, dependiendo de si el orden de los elementos es relevante o no.

4.4.1. Listas ordenadas

Las listas ordenadas son aquellas en las que los elementos siguen un orden específico y se numeran automáticamente. Se utilizan cuando el orden de los elementos es importante, como en una secuencia de pasos o una lista jerárquica.

Para crear una lista ordenada en HTML, se utiliza la etiqueta **** para definir la lista, y dentro de ella, se colocan las etiquetas **** para cada uno de los elementos de la lista. Los elementos se numeran automáticamente en función de su posición.

Ejemplo

```
<ol>
  <li>Primero, enciende el dispositivo.</li>
  <li>Luego, abre la aplicación.</li>
  <li>Finalmente, selecciona tu opción preferida.</li>
</ol>
```

En este ejemplo, los elementos de la lista aparecerán numerados en orden secuencial.

4.4.2. Listas desordenadas

Las listas desordenadas son aquellas en las que el orden de los elementos no es importante. Los elementos de una lista desordenada se representan con viñetas o puntos. Este tipo de lista es adecuado para mostrar elementos que no necesitan estar organizados de manera secuencial.

Para crear una lista desordenada en HTML, se utiliza la etiqueta **** para definir la lista, y dentro de ella, se colocan las etiquetas **** para cada uno de los elementos de la lista.

Ejemplo

```
<ul>
  <li>Frutas</li>
  <li>Verduras</li>
  <li>Carnes</li>
</ul>
```

En este caso, los elementos aparecerán con viñetas, sin importar el orden en el que están listados.

4.4.3. Listas anidadas

También es posible crear listas dentro de otras listas, lo que permite organizar la información de manera jerárquica. Se pueden anidar tanto listas ordenadas como desordenadas, según sea necesario.

Ejemplo

```
<ul>
  <li>Frutas
    <ul>
      <li>Manzana</li>
      <li>Plátano</li>
    </ul>
  </li>
  <li>Verduras
    <ul>
      <li>Lechuga</li>
      <li>Zanahoria</li>
    </ul>
  </li>
</ul>
```

En este ejemplo, se ha creado una lista desordenada con elementos que contienen otras listas desordenadas, proporcionando un formato jerárquico.

Consideraciones importantes

- **Listas ordenadas:** Son útiles cuando el orden de los elementos es importante, como en tutoriales, instrucciones o clasificaciones.
- **Listas desordenadas:** Se utilizan cuando el orden de los elementos no es relevante, como en listas de cosas que no necesitan seguir un orden específico, como una lista de compras o categorías.

- **Listas anidadas:** Permiten organizar mejor la información, creando subcategorías dentro de una lista.

4.5. TABLAS

Las tablas son una de las herramientas más útiles y flexibles en la construcción de páginas web, ya que permiten organizar y presentar el contenido de manera estructurada. Son especialmente prácticas cuando se necesita categorizar información o mostrar datos en un formato claro y accesible.

Para saber más: [Crear tablas en HTML, truco fácil y rápido y luego cómo dar formato a la tabla usando CSS.](#)

4.5.1. Elementos Principales de una Tabla

Para crear una tabla en HTML, utilizamos varias etiquetas clave, que nos permiten definir tanto la estructura como la apariencia de la tabla:

- **<TABLE>**: Esta etiqueta se utiliza para definir el inicio de una tabla. También permite establecer algunas propiedades generales, como el borde o el ancho de la tabla.
- **<TR>**: Se emplea para definir cada fila dentro de la tabla. Cada fila debe estar contenida dentro de una etiqueta <tr>, y las celdas que forman esa fila estarán dentro de la etiqueta <td>.
- **<TD>**: Es la etiqueta que se usa para definir las celdas dentro de una fila. Cada celda puede contener texto, imágenes, enlaces o cualquier otro tipo de contenido.

A continuación, te mostramos un ejemplo sencillo de una tabla con tres filas y tres columnas. Cada etiqueta está comentada para explicar su función:

Ejemplo

- La tabla tiene un borde definido por `border="1"`, lo que la hace visible.
- La etiqueta `<tr>` define las filas de la tabla.
- Dentro de cada fila, las etiquetas `<td>` definen las celdas que contienen los datos.

```
<table border="1">
  <tr>
    <td>Mes</td>
    <td>Ingresos</td>
    <td>Gastos</td>
  </tr>
  <tr>
    <td>Enero</td>
    <td>1050€</td>
    <td>150€</td>
  </tr>
  <tr>
    <td>Febrero</td>
    <td>1150€</td>
    <td>450€</td>
  </tr>
</table>
```

4.5.2. Modificadores de las Etiquetas de Tabla

HTML también permite personalizar las tablas con diversos atributos que modifican el comportamiento y la apariencia de las celdas, filas o la tabla completa. Algunos de los atributos más comunes son:

- **colspan:** Permite que una celda ocupe varias columnas. Por ejemplo, si una celda necesita abarcar dos o más columnas, se utiliza este atributo.

- **rowspan**: Similar al colspan, pero en este caso permite que una celda ocupe varias filas.
- **width** y **height**: Permiten establecer el ancho y la altura de una celda, fila o tabla.
- **bgcolor**: Se usa para definir el color de fondo de una celda, fila o tabla.
-

Ejemplo

```
<table border="1">
  <tr>
    <td colspan="2">Mes</td>
    <td>Ingresos</td>
  </tr>
  <tr>
    <td rowspan="2">Enero</td>
    <td>1050€</td>
    <td>150€</td>
  </tr>
  <tr>
    <td>1150€</td>
    <td>450€</td>
  </tr>
</table>
```

ELEMENTOS ESTRUCTURALES

Los elementos estructurales son fundamentales en el diseño de una página web. Ayudan a organizar y dividir el contenido de manera clara y lógica, facilitando tanto la navegación para el usuario como la indexación por parte de los motores de búsqueda. A continuación, se describen algunos de los elementos estructurales más utilizados en HTML.

<header>

El elemento <header> define la sección de cabecera de un documento o una sección dentro de la página. Por lo general, se utiliza para contener elementos como el logotipo, el título principal, los menús de navegación y otros elementos introductorios.

```
<header>
  <h1>Bienvenido a Mi Página Web</h1>
  <nav>
    <ul>
      <li><a href="#home">Inicio</a></li>
      <li><a href="#about">Sobre Nosotros</a></li>
      <li><a href="#contact">Contacto</a></li>
    </ul>
  </nav>
</header>
```

En este ejemplo, el encabezado contiene un título principal <h1> y un menú de navegación <nav> con enlaces a diferentes secciones de la página.

<nav>

El elemento <nav> se utiliza para definir una sección de navegación dentro de la página web. Este elemento contiene los enlaces principales que permiten a los usuarios moverse entre las distintas secciones de la página o entre diferentes páginas de un mismo sitio.

```
<nav>
  <ul>
    <li><a href="#home">Inicio</a></li>
    <li><a href="#services">Servicios</a></li>
    <li><a href="#contact">Contacto</a></li>
  </ul>
</nav>
```

Aquí, el `<nav>` contiene una lista de enlaces que dirigen a diferentes secciones de la página.

`<section>`

El elemento `<section>` se utiliza para agrupar contenido relacionado en bloques dentro de una página. Cada sección debe ser autónoma, es decir, contener información relacionada entre sí, y generalmente tiene un encabezado propio.

```
<section>
  <h2>Servicios</h2>
  <p>Ofrecemos una amplia gama de servicios para satisfacer
tus necesidades.</p>
</section>
```

En este caso, la sección contiene un título y un párrafo que describen los servicios ofrecidos.

`<article>`

El elemento `<article>` se utiliza para definir contenido autónomo e independiente, como publicaciones de blog, noticias o comentarios. Un artículo puede estar completo por sí mismo, sin necesidad de depender del resto del contenido de la página.

```
<article>
  <h2>Noticias Recientes</h2>
  <p>Este es un artículo sobre los eventos recientes en
nuestra empresa...</p>
</article>
```

Aquí, el contenido dentro del `<article>` es un artículo de noticias con su propio título y párrafo.

<aside>

El elemento <aside> se usa para definir contenido relacionado pero que no es esencial para el contenido principal de la página. Se utiliza típicamente para incluir elementos como barras laterales, citas, enlaces adicionales o anuncios.

```
<aside>
  <h3>Enlaces útiles</h3>
  <ul>
    <li><a href="#link1">Enlace 1</a></li>
    <li><a href="#link2">Enlace 2</a></li>
  </ul>
</aside>
```

Este <aside> contiene enlaces adicionales que complementan el contenido principal de la página pero no son estrictamente necesarios.

<footer>

El elemento <footer> se utiliza para definir la parte final de una página o sección. Normalmente, contiene información como el aviso de derechos de autor, enlaces de contacto, enlaces a políticas de privacidad, entre otros.

```
<footer>
  <p>&copy; 2025 Mi Página Web. Todos los derechos
reservados.</p>
</footer>
```

En este caso, el <footer> contiene una nota de derechos de autor.

ESTRUCTURACIÓN SEMÁNTICA

La estructuración semántica en HTML se refiere al uso de etiquetas que describen el significado del contenido que contienen, más allá de su presentación visual. Estas etiquetas proporcionan un contexto adicional a los navegadores y motores de búsqueda, lo que mejora la accesibilidad, la optimización para motores de búsqueda (SEO) y la comprensión general del

contenido por parte de los usuarios. A continuación, se describen algunos de los elementos semánticos más utilizados en HTML.

<header>

El elemento `<header>` se utiliza para definir la cabecera de un documento o una sección dentro de la página. Su función semántica es indicar que el contenido en su interior es introducción o información de navegación general. A menudo, se encuentra en la parte superior de la página y puede incluir el logo, el título principal, o un menú de navegación.

```
<header>
  <h1>Bienvenido a Nuestro Sitio Web</h1>
  <nav>
    <ul>
      <li><a href="#home">Inicio</a></li>
      <li><a href="#about">Sobre Nosotros</a></li>
      <li><a href="#contact">Contacto</a></li>
    </ul>
  </nav>
</header>
```

En este ejemplo, el `<header>` contiene un título y un menú de navegación, que brindan contexto sobre la página y sus secciones.

<nav>

El elemento `<nav>` es fundamental en la estructuración semántica, ya que define una sección específica para enlaces de navegación. Esto permite a los usuarios acceder rápidamente a otras partes del sitio web. Los motores de búsqueda también pueden identificar esta sección y considerarla en su índice.

```
<nav>
  <ul>
    <li><a href="#services">Servicios</a></li>
    <li><a href="#portfolio">Portafolio</a></li>
    <li><a href="#contact">Contacto</a></li>
  </ul>
</nav>
```

El `<nav>` proporciona una agrupación lógica de enlaces relacionados con la navegación.

`<section>`

El elemento `<section>` se utiliza para agrupar contenido relacionado en bloques autónomos. Este elemento es semánticamente significativo porque indica que el contenido dentro de él está relacionado temáticamente y que puede considerarse como una sección independiente de la página.

```
<section>
  <h2>Servicios</h2>
  <p>Ofrecemos soluciones personalizadas para tu
negocio.</p>
</section>
```

Aquí, el contenido de la sección sobre "Servicios" está claramente marcado con la etiqueta `<section>`, lo que facilita la comprensión de su propósito.

`<article>`

El elemento `<article>` define contenido independiente y autónomo que tiene sentido por sí mismo, como un artículo de blog, una noticia, o una entrada en un foro. Este tipo de contenido se puede distribuir de manera independiente sin perder su contexto.


```
<article>
  <h2>Noticia Destacada</h2>
  <p>Hoy lanzamos una nueva versión de nuestro
producto...</p>
</article>
```

El uso del `<article>` señala que este contenido es autónomo y puede ser comprendido independientemente del resto de la página.

`<aside>`

El elemento `<aside>` se utiliza para identificar contenido complementario, pero que no es esencial para la comprensión del contenido principal. Usualmente, se coloca en una barra lateral o como un bloque adicional que contiene información relacionada, pero no imprescindible.

```
<aside>
  <h3>Artículos Relacionados</h3>
  <ul>
    <li><a href="#article1">Artículo 1</a></li>
    <li><a href="#article2">Artículo 2</a></li>
  </ul>
</aside>
```

Este contenido complementario está marcado con `<aside>`, lo que indica que no es parte del flujo principal del contenido, pero es relevante para el usuario.

`<footer>`

El elemento `<footer>` define el pie de página de un documento o sección. Generalmente, contiene información de contacto, derechos de autor, enlaces a políticas, o cualquier otro detalle adicional que no esté directamente relacionado con el contenido principal, pero que es importante incluir.

```
<footer>
  <p>&copy; 2025 Mi Empresa. Todos los derechos
reservados.</p>
  <p><a href="#privacy-policy">Política de
privacidad</a></p>
</footer>
```

En este caso, el `<footer>` contiene una nota de derechos de autor y un enlace a la política de privacidad del sitio.

Encontramos una serie de ventajas:

- **Mejora de la accesibilidad:** Las etiquetas semánticas permiten a los usuarios con discapacidades, que usan tecnologías de asistencia como lectores de pantalla, comprender mejor la estructura y el propósito del contenido.
- **Optimización en motores de búsqueda (SEO):** Los motores de búsqueda utilizan la semántica de un documento para indexar su contenido de manera más eficiente. Las etiquetas adecuadas permiten que el contenido se clasifique mejor en los resultados de búsqueda.
- **Mantenimiento y organización:** El uso de etiquetas semánticas mejora la legibilidad y mantenibilidad del código, haciendo que sea más fácil entender y modificar la estructura de una página.

Ejemplo: [CREA TU PROPIO FOOTER HACIENDO CLICK](#)

UNIDAD 2. Elementos de multimedia y formularios

INSERCIÓN DE ELEMENTOS MULTIMEDIA

La integración de elementos multimedia en una página web, como videos, audios o animaciones, puede mejorar significativamente la experiencia del usuario, siempre que se utilicen de forma adecuada. Es esencial seleccionar formatos compatibles con los navegadores más utilizados y optimizar los archivos para garantizar un rendimiento óptimo de la página.

Entre los formatos de video más comunes para la web se encuentran MP4 (con codificación H.264) y WebM, debido a su amplia compatibilidad y compresión eficiente. Para el audio, los formatos recomendados son MP3, OGG y AAC, ya que ofrecen buena calidad con tamaños de archivo reducidos.

Al incorporar estos elementos, es importante utilizar etiquetas HTML específicas, como `<video>` y `<audio>`, que permiten integrar el contenido de manera nativa sin requerir complementos externos. Además, se debe configurar correctamente las opciones de reproducción, como controles, reproducción automática o bucle, para mejorar la accesibilidad y usabilidad.

También es aconsejable proporcionar alternativas al contenido multimedia, como subtítulos o transcripciones, para cumplir con las pautas de accesibilidad y garantizar que todos los usuarios puedan interactuar con el sitio web, independientemente de sus limitaciones tecnológicas o físicas.

Por último, es crucial tener en cuenta la optimización de los archivos multimedia, ya que tamaños excesivamente grandes pueden afectar negativamente el rendimiento de la página. Reducir la resolución o la tasa de bits, cuando sea posible, es una práctica que contribuye a una experiencia de usuario más fluida y eficiente.

TRATAMIENTO WEB DE IMÁGENES

El tratamiento de imágenes para su uso en la web es una tarea fundamental para garantizar que las páginas sean visualmente atractivas, rápidas y funcionales. Esto implica optimizar el tamaño, formato y resolución de las imágenes para adaptarlas a las necesidades específicas del sitio web.

Elección del formato

La elección del formato de imagen en la construcción de páginas web es crucial para garantizar una experiencia de usuario óptima, el rendimiento del sitio y la compatibilidad con diferentes dispositivos y navegadores. Cada formato tiene características específicas que lo hacen adecuado para ciertos usos:

- **Rendimiento:** Formatos como JPEG y WEBP ofrecen una buena calidad con tamaños de archivo reducidos, lo que acelera los tiempos de carga, especialmente en dispositivos móviles o conexiones lentas.
- **Calidad visual:** PNG y SVG son ideales para gráficos, logotipos e imágenes que requieren alta calidad o transparencia, mientras que formatos como GIF son útiles para animaciones simples.
- **Compatibilidad:** Algunos formatos, como AVIF o WEBP, son modernos y muy eficientes, pero pueden no ser compatibles con navegadores antiguos, lo que requiere considerar alternativas o medidas adicionales.
- **Accesibilidad y estética:** La correcta elección garantiza que los elementos visuales sean claros, atractivos y funcionales, mejorando la accesibilidad y percepción del sitio.

Optimización del tamaño de archivo

Para garantizar tiempos de carga rápidos, es esencial reducir el tamaño de las imágenes. Esto se puede lograr utilizando herramientas o programas específicos para comprimir las imágenes sin afectar significativamente su calidad. La optimización también mejora el rendimiento del sitio y su posicionamiento en motores de búsqueda.

Ajuste de tamaño y resolución

Dado que las imágenes en una página web están destinadas a visualizarse en

pantalla, no necesitan resoluciones extremadamente altas. Una resolución estándar, como 72 ppp (píxeles por pulgada), suele ser suficiente. Además, es importante redimensionar las imágenes al tamaño exacto que se mostrarán en la página, evitando el escalado automático desde el navegador, que podría ralentizar la carga.

Uso de herramientas específicas

Existen múltiples herramientas y plataformas que facilitan el tratamiento de imágenes para la web, como Adobe Photoshop, GIMP o servicios en línea como TinyPNG. Estas herramientas permiten recortar, redimensionar, ajustar colores y optimizar imágenes de manera sencilla.

Implementación en el diseño web

Además del tratamiento técnico, es importante considerar el uso estratégico de las imágenes en el diseño de la página. Imágenes bien tratadas y adaptadas no solo mejoran la estética, sino que también influyen en la experiencia del usuario y la accesibilidad del sitio.

Un tratamiento adecuado de las imágenes garantiza un equilibrio entre calidad visual y rendimiento, lo que contribuye a una experiencia de navegación más eficiente y agradable.

2.1. MAPA DE IMÁGENES

Los mapas de imágenes permiten asignar enlaces específicos a diferentes zonas dentro de una misma imagen. Esto se logra mediante la combinación de las etiquetas `` y `<map>`, donde cada área definida dentro de la imagen está vinculada a una URL distinta.

La etiqueta `` incluye un atributo llamado `usemap`, que relaciona la imagen con un mapa definido por la etiqueta `<map>`. Este mapa utiliza el atributo `name` para identificarlo, asegurando que coincida con el valor especificado en `usemap`.

Dentro del mapa, se utiliza la etiqueta `<area>` para delimitar las zonas interactivas de la imagen. Cada área puede tener diferentes formas, como

rectángulos (rect), círculos (circle) o polígonos (poly), definidas mediante el atributo shape. Las coordenadas necesarias para delimitar las áreas se establecen con el atributo coords, mientras que el atributo href permite enlazar cada zona con una página web específica.

Por ejemplo, un mapa de imagen podría dividirse en tres áreas interactivas: una rectangular, una circular y una poligonal, cada una con un enlace distinto. Este método es útil para mejorar la funcionalidad de las imágenes en un sitio web, permitiendo a los usuarios interactuar con diferentes partes de una misma imagen.

2.2. TRATAMIENTO WEB DE VÍDEOS

Al igual que sucede con los archivos de audio, los vídeos están disponibles en una variedad de formatos y codificaciones. Entre los más utilizados se encuentran MPEG, MOV, OGG, FLV, WMV y AVI.

Con la evolución de Internet, el consumo de contenido en vídeo ha experimentado un gran avance, destacándose especialmente el uso del **streaming**, que permite reproducir vídeos en tiempo real sin necesidad de descargar el archivo completo.

2.3. FORMATOS DE IMAGEN, AUDIO Y VÍDEO

2.3.1. Imagen

En el diseño y desarrollo de páginas web, los formatos de imagen cumplen un papel fundamental al permitir la representación visual de contenido de manera efectiva. A continuación, se describen algunos de los formatos más utilizados:

JPEG (Joint Photographic Experts Group): Es uno de los formatos más comunes para imágenes fotográficas debido a su capacidad de compresión con pérdida, que reduce significativamente el tamaño del archivo sin sacrificar demasiada calidad visual. Ideal para fotografías y contenido con muchos colores.

PNG (Portable Network Graphics): Ofrece compresión sin pérdida, lo que mantiene la calidad original de la imagen. Soporta transparencias, lo que lo hace ideal para logotipos, íconos y gráficos.

GIF (Graphics Interchange Format): Limitado a 256 colores, es popular para animaciones simples debido a su capacidad de soportar cuadros consecutivos. Sin embargo, no es eficiente para imágenes con una amplia gama de colores.

SVG (Scalable Vector Graphics): Es un formato basado en vectores que permite redimensionar imágenes sin perder calidad. Es perfecto para gráficos, logotipos y elementos interactivos.

WEBP: Un formato moderno que combina compresión con y sin pérdida, logrando archivos de menor tamaño con alta calidad visual. Es cada vez más popular debido a su eficiencia y compatibilidad creciente con navegadores modernos.

BMP (Bitmap): Un formato antiguo con imágenes sin compresión. No es eficiente en términos de tamaño de archivo, por lo que su uso en la web es muy limitado.

TIFF (Tagged Image File Format): Ofrece alta calidad y se utiliza principalmente en edición profesional y almacenamiento de imágenes, pero no es común en la web debido a su tamaño de archivo.

AVIF (AV1 Image File Format): Un formato reciente que utiliza compresión avanzada para ofrecer imágenes de alta calidad con tamaños significativamente reducidos. Su adopción está en crecimiento, especialmente en entornos modernos.

2.3.2. Audio

Con el avance de las tecnologías web y la creciente demanda de interactividad, los contenidos multimedia, como el audio, han adquirido un papel esencial. Estos se utilizan en diversas aplicaciones, como la reproducción de música de fondo mientras el usuario navega por un sitio web.

Los formatos de audio se clasifican principalmente en función de la compresión aplicada durante su grabación. Esta clasificación se divide en tres grupos:

- **Audio sin comprimir:** Estos formatos almacenan el audio tal como fue grabado, ofreciendo excelente calidad, pero ocupando un espacio significativo. Algunos ejemplos comunes son WAV, AIFF, AU y MIDI.
- **Audio comprimido:** En estos formatos, el audio se codifica para ahorrar espacio, aunque requiere mayor procesamiento durante su reproducción. El formato más conocido en esta categoría es el MP3.

A continuación, se describen los formatos más utilizados, junto con sus principales características:

- **WAV (Waveform Audio File Format):** Creado por Microsoft e IBM, este formato generalmente no utiliza compresión, manteniendo la calidad original del sonido. Es ampliamente compatible con navegadores y entornos profesionales. Sin embargo, su principal desventaja es el gran espacio de almacenamiento que requiere.
- **AIFF (Audio Interchange File Format):** Este formato, desarrollado por Apple, es un estándar para audio sin comprimir. Es especialmente popular en sistemas Apple debido a su participación en su creación.
- **AU:** Un formato sin comprimir diseñado por Sun Microsystems, ampliamente utilizado en sistemas operativos GNU/Linux.
- **MIDI (Musical Instrument Digital Interface):** A diferencia de otros formatos, los archivos MIDI no contienen audio grabado, sino instrucciones que indican al ordenador cómo generar el sonido en tiempo real. Son extremadamente ligeros en tamaño, pero la calidad del sonido depende de la tarjeta de audio del dispositivo, lo que puede resultar en una experiencia menos satisfactoria en equipos básicos.
- **MP3 (MPEG-1/2 Audio Layer III):** Este formato es famoso por su capacidad de comprimir audio significativamente, permitiendo almacenar contenido de alta calidad en tamaños reducidos. Por ejemplo, un CD completo puede comprimirse en aproximadamente 50 MB, lo que lo

convierte en la opción preferida para compartir y almacenar música en la web.

La elección del formato de audio adecuado dependerá de las necesidades específicas del proyecto, equilibrando factores como calidad, espacio de almacenamiento y compatibilidad con los navegadores y dispositivos.

2.3.3. Vídeo

En la creación de páginas web, el formato de video es crucial para garantizar que los contenidos multimedia se muestren de forma óptima en diversos dispositivos y navegadores. Los formatos más comunes son:

- **MPEG (Movie Picture Experts Group):** Este es un códec de compresión de vídeo que logra reducir el tamaño de los archivos con mínimas pérdidas de calidad. Incluye una variedad de estándares, como MPEG-1, MPEG-2 y MPEG-4, utilizados para diferentes aplicaciones, desde videos caseros hasta transmisiones digitales.
- **OGG:** Este es un formato abierto y libre de patentes que se ha vuelto cada vez más popular, especialmente en el ámbito del streaming. Ofrece una excelente relación entre compresión y calidad, lo que lo hace ideal para aplicaciones en línea.
- **FLV (Flash Video):** Este formato, desarrollado por Adobe, se popularizó ampliamente debido a su uso en plataformas como YouTube. Para reproducirlo, es necesario tener instalado el reproductor Adobe Flash, aunque su uso ha disminuido con la llegada de nuevos estándares como HTML5.
- **AVI (Audio Video Interleave):** Este es un formato contenedor que permite combinar un flujo de vídeo con varios flujos de audio, cada uno codificado con un códec diferente. Su flexibilidad lo ha convertido en un formato ampliamente utilizado, aunque puede generar archivos más grandes en comparación con otros formatos modernos.

El conocimiento y la correcta selección del formato de vídeo son esenciales para garantizar una buena experiencia de usuario, tanto en términos de calidad visual como de compatibilidad y rendimiento en diferentes plataformas web. Se

detallan a continuación los tipos más comunes de elementos incrustados y su forma de uso.

ELEMENTOS INCRUSTADOS

Los elementos incrustados permiten integrar contenido interactivo o multimedia dentro de una página web, enriqueciendo la experiencia del usuario. Estos elementos son esenciales para incorporar funcionalidades como videos, audios, aplicaciones interactivas y otros recursos externos.

1. VÍDEOS

Los videos pueden incrustarse directamente en una página web utilizando la etiqueta `<video>`. Este elemento permite reproducir contenido multimedia sin necesidad de plugins externos.

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

Los navegadores modernos soportan formatos populares como MP4, WEBM y OGG, lo que asegura compatibilidad. Además, se pueden añadir atributos como autoplay, loop, y muted para personalizar su comportamiento.

2. AUDIO

El contenido de audio se puede integrar mediante la etiqueta `<audio>`. Al igual que con los videos, esta etiqueta soporta los formatos más utilizados, como MP3, OGG y WAV. Ejemplo de uso:

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

Se pueden incluir opciones adicionales, como reproducción automática o bucle continuo, para adaptarse a las necesidades del sitio.

3. IMÁGENES SVG

Los gráficos vectoriales escalables (SVG) permiten incrustar ilustraciones, iconos o gráficos interactivos directamente en el código HTML. Este formato es ideal para contenido que debe mantenerse nítido en cualquier tamaño. Ejemplo de uso:

```
<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red"  
/>  
</svg>
```

Los elementos SVG son altamente personalizables y pueden animarse con CSS o JavaScript.

4. IFRAMES

La etiqueta <iframe> permite incrustar contenido externo, como videos de YouTube, mapas de Google o aplicaciones web completas. Ejemplo:

```
<iframe src="https://www.youtube.com/embed/VIDEO_ID" width="560"  
height="315" frameborder="0" allowfullscreen></iframe>
```

Aunque es una herramienta poderosa, su uso debe ser moderado, ya que puede afectar el rendimiento de la página y la seguridad.

5. OBJETOS MULTIMEDIA AVANZADOS

La etiqueta <object> permite incrustar archivos multimedia avanzados, como documentos PDF, aplicaciones Flash (aunque su uso ha caído en desuso), y otros formatos no soportados directamente por HTML5. Ejemplo:

```
<object data="document.pdf" type="application/pdf"  
width="600" height="400"></object>
```

Los elementos incrustados son una herramienta clave para enriquecer las páginas web y ofrecer experiencias dinámicas y atractivas, siempre que se usen de manera adecuada y optimizada.

HERRAMIENTAS PARA EL TRATAMIENTO DE ELEMENTOS MULTIMEDIA

El manejo de elementos multimedia, como imágenes, audio y video, requiere herramientas especializadas que permitan editarlos, optimizarlos y adaptarlos para su integración en la web. Estas herramientas son esenciales para garantizar calidad visual y auditiva, una carga eficiente y una experiencia de usuario adecuada.

A continuación, se describen las principales herramientas utilizadas para el tratamiento de elementos multimedia, clasificadas por su función:

1. Herramientas de edición de imágenes

Estas aplicaciones permiten ajustar el tamaño, calidad, y otros aspectos de las imágenes para adaptarlas a las necesidades de un sitio web.

- **Adobe Photoshop:** Software profesional para edición avanzada de imágenes. Permite manipular gráficos, optimizar formatos y exportar en distintos tamaños y resoluciones.
- **GIMP (GNU Image Manipulation Program):** Una alternativa gratuita y de código abierto que ofrece funciones similares a Photoshop, ideal para ajustes de color, recorte y optimización.
- **TinyPNG:** Herramienta en línea para comprimir imágenes PNG y JPEG sin perder calidad, mejorando los tiempos de carga en la web.
- **Canva:** Plataforma intuitiva para crear y editar gráficos directamente en línea, ideal para diseños rápidos y sencillos.

2. Herramientas de edición de audio

Estas herramientas ayudan a recortar, mezclar y optimizar archivos de audio para su uso en proyectos web.

- **Audacity:** Software gratuito y de código abierto para edición de audio. Permite cortar pistas, ajustar volúmenes y convertir formatos como WAV, MP3 y OGG.
- **Adobe Audition:** Una herramienta profesional para la edición avanzada de audio, ideal para proyectos complejos como podcasts o bandas sonoras.
- **Online Audio Converter:** Aplicación web que facilita la conversión de formatos de audio y ajustes básicos de compresión.

3. Herramientas de edición de video

Para integrar videos en la web, estas herramientas permiten ajustar formatos, calidad y tamaños de archivo.

- **Adobe Premiere Pro:** Programa profesional para edición de video, ampliamente utilizado en la industria para crear contenido multimedia de alta calidad.
- **DaVinci Resolve:** Software gratuito que combina edición, efectos visuales y corrección de color, ideal para proyectos avanzados.
- **HandBrake:** Herramienta gratuita para convertir y comprimir videos en formatos como MP4, WEBM y otros compatibles con la web.
- **Kapwing:** Plataforma en línea para editar videos de manera rápida, con opciones para subtítulos, recortes y optimización.

4. Herramientas de conversión y optimización

Son esenciales para adaptar elementos multimedia al formato y tamaño ideales para su uso en la web.

- **FFmpeg:** Herramienta de línea de comandos potente y versátil para convertir formatos de audio y video, extraer pistas y realizar optimizaciones avanzadas.
- **CloudConvert:** Servicio en línea para convertir una amplia gama de archivos multimedia, incluyendo imágenes, audio y video.

- **Compressor.io:** Plataforma en línea para reducir el tamaño de imágenes y videos sin comprometer la calidad visual o auditiva.

5. Herramientas para diseño interactivo

Además de editar, algunas herramientas permiten crear elementos multimedia interactivos para enriquecer la experiencia web.

- **Adobe Animate:** Software para desarrollar animaciones y gráficos interactivos que se integran fácilmente con sitios web.
- **Blender:** Herramienta de código abierto para crear modelos y animaciones 3D, útiles en proyectos multimedia avanzados.

FORMULARIOS: CARACTERÍSTICAS, MÉTODOS DE ENVÍO, PROCESAMIENTO DE LOS DATOS

Los formularios son herramientas esenciales para que los usuarios puedan enviar información a través de una página web e interactuar con el servidor. Por ejemplo, a través de un formulario, un usuario puede ingresar su nombre y contraseña, lo que permite al sistema mostrarle su perfil o datos personales como correos electrónicos.

Aunque los formularios se utilizan principalmente para recopilar datos de los usuarios, su uso más común es justamente este: obtener información como nombres, preferencias o detalles personales.

Generalmente, los formularios envían los datos a una página dinámica (por ejemplo, usando PHP) que procesa la información y, si es necesario, la almacena en una base de datos. También existe la opción más sencilla de hacer que el formulario envíe los datos directamente a una dirección de correo electrónico, dependiendo de la necesidad del sitio web.

Características

Un formulario se compone de cuatro partes principales:

1. **Inicio del formulario:** Esta sección se define con la etiqueta `<form>`, y en ella se especifica cómo se enviarán los datos y cuál es el destino de la información (por ejemplo, la URL a la que se enviarán los datos).
2. **Elementos del formulario:** Es el núcleo del formulario, donde se definen los campos para que el usuario ingrese información. Los elementos más comunes incluyen las etiquetas `<input>`, `<textarea>` o `<select>`, que permiten recoger diferentes tipos de datos.
3. **Botones de acción:** Son los botones que permiten al usuario enviar el formulario o resetear los campos. Estos botones se pueden definir utilizando las etiquetas `<input>` o `<button>`, dependiendo de la funcionalidad deseada.
4. **Fin del formulario:** El formulario se cierra con la etiqueta `</form>`, que indica el final de la estructura y asegura que los datos se enviarán correctamente al destino establecido.

La etiqueta `<form>` marca el comienzo de un formulario en una página web y permite definir diversos atributos para configurar su funcionamiento. A continuación, se describen los atributos más comunes que esta etiqueta puede incluir:

- **Name:** Este atributo define el nombre del formulario. Es especialmente útil cuando hay varios formularios en una misma página o cuando se usan scripts para verificar los datos ingresados por el usuario.
- **Method:** Especifica el método de envío de los datos al servidor. Los dos métodos más comunes son:
 - **GET:** En este caso, los datos se agregan directamente a la URL, separados por un signo de interrogación. Por ejemplo, en una URL como `http://miweb.es/formulario.html?nombre=Javier`, el dato "nombre" con el valor "Javier" se envía como parte de la dirección web.
 - **POST:** Este método es más seguro que el GET, ya que los datos no se exponen en la URL, sino que se envían directamente al servidor de manera más confidencial.

- **Action:** Define la dirección a la que se envían los datos del formulario una vez que se envíen. La opción más común es enviar los datos a otra página web (por ejemplo, `action="http://miweb.es/destino.php"`) aunque también se pueden enviar a una dirección de correo electrónico (por ejemplo, `action="mailto:usuario@miweb.es"`).
- **Enctype:** Es un atributo opcional, pero recomendable, que especifica cómo se codifican los datos antes de enviarlos. Puede tomar varios valores, siendo los más comunes:
 - **text/plain:** Envía la información como texto plano.
 - **application/x-www-form-urlencoded:** Es el valor predeterminado y codifica los caracteres de los datos antes de enviarlos.
 - **multipart/form-data:** Utilizado principalmente para enviar archivos, sin realizar codificación sobre los datos.

Ejemplo:

A continuación, se muestra un ejemplo de un formulario que envía los datos a una dirección de correo electrónico:

```
<FORM action="mailto:usuario@miweb.es" method="post"
enctype="text/plain">
  <!-- Campos del formulario aquí -->
</FORM>
```

ELEMENTOS Y ATRIBUTOS DE FORMULARIO

Los formularios web permiten interactuar con los usuarios y recopilar datos, y para lograrlo, se utilizan diversos elementos y atributos. A continuación, se describen los elementos más comunes de un formulario, así como los atributos que se pueden emplear para personalizar su comportamiento.

Elementos del formulario

1. <input>

El elemento <input> es el más versátil y común en los formularios.

Permite crear distintos tipos de campos de entrada según el valor de su atributo `type`. Algunos de los tipos más comunes son:

- **text**: Campo para ingresar texto de una sola línea.
- **password**: Campo para ingresar contraseñas, mostrando los caracteres como asteriscos.
- **radio**: Botones de opción que permiten seleccionar una sola opción de un conjunto.
- **checkbox**: Casillas de verificación que permiten seleccionar múltiples opciones.
- **submit**: Un botón que envía el formulario.
- **reset**: Un botón que restablece los valores de todos los campos del formulario a sus valores predeterminados.
- **button**: Un botón que puede ejecutar una acción definida con JavaScript.

2. **<textarea>**

El elemento `<textarea>` se utiliza para crear un campo de texto que permita la entrada de varias líneas. Este campo es útil para comentarios, descripciones o mensajes largos.

3. **<select>**

El elemento `<select>` se utiliza para crear una lista desplegable. Dentro de este elemento, se deben incluir varias opciones definidas con la etiqueta `<option>`, y el usuario puede elegir una de ellas.

4. **<button>**

La etiqueta `<button>` es una alternativa al `<input>` de tipo "submit" o "reset". Permite crear botones con un contenido personalizado y es más flexible, ya que permite incluir texto, imágenes y otros elementos dentro del botón.

5. **<label>**

La etiqueta `<label>` se usa para proporcionar una descripción o etiqueta a los campos del formulario, mejorando la accesibilidad. Se puede asociar con un campo específico mediante el atributo `for`, que corresponde al id del campo.

Atributos de los elementos de formulario

1. **name**

Este atributo define el nombre del campo y es esencial para identificar los datos que se envían al servidor. Se utiliza en casi todos los elementos del formulario, como en `<input name="nombre" />` o `<textarea name="comentarios"></textarea>`.

2. **value**

El atributo value define el valor inicial del campo. En los campos de texto, radio y checkbox, value determina lo que se enviará al servidor cuando el formulario sea enviado. Por ejemplo, en un campo de entrada tipo "radio", `value="opcion1"` define el valor que se enviará si esa opción está seleccionada.

3. **placeholder**

Este atributo proporciona un texto guía dentro de un campo de entrada, que desaparece cuando el usuario comienza a escribir. Es útil para mostrar ejemplos o instrucciones breves.

4. **required**

El atributo required indica que un campo es obligatorio. El formulario no se enviará hasta que el usuario haya completado ese campo.

5. **disabled**

El atributo disabled desactiva un campo, evitando que el usuario interactúe con él o lo modifique.

6. **readonly**

Similar al atributo disabled, pero mientras que los campos con readonly son visibles y pueden ser copiados, no pueden ser modificados.

7. **maxlength**

Define el número máximo de caracteres que un usuario puede ingresar en un campo de texto.

8. **pattern**

Este atributo permite establecer una expresión regular que el valor del campo debe cumplir. Es útil para validar el formato de la entrada, como correos electrónicos o números de teléfono.

9. autofocus

El atributo autofocus hace que el campo de formulario obtenga el enfoque automáticamente cuando se carga la página, permitiendo al usuario comenzar a escribir inmediatamente.

10. accept

Este atributo se utiliza en el campo de tipo file para especificar los tipos de archivos que el usuario puede cargar. Por ejemplo, accept="image/png,image/jpeg" restringe la carga solo a imágenes en formato PNG y JPEG.

EJEMPLO

Vamos a ver un ejemplo de un formulario con varios elementos y atributos:

```
<form action="procesar_formulario.php" method="post">
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre"
placeholder="Ingresa tu nombre" required />

  <label for="email">Correo electrónico:</label>
  <input type="email" id="email" name="email" required
/>

  <label for="comentarios">Comentarios:</label>
  <textarea id="comentarios" name="comentarios"
placeholder="Escribe tus comentarios aquí"
maxlength="500"></textarea>

  <label for="newsletter">¿Deseas recibir nuestro
boletín?</label>
  <input type="checkbox" id="newsletter"
name="newsletter" value="si" />

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

6.1. CONTROLES DE FORMULARIO

Los controles de formulario son los elementos interactivos que permiten a los usuarios ingresar, seleccionar o modificar la información en un formulario web. Cada tipo de control se utiliza para recopilar un tipo específico de datos. A continuación, se describen los controles más comunes, sus características y usos.

Controles básicos de formulario

1. **<input>**

El control **<input>** es uno de los más versátiles, ya que permite crear diferentes tipos de campos de entrada según el valor de su atributo **type**.

Los controles más utilizados dentro de este tipo son:

- **text**: Un campo de texto donde el usuario puede escribir una sola línea de texto.
- **password**: Un campo para introducir contraseñas, donde los caracteres se muestran como asteriscos o puntos para ocultarlos.
- **radio**: Botones de opción que permiten al usuario seleccionar una única opción entre varias, generalmente se agrupan por nombre.
- **checkbox**: Casillas de verificación que permiten seleccionar una o más opciones de un conjunto.
- **submit**: Un botón que envía el formulario al servidor.
- **reset**: Un botón que restablece todos los campos del formulario a su valor predeterminado.
- **button**: Un botón personalizable que puede ejecutar una acción mediante JavaScript.

2. **<textarea>**

Este control se utiliza para ingresar texto de varias líneas. Es ideal para campos donde el usuario necesita escribir un texto más largo, como comentarios o descripciones. Se puede configurar el tamaño del área de texto usando los atributos **rows** y **cols**.

3. **<select>**

El control **<select>** crea una lista desplegable desde la cual el usuario puede seleccionar una opción. Este control se utiliza generalmente para

campos con un número limitado de opciones predefinidas. Las opciones dentro del `<select>` se definen usando la etiqueta `<option>`. Además, se puede permitir la selección múltiple utilizando el atributo `multiple`.

Controles avanzados de formulario

1. `<input type="file">`

Este control permite que el usuario cargue archivos desde su dispositivo. Se utiliza comúnmente para permitir que los usuarios suban imágenes, documentos u otros tipos de archivos a través del formulario. A través de este control, se puede especificar un tipo de archivo específico utilizando el atributo `accept`.

2. `<input type="date">`

Este control permite que el usuario seleccione una fecha a través de un calendario, garantizando que la entrada sea en el formato correcto. El navegador mostrará un selector de fecha adecuado para facilitar la elección de una fecha.

3. `<input type="email">`

Este control es específico para la entrada de direcciones de correo electrónico. Los navegadores modernos validan automáticamente que el texto ingresado tenga el formato correcto de un correo electrónico (por ejemplo, `usuario@dominio.com`).

4. `<input type="number">`

Un campo numérico que permite al usuario ingresar solo números. Este tipo de campo puede incluir atributos adicionales como `min`, `max` y `step` para definir un rango de valores permitidos o especificar incrementos.

5. `<input type="tel">`

Este control está diseñado para ingresar números de teléfono. Aunque no realiza una validación estricta como el control de correo electrónico, algunos navegadores pueden mostrar un teclado numérico en dispositivos móviles para facilitar la entrada.

6. `<input type="range">`

Un control deslizante que permite seleccionar un valor dentro de un

rango. Los usuarios pueden arrastrar el control para elegir un valor entre un mínimo y un máximo definidos por los atributos min y max.

Controles de formulario con opciones múltiples

1. **<select> (con atributo multiple)**

Cuando se utiliza el atributo multiple, el control <select> permite seleccionar varias opciones a la vez. El usuario puede hacer clic en varias opciones mientras mantiene presionada la tecla Ctrl (en Windows) o Cmd (en Mac).

2. **<input type="checkbox">**

Las casillas de verificación permiten seleccionar múltiples opciones. Cada casilla funciona de manera independiente, y el formulario puede enviar una lista de opciones seleccionadas cuando se envíe.

3. **<input type="radio">**

Los botones de opción permiten seleccionar una sola opción de un conjunto. Todos los botones de opción que pertenecen al mismo grupo deben compartir el mismo valor para el atributo name, lo que asegura que solo una opción pueda ser seleccionada al mismo tiempo.

Controles de formulario con validación

1. **required**

Este atributo se usa en varios controles para indicar que un campo debe ser completado antes de enviar el formulario. Si el campo está vacío, el navegador mostrará un mensaje de advertencia y evitará el envío del formulario.

2. **pattern**

El atributo pattern permite especificar una expresión regular que los datos del campo deben cumplir para ser enviados. Por ejemplo, puede utilizarse para asegurar que un campo de texto contenga solo números o que un campo de correo electrónico tenga el formato adecuado.

3. min y max

Los atributos min y max se utilizan en los controles de tipo numérico o de fecha para definir el valor mínimo y máximo aceptable. Estos atributos son útiles para restringir las entradas a un rango específico de valores.

4. step

El atributo step especifica el incremento entre valores permitidos en un campo numérico o de fecha. Por ejemplo, si se establece un paso de 5, solo se permitirán valores que sean múltiplos de 5.

Ejemplo

```
<form action="procesar_formulario.php" method="post">
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre"
required />

  <label for="email">Correo electrónico:</label>
  <input type="email" id="email" name="email" required
/>

  <label for="comentarios">Comentarios:</label>
  <textarea id="comentarios" name="comentarios"
rows="4" cols="50"></textarea>

  <label for="suscripcion">¿Deseas suscribirte a
nuestro boletín?</label>
  <input type="checkbox" id="suscripcion"
name="suscripcion" value="si" />

  <label for="opciones">Selecciona una opción:</label>
  <select id="opciones" name="opciones">
    <option value="opcion1">Opción 1</option>
    <option value="opcion2">Opción 2</option>
    <option value="opcion3">Opción 3</option>
```

```
</select>

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

6.2. FORMULARIOS Y EVENTOS

Los formularios son elementos esenciales en la creación de sitios web interactivos, ya que permiten a los usuarios enviar información al servidor o realizar acciones específicas. Sin embargo, un formulario por sí solo no es suficiente para ofrecer una experiencia de usuario dinámica y eficiente. Los **eventos** juegan un papel clave en la gestión de interacciones dentro de un formulario, permitiendo que el comportamiento del formulario responda de manera automática a las acciones del usuario, como la introducción de datos, la validación o el envío de la información.

Eventos en formularios

Un evento es una acción que ocurre en la página web como resultado de la interacción del usuario con un formulario o sus elementos. Estos eventos pueden ser capturados y gestionados mediante JavaScript para proporcionar una retroalimentación inmediata o realizar validaciones antes de que se envíe el formulario. Algunos de los eventos más comunes en los formularios incluyen:

1. **submit**

El evento submit se activa cuando el usuario intenta enviar un formulario. Es útil para realizar validaciones antes de que los datos se envíen al servidor, para asegurarse de que la información introducida es correcta. En muchos casos, este evento puede prevenir el envío del formulario si los datos no cumplen con los requisitos.


```
document.getElementById("miFormulario").addEventListener("submit", function(event) {  
    if (!confirm("¿Estás seguro de que deseas enviar el formulario?")) {  
        event.preventDefault();  
    }  
});
```

2. **change**

El evento **change** se activa cuando un valor de un campo de formulario cambia. Este evento es útil cuando se necesita realizar alguna acción al modificar los datos en un campo, como actualizar un resumen o realizar una validación.

```
document.getElementById("miCampo").addEventListener("change", function() {  
    alert("¡Has cambiado el valor!");  
});
```

3. **focus y blur**

El evento **focus** ocurre cuando un campo de formulario recibe el enfoque, es decir, cuando el usuario hace clic o usa la tecla de tabulación para seleccionar un campo. El evento **blur**, por su parte, se activa cuando el campo pierde el enfoque. Ambos eventos se utilizan para realizar acciones como la validación en tiempo real o resaltar los campos con errores.

```
document.getElementById("miCampo").addEventListener("focus", function() {  
    console.log("Campo en foco");  
});  
  
document.getElementById("miCampo").addEventListener("blur", function() {  
    console.log("Campo fuera de foco");  
});
```

4. **keydown, keyup y keypress**

Los eventos de teclado como `keydown`, `keyup` y `keypress` se activan cuando el usuario presiona o suelta una tecla mientras está interactuando con un campo del formulario. Estos eventos pueden ser útiles para realizar acciones inmediatas basadas en la entrada del teclado, como la validación de caracteres en un campo de texto.

```
document.getElementById("miCampoTexto").addEventListener("keydown", function(event) {  
    if (event.key === "Enter") {  
        console.log("Has presionado Enter");  
    }  
});
```

5. **input**

El evento `input` se activa cada vez que el valor de un campo de texto cambia, lo que lo hace más preciso que el evento `change`, ya que este último solo se activa cuando el campo pierde el foco. El evento `input` es ideal para realizar validaciones en tiempo real o para realizar acciones dinámicas mientras el usuario introduce datos.

```
document.getElementById("miCampo").addEventListener("input", function() {  
    console.log("El valor ha cambiado");  
});
```

6.3. EVENTOS COMUNES CON FORMULARIOS Y SUS CONTROLES

1. submit con validación

El evento submit es fundamental para la validación de formularios. A través de JavaScript, podemos capturar este evento para asegurarnos de que los datos sean correctos antes de que el formulario se envíe al servidor. Si alguna validación falla, podemos usar `preventDefault()` para evitar que el formulario se envíe.

```
document.getElementById("miFormulario").addEventListener("submit", function(event) {  
    const nombre =  
document.getElementById("nombre").value;  
    if (nombre === "") {  
        alert("El campo nombre es obligatorio.");  
        event.preventDefault();  
    }  
});
```

2. Eventos de entrada y salida de datos

Los eventos focus y blur se pueden usar para mejorar la experiencia del usuario, por ejemplo, agregando un mensaje que le indique al usuario que debe completar un campo o resaltar los campos cuando reciben o pierden el enfoque.

```
document.getElementById("nombre").addEventListener("focus", function() {  
    this.style.backgroundColor = "#f0f0f0";  
});  
  
document.getElementById("nombre").addEventListener("blur", function() {  
    this.style.backgroundColor = "#fff";  
});
```

3. Eventos para manejo de múltiples opciones

En campos como los de tipo checkbox y radio, los eventos change permiten detectar cuándo se ha seleccionado o desmarcado una opción. Es útil cuando se necesita realizar acciones dependiendo de las elecciones del usuario.

```
document.getElementById("suscripcion").addEventListener("change", function() {  
    if (this.checked) {  
        alert("Te has suscrito al boletín");  
    } else {  
        alert("Has cancelado la suscripción");  
    }  
});
```

4. Eventos de teclado para control de entradas

Los eventos keydown, keyup o keypress se utilizan en formularios que requieren una entrada rápida o restricción de caracteres. Por ejemplo, podemos limitar el número de caracteres permitidos o prevenir que el usuario ingrese caracteres no válidos.

```
document.getElementById("telefono").addEventListener("keydown", function(event) {  
    if (event.key === "e" || event.key === "E") {  
        event.preventDefault();  
    }  
});
```

Prevención del comportamiento por defecto

En algunos casos, podemos querer evitar el comportamiento predeterminado del formulario, como el envío al servidor o la recarga de la página. Para ello, podemos usar `event.preventDefault()` dentro de cualquier manejador de eventos. Esto es especialmente útil cuando estamos realizando validaciones o manipulaciones personalizadas antes de procesar los datos.

```
document.getElementById("miFormulario").addEventListener("submit", function(event) {  
    // Realizar validaciones  
    if (formularioValido) {  
        // Evitar que el formulario se envíe si todo está  
        bien  
    } else {  
        event.preventDefault();  
    }  
});
```

UNIDAD 3. Técnicas de accesibilidad y usabilidad

ACCESIBILIDAD WEB

Los formularios son elementos clave para la interacción entre los usuarios y los servidores en una página web. Permiten a los usuarios ingresar y enviar datos, como nombres, direcciones, o cualquier otro tipo de información. Existen varios atributos que se pueden aplicar a los elementos de un formulario para mejorar su funcionalidad y personalización.

1. **name:** Este atributo es fundamental para identificar el campo dentro del formulario. Cada campo que se envía al servidor necesita un name único para que los datos se puedan procesar correctamente. Ejemplos comunes de su uso incluyen `<input name="nombre">` o `<textarea name="comentarios">`.
2. **value:** Define el valor por defecto de un campo en el formulario. Este atributo es importante para elementos como los campos de texto, botones de opción (radio) o casillas de verificación (checkbox). En un campo de tipo "radio", por ejemplo, el atributo `value="opcion1"` determina el valor que se enviará cuando el usuario seleccione esa opción.
3. **placeholder:** Es un texto informativo que se muestra dentro de un campo de entrada antes de que el usuario comience a escribir. Sirve como una pista o sugerencia sobre el tipo de información que se espera. Por ejemplo, `<input placeholder="Introduce tu nombre">`.
4. **required:** Este atributo marca un campo como obligatorio. El formulario no podrá ser enviado hasta que todos los campos requeridos estén correctamente completados. Es una manera sencilla de garantizar que el usuario proporcione información clave.
5. **disabled:** Desactiva un campo, de modo que no sea accesible ni modificable por el usuario. Es útil cuando se necesita mostrar un dato de solo lectura o desactivar temporalmente un campo.
6. **readonly:** Similar al atributo disabled, pero en este caso el campo sigue siendo visible y se puede copiar, aunque no se puede modificar. Esto es útil cuando se quiere mostrar información sin permitir cambios en ella.

7. **maxlength**: Establece el número máximo de caracteres que se pueden ingresar en un campo de texto. Por ejemplo, si se desea que un campo acepte solo 10 caracteres, se utilizaría `maxlength="10"`.
8. **pattern**: Permite establecer una expresión regular para validar el formato de la entrada. Es ideal para campos como correos electrónicos o números de teléfono, ya que asegura que el usuario ingrese la información en el formato correcto.
9. **autofocus**: Este atributo hace que el campo reciba automáticamente el foco cuando se carga la página. Es útil cuando se quiere facilitar la experiencia del usuario al permitirle comenzar a escribir de inmediato en el campo deseado.
10. **accept**: En campos de tipo file, este atributo define los tipos de archivos que el usuario puede cargar. Por ejemplo, al usar `accept="image/png,image/jpeg"`, se restringe la carga de archivos solo a imágenes en esos formatos.

Niveles de accesibilidad

Para definir la accesibilidad de una página web, el W3C establece tres niveles de prioridad, que deben ser considerados por los desarrolladores para garantizar que todos los usuarios, independientemente de sus capacidades, puedan acceder a la información de forma equitativa:

1. Prioridad 1

Estos puntos son de cumplimiento obligatorio. Si no se cumplen, los usuarios con ciertas discapacidades no podrán acceder al contenido de la web. Es esencial cumplir con estos aspectos para asegurar que los elementos más críticos estén accesibles para todos.

2. Prioridad 2

Estos puntos deben ser atendidos, ya que, de no cumplirse, algunos usuarios pueden encontrar dificultades al navegar en el sitio web. Aunque no son tan

fundamentales como los de Prioridad 1, siguen siendo importantes para mejorar la experiencia de accesibilidad.

3. Prioridad 3

Los puntos de prioridad 3 son aquellos que, si no se cumplen, pueden causar dificultades menores para algunos usuarios, pero no afectan gravemente la accesibilidad general de la web.

Niveles de conformidad

De acuerdo con la compatibilidad de estos puntos, se establecen tres niveles de conformidad, que van desde el nivel de conformidad **A**, hasta el nivel **AAA**:

- **Nivel A:** Se cumple con todos los puntos de la prioridad 1.
- **Nivel AA:** Se cumplen los puntos de las prioridades 1 y 2.
- **Nivel AAA:** Se cumplen todos los puntos de las tres prioridades.

Este esquema permite a los desarrolladores comprender la accesibilidad de su sitio web y cómo mejorarlo en función de las prioridades de cada punto.

Verificación de la accesibilidad

El W3C ofrece una serie de puntos de verificación para cada uno de los niveles de conformidad, con el objetivo de garantizar que las páginas web sean accesibles para todos los usuarios. A continuación, se detallan los aspectos clave de cada nivel de conformidad:

Nivel de conformidad a

- **Texto alternativo:** Todos los elementos no textuales deben tener un texto equivalente.
- **Colores:** Asegurarse de que la información transmitida mediante el color sea accesible sin depender solo de él.
- **Lenguaje claro:** Utilizar un lenguaje sencillo y comprensible.
- **Accesibilidad sin hojas de estilo:** Verificar que el contenido sea accesible sin el uso de hojas de estilo.

- **Contenido multimedia:** Proporcionar descripciones alternativas como subtítulos para vídeos.

Nivel de conformidad aa

Para asegurar que se cumple con este nivel de conformidad, además de los requisitos previos, es necesario verificar los siguientes aspectos:

- **Comprobaciones generales:**
 - Asegúrese de que las combinaciones de colores de fondo y primer plano tengan un contraste suficiente para garantizar la legibilidad.
 - Verifique que se utilicen marcadores adecuados en lugar de imágenes para facilitar la comprensión.
 - Emplee hojas de estilo (CSS) para la presentación y maquetación del contenido.
 - Utilice unidades relativas en lugar de absolutas para las propiedades dimensionales, tanto en el texto como en otros elementos.
 - Use los encabezados HTML (<h1> ... <h6>) para organizar y estructurar el contenido de manera lógica.
 - Las listas y los ítems deben estar marcados correctamente.
 - Si hay citas, asegúrese de que estén correctamente etiquetadas.
 - Si existen contenidos dinámicos, confirme que estos sean accesibles o, si no lo son, proporcione una alternativa accesible.
- **Comprobaciones adicionales:**
 - Asegúrese de que la página no se actualice automáticamente sin previo aviso.
 - Evite las redirecciones automáticas hacia otros sitios web sin el consentimiento del usuario.
 - No cambie de ventana o página sin notificar previamente al usuario.
 - Proporcione un mapa del sitio o una tabla de contenido para facilitar la navegación.

- Garantice que la navegación en el sitio sea coherente en todas las páginas.
 - Incluya metadatos que aporten información semántica a las páginas y al sitio en general.
- **Comprobaciones específicas para elementos como tablas, marcos y formularios:**
 - Si se utilizan tablas, asegúrese de que no se empleen para la maquetación del sitio, sino solo para la representación de datos tabulares.
 - Si se utilizan marcos, verifique que se describa adecuadamente su función y cómo se interrelacionan entre sí.
 - En el caso de los formularios, compruebe lo siguiente:
 - Cada elemento del formulario debe estar vinculado de manera explícita con su etiqueta correspondiente.
 - Asegúrese de que todos los campos y elementos del formulario tengan una asociación clara con su etiqueta.
- **Comprobaciones de scripts:**
 - Verifique que los manejadores de eventos sean independientes del dispositivo de entrada.
 - Utilice manejadores de eventos basados en lógica en lugar de depender exclusivamente de dispositivos específicos de entrada.

Nivel de conformidad aaa

- **Atajos de teclado:** Proporcionar atajos de teclado en los vínculos más importantes.
- **Explicación de abreviaturas:** Incluir la explicación de las abreviaturas cuando se mencionen por primera vez.
- **Mecanismos de búsqueda:** Proporcionar mecanismos de búsqueda eficientes en el sitio web.
- **Imágenes y vínculos:** Asegurar que todas las imágenes tengan una descripción textual y que los vínculos sean fácilmente comprensibles.

1.1. ESTÁNDARES DE ACCESIBILIDAD

La accesibilidad web se refiere a la capacidad de un sitio para ser utilizado por todas las personas, sin importar el tipo de tecnología que empleen, como hardware, software, idioma o ubicación. Además, se enfoca en permitir que usuarios con diferentes capacidades, incluidas aquellas con discapacidades, puedan acceder y navegar por el contenido de manera efectiva.

El principal objetivo de la accesibilidad web es garantizar que todos los usuarios, sin distinción, tengan igualdad de oportunidades para acceder a la información. Con este fin, varios gobiernos e instituciones alrededor del mundo han comenzado a implementar leyes y normativas que promuevan la accesibilidad en los sitios web.

A nivel global, el organismo líder en la creación de pautas para la accesibilidad web es el W3C (World Wide Web Consortium), a través de su grupo de trabajo WAI (Web Accessibility Initiative). Este grupo establece directrices y recursos para ayudar a los desarrolladores a crear sitios web accesibles para todos los usuarios, independientemente de sus habilidades o dispositivos.

La accesibilidad web se refiere a la capacidad de un sitio para permitir que todas las personas, sin importar el tipo de hardware, software, idioma, localización o capacidades individuales, puedan acceder a su contenido. Esto busca garantizar que cualquier persona, independientemente de sus circunstancias, pueda disfrutar de la misma experiencia en línea.

El principal objetivo de la accesibilidad es asegurar que todos los usuarios tengan las mismas oportunidades de acceder y utilizar los contenidos, servicios y herramientas en la web. Para promover esta igualdad de acceso, diversos gobiernos e instituciones alrededor del mundo han comenzado a implementar leyes y regulaciones que exigen cumplir con estos estándares.

A nivel global, la organización principal encargada de supervisar y establecer los lineamientos para la accesibilidad web es el W3C (World Wide Web Consortium). Este organismo trabaja a través de un grupo especializado conocido como la Iniciativa de Accesibilidad en la Web (WAI, por sus siglas en

inglés), cuyo propósito es facilitar la creación de páginas web accesibles para personas con diferentes tipos de discapacidades y necesidades.

Para garantizar que un sitio web sea accesible para todos, la Web Accessibility Initiative (WAI) del W3C establece las *Pautas de Accesibilidad al Contenido en la Web* (WCAG). Estas pautas tienen como objetivo principal guiar a los diseñadores para crear sitios web accesibles, superando barreras y facilitando el acceso a la información. La WCAG se basa en una serie de recomendaciones clave para lograr este objetivo:

1. **Pauta 1:** Ofrecer alternativas equivalentes para contenidos audiovisuales.
2. **Pauta 2:** No depender exclusivamente del color; los textos y gráficos deben ser comprensibles sin necesidad de color.
3. **Pauta 3:** Utilizar hojas de estilo para la presentación del contenido.
4. **Pauta 4:** Identificar el idioma utilizado en la página.
5. **Pauta 5:** Las tablas deben utilizarse solo para mostrar datos, no para la maquetación del contenido.
6. **Pauta 6:** Asegurarse de que el contenido sea accesible independientemente de la tecnología utilizada. Por ejemplo, si se emplean scripts para formatear contenido, debe seguir siendo accesible incluso si los scripts se desactivan.
7. **Pauta 7:** Evitar el movimiento y la actualización automática de la página.
8. **Pauta 8:** Asegurar la accesibilidad directa a las interfaces de usuario de los objetos embebidos.
9. **Pauta 9:** Diseñar de manera independiente del hardware.
10. **Pauta 10:** Ofrecer siempre alternativas en caso de fallo, como para navegadores antiguos.
11. **Pauta 11:** Usar las tecnologías y pautas recomendadas por el W3C.
12. **Pauta 12:** Proporcionar información de contexto y orientación.
13. **Pauta 13:** Incluir mecanismos de navegación adecuados, como mapas del sitio y barras de navegación.
14. **Pauta 14:** Usar un lenguaje claro y sencillo en los documentos.

Aunque estas pautas puedan parecer difíciles de implementar inicialmente, diseñarlas desde el principio y seguir las recomendaciones adecuadas puede hacer que el proceso sea más sencillo. Cumplir con estas pautas no solo hace que un sitio sea más accesible, sino que también ofrece ventajas significativas para los propietarios de los sitios web.

Entre las principales ventajas de garantizar la accesibilidad se encuentran:

- **Mayor satisfacción del usuario:** Una web accesible, generalmente bien estructurada y adaptable a diferentes dispositivos, mejora la comodidad del usuario, lo que aumenta su confianza.
- **Mayor número de visitantes:** Aproximadamente el 10% de la población mundial tiene algún tipo de discapacidad, por lo que un sitio accesible puede atraer a una audiencia mayor, lo que representa una ventaja competitiva.
- **Reducción de errores:** Al tener una web bien organizada y fácil de navegar, los usuarios cometen menos errores, lo que reduce su estrés y aumenta la fidelidad hacia el sitio.
- **Mejor posicionamiento en los motores de búsqueda:** Las páginas accesibles suelen estar mejor estructuradas, con más texto que imágenes, lo que facilita su rastreo por los motores de búsqueda y mejora su clasificación.

NORMATIVA EUROPEA Y ESPAÑOLA

- La normativa relacionada con la accesibilidad web en Europa y España tiene como objetivo garantizar que los sitios web sean accesibles para todas las personas, independientemente de sus capacidades. Existen diversas leyes y directrices que regulan este aspecto, y su cumplimiento es fundamental para lograr una web inclusiva.
- A nivel europeo, la **Directiva Europea de Accesibilidad Web (2016/2102)** es una de las principales regulaciones. Esta directiva establece que todos los organismos públicos deben asegurarse de que sus sitios web y aplicaciones móviles sean accesibles para las personas

con discapacidad. Además, la directiva exige que estos sitios sean evaluados y que se publiquen informes de accesibilidad.

- En España, la **Ley 34/2002, de Servicios de la Sociedad de la Información y Comercio Electrónico (LSSI-CE)** establece que las páginas web deben ser accesibles para las personas con discapacidad. Esta ley se complementa con el **Real Decreto 1112/2018**, que adapta la normativa europea a la legislación española. El Real Decreto establece los requisitos técnicos específicos que deben cumplir los sitios web y aplicaciones móviles de las entidades públicas y privadas que presten servicios al público.
- Adicionalmente, las **Pautas de Accesibilidad al Contenido Web (WCAG)** de la W3C, adoptadas a nivel internacional, también guían el desarrollo de sitios accesibles, y son la referencia para cumplir con los estándares de accesibilidad tanto en Europa como en España.
- Estas normativas buscan promover la igualdad de acceso a la información y los servicios en línea, permitiendo que todas las personas, sin importar sus capacidades, puedan navegar y beneficiarse de los contenidos disponibles en la web.

HERRAMIENTAS "SOFTWARE" DE VALIDACIÓN DE LA ACCESIBILIDAD

Además de la comprobación manual del nivel de accesibilidad de una página web, existen diversas herramientas que permiten validar el nivel de conformidad del sitio. A continuación, vamos a ver las funcionalidades de TAW, una herramienta utilizada para la validación de la accesibilidad web.

TAW está disponible en dos versiones: en línea y de escritorio. La versión en línea es ideal para realizar un análisis básico de la web de forma rápida y sencilla. Sin embargo, si se requiere un análisis más exhaustivo, se recomienda utilizar la aplicación TAW de escritorio.

Taw online

TAW (herramienta de accesibilidad web) tiene como objetivo verificar el nivel de accesibilidad de una página web, asegurando que sea accesible para todos los usuarios, independientemente de sus características. Para utilizar esta herramienta, basta con acceder a su página [TAW Online](#), donde se debe seleccionar la especificación deseada, introducir la URL de la página a analizar y hacer clic en el botón de análisis.

Tras realizar el análisis, TAW muestra un resumen de los resultados en la parte superior de la página, junto con la localización de los elementos afectados. Los problemas de accesibilidad se indican en colores, con el rojo para los errores de nivel A y el amarillo para los de nivel AA, lo que facilita la identificación de las áreas que requieren atención. Además, TAW clasifica los errores en automáticos y manuales, siendo los automáticos los que deben corregirse de forma prioritaria para cumplir con las pautas de accesibilidad.

Taw versión de escritorio

Si se planea usar TAW de manera frecuente, se recomienda descargar su versión de escritorio, que permite un proceso de validación más detallado y eficiente. Esta versión tiene la ventaja de que puede escanear todo el sitio web de una sola vez, mientras que la herramienta en línea requiere analizar cada página individualmente.

La interfaz de la aplicación es muy simple: solo es necesario ingresar la URL de la página web y seleccionar el tipo de informe que se desea generar.

Existen cuatro tipos de informes disponibles:

- **Resumen:** Ofrece un resumen general de los resultados del análisis para cada nivel de conformidad.

- **Prioridad 1:** Muestra todos los errores detectados para cumplir con el nivel de conformidad A.
- **Prioridad 2:** Muestra todos los errores detectados para cumplir con el nivel de conformidad AA.
- **Prioridad 3:** Muestra todos los errores detectados para cumplir con el nivel de conformidad AAA.

En el informe generado, se incluyen detalles sobre el nivel de prioridad de los errores, así como el índice y la descripción de cada pauta analizada. Además, se especifican los fallos encontrados y la línea del documento donde se localizan.

Tipos de informes generados

Al finalizar el análisis, TAW genera tres tipos de informes:

1. **Informe TAW:** Es el informe más básico y es el que proporciona la herramienta en línea.
2. **Informe EARL:** Este informe contiene los resultados de la evaluación de accesibilidad según las pautas WCAG 1.0 y se presenta en formato XML, a diferencia del informe TAW, que está en formato HTML.
3. **Informe Resumen:** Un informe detallado en formato HTML que se divide en dos secciones. La primera muestra una tabla con los parámetros del análisis y los resultados del análisis en general. La segunda parte ofrece un listado con la cantidad de problemas automáticos y manuales encontrados en cada página web.

Al realizar el análisis con la versión de escritorio, también es posible configurar las pautas a analizar y realizar comprobaciones definidas por el usuario mediante expresiones regulares.

Importante

Con el uso de estas herramientas, se puede obtener un diagnóstico detallado del nivel de accesibilidad de una página web, lo que facilita la tarea de asegurar que el sitio sea accesible para todos los usuarios.

USABILIDAD WEB

La usabilidad se refiere a la facilidad con la que las personas pueden aprender a utilizar una herramienta u objeto, y en el contexto web, se enfoca en la claridad y simplicidad de un sitio para que cualquier usuario pueda navegar por él sin dificultades. Aunque diseñar una web que sea completamente clara y fácil de usar para todos los usuarios es un reto, debido a que cada uno puede tener diferentes perspectivas y necesidades, el diseñador debe procurar hacer que el sitio sea lo más accesible y sencillo posible.

Los factores clave que influyen en la usabilidad de una página web son los siguientes:

- **Facilidad de aprendizaje:** El sitio debe permitir a un usuario nuevo comprender rápidamente cómo navegar y encontrar la información que busca.
- **Eficiencia de uso:** Una vez que el usuario ha aprendido a utilizar el sitio, debe ser capaz de acceder y trabajar con los contenidos de manera ágil.
- **Robustez:** Es importante minimizar los errores que puedan surgir durante la interacción con la web, lo que contribuye a que el usuario se sienta más seguro y fidelizado.
- **Flexibilidad:** El diseño debe ofrecer diversas formas para que el usuario logre su objetivo, adaptándose a sus preferencias y necesidades.

A medida que el uso de internet ha crecido, la usabilidad se ha vuelto cada vez más importante, ya que los usuarios esperan aplicaciones fáciles de usar y de aprender, sin interfaces complicadas o que requieran largos tiempos de familiarización.

Por lo tanto, garantizar una buena usabilidad en un sitio web ofrece numerosos beneficios, entre los que se incluyen:

- **Mejora de la imagen corporativa:** Un sitio bien diseñado que sea fácil de usar aumenta la satisfacción de los usuarios, lo que favorece la percepción de la marca.

- **Reducción de costos:** Una web con buena usabilidad reduce la necesidad de realizar constantes rediseños y ajustes.
- **Reducción del tiempo de aprendizaje:** Los usuarios se familiarizan más rápidamente con el sitio, mejorando su experiencia.
- **Mayor satisfacción del cliente:** Un sitio intuitivo contribuye a que los usuarios disfruten más de la experiencia de navegación.
- **Fidelización de usuarios:** Si los usuarios tienen una experiencia positiva, es más probable que regresen y recomienden el sitio a otros.

HERRAMIENTAS “SOFTWARE” DE MEDICIÓN DE USABILIDAD

Las herramientas de medición de usabilidad son aplicaciones diseñadas para evaluar y mejorar la experiencia del usuario en sitios web, aplicaciones móviles y otros productos digitales. Estas herramientas proporcionan información valiosa sobre cómo los usuarios interactúan con el contenido y las funcionalidades, lo que permite a los diseñadores y desarrolladores hacer ajustes para optimizar la accesibilidad, eficiencia y satisfacción del usuario.



¿Qué son las Herramientas de Medición de Usabilidad?

Las herramientas de medición de usabilidad se utilizan para evaluar la efectividad, facilidad de uso y satisfacción de los usuarios al interactuar con un producto digital. Estas herramientas proporcionan datos cuantitativos y cualitativos que permiten detectar problemas en la interfaz, los flujos de navegación, la comprensión del contenido y la velocidad de realización de tareas.

Algunas de las métricas más comunes que se miden son:

- **Tiempo en tarea:** Cuánto tiempo tarda un usuario en completar una tarea específica.
- **Tasa de éxito:** El porcentaje de usuarios que logran completar una tarea con éxito.
- **Errores:** El número de veces que los usuarios cometen errores durante la interacción.
- **Satisfacción del usuario:** Evaluación subjetiva del usuario sobre su experiencia general.

Tipos de Herramientas de Medición de Usabilidad

1. Herramientas de Pruebas de Usuario

- Permiten observar cómo los usuarios interactúan con el sitio o la aplicación en tiempo real, proporcionando datos valiosos sobre los puntos problemáticos de la interfaz.

Ejemplos:

- **Lookback:** Permite grabar las sesiones de los usuarios mientras navegan por la web, capturando clics, movimientos del ratón y comentarios en vivo.
- **UserTesting:** Proporciona grabaciones y transcripciones de las interacciones de los usuarios, ofreciendo insights sobre su experiencia.

- **Hotjar:** Ofrece grabaciones de sesiones y mapas de calor para ver cómo los usuarios interactúan con los elementos en la página.

2. Herramientas de Análisis de Mapas de Calor

- Muestran visualmente qué áreas de una página web reciben más atención, mediante colores que indican las zonas de mayor interacción, como clics, desplazamientos y toques.

Ejemplos:

- **Crazy Egg:** Ofrece mapas de calor, grabaciones de sesiones y análisis de clics para identificar qué partes de la página son las más efectivas y cuáles necesitan mejoras.
- **Heatmap.me:** Permite visualizar los mapas de calor y ajustar el diseño para mejorar la usabilidad y la experiencia del usuario.

3. Herramientas de Encuestas y Retroalimentación

- Recogen datos cualitativos y cuantitativos sobre la satisfacción de los usuarios mediante encuestas o formularios dentro del sitio o la aplicación.

Ejemplos:

- **Qualaroo:** Permite colocar encuestas interactivas dentro de un sitio web para recopilar comentarios en tiempo real de los usuarios.
- **Survicate:** Ofrece encuestas que pueden integrarse dentro de las aplicaciones web o móviles para recibir retroalimentación instantánea.

4. Herramientas de Análisis de la Velocidad de Carga y Rendimiento

- Miden el tiempo de carga de las páginas web, ya que la velocidad de un sitio impacta directamente en la experiencia del usuario.

Ejemplos:

- **Google PageSpeed Insights:** Analiza el rendimiento de las páginas web y ofrece recomendaciones para mejorar la velocidad de carga.

- **Pingdom:** Proporciona un análisis detallado de la velocidad de carga de las páginas, identificando los cuellos de botella en el rendimiento.

5. Herramientas de Test A/B y Multivariantes

- Permiten realizar pruebas A/B y multivariantes para evaluar qué versión de una página web o elemento de la interfaz proporciona mejores resultados en términos de usabilidad y conversión.

Ejemplos:

- **Optimizely:** Proporciona una plataforma para realizar pruebas A/B en tiempo real y obtener resultados sobre el comportamiento de los usuarios.
- **VWO (Visual Website Optimizer):** Ofrece pruebas A/B, pruebas multivariantes y análisis de embudos de conversión para mejorar la experiencia del usuario.

Ventajas de Utilizar Herramientas de Medición de Usabilidad

1. Mejora de la Experiencia del Usuario:

- Identificar puntos de fricción en la interacción permite hacer ajustes para mejorar la satisfacción del usuario y reducir la tasa de rebote.

2. Optimización de la Conversión:

- Al comprender cómo los usuarios navegan por el sitio, es posible realizar cambios para mejorar la tasa de conversión y alcanzar los objetivos del negocio.

3. Reducción de Errores y Problemas de Navegación:

- Al medir cómo los usuarios interactúan con el sistema, es más fácil detectar y corregir errores de navegación, mejorando la accesibilidad y la usabilidad.

4. Decisiones Basadas en Datos:

- Estas herramientas proporcionan datos concretos sobre el comportamiento del usuario, lo que permite tomar decisiones informadas y basadas en hechos en lugar de suposiciones.

5. Incremento en la Retención de Usuarios:

- Mejorar la usabilidad a través de datos medibles puede llevar a una mayor satisfacción y, en consecuencia, a una mayor retención de usuarios.

UNIDAD 4. Herramientas de edición web y optimización de páginas web

FUNCIONES Y CARACTERÍSTICAS DE HERRAMIENTAS DE DESARROLLO WEB

Para desarrollar páginas web, se emplean diversas herramientas que simplifican el proceso de creación. No es indispensable limitarse a editores de HTML, ya que las páginas web suelen integrar elementos como imágenes, sonidos y otros recursos multimedia.

Estas herramientas pueden clasificarse según su tipo de licencia. Existen tanto herramientas comerciales como opciones de software libre. Aunque las herramientas comerciales suelen ofrecer funcionalidades avanzadas, en la actualidad el software libre ha evolucionado hasta ofrecer capacidades destacadas, que permiten desarrollar sitios web de manera eficiente y profesional.

1.1. Características de las herramientas de desarrollo web

Las herramientas de desarrollo web se caracterizan por ofrecer un conjunto de funcionalidades diseñadas para facilitar la creación y mantenimiento de sitios web. Estas herramientas varían según su propósito específico, pero comparten ciertas características fundamentales que las hacen indispensables para desarrolladores y diseñadores.

1. **Interfaz Intuitiva**

La mayoría de estas herramientas están diseñadas con interfaces gráficas fáciles de usar que simplifican tareas complejas, permitiendo que tanto principiantes como expertos puedan utilizarlas con eficacia.

2. **Compatibilidad Multiplataforma**

Muchas herramientas son compatibles con diferentes sistemas operativos y navegadores, asegurando que los proyectos puedan desarrollarse y visualizarse correctamente en distintos entornos.

3. **Soporte para Tecnologías Modernas**

Incluyen compatibilidad con lenguajes y estándares actuales, como HTML5, CSS3, JavaScript, y frameworks populares como React,

Angular o Vue.js, lo que permite la creación de sitios web dinámicos y adaptables.

4. Integración con Otros Recursos

Estas herramientas facilitan la incorporación de recursos multimedia, como imágenes, videos y sonidos, así como la integración con bases de datos y servicios de terceros, para enriquecer las funcionalidades del sitio.

5. Edición en Tiempo Real

Ofrecen opciones de vista previa en tiempo real, lo que permite a los desarrolladores ver los cambios inmediatamente mientras trabajan en el código o diseño de la página.

6. Optimización para el Rendimiento

Incorporan funciones para optimizar el rendimiento de los sitios web, como la compresión de imágenes, la minificación de código y herramientas para analizar la velocidad de carga.

7. Seguridad

Proporcionan herramientas y prácticas para reforzar la seguridad de los sitios web, como la gestión de certificados SSL, el análisis de vulnerabilidades y la implementación de sistemas de autenticación.

8. Flexibilidad en Licencias

Las herramientas de desarrollo web pueden ser comerciales, ofreciendo soporte técnico y características premium, o basadas en software libre, con comunidades activas que contribuyen a su mejora continua.

CONFIGURACIÓN DE MÓDULOS Y MENÚS

2.1. Configuración de módulos

La configuración de módulos y menús es una tarea esencial en el desarrollo y administración de sitios web, ya que permite estructurar el contenido y organizar la navegación de manera eficiente para los usuarios. A continuación, se destacan los aspectos clave relacionados con esta configuración:

Características principales de los módulos

1. **Funcionalidad Específica:**

Cada módulo está diseñado para cumplir un propósito concreto, como mostrar un formulario de contacto, un calendario, una galería de imágenes, o un cuadro de búsqueda.

2. **Ubicación Personalizable:**

Los módulos pueden ubicarse en diferentes secciones del diseño del sitio, como encabezados, barras laterales, áreas centrales o pies de página, dependiendo de su funcionalidad.

3. **Configuración Dinámica:**

Los módulos suelen contar con opciones de configuración que permiten ajustar su comportamiento y apariencia, como cambiar estilos, definir filtros de contenido o establecer condiciones de visibilidad.

4. **Actualización de Contenido:**

Muchos módulos permiten la integración con bases de datos o sistemas externos para mostrar contenido dinámico, como noticias, productos de una tienda en línea o publicaciones recientes de un blog.

Tipos de módulos más comunes

1. **Módulos de Navegación:**

- Incluyen menús desplegables, breadcrumbs o barras de navegación.
- Ayudan a los usuarios a moverse por el sitio de manera eficiente.

2. **Módulos de Multimedia:**

- Incorporan imágenes, videos o carruseles.
- Enriquecen visualmente el contenido del sitio.

3. **Módulos de Interacción:**

- Ejemplo: formularios de contacto, encuestas o foros.
- Permiten la comunicación y el feedback de los usuarios.

4. **Módulos Informativos:**

- Muestran noticias, anuncios o calendarios.
- Se actualizan frecuentemente para mantener el sitio relevante.

5. **Módulos de Redes Sociales:**

- Integran botones para compartir, feeds en vivo o enlaces a perfiles sociales.
- Ayudan a mejorar la presencia digital y el alcance del contenido.

2.2. Configuración de menús

Los menús son elementos fundamentales en la estructura de un sitio web, ya que organizan y facilitan la navegación para los usuarios. Un menú bien diseñado garantiza que el contenido sea accesible de manera intuitiva y rápida, mejorando la experiencia del visitante.

Características Principales de los Menús

1. Estructura Jerárquica:

Los menús suelen organizarse en niveles, desde enlaces principales hasta submenús desplegables, para reflejar la arquitectura del contenido del sitio.

2. Enlaces Internos y Externos:

Los elementos del menú pueden redirigir a páginas dentro del sitio web o a recursos externos, como documentos o enlaces a otros dominios.

3. Diseño Personalizable:

Se pueden ajustar los estilos, colores, tipografías y comportamientos (como animaciones o transiciones) para integrarlos perfectamente con el diseño del sitio.

4. Compatibilidad Responsive:

Los menús se adaptan automáticamente a diferentes dispositivos, mostrando versiones simplificadas o desplegables en pantallas pequeñas, como teléfonos móviles.

Tipos de menús

1. Menús Horizontales:

- Se colocan en la parte superior de la página.
- Son ideales para destacar las secciones principales del sitio.

2. Menús Verticales:

- Se ubican en barras laterales.
- Son útiles para sitios con una gran cantidad de secciones o categorías.

3. Menús Desplegables o Anidados:

- Permiten organizar enlaces en varios niveles.
- Mejoran la navegación al agrupar contenido relacionado bajo una misma sección.

4. Menús de Hamburguesa:

- Comunes en diseños responsivos.
- Se activan con un ícono y expanden los enlaces en dispositivos móviles o en sitios minimalistas.

5. Menús de Pie de Página:

- Ubicados al final del sitio.
- Incluyen enlaces secundarios como políticas de privacidad, términos de uso o información de contacto.

Configuración de Menús

1. Definición de Enlaces:

- Se establecen los elementos que compondrán el menú, asegurándose de que reflejen las secciones más relevantes y necesarias del sitio.

2. Ordenación y Jerarquía:

- Los enlaces se organizan de forma lógica, priorizando la información más importante y estructurándola en niveles si es necesario.

3. Diseño e Integración:

- Se configuran colores, fuentes, tamaños y estilos para que el menú sea visualmente coherente con el diseño del sitio.

4. Pruebas de Usabilidad:

- El menú se evalúa para garantizar que sea fácil de usar en todos los dispositivos y que los enlaces sean funcionales.

5. Actualización Continua:

- A medida que el sitio crece o cambia, se revisa y ajusta el menú para mantener su relevancia y funcionalidad.

2.3. Integración

La integración entre módulos y menús es un aspecto crucial en el diseño y desarrollo de sitios web, ya que permite una navegación fluida y coherente, conectando elementos funcionales y estructurales del sitio. Este vínculo garantiza que los usuarios puedan interactuar con el contenido de manera intuitiva y que las funcionalidades del sitio estén accesibles de forma organizada.

Importancia de la Integración

- **Coherencia Visual y Funcional:** La combinación de módulos y menús debe mantener un diseño uniforme y una funcionalidad consistente en todo el sitio.
- **Facilita la Navegación:** Los menús proporcionan acceso a los módulos, mientras que los módulos pueden presentar contenido o herramientas vinculadas directamente a los menús.
- **Optimización de Recursos:** Aprovecha al máximo las capacidades de ambos elementos, evitando redundancias y simplificando la administración del sitio.

Formas de Integración

1. Enlaces Dinámicos desde los Menús a los Módulos:

- Los menús pueden configurarse para redirigir a secciones específicas gestionadas por módulos, como galerías de imágenes, formularios de contacto o publicaciones de un blog.
- Ejemplo: Un ítem del menú "Blog" puede enlazar directamente con un módulo que muestra las entradas más recientes.

2. Visibilidad Condicional de Módulos Basada en Menús:

- Algunos módulos pueden configurarse para mostrarse únicamente cuando se selecciona un enlace específico del menú.
- Ejemplo: Un módulo de ofertas especiales puede activarse al seleccionar "Promociones" en el menú principal.

3. Módulos que Generan Menús Dinámicos:

- Existen módulos que automáticamente crean menús en función del contenido disponible.
- Ejemplo: Un módulo de categorías de productos en una tienda en línea puede generar un menú de navegación para cada categoría.

4. Módulos Integrados en Estructuras de Menús:

- Los módulos pueden ser incrustados directamente dentro de los menús, proporcionando funcionalidades adicionales.
- Ejemplo: Un módulo de búsqueda o un carrito de compras se puede integrar como un elemento del menú principal.

5. Interacción con Módulos Multimedia:

- Los menús pueden dirigir a módulos que muestren contenido multimedia específico, como videos, galerías o mapas interactivos.
- Ejemplo: Al seleccionar "Ubicación" en el menú, se carga un módulo con un mapa interactivo de Google Maps.

Ventajas de una Integración Efectiva

- **Mejora la Experiencia del Usuario:** Proporciona una navegación más intuitiva y acceso directo a las funciones importantes del sitio.
- **Simplifica la Administración:** Permite una gestión centralizada y sincronizada de menús y módulos, reduciendo esfuerzos redundantes.
- **Adapta el Sitio a las Necesidades del Usuario:** Ofrece una presentación personalizada y relevante de contenido y funcionalidades.

Pasos para una Integración Eficiente

1. Planificación Inicial:

- Identificar qué módulos necesitan ser accesibles desde los menús y cómo se relacionan con la estructura del sitio.

2. **Diseño Cohesivo:**

- Asegurar que los módulos y menús mantengan un estilo visual unificado, respetando la identidad gráfica del sitio.

3. **Configuración Técnica:**

- Configurar los módulos y menús para que se comuniquen entre sí mediante enlaces, condiciones de visibilidad o integración dinámica.

4. **Pruebas y Ajustes:**

- Evaluar la funcionalidad para garantizar que los usuarios puedan acceder fácilmente a los módulos a través de los menús, y viceversa.

5. **Monitoreo y Actualización:**

- Revisar periódicamente la integración para adaptarla a nuevos contenidos o cambios en el diseño del sitio.
-

EDITORES DE LENGUAJE DE MARCADO

Los editores de lenguaje de marcado son herramientas fundamentales para la creación, edición y mantenimiento de documentos estructurados en lenguajes como HTML, XML o Markdown. Estas herramientas facilitan la escritura del código, ofreciendo funciones específicas que ayudan a optimizar el trabajo de desarrolladores y creadores de contenido.

Características de los editores de lenguaje de marcado

1. **Resaltado de Sintaxis:**

- Los editores destacan visualmente los elementos del código según su función (etiquetas, atributos, valores, etc.), facilitando la comprensión y detección de errores.

2. Autocompletado:

- Ayudan a ahorrar tiempo ofreciendo sugerencias mientras se escribe, como el cierre automático de etiquetas o la inserción de atributos comunes.

3. Compatibilidad con Múltiples Lenguajes:

- Muchos editores soportan no solo HTML o XML, sino también CSS, JavaScript y otros lenguajes relacionados, permitiendo trabajar en un entorno unificado.

4. Previsualización en Tiempo Real:

- Algunos editores permiten ver cómo se verá el documento en un navegador mientras se escribe, proporcionando una retroalimentación inmediata.

5. Validación de Código:

- Integran herramientas que analizan el código para identificar errores de sintaxis o incumplimientos de estándares establecidos.

6. Integración con Otras Herramientas:

- Facilitan la conexión con servidores, repositorios de código o frameworks, agilizando el flujo de trabajo.

7. Personalización:

- Permiten modificar el entorno de trabajo, como cambiar temas de colores, atajos de teclado o añadir extensiones para funciones adicionales.

Tipos de Editores de Lenguaje de Marcado

1. Editores de Texto Plano:

- Herramientas básicas como Notepad o Vim, que no incluyen funciones avanzadas, pero son útiles para proyectos simples.

2. Editores Específicos para Marcado:

- Diseñados específicamente para trabajar con lenguajes de marcado, como Sublime Text, Visual Studio Code o Brackets.
- Incluyen funciones avanzadas como extensiones y soporte para varios lenguajes.

3. Editores WYSIWYG:

- “What You See Is What You Get” (Lo que ves es lo que obtienes). Estos editores permiten crear contenido sin escribir código directamente, generándolo automáticamente en segundo plano. Ejemplos: Adobe Dreamweaver o Microsoft FrontPage.

4. Editores Basados en Navegadores:

- Herramientas en línea como CodePen o JSFiddle, ideales para probar y compartir código rápidamente sin necesidad de instalar software.

Ventajas de Utilizar Editores de Lenguaje de Marcado

1. Eficiencia en el Trabajo:

- Automatizan tareas repetitivas, como cerrar etiquetas o formatear el código, reduciendo errores y tiempos de desarrollo.

2. Accesibilidad a Herramientas Avanzadas:

- Ofrecen funciones que no están disponibles en simples editores de texto, como la depuración integrada o la gestión de proyectos.

3. Facilitan el Cumplimiento de Estándares:

- Ayudan a mantener un código limpio y compatible con los estándares internacionales, mejorando la calidad del proyecto.

4. Compatibilidad Multiplataforma:

- Muchos editores están disponibles para diferentes sistemas operativos o en versiones web, permitiendo trabajar desde cualquier lugar.

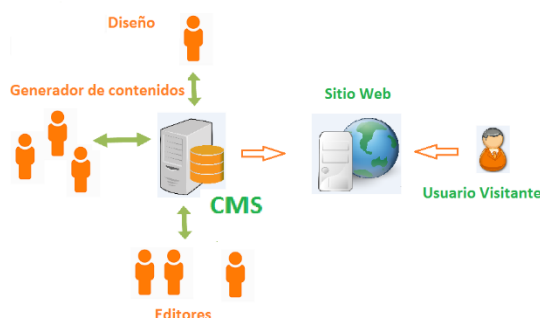
Elección del Editor Adecuado

La elección de un editor dependerá del nivel de experiencia del usuario, las necesidades específicas del proyecto y el flujo de trabajo preferido. Por ejemplo:

- Para principiantes, un editor WYSIWYG puede ser ideal para aprender.
- Para desarrolladores avanzados, herramientas como Visual Studio Code o Atom ofrecen mayor flexibilidad y personalización.

INSTALACIÓN HERRAMIENTAS DE PUBLICACIÓN

La instalación de herramientas de publicación es un paso fundamental para gestionar y poner en línea contenidos en un sitio web. Estas herramientas permiten crear, editar y actualizar contenido de forma eficiente, facilitando la administración del proyecto web. Su correcta instalación garantiza el funcionamiento óptimo del sistema y la disponibilidad de las funcionalidades necesarias para la publicación.



Tipos de herramientas de publicación

1. Sistemas de Gestión de Contenidos (CMS):

- Plataformas como WordPress, Joomla o Drupal que ofrecen soluciones completas para la administración de sitios web.
- Son ideales para usuarios que necesitan publicar y organizar contenido sin profundos conocimientos técnicos.

2. Editores de Publicación Locales:

- Herramientas como Adobe Dreamweaver o Visual Studio Code que permiten desarrollar y publicar sitios web desde un entorno local hacia un servidor remoto.

3. Publicadores Basados en Línea:

- Plataformas como Wix, Squarespace o Blogger que no requieren instalación, ya que funcionan directamente desde el navegador.

4. Herramientas de Línea de Comandos:

- Software como Git, FTP clients o herramientas de despliegue automatizado, utilizadas por desarrolladores avanzados para gestionar proyectos en servidores.

Pasos Generales para la Instalación de Herramientas de Publicación

1. Selección de la Herramienta Adecuada:

- Evaluar las necesidades del proyecto: tipo de sitio web, nivel de personalización requerido y el nivel de experiencia del usuario.

2. Descarga del Software:

- Obtener la herramienta desde el sitio oficial o una fuente confiable para garantizar la seguridad y compatibilidad con el sistema operativo.

3. Requisitos Previos:

- Verificar los requisitos del sistema:
 - **CMS:** Requieren un servidor web (Apache, Nginx), un gestor de bases de datos (MySQL, PostgreSQL) y un lenguaje de scripting (PHP).
 - **Aplicaciones Locales:** Es necesario disponer de espacio de almacenamiento suficiente y compatibilidad con el sistema operativo.
 - **Herramientas Basadas en Línea:** Solo requieren un navegador compatible y conexión a Internet.

4. Instalación:

- **Instalación Local:**
 - Ejecutar el instalador descargado y seguir las instrucciones proporcionadas.
 - Configurar carpetas de destino y ajustes básicos durante el proceso.
- **Instalación en el Servidor (CMS):**
 - Subir los archivos de instalación al servidor mediante FTP o herramientas similares.
 - Configurar la base de datos y realizar la instalación inicial desde el navegador.

5. Configuración Inicial:

- Personalizar la herramienta con configuraciones específicas del proyecto, como ajustes de idioma, plantillas o extensiones.

- Establecer permisos de usuario, configuraciones de seguridad y preferencias de publicación.

6. Pruebas:

- Verificar que la herramienta funcione correctamente, revisando la navegación, publicación y visualización de contenido.

7. Mantenimiento:

- Mantener el software actualizado para garantizar la compatibilidad y seguridad.

SISTEMAS GESTORES DE CONTENIDO (CMS)

Los **Gestores de Contenido** o **Content Management Systems (CMS)** son herramientas diseñadas para crear, gestionar y publicar contenido digital de manera sencilla. Estas plataformas permiten a los usuarios, incluso sin conocimientos técnicos avanzados, administrar sitios web, blogs, tiendas en línea y portales corporativos con eficiencia y flexibilidad.

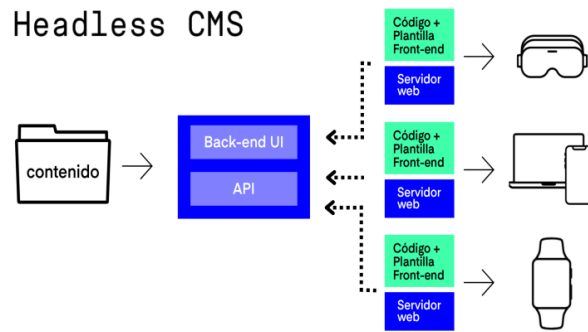
¿Qué es un CMS?

Un CMS es un software que proporciona una interfaz amigable para la creación y organización de contenido. Funciona como un sistema de administración donde los usuarios pueden:

- Diseñar la estructura de un sitio web.
- Añadir, editar o eliminar texto, imágenes y otros elementos multimedia.
- Gestionar usuarios, permisos y roles.
- Optimizar el contenido para buscadores (SEO).

Un CMS divide su funcionamiento en dos partes principales:

1. **Front-end:** La parte visible del sitio web, que interactúa con los visitantes.
2. **Back-end:** La interfaz de administración donde los usuarios gestionan el contenido.



Características principales de un CMS

1. Gestión de Contenido:

- Edición de texto e imágenes mediante editores visuales WYSIWYG.
- Organización del contenido en páginas, categorías o etiquetas.

2. Personalización:

- Posibilidad de usar temas y plantillas para definir la apariencia del sitio.
- Configuración de menús, encabezados y pies de página.

3. Extensibilidad:

- Uso de plugins, módulos o extensiones para añadir nuevas funcionalidades como formularios, galerías o comercio electrónico.

4. Control de Acceso:

- Asignación de roles y permisos para diferentes usuarios (administradores, editores, autores).

5. Optimización SEO:

- Herramientas integradas para mejorar la visibilidad en buscadores mediante la personalización de URLs, metadatos y etiquetas.

6. Multimedia:

- Soporte para la integración de imágenes, videos, audios y otros tipos de contenido interactivo.

7. Multilingüismo:

- Facilidades para crear contenido en varios idiomas, adaptando el sitio a diferentes audiencias.

8. Gestión de Bases de Datos:

- Almacenamiento y recuperación eficiente de contenido mediante bases de datos como MySQL o PostgreSQL.

Tipos de cms

1. CMS Generalistas:

- Diseñados para una amplia gama de usos, como blogs, sitios corporativos o portafolios. Ejemplos:
 - **WordPress:** Versátil, con una gran comunidad y extensiones.
 - **Joomla:** Ideal para sitios más complejos con múltiples usuarios.
 - **Drupal:** Potente y altamente personalizable.

2. CMS Especializados:

- Enfocados en necesidades específicas, como:
 - **eCommerce:** Magento, Shopify, WooCommerce.
 - **Educación:** Moodle.
 - **Gestión de contenido multimedia:** TYPO3.

3. CMS de Código Abierto:

- Son gratuitos y ofrecen acceso al código fuente, permitiendo personalizaciones avanzadas. Ejemplo: WordPress, Drupal, Joomla.

4. CMS Comerciales:

- Licenciados y con soporte técnico dedicado. Ejemplo: Adobe Experience Manager, Sitecore.

VENTAJAS DE USAR UN CMS
<ul style="list-style-type: none">• Facilidad de Uso: Permite a usuarios no técnicos gestionar contenido sin necesidad de escribir código.
<ul style="list-style-type: none">• Rapidez en el Desarrollo: Reduce el tiempo necesario para crear un sitio funcional.

- | |
|---|
| <ul style="list-style-type: none">• Flexibilidad y Escalabilidad: Puede adaptarse a proyectos pequeños o grandes, y crecer junto con el sitio. |
| <ul style="list-style-type: none">• Comunidad y Recursos: Muchos CMS tienen una comunidad activa que ofrece soporte, plugins y plantillas. |

DESVENTAJAS DE USAR UN CMS
<ul style="list-style-type: none">• Curva de Aprendizaje: Aunque accesible, algunas plataformas avanzadas requieren tiempo para dominarse.
<ul style="list-style-type: none">• Mantenimiento: Requieren actualizaciones regulares para garantizar seguridad y rendimiento.
<ul style="list-style-type: none">• Limitaciones de Personalización: En algunos casos, las funciones avanzadas pueden requerir conocimientos técnicos o desarrollo adicional.

NORMATIVA EUROPEA Y ESPAÑOLA SOBRE PROTECCIÓN DE DATOS Y LSSI-CE

El Reglamento General de Protección de Datos (RGPD), aprobado por la Unión Europea, es la normativa marco en materia de protección de datos personales. Su objetivo principal es proteger los derechos fundamentales de los ciudadanos europeos y garantizar la seguridad y privacidad de sus datos personales. Entró en vigor el 25 de mayo de 2018 y se aplica a todas las empresas y organizaciones que gestionen datos personales de residentes en la UE, independientemente de dónde se encuentren.

6.1. LSSI-CE (LEY DE SERVICIOS DE LA SOCIEDAD DE LA INFORMACIÓN Y COMERCIO ELECTRÓNICO)

La **Ley de Servicios de la Sociedad de la Información y Comercio Electrónico (LSSI-CE)** es una normativa española que regula los servicios prestados por medios electrónicos y la actividad comercial en internet. Su objetivo principal es regular el uso de internet y otros medios electrónicos para garantizar la protección de los derechos de los usuarios, fomentar la confianza digital y asegurar la seguridad en las transacciones comerciales.

La LSSI-CE establece una serie de obligaciones tanto para los prestadores de servicios como para los usuarios, y su ámbito de aplicación incluye cualquier persona o entidad que ofrezca servicios en línea o realice actividades comerciales a través de internet dentro del territorio español, incluyendo empresas extranjeras que operen en España.

POSICIONAMIENTO ORGÁNICO

El **posicionamiento orgánico** (SEO, por sus siglas en inglés, *Search Engine Optimization*) se refiere a la estrategia para mejorar la visibilidad de un sitio web en los motores de búsqueda de manera natural, es decir, sin recurrir a publicidad pagada. Aunque no está directamente relacionado con la protección de datos personales, la implementación adecuada del SEO debe tener en cuenta los principios de privacidad y protección de los datos que establece el **Reglamento General de Protección de Datos (RGPD)**.

Principales consideraciones sobre el posicionamiento orgánico y la protección de datos:

1. Recopilación y Uso de Datos Personales:

Al aplicar técnicas de SEO, es común recoger datos de los usuarios a través de cookies, formularios de contacto y otras herramientas de análisis web (como Google Analytics). Es esencial que los sitios web cumplan con el RGPD al asegurarse de que:

- Obtienen el **consentimiento explícito** de los usuarios antes de procesar datos personales.
- Incluir una **política de privacidad** clara, que explique qué datos se recogen y cómo se utilizan para fines de SEO, como el análisis del tráfico y la personalización de la experiencia del usuario.

2. **Transparencia y Control sobre los Datos:**

Los usuarios deben tener el **derecho a gestionar** sus preferencias de privacidad. Por ejemplo, pueden aceptar o rechazar las cookies a través de un **banner de cookies**. Este control es esencial no solo para cumplir con la normativa, sino también para mejorar la confianza de los usuarios en el sitio web, lo que puede impactar positivamente en el SEO.

3. **Uso de Datos Anónimos o Pseudonimizados:**

Cuando sea posible, se debe fomentar el uso de datos **anonimizados o pseudonimizados** para analizar el tráfico y el comportamiento del usuario. De esta manera, se pueden cumplir los principios de minimización de datos y garantizar que la información personal no sea utilizada sin el consentimiento adecuado. Esto también ayuda a mitigar los riesgos asociados con el tratamiento de datos personales y mejora la seguridad del sitio web.

4. **Impacto en la Usabilidad y la Experiencia del Usuario (UX):**

El cumplimiento con el RGPD y las normativas de protección de datos mejora la **experiencia del usuario**, un factor importante en las estrategias de SEO. Un sitio web que respeta la privacidad y ofrece transparencia en el manejo de datos genera mayor confianza, lo que puede aumentar el tiempo de permanencia de los usuarios en el sitio y reducir las tasas de rebote, lo cual impacta positivamente en los algoritmos de los motores de búsqueda.

5. **Optimización de Contenidos y Consentimiento:**

En cuanto a la creación de contenido, es crucial que cualquier formulario, encuesta o interacción con el usuario que implique la recopilación de datos personales esté claramente indicado y cuente con un mecanismo de aceptación, de manera que no se vulnere el consentimiento requerido por la ley. Esto también impacta en la **indexación de contenidos**, ya que los motores de búsqueda valoran

positivamente la transparencia y la legalidad en el manejo de los datos de los usuarios.

6. **Riesgo de Penalización por Incumplimiento:**

El incumplimiento del RGPD puede generar **multas severas**, lo que afectaría negativamente al posicionamiento orgánico de un sitio web. Los motores de búsqueda como Google consideran la seguridad y la privacidad como factores clave para determinar la calidad de un sitio web. Por lo tanto, no cumplir con las normativas de protección de datos podría resultar en una disminución de la visibilidad en los resultados de búsqueda.

En el contexto del SEO, el **posicionamiento orgánico** no solo debe centrarse en las estrategias tradicionales de optimización, sino también en la **protección de datos personales**. El cumplimiento con el RGPD es esencial para evitar penalizaciones legales y mejorar la confianza de los usuarios, lo cual, a su vez, impactará de manera positiva en el rendimiento del sitio web en los motores de búsqueda.

TÉCNICAS SEO

El **SEO (Search Engine Optimization)** engloba un conjunto de estrategias para mejorar la visibilidad y el posicionamiento de un sitio web en los motores de búsqueda. Se divide en dos áreas principales: **SEO On-Page** (dentro de la página) y **SEO Off-Page** (fuera de la página).

SEO On-Page

1. **Investigación de Palabras Clave:** Identificar palabras clave relevantes y con buen volumen de búsqueda para atraer tráfico cualificado.
2. **Optimización de Títulos y Meta Descripciones:** Incluir palabras clave y crear descripciones atractivas para mejorar el clic en los resultados.

3. **Optimización de Contenidos:** Crear contenido único y relevante con palabras clave, utilizando encabezados adecuados.
4. **Optimización de Imágenes:** Comprimir imágenes, usar texto alternativo (alt text) y formatos adecuados para mejorar la velocidad y accesibilidad.
5. **Velocidad del Sitio Web:** Mejorar la carga de la página a través de la compresión de archivos, uso de caché y CDN.
6. **Optimización Móvil:** Asegurar que el sitio sea responsivo y fácil de usar en dispositivos móviles.

SEO Off-Page

1. **Link Building:** Obtener enlaces de calidad desde sitios relevantes para aumentar la autoridad.
2. **Marketing en Redes Sociales:** Promocionar contenido en plataformas sociales para generar tráfico y visibilidad.
3. **Influencer Marketing:** Colaborar con influencers para obtener menciones y enlaces a tu sitio.
4. **Participación en Foros y Comunidades:** Generar enlaces y tráfico al responder preguntas o compartir contenido valioso.

URL SEMÁNTICAS

Las **URL semánticas** son aquellas que están estructuradas de manera clara y descriptiva, de forma que reflejan el contenido de la página de manera comprensible tanto para los usuarios como para los motores de búsqueda. A diferencia de las URLs generadas de forma automática o con parámetros numéricos (como `www.ejemplo.com/?id=123`), las URLs semánticas utilizan palabras clave relevantes y legibles para mejorar la experiencia de navegación y optimizar el SEO.

Características de las URLs Semánticas

1. **Descriptivas y Claras:** Las URLs deben reflejar de forma concisa y precisa el contenido de la página. Por ejemplo, una URL como `www.ejemplo.com/blog/como-crear-una-web` es más clara que `www.ejemplo.com/pagina?id=23`.

2. **Uso de Palabras Clave Relevantes:** Incluir **palabras clave** relacionadas con el contenido de la página ayuda tanto a los usuarios como a los motores de búsqueda a entender de qué trata la página. Por ejemplo, www.ejemplo.com/seo/guia-practica es más útil que www.ejemplo.com/seo123.
3. **Simplicidad y Brevedad:** Las URLs semánticas deben ser fáciles de recordar y compartir. Evitar el uso excesivo de caracteres o palabras innecesarias es clave para mantenerlas simples.
4. **Uso de Guiones para Separar Palabras:** En lugar de utilizar guiones bajos (_) o espacios, es recomendable usar guiones (-) para separar las palabras en las URLs, ya que los motores de búsqueda los interpretan correctamente.
5. **Estructura Jerárquica:** Organizar la URL de forma jerárquica refleja la estructura del sitio web y mejora la navegación. Por ejemplo, www.ejemplo.com/cursos/marketing-digital muestra que el curso sobre marketing digital pertenece a una sección de "cursos".
6. **Evitar Caracteres Especiales y Parámetros Complejos:** Evitar el uso de caracteres especiales como &, ?, o parámetros complejos en la URL mejora tanto la legibilidad como el SEO.

Beneficios de las URLs Semánticas

1. **Mejora del SEO:** Las URLs claras y descriptivas facilitan la indexación del contenido por parte de los motores de búsqueda, lo que puede mejorar el **posicionamiento** en los resultados orgánicos.
2. **Mejor Experiencia del Usuario:** Los usuarios pueden identificar fácilmente el contenido de la página a través de la URL, lo que mejora la **usabilidad** y facilita la navegación.
3. **Mayor Facilidad de Compartir:** Las URLs semánticas son más fáciles de recordar y compartir en redes sociales, correos electrónicos y otros canales.
4. **Reducción de Errores de Navegación:** Al ser más legibles, las URLs semánticas permiten una mejor organización y menor confusión a la hora de navegar por el sitio web.

Ejemplo:

Incorrecta (no semántica):

- www.ejemplo.com/?id=12345

Correcta (semántica):

- www.ejemplo.com/como-crear-una-web

En el segundo caso, la URL es más clara, ya que contiene las palabras clave relevantes y proporciona una mejor idea del contenido de la página.

PRUEBAS DE VALIDACIÓN

Las **pruebas de validación** son un conjunto de procesos y técnicas que se utilizan para verificar si un sistema, producto o servicio cumple con los requisitos y expectativas del usuario final o con las especificaciones establecidas. En el contexto del desarrollo de software y páginas web, las pruebas de validación son esenciales para garantizar que el producto final funcione correctamente, sea accesible, y cumpla con los estándares de calidad.

Tipos de Pruebas de Validación

1. Pruebas de Validación de Funcionalidad (Funcionales)

Este tipo de prueba tiene como objetivo asegurarse de que todas las funcionalidades de la página web o la aplicación se comporten como se espera y cumplan con los requisitos iniciales.

Por ejemplo, encontramos:

- Verificar que los formularios de contacto se envíen correctamente.
- Comprobar que los botones de navegación lleven a las páginas correctas.

- Asegurar que los enlaces internos y externos funcionen y no generen errores 404.

2. Pruebas de Validación de Usabilidad

Estas pruebas verifican la **experiencia del usuario (UX)**, asegurando que la interfaz sea intuitiva y fácil de usar.

Por ejemplo, encontramos:

- Realizar pruebas con usuarios reales para observar su interacción con el sitio.
- Validar que los elementos interactivos sean fáciles de encontrar y usar (por ejemplo, formularios, botones).
- Asegurar que la navegación sea clara y que los usuarios puedan realizar tareas sin confusión.

3. Pruebas de Validación de Compatibilidad

Las pruebas de compatibilidad validan que el sistema o sitio web funcione correctamente en diversos dispositivos, navegadores y sistemas operativos.

Por ejemplo, encontramos:

- Verificar que el sitio web se visualice correctamente en diferentes navegadores como **Chrome**, **Firefox**, **Safari**, etc.
- Asegurar que el sitio sea completamente funcional en dispositivos móviles, tablets y computadoras de escritorio.
- Comprobar la compatibilidad con diferentes versiones de sistemas operativos como **Windows**, **macOS**, y **Linux**.

4. Pruebas de Validación de Seguridad

Este tipo de prueba tiene como fin verificar que el sitio web o aplicación

sea seguro y proteja adecuadamente los datos sensibles del usuario, cumpliendo con normativas de seguridad.

Por ejemplo, encontramos:

- Verificar que las contraseñas se almacenen de forma segura (usando cifrado adecuado).
- Asegurar que las transacciones financieras en línea estén protegidas mediante protocolos como **SSL/TLS**.
- Comprobar que el sitio esté protegido contra ataques comunes como **SQL Injection** o **Cross-Site Scripting (XSS)**.

5. Pruebas de Validación de Rendimiento

Estas pruebas validan que el sitio web o la aplicación puede manejar un alto volumen de tráfico o uso sin experimentar caídas o tiempos de carga excesivos.

Por ejemplo, encontramos:

- Realizar pruebas de carga para verificar cómo se comporta el sitio web cuando recibe una gran cantidad de usuarios simultáneamente.
- Medir la velocidad de carga de las páginas y asegurarse de que esté dentro de los parámetros ideales (menos de 3 segundos).

6. Pruebas de Validación de Accesibilidad

Estas pruebas aseguran que el sitio web sea accesible para todos los usuarios, incluidos aquellos con discapacidades.

Por ejemplo, encontramos:

- Validar que el contenido sea legible para personas con dificultades visuales utilizando herramientas de accesibilidad, como lectores de pantalla.
- Asegurarse de que el sitio sea completamente funcional utilizando solo el teclado (sin mouse).

- Verificar que los contrastes de color sean adecuados para personas con daltonismo y que el contenido sea fácil de navegar para usuarios con discapacidades motoras.

7. Pruebas de Validación de Localización e Internacionalización

Estas pruebas son necesarias cuando el sitio web se dirige a un público global o se traduce a varios idiomas. El objetivo es verificar que el contenido esté adaptado correctamente a las diversas regiones y culturas.

Por ejemplo, encontramos:

- Comprobar que el contenido traducido esté correcto y sea coherente con la cultura local.
- Verificar que las fechas, monedas y formatos de número sean apropiados según la región.
- Validar que los caracteres especiales en otros idiomas (como los acentos en español o los caracteres chinos) se muestren correctamente.

PRUEBAS DE VERIFICACIÓN

Las **pruebas de verificación** son un conjunto de procesos en los que se evalúa si un sistema, producto o componente cumple con los requisitos y especificaciones previamente establecidos. El objetivo principal de estas pruebas es **asegurar que el desarrollo se ha hecho correctamente** y que el sistema funciona según lo planeado, antes de pasar a las pruebas de validación, que se enfocan en la satisfacción de las necesidades del usuario.

A diferencia de las pruebas de validación, que se centran en verificar si el producto satisface las expectativas del usuario final, las pruebas de verificación se enfocan en garantizar que el producto cumpla con los requisitos técnicos, funcionales y de diseño establecidos por los desarrolladores y las partes interesadas.

Tipos de Pruebas de Verificación

1. **Revisión de Requisitos** Esta prueba se realiza para verificar que los **requisitos del sistema** (tanto funcionales como no funcionales) estén claramente definidos, sean completos y sean comprensibles. El objetivo es garantizar que no haya ambigüedades o contradicciones en los requisitos.

Por ejemplo, encontramos:

- Revisar la documentación del proyecto para asegurar que todos los requisitos del cliente estén presentes.
- Comprobar que no existan conflictos o incoherencias entre los requisitos.

2. **Revisión de Diseño** Las revisiones de diseño se llevan a cabo para verificar que el **diseño arquitectónico** y técnico del sistema se alinee correctamente con los requisitos previamente definidos. Esto incluye tanto el diseño funcional como el diseño de la interfaz de usuario.

Por ejemplo, encontramos:

- Validar la arquitectura de la aplicación para asegurarse de que todos los componentes estén correctamente distribuidos y sean escalables.
- Revisar el diseño de las bases de datos y las estructuras de datos para asegurar que se cumplan los requisitos de rendimiento y eficiencia.

3. **Revisión de Código** En este tipo de prueba, se lleva a cabo una **inspección del código fuente** para asegurarse de que se sigue la **mejor práctica de programación** y que el código esté libre de errores. Estas revisiones se realizan generalmente de forma manual o mediante herramientas automatizadas.

Por ejemplo, encontramos:

- Comprobar que el código esté estructurado adecuadamente y que siga las convenciones de codificación.
- Verificar que se hayan implementado las funciones según lo especificado en los requisitos del proyecto.
- Detectar posibles **errores de sintaxis** o **problemas de rendimiento** en el código.

4. **Análisis Estático** El **análisis estático** es una técnica en la que se analiza el código sin ejecutarlo. Se utilizan herramientas automatizadas para examinar el código y encontrar posibles errores, vulnerabilidades o defectos.

Por ejemplo, encontramos:

- Utilizar herramientas como **SonarQube** o **Checkmarx** para encontrar errores en el código o problemas de seguridad.
- Analizar el código para detectar violaciones de las reglas de codificación o detectar áreas que podrían mejorar en términos de rendimiento.

5. **Pruebas Unitarias** Las **pruebas unitarias** se enfocan en verificar que **cada unidad o componente individual del sistema** funcione correctamente. Son pruebas automáticas que se ejecutan a nivel de función o clase.

Por ejemplo, encontramos:

- Probar funciones individuales de una API para asegurarse de que devuelvan los resultados esperados.
- Verificar que las clases y métodos en una aplicación orientada a objetos se comporten de acuerdo con las especificaciones.

6. **Pruebas de Integración** Las pruebas de integración verifican que los componentes del sistema funcionen correctamente cuando se combinan. Estas pruebas son esenciales para asegurarse de que las diferentes partes del sistema interactúen adecuadamente.

Por ejemplo, encontramos:

- Verificar que las APIs se comuniquen correctamente con la base de datos.
- Asegurar que los diferentes módulos de un sistema se integren bien sin causar errores de compatibilidad.

7. **Pruebas de Configuración** Las pruebas de configuración se realizan para verificar que la aplicación pueda **ejecutarse correctamente** en los entornos de producción y de prueba. Esto incluye verificar las dependencias de software, bibliotecas y sistemas operativos necesarios para ejecutar la aplicación.

Por ejemplo, encontramos:

- Validar que la aplicación funcione en diferentes sistemas operativos (Windows, Linux, macOS).
- Comprobar que las dependencias, como las bases de datos y servidores web, estén configuradas correctamente.