

# Tema 2: HTML

---

Diseño de Interfaces Web - 2º DAW  
Juana Mª Rodríguez García

# 0. Lenguaje de Marcas

- Los **lenguajes de marcado** o de **marcas** son utilizados para estructurar y organizar la información en la web y otros entornos digitales.
- Los lenguajes de marcado **suelen confundirse con lenguajes de programación**. Sin embargo, no son lo mismo, ya que el lenguaje de marcado no tiene funciones aritméticas o variables, como tienen los lenguajes de programación.

# 0.1. Ejemplos de LM

- **HTML (HyperText Markup Language)**: Utilizado para crear páginas web y estructurar su contenido.
- **Markdown**: Empleado para escribir documentos de texto plano con formato ligero, como READMEs en GitHub.
- **LaTeX**: Utilizado en la producción de documentos científicos y técnicos de alta calidad tipográfica.
- **XML (eXtensible Markup Language)**: Usado para el intercambio de datos estructurados entre diferentes sistemas.

## 0.2. Características de LM

Los lenguajes de marcado utilizan **archivos de texto plano**, lo que permite que sean editados fácilmente con cualquier editor de texto y sean independientes de la plataforma o sistema operativo. Esto evita las restricciones de los archivos binarios, que requieren programas específicos para su edición.

Las instrucciones de marcado **se mezclan con el contenido en un único archivo**. Veamos algunos ejemplos de HTML, XML y Markdown:

- **HTML:**

1. <h1>Título de la página</h1>
2. <p>Este es un párrafo de ejemplo.</p>

- **Markdown:**

1. # Título de la página
2. Este es un párrafo de ejemplo.

- **XML:**

1. <documento>
2.     <titulo>Título de la página</titulo>
3.     <contenido>Este es un párrafo de ejemplo.</contenido>
4. </documento>

# 1. Lenguaje de marcas HTML

- El lenguaje de marcas HTML es un estándar reconocido en todo el mundo y sus normas están definidas en la página web del organismo W3C (World Wide Web Consortium)
- HTML fue desarrollado por **Tim Berners-Lee** en 1991 como parte del proyecto World Wide Web en el CERN (Consejo Europeo para la Investigación Nuclear).
- A lo largo de los años, HTML ha evolucionado con nuevas versiones y características para adaptarse a las necesidades cambiantes de la web.
- La última versión estable es HTML5, que ofrece mejoras en multimedia, interactividad y accesibilidad. HTML ha sido **fundamental en el crecimiento y la expansión de Internet**, permitiendo la creación de contenido digital accesible a nivel mundial.

# 1.1 Documentación oficial

Fuente	Descripción
<b>WHATWG (Web Hypertext Application Technology Working Group)</b>	Esta es la versión continuamente actualizada del estándar HTML mantenido por WHATWG.
<b>W3C (World Wide Web Consortium)</b>	Incluye versiones históricas y documentos sobre el desarrollo de HTML.
<b>MDN Web Docs (Mozilla Developer Network)</b>	Una referencia muy utilizada por desarrolladores, con documentación, tutoriales y ejemplos prácticos.
<b>Can I use</b>	Información sobre la compatibilidad de características HTML en diferentes navegadores.
<b>HTML5 Doctor</b>	Ofrece artículos y recursos sobre las características de HTML5, con ejemplos prácticos y explicaciones.
<b>W3Schools</b>	Tutoriales interactivos y referencias fáciles de seguir para aprender y practicar HTML.

## 1.2. Etiquetas HTML

- Todas las etiquetas HTML disponen de los símbolos mayor que y menor que ( < > ), y normalmente se utilizan en parejas. Una al principio (etiqueta de apertura) y otra al final del texto al que afectan (etiqueta de cierre).
- En el siguiente ejemplo se define un párrafo mediante la etiqueta `<p>`. Observa que la etiqueta de cierre del final incluye una barra inclinada.

```
<p>Párrafo</p>
```

## 1.3. Etiquetas vacías

Las etiquetas vacías o de autocierre en HTML son aquellas que **no contienen contenido entre una etiqueta de apertura y una de cierre**. Estas etiquetas se cierran automáticamente dentro de la etiqueta de apertura. Algunos ejemplos de etiquetas vacías o de autocierre son:

- `<br>`: Salto de línea o línea de quiebre.
- `<hr>`: Línea horizontal.
- `<img>`: Imagen.
- `<input>`: Campo de entrada de datos.
- `<meta>`: Metadatos del documento.
- `<link>`: Enlace a un recurso externo.
- `<wbr>`: Salto de línea sugerido.

# 1.4. Elementos estructurales

`<!DOCTYPE html>`

Indica que el documento está bajo el estándar de HTML5.

`<html>`

Representa la raíz de un documento HTML y sirve como contenedor para los demás elementos HTML, como `<head>` y `<body>`. El atributo lang especifica el idioma del documento, lo que ayuda con la accesibilidad y el SEO.

`<head>`

En su interior se incluye la colección de metadatos sobre el documento, los enlaces a scripts y hojas de estilo.

`<title>`

Representa el título del documento. Se muestra en la barra superior del navegador o en las pestañas de página.

`<link>`

Enlaza documentos externos, por ejemplo CSS.

`<meta>`

Define los metadatos que no pueden ser definidos usando otro elemento HTML.

`<style>`

Usada para escribir CSS interno.

`<script>`

Define un script interno o un enlace hacia un script externo de JavaScript.

`<body>`

Recoge todo el contenido que puede ver un usuario de un documento HTML. Sólo puede existir un elemento `<body>` en el documento.

# 1.5. Etiquetas y atributos

- Una **etiqueta** es un elemento que define la estructura y el contenido de una página web. Las etiquetas se utilizan para marcar o envolver diferentes partes del contenido, como textos, imágenes, enlaces, listas, entre otros elementos. Por ejemplo, `<p>` define un párrafo y `<img>` define una imagen.
- Por otro lado, un **atributo** proporciona información adicional sobre una etiqueta y suele especificarse dentro de la etiqueta de apertura. Los atributos modifican el comportamiento o la apariencia de las etiquetas y pueden ser obligatorios u opcionales dependiendo del contexto. Por ejemplo, en la etiqueta `<html>`, el atributo `lang` especifica el idioma del documento HTML, lo cual es crucial para la accesibilidad y el SEO:

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

# 1.6. Comentarios

```
1. <!-- Esto es un comentario dentro de un documento HTML -->
2. <!--
3. /$$  /$$ /$$$$$$$$$/ $$      /$$ /$$
4. | $$ | $$|__ $$__/| $$$ /$$$| $$
5. | $$ | $$| $$ | $$$$/ $$| $$
6. |$$$$$$$| $$ | $$ $$/$$ $$| $$
7. | $$__ $$| $$ | $$ $$| $$| $$
8. | $$ | $$| $$ | $$\ $| $$| $$
9. | $$ | $$| $$ | $$ \/| $$|$$$$$$$
10. |__/_/|__/_/|__/_/|__/_/|__/_/|_____/|
11. -->
12. <!--¿Te están entrando ganas de hacer ASCII Art? Puedes hacerlo en:
    http://patorjk.com/software/taag/#p=display&f=Big%20Money-ne&t=ENIUN -->
```

# 1.7. Ejemplo práctico

```
1. <!DOCTYPE html>
2. <html lang="es">
3. <head>
4.     <title>Título de la WEB</title>
5.     <meta charset="UTF-8">
6.     <meta name="title" content="Título de la WEB">
7.     <meta name="description" content="Descripción de la WEB">
8.     <link href="estilos.css" rel="stylesheet">
9.     <style>
10.        header{background-color:yellow;} /* Código CSS */
11.    </style>
12.    <script src="script.js"></script>
13.    <script type="text/javascript">
14.        /* Código JS */
15.    </script>
16. </head>
17. <body>
18.     ¡Hola Mundo!
19.     <!-- Añade tu código HTML aquí -->
20. </body>
21. </html>
```

## 2. HTML5

### 2.1. Reglas y normas HTML5

- **Estructura básica:** Todo documento HTML5 debe tener una estructura básica que incluya las etiquetas `<!DOCTYPE html> <html>`, `<head>`, y `<body>`. La etiqueta `<html>` es el elemento raíz y debe contener todas las demás etiquetas.
- **Etiquetas en mayúsculas o minúsculas:** Las etiquetas HTML no distinguen entre mayúsculas y minúsculas. Por lo tanto, `<HTML>` y `<html>` son equivalentes. Se recomienda seguir siempre la misma sintaxis, preferiblemente letras en minúsculas.
- **Caracteres especiales:** Algunos caracteres especiales, como `<`, `>`, `&`, `"`, y `'`, tienen significados especiales en HTML. Para incluir estos caracteres en tu código, puedes usar entidades HTML, su código decimal o hexadecimal.

## 2. HTML5

### 2.1. Reglas y normas HTML5

- **Código con tabulados y sangrías:** El código HTML indentado con tabulados facilita la lectura y su comprensión. Además, puede ayudar a evitar errores, ya que facilita la identificación de los elementos HTML. Sin embargo, hay que tener en cuenta también que la **velocidad de carga** es uno de los factores más influyentes en la experiencia que reciben los usuarios. Por ello, para **reducir el tamaño de los ficheros HTML** hay que evitar los tabulados, los espacios en blanco y los comentarios innecesarios. Una vez implementado nuestro código, se puede minificar mediante herramientas específicas.
- **Los saltos de línea y los espacios no tienen ningún efecto en el código HTML:** Los navegadores web ignoran los saltos de línea y los tabuladores en el código HTML contenido entre las etiquetas `<body></body>`. Para escribir espacios en blanco podemos usar `&nbsp;` y para escribir saltos de línea la etiqueta `<br>`

## 2. HTML5

### 2.1. Reglas y normas HTML5

- **Uso de comillas:** Aunque HTML5 permite el uso de comillas simples (' ) alrededor de los valores de los atributos, es una buena práctica utilizar comillas dobles ("") para que el código sea más consistente. `<a href="pagina.html">Enlace</a>`
- **Validación del código:** Utiliza herramientas de validación HTML para garantizar que tu código cumpla con las especificaciones y estándares HTML. Un código HTML válido es más compatible y menos propenso a errores. Te recomendamos utilizar el **Validador de W3C** para comprobar la calidad y estructura de tu código HTML.
- **Pruebas en diferentes navegadores:** Prueba tu sitio web en diferentes navegadores y dispositivos para garantizar que se vea y funcione correctamente en todas partes.
- **Implementar los factores HTML clave para el SEO:** La estructura y las prácticas de HTML5 pueden influir en la optimización de motores de búsqueda (SEO). Es importante asegurarse de que el código esté optimizado para SEO para mejorar la visibilidad y el ranking de los motores de búsqueda.

## 2. HTML5

### 2.2. Cierre de etiquetas

Las **etiquetas de apertura y cierre** en HTML son aquellas que tienen contenido entre una etiqueta de apertura y una etiqueta de cierre. Para cerrar estas etiquetas, es necesario utilizar el símbolo de barra inclinada (/) o *forward slash* seguido del nombre de la etiqueta. Por ejemplo:

1. <p>Este es un párrafo de texto.</p>
2. <a href="https://www.ejemplo.com">Enlace</a>
3. <div class="contenedor">
4.     Contenido dentro del div.
5. </div>

## 2. HTML5

### 2.2. Cierre de etiquetas

Las **etiquetas de autocierre o vacías** en HTML5 son aquellas que no requieren contenido adicional y se cierran de manera compacta utilizando solo el símbolo `>` al final de la etiqueta de apertura. Algunos ejemplos comunes de etiquetas de autocierre son `<img>`, `<br>`, y `<input>`. Por ejemplo:

1. ``
2. `<br>`
3. `<input type="text">`

En HTML5, es estándar y recomendado cerrar estas etiquetas de esta manera sin utilizar la barra inclinada y el corchete de cierre (`/>`).

## 2. HTML5

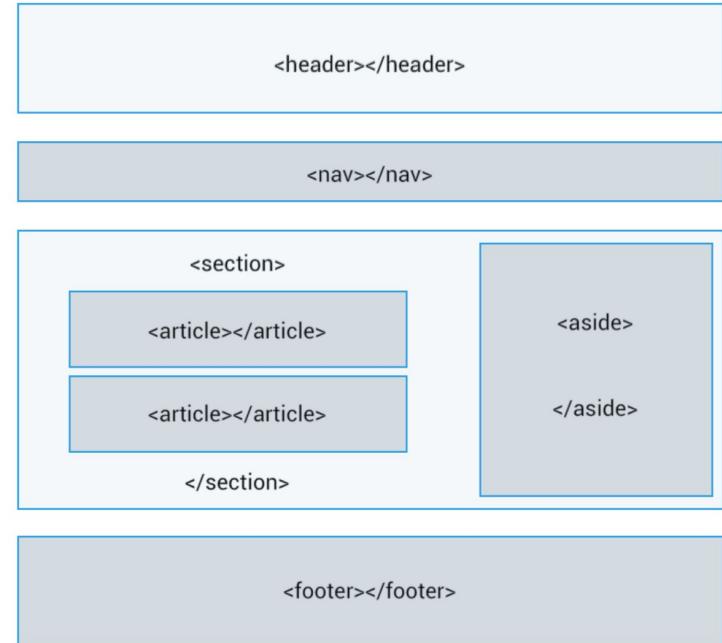
### 2.2. Cierre de etiquetas: etiquetas de autocierre

Etiqueta	Descripción
 	Salto de línea o quiebre de línea
<hr>	Línea horizontal
<img>	Imagen
<input>	Campo de entrada (input)
<link>	Enlace a un recurso externo
<meta>	Metadatos de la página
<col>	Columna en una tabla
<embed>	Contenido embebido
<param>	Parámetro para un objeto <object> o <embed>
<source>	Fuente de medios para elementos multimedia
<wbr>	Oportunidad de salto de línea condicional

# 3. Estructura y elementos semánticos

## 3.1. Elementos semánticos HTML5

- Una etiqueta semántica describe su significado. Por ejemplo: `<header>`, `<section>`, `<footer>`, `<form>`, `<table>`, `<article>`, `<nav>`... Ejemplos de elementos no semánticos (no dicen nada sobre su contenido): `<div>` y `<span>`.
- Muchos sitios web contienen código HTML como: `<div class="header"> <div id="nav"> <div id="section"> <div id="aside"> <div id="footer">`. Es muy importante cambiar ese código y usar etiquetas semánticas: `<header> <nav> <section> <aside> <footer>`.



Elemento	Descripción
<code>&lt;section&gt;</code>	Define secciones de una web.
<code>&lt;nav&gt;</code>	Especifica una sección que contiene un menú de navegación.
<code>&lt;article&gt;</code>	Representa un bloque de contenido independiente que podría ser distribuido y reutilizado por separado, como una publicación de blog, un artículo de revista o un comentario.
<code>&lt;aside&gt;</code>	Define la barra lateral de una página web.
<code>&lt;h1&gt;</code> <code>&lt;h2&gt;</code> <code>&lt;h3&gt;</code> <code>&lt;h4&gt;</code> <code>&lt;h5&gt;</code> <code>&lt;h6&gt;</code>	Describe el tema de la sección. Disponemos de seis niveles: de h1 a h6, siendo <code>&lt;h1&gt;</code> la cabecera de mayor importancia. Sólo puede existir una etiqueta <code>&lt;h1&gt;</code> en el documento.
<code>&lt;header&gt;</code>	Determina la cabecera de una web o de un elemento.
<code>&lt;footer&gt;</code>	Define el pie de página. También se puede utilizar para definir el pie de otros elementos.
<code>&lt;address&gt;</code>	Especifica una sección que contiene información de contacto.
<code>&lt;main&gt;</code>	Determina el contenido principal del documento. Solo puede existir un elemento <code>&lt;main&gt;</code> en el documento.

### **3. Estructura y elementos semánticos**

#### **3.1. Elementos semánticos HTML5**

Hay varias ventajas de usar elementos semánticos en HTML:

- Ayuda a los motores de búsqueda y otros dispositivos de usuario a determinar la importancia y el contexto de las páginas web. Es importante para el SEO (optimización en motores de búsqueda).
- Es mucho más fácil de leer.
- Tiene mayor accesibilidad.

Ejemplo de uso: <https://codepen.io/Eniun/pen/ymNam0>

# 4. Etiquetas de contenido y texto

## 4.1. Etiquetas de contenido

<code>&lt;p&gt;</code>	Define una parte que debe mostrarse como un párrafo.
<code>&lt;hr&gt;</code>	Representa un cambio temático entre párrafos. Suele representar una línea horizontal.
<code>&lt;pre&gt;</code>	Indica que su contenido está preformatado y que este formato debe ser preservado.
<code>&lt;blockquote&gt;</code>	Representa un contenido citado desde otra fuente.
<code>&lt;ol&gt;</code>	Especifica una lista ordenada de elementos.
<code>&lt;ul&gt;</code>	Determina una lista de elementos sin orden.
<code>&lt;li&gt;</code>	Define un elemento de una lista.
<code>&lt;dl&gt;</code>	Especifica una lista de definiciones, es decir, una lista de términos y sus definiciones asociadas.
<code>&lt;dt&gt;</code>	Representa un término definido por el siguiente <code>&lt;dd&gt;</code> .
<code>&lt;dd&gt;</code>	Indica la definición de los términos listados antes que él.
<code>&lt;figure&gt;</code>	Establece una figura ilustrada como parte del documento.
<code>&lt;figcaption&gt;</code>	Representa la leyenda de una figura.
<code>&lt;div&gt;</code>	Representa un contenedor genérico sin ningún significado especial.

EJEMPLO:

<https://codepen.io/Eniun/pen/oKzgEb>

## 4. Etiquetas de contenido y texto

### 4.1.1. Atributos más usados

Los atributos más usados son *id* y *class*.

- ***id***: se utiliza para identificar de manera única a un elemento. Como es un atributo identificativo único, no debemos añadir dos bloques con el mismo valor de id en el mismo documento HTML.

```
<div id="identificador">Contenedor</div>
```

- ***class***: se utiliza para asignar una o varias clases a un elemento HTML. Las clases son muy útiles para aplicar estilos CSS. El mismo valor de class se puede asignar a varios elementos.

```
<div class="clase1 clase2">Contenedor</div>
```

## 4. Etiquetas de contenido y texto

### 4.1.2. Atributos de listas

Atributo «**type**» (para listas ordenadas):

- «1»: Numeración decimal (1, 2, 3, ...).
- «A»: Letras mayúsculas (A, B, C, ...).
- «a»: Letras minúsculas (a, b, c, ...).
- «I»: Números romanos mayúsculos (I, II, III, ...).
- «i»: Números romanos minúsculos (i, ii, iii, ...).

EJEMPLO:

<https://codepen.io/Eniun/pen/qBLrRR>

Atributo «**type**» (para listas no ordenadas):

- «disc»: Puntos sólidos (●, ●, ●, ...).
- «circle»: Círculos (○, ○, ○, ...).
- «square»: Cuadrados (■, ■, ■, ...).

Atributo «**start**» (solo para listas ordenadas): Establece el valor inicial de la numeración de la lista.

Atributo «**reversed**» (solo para listas ordenadas): Invierte el orden de numeración de la lista.

# 4. Etiquetas de contenido y texto

## 4.2. Etiquetas de texto

Las etiquetas de texto son las que dan significado a los textos que contienen.

<a>

Representa un *hiperenlace*.

<em>

Especifica un texto *enfatizado*.

<strong>

Establece un texto *importante*.

<small>

Define un comentario aparte, es decir, textos de políticas de responsabilidad o una nota de derechos de autoría, que no son esenciales para la comprensión del documento.

<s>

Representa contenido que no es exacto, tiene el estilo *tachado*.

<cite>

Indica el título de una obra.

<q>

Representa una cita textual entre comillas.

<dfn>

Sirve para marcar el término que se quiere definir.

<abbr>

Envuelve un texto que al pasar el puntero por encima despliega un tooltip. El contenido del tooltip se detalla mediante el atributo *title*.

<time>

Determina un valor de fecha y hora.

<code>

Establece un código de programación.

<sub>

Establece un subíndice y un superíndice respectivamente.

<i>

Muestra el texto marcado con un estilo en cursiva o itálica.

<b>

Representa el texto marcado con un estilo en negrita.

<u>

Muestra el texto subrayado.

<mark>

Representa un texto marcado o resaltado.

<span>

Especifica texto en línea. Sirve para aplicar estilo al texto o agrupar elementos en línea.

<br>

Inserta un salto de línea.

<wbr>

Indica una oportunidad de salto de línea, es decir, un punto sugerido donde el texto puede ser dividido para mejorar su legibilidad.

## 4. Etiquetas de contenido y texto

### 4.2.2 Hiperenlaces: valores del atributo href

Como hemos visto, el elemento `<a>` o hiperenlace crea un enlace a otra página o archivo. Además, también puede enlazar a una sección de la misma página mediante marcadores, a un correo electrónico o a un teléfono, entre otras opciones.

```
<!-- Enviar un correo con mailto-->
```

```
<a href="mailto:info@eniun.com">Enviar correo a Eniun</a><br>
```

```
<!-- Llamar por teléfono con tel-->
```

```
<a href="tel:+34666666666">Llamar por teléfono a 666 666 666</a>
```

## **4. Etiquetas de contenido y texto**

### **4.2.1 Hiperenlaces: valores del atributo target**

Valor	Descripción
«_blank»	Abre el enlace en una nueva ventana o pestaña del navegador.
«_self»	Abre el enlace en la misma ventana o pestaña del navegador (valor predeterminado).
«_parent»	Abre el enlace en el marco principal o padre en caso de que se esté utilizando frames.
«_top»	Abre el enlace en la ventana principal del navegador, reemplazando cualquier frame existente.

## 4. Etiquetas de contenido y texto

### 4.2.1 Hiperenlaces: valores del atributo download

El atributo «download» en HTML se utiliza para especificar que un enlace debe ser descargado al hacer clic en él en lugar de navegar hacia él. Permite proporcionar un nombre de archivo sugerido para el archivo descargado. Así es como funciona el atributo «download»:

```
<a href="ruta/al/archivo.pdf" download="miarchivo.pdf">Descargar PDF</a>
```

En este ejemplo, cuando el usuario hace clic en el enlace «Descargar PDF», en lugar de abrirse en el navegador, se descargará el archivo especificado en el atributo «href» (ruta/al/archivo.pdf).

El atributo «download» especifica el nombre de archivo sugerido para el archivo descargado (miarchivo.pdf).

## 4. Etiquetas de contenido y texto

### 4.3 Caracteres especiales

Símbolo	Código decimal	Entidad
&	&#38;	&amp;
<	&#60;	&lt;
"	&#34;	&quot;
>	&#62;	&gt;
©	&#169;	&copy;
÷	&#247;	&divide;
€	&#8364;	&euro;
←	&#8592	&larr;

## **4. Etiquetas de contenido y texto**

EJERCICIO BÁSICO ESTRUCTURA (Moodle)

## 5. Formularios

El principio y final de un formulario se define con las etiquetas `<form>` y `</form>`. Dentro de las etiquetas de apertura y cierre de form se pueden incluir diferentes elementos que son enviados para ser procesados por el servidor web. Ejemplo:

```
1. <form action="procesar-registro.php" method="POST">
2.   <label for="name">Name:</label>
3.   <input type="text" id="name" required>
4.
5.   <label for="email">Email:</label>
6.   <input type="email" id="email" required>
7.
8.   <input type="submit" value="Submit">
9. </form>
```

# 5. Formularios

## 5.1. Atributos de la etiqueta form

- *action*: Puede ser una URL relativa o absoluta que indica dónde se enviarán los datos del formulario para su procesamiento.
- *method* puede ser «*GET*» o «*POST*» y determina el método HTTP que se utilizará para enviar los datos del formulario al servidor. La diferencia principal radica en cómo se transmiten los datos.
  - ***GET***: Los datos del formulario se adjuntan a la URL como parámetros de consulta. Es la forma predeterminada si no se especifica el atributo *method*. Es útil cuando se desea que los datos del formulario sean visibles en la URL, por ejemplo, al realizar búsquedas.
  - ***POST***: Los datos del formulario se envían en el cuerpo de la solicitud HTTP de manera más segura y no se muestran en la URL. Es adecuado para enviar datos sensibles, como contraseñas o información personal.

# 5. Formularios Petición HTTP

## 1. Cabecera (Headers):

Piensa en las cabeceras como la "información extra" o las **instrucciones** sobre cómo se debe procesar la solicitud. Contienen datos importantes como:

- El **tipo de navegador** que estás usando.
- El **idioma** que prefieres.
- Si tienes **cookies** guardadas para ese sitio web.
- El tipo de solicitud que estás haciendo (GET, POST, etc.).

Las cabeceras son invisibles para el usuario, pero el servidor las utiliza para entender mejor qué estás pidiendo y cómo responder.

### Ejemplo de cabecera:

```
css
```

Copiar código

```
GET /index.html HTTP/1.1
```

```
Host: www.ejemplo.com
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
```

```
Accept-Language: es-ES,es;q=0.9
```

## 2. URL (línea de solicitud):

Es la dirección web que indica lo que estás solicitando. En una solicitud de tipo **GET**, los datos del formulario se agregan aquí como parte de la URL, justo después del símbolo **?**.

### Ejemplo de URL en una solicitud GET:

```
bash
```

Copiar código

```
www.ejemplo.com/busqueda?producto=libro&orden=precio
```

- Aquí, **producto=libro** y **orden=precio** son datos que le estás enviando al servidor como parte de la URL.

## 3. Cuerpo (Body):

El **cuerpo de la solicitud** es donde viajan los **datos reales** que quieras enviar al servidor, especialmente en solicitudes **POST**. A diferencia de **GET**, donde los datos van en la URL, con **POST** los datos se "esconden" en el cuerpo de la solicitud, lo que es más seguro porque no se ven en la barra de direcciones.

Imagina que estás llenando un formulario con tu **nombre**, **correo electrónico** y **contraseña**. En una solicitud **POST**, esta información va dentro del cuerpo y viaja directamente al servidor.

### Ejemplo de cuerpo de solicitud en POST:

```
css
```

Copiar código

```
nombre=Juan&email=juan@example.com&password=1234
```

# 5. Formularios

## 5.2. Etiquetas para la creación de formularios

Elemento	Descripción
<code>&lt;form&gt;</code>	Define un formulario.
<code>&lt;fieldset&gt;</code>	Permite organizar en grupos los campos de un formulario.
<code>&lt;legend&gt;</code>	Representa el título de un <code>&lt;fieldset&gt;</code> .
<code>&lt;label&gt;</code>	Representa el título de un elemento de control de un formulario.
<code>&lt;input&gt;</code>	Se usa para crear controles interactivos que reciben datos del usuario.
<code>&lt;button&gt;</code>	Representa un botón.
<code>&lt;select&gt;</code>	Representa un elemento de control que permite la selección entre un conjunto de opciones <code>&lt;option&gt;</code> .
<code>&lt;option&gt;</code>	Representa una opción en un elemento <code>&lt;select&gt;</code> o <code>&lt;datalist&gt;</code> .
<code>&lt;optgroup&gt;</code>	Representa un conjunto de opciones, agrupadas lógicamente dentro de un <code>&lt;select&gt;</code> .
<code>&lt;datalist&gt;</code>	Se utiliza junto con la etiqueta <code>&lt;input&gt;</code> para proporcionar una lista de opciones <code>&lt;option&gt;</code> que ayudan a los usuarios a seleccionar un valor mientras escriben en un campo de entrada <code>&lt;input&gt;</code> .
<code>&lt;textarea&gt;</code>	Representa una caja de edición de texto en varias líneas.

EJEMPLO:

<https://codepen.io/Eniun/pen/xvRVEZ>

# 5. Formularios

## 5.2.1 Etiquetas `fieldset` y `legend`

Las etiquetas `<fieldset>` y `<legend>` se utilizan para agrupar elementos de formulario relacionados y proporcionar un título o leyenda descriptiva para el grupo.

El uso de las etiquetas `<fieldset>` y `<legend>` ayuda a mejorar la accesibilidad y usabilidad de los formularios al proporcionar una estructura clara y agrupación visual de campos relacionados. También mejora el estilo general y la organización del formulario.

```
<fieldset>
  <legend>Información de contacto</legend>
  <label for="name">Nombre:</label>
  <input type="text" id="name"><br>
  <label for="email">Email:</label>
  <input type="email" id="email"><br>
  <label for="message">Mensaje:</label>
  <textarea id="message"></textarea>
</fieldset>
```

# 5. Formularios

## 5.2.2 Etiqueta label

La etiqueta `<label>` ayuda a **asociar una descripción** de texto con un campo de entrada de formulario, lo que lo hace más accesible y fácil de usar. Ejemplo:

```
<label for="username">Nombre de usuario:</label>

<input type="text" id="username">
```

El atributo `for` de la etiqueta `<label>` especifica con qué elemento de entrada está asociado. En este caso, el valor del atributo `for` coincide con el atributo `id` del campo de entrada.

Cuando un usuario hace clic en la etiqueta, el campo de entrada correspondiente se enfoca o activa. Esto mejora la **usabilidad y accesibilidad**, ya que los usuarios pueden hacer clic en el texto de la etiqueta en lugar de hacer clic directamente en el campo de entrada.

# 5. Formularios

## 5.2.3 Etiqueta input

La etiqueta `<input>` en HTML se utiliza para crear campos de entrada en un formulario que permiten a los usuarios ingresar datos.

El tipo de dato que se ingresa depende del valor del atributo `type` de la etiqueta `<input>`.

1. **Campo de Texto:** Permite a los usuarios ingresar texto, como su nombre o dirección de correo electrónico.

```
1. | <input type="text" id="nombre" name="nombre" placeholder="Tu nombre">
```

2. **Campo de Contraseña:** Similar al campo de texto, pero los caracteres ingresados se ocultan por asteriscos o puntos para proteger la privacidad de la contraseña.

```
1. | <input type="password" id="contrasena" name="contrasena" placeholder="Contraseña">
```

3. **Casilla de Verificación (Checkbox):** Permite a los usuarios seleccionar o deseleccionar una opción.

```
1. | <input type="checkbox" id="suscripcion" name="suscripcion" value="aceptar">
```

```
2. | <label for="suscripcion">Acepto los términos y condiciones</label>
```

# 5. Formularios

## 5.2.3 Etiqueta input

4. **Botón de Radio (Radio Button)**: Permite a los usuarios seleccionar una opción de un conjunto de opciones mutuamente excluyentes.

```
1. <input type="radio" id="opcion1" name="opcion" value="opcion1">
2. <label for="opcion1">Opción 1</label><br>
3. <input type="radio" id="opcion2" name="opcion" value="opcion2">
4. <label for="opcion2">Opción 2</label>
```

5. **Botón (Button)**: Crea un botón que puede utilizarse para enviar un formulario o realizar otras acciones mediante JavaScript.

```
1. <input type="button" value="Enviar" onclick="enviarFormulario()">
```

6. **Campo de Correo Electrónico**: Ayuda a validar que se ingrese una dirección de correo electrónico válida

```
1. <input type="email" id="correo" name="correo" placeholder="Correo electrónico">
```

7. **Campo de Número**: Permite ingresar valores numéricos y puede incluir restricciones como mínimo y máximo.

```
1. <input type="number" id="edad" name="edad" min="18" max="99" placeholder="Edad">
```

# 5. Formularios

## 5.2.4 Etiqueta button

La etiqueta `<button>` en HTML se utiliza para crear un botón.

```
<button>Haz clic</button>
```

También puedes utilizar el atributo `type` para definir el tipo de botón. Algunos valores comunes para el atributo `type` son:

- `submit`: Un botón que envía un formulario.
- `reset`: Un botón que reinicia los valores iniciales de los campos de un formulario.

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

## 5. Formularios

### 5.2.4 Diferencias entre `input` y `button`

Los elementos `<input>` y `<button>` se utilizan para crear botones, pero tienen características distintas que afectan su uso y apariencia.

- `<input>`: Su apariencia es más limitada, ya que solo puede mostrar texto o imágenes y no puede contener otros elementos HTML.
- `<button>`: Es un elemento de botón más versátil que permite incluir contenido HTML, como texto, imágenes o iconos. Esto permite una mayor personalización en la presentación visual.

Por tanto, utiliza `<input>` para botones de formulario con funciones específicas y limitadas, mientras que `<button>` es preferible cuando se necesita más personalización o cuando se requiere incluir contenido HTML adicional.

# 5. Formularios

## 5.2.5 Etiqueta select

La etiqueta `<select>` en HTML se utiliza para crear un menú desplegable o una lista de opciones seleccionables. Permite al usuario elegir una opción de entre varias disponibles.

Cada `<option>` dentro de la etiqueta `<select>` representa una opción en el menú desplegable. El atributo `value` se utiliza para especificar el valor asociado a cada opción, que se enviará al servidor cuando se envíe el formulario.

También puedes agregar el atributo `selected` a una opción específica para que se seleccione automáticamente cuando se muestre el menú desplegable:

```
<select>
  <option value="option1">Opción 1</option>
  <option value="option2">Opción 2</option>
  <option value="option3">Opción 3</option>
</select>
```

# 5. Formularios

## 5.2.5 Etiqueta select

La etiqueta `<select>` en HTML también admite el atributo ***multiple***, que permite al usuario seleccionar múltiples opciones del menú desplegable manteniendo presionada la tecla Ctrl (en Windows) o Cmd (en Mac) mientras se hacen clic en las opciones deseadas.

La etiqueta `<optgroup>` se utiliza dentro de un elemento `<select>` para agrupar varias opciones relacionadas en una categoría o conjunto. Esto proporciona una organización visual a las opciones dentro de la lista desplegable. La etiqueta `<optgroup>` debe contener una o más etiquetas `<option>` que representan las opciones dentro de ese grupo.

```
<select multiple>
  <option value="option1">Opción 1</option>
  <option value="option2">Opción 2</option>
  <option value="option3">Opción 3</option>
</select>
```

```
<select>
  <optgroup label="Frutas">
    <option value="manzana">Manzana</option>
    <option value="naranja">Naranja</option>
    <option value="platano">Plátano</option>
  </optgroup>
  <optgroup label="Verduras">
    <option value="zanahoria">Zanahoria</option>
    <option value="brocoli">Brócoli</option>
    <option value="espinaca">Espinaca</option>
  </optgroup>
</select>
```

# 5. Formularios

## 5.2.6 Etiqueta `datalist`

La etiqueta `<datalist>` se utiliza junto con la etiqueta `<input>` para proporcionar una lista de opciones predefinidas que ayudan a los usuarios a seleccionar un valor mientras escriben en un campo de entrada `<input>`.

Esta etiqueta crea una lista desplegable de sugerencias basadas en las opciones que defines, lo que facilita la entrada de datos y **mejora la experiencia del usuario.**

El uso de `<datalist>` mejora la usabilidad y facilita a los usuarios la entrada de datos, especialmente en dispositivos móviles y pantallas táctiles.

```
<label for="frutas">Selecciona una fruta:</label>
<input list="frutas" id="fruta" name="fruta" />

<datalist id="frutas">
  <option value="Manzana"></option>
  <option value="Banana"></option>
  <option value="Naranja"></option>
  <option value="Uva"></option>
  <option value="Pera"></option>
</datalist>
```

## 5. Formularios

### 5.2.7 Etiqueta textarea

La etiqueta `<textarea>` en HTML se utiliza para crear un área de entrada de texto de varias líneas. Permite a los usuarios ingresar y editar texto más largo que un solo campo de entrada de texto.

Ejemplo básico de cómo utilizar la etiqueta `<textarea>`:

```
<textarea rows="4" cols="40">
```

Esto es un textarea con 4 filas y 40 columnas.

```
</textarea>
```

# 5. Formularios

## 5.3. Atributos de la etiqueta input

Atributos	Descripción
<b>type</b>	Tipo de input: text, password, checkbox, radio, etc.
<b>value</b>	Valor inicial del input.
<b>readonly</b>	El input es de solo lectura.
<b>disabled</b>	El input no se puede modificar.
<b>size</b>	Cantidad de caracteres visibles en un input. Su valor por defecto es 20. Funciona en los inputs de los siguientes tipos: text, search, tel, url, email, and password.
<b>maxlength</b>	Máximo número de caracteres del input.
<b>min / max</b>	Mínimo y máximo número de caracteres del input.
<b>pattern</b>	Nos permite indicar un patrón.
<b>placeholder</b>	Describe el valor esperado en un campo.
<b>required</b>	Especifica que es obligatorio completar el input.
<b>autofocus</b>	Indica que se hace foco en ese input cuando se carga la página.

## 5. Formularios

### 5.3. Atributos de la etiqueta input

El atributo **name** de un elemento <input> en un formulario HTML es fundamental para enviar los datos del formulario al servidor.

Cuando un formulario es enviado (por ejemplo, mediante un método POST o GET), los datos de los distintos campos se envían al servidor. El valor del atributo name en cada campo se utiliza como clave para identificar ese dato específico.

Al enviar el formulario, el navegador crea pares clave-valor, donde:

- La clave es el valor del atributo name del <input>.
- El valor es lo que el usuario ingresó en ese campo

# 5. Formularios

## 5.3. Atributos de la etiqueta input

```
<form>
  <label for="name">Nombre:</label>
  <input type="text" id="name" name="name" placeholder="Ingresa tu nombre" required>

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

  <label for="password">Contraseña:</label>
  <input type="password" id="password" name="password" minlength="8" maxlength="16" required>

  <label for="age">Edad:</label>
  <input type="number" id="age" name="age" min="18" max="99" required>

  <label for="terms">Acepto los términos y condiciones:</label>
  <input type="checkbox" id="terms" name="terms" required>

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

# 5. Formularios

## 5.4. Tipos de input: atributo type

<https://codepen.io/Eniun/pen/gOPGJGb>

Type	Description	Type	Description
checkbox	Casilla de verificación que permite seleccionar o deseleccionar distintos valores.	radio	Permite seleccionar un valor único entre múltiples opciones.
color	Control para especificar un color; abre un selector de colores cuando está activo.	range	Nos deja ingresar un número cuyo valor exacto no es importante. Se muestra como un widget de rango con un valor predeterminado en el medio. Se utiliza en conjunto con «min» y «max» para definir el rango de valores aceptados.
date	Permite indicar una fecha (año, mes y día, sin hora). Abre un selector de fechas para año, mes y día.	search	Campo para ingresar cadenas de búsqueda.
email	Permite insertar un correo electrónico haciendo su validación.	submit	Botón que envía el formulario
file	Permite seleccionar un archivo.	tel	Control para ingresar un número de teléfono y agrega validación predeterminada.
image	Botón con imagen de tipo submit. La imagen se define en el atributo src.	text	Control para campo de texto con validación predeterminada.
month	Permite ingresar mes y año.	time	Control para hora.
number	Permite ingresar un número. Muestra un selector y agrega validación predeterminada.	url	Nos permite ingresar una URL.
password	Valor oculto. Usado para contraseñas.	week	Control para ingresar una fecha.

# **5. Formularios**

EJERCICIO BÁSICO DE FORMULARIOS (MOODLE)

## 5. Formularios

### 5.4. Atributo pattern y expresiones regulares

El atributo *pattern* nos permite definir nuestras propias reglas para validar el valor de entrada de los campos usando expresiones regulares o *regexp* (contracción de las palabras inglesas *regular expression*).

Clases de caracteres	
.	Cualquier carácter excepto salto de línea
[abc]	Cualquiera de los caracteres entre corchetes
[^abc]	Que NO sea cualquiera de los caracteres entre corchetes
[a-g]	Cualquier carácter entre a y g (en minúscula)
(a b)	a o b
Anclas	
^abc	Comienzo de una línea
abc\$	Final de una línea

## 5. Formularios

### 5.4. Atributo pattern y expresiones regulares

El atributo *pattern* nos permite definir nuestras propias reglas para validar el valor de entrada de los campos usando expresiones regulares o *regexp* (contracción de las palabras inglesas *regular expression*).

Caracteres específicos	
\w\ d \s	Palabra, dígito, espacio en blanco
\W \D \S	Que NO sea palabra, dígito o espacio en blanco
\t \n \r	Tabulador, salto de línea, retorno de carro
Cuantificadores	
a* a+ a?	0 o más veces, 1 o más veces, 0 o 1 vez
a{5} a{2,}	Solo 5, 2 o más
a{1,3}	Entre 1 y 3

## 5. Formularios

### 5.4. Atributo pattern y expresiones regulares

#### Expresiones comunes

- **Entrada numérica:** Este patrón permite solo valores numéricos en el campo.
  - <input type="number" pattern="[0-9]+>
- **Patrón personalizado** donde la entrada debe comenzar con tres letras seguidas de tres dígitos.
  - <input type="text" pattern="[A-Za-z]{3}\d{3}>
- **Validación de código postal:** Este patrón asegura que el valor de entrada sea un código postal válido de 5 dígitos.
  - <input type="text" pattern="[0-9]{5}>
- **Validación de fecha** (formato AAAA-MM-DD): Este patrón valida que el valor de entrada sea una fecha en el formato AAAA-MM-DD. No valida la lógica de los valores de fecha, solo el formato de la cadena.
  - <input type="text" pattern="\d{4}-\d{2}-\d{2}>
- **Validación de nombre de usuario** (solo caracteres alfanuméricos): Este patrón permite solo caracteres alfanuméricos para el nombre de usuario.
  - <input type="text" pattern="[A-Za-z0-9\w]+>

## 5. Formularios

### 5.4. Atributo pattern y expresiones regulares

EJERCICIO: haz un formulario con campos que tengan las siguientes validaciones:

**Nombre de 40 caracteres como máximo y sólo permite letras(mayúsculas y minúsculas).**

**Edad (solo dos dígitos numéricos).**

**Validación de un formato de hora (formato HH:SS)**

**Validación de dirección de correo electrónico simple:** El correo debe seguir el formato estándar con una parte local, el símbolo @ y un dominio.

**Validación de un número de teléfono internacional (formato: + seguido de entre 1 y 3 dígitos para el código de país, seguido de 9 a 12 dígitos):**

**Validación de una matrícula de coche que debe comenzar con tres letras, seguidas de cuatro dígitos:**

## 5. Formularios

### 5.4. Atributo pattern y expresiones regulares

EJERCICIO: haz un formulario con campos que tengan las siguientes validaciones:

Validación de una dirección IP (IPv4)

Validación de una URL que debe empezar con `http://` o `https://` seguido de un dominio:

Validación de un número hexadecimal de 6 dígitos (usualmente utilizado en colores HTML):

Validación de una contraseña con al menos 8 caracteres, que incluya al menos una letra mayúscula, una letra minúscula, un número y un carácter especial

# 6. Tablas de contenido

El principio y final de una tabla se define con las etiquetas `<table>` y `</table>`. Las filas se engloban con las etiquetas `<tr>` `</tr>` y las columnas se crean mediante las etiquetas `<td>` `</td>` (las etiquetas `<td>` siempre estarán dentro de una fila `<tr>`). Las etiquetas `<th>` `</th>` se utilizan para representar los encabezados. Ejemplo:

```
<table>
  <tr>
    <th>Encabezado 1</th>
    <th>Encabezado 2</th>
    <th>Encabezado 3</th>
  </tr>
  <tr>
    <td>Fila 1, Celda 1</td>
    <td>Fila 1, Celda 2</td>
    <td>Fila 1, Celda 3</td>
  </tr>
  <tr>
    <td>Fila 2, Celda 1</td>
    <td>Fila 2, Celda 2</td>
    <td>Fila 2, Celda 3</td>
  </tr>
</table>
```

# 6. Tablas de contenido

## 6.1 Etiquetas

<https://codepen.io/Eniun/pen/qeqxrZ>

Elemento	Descripción
<code>&lt;table&gt;</code>	Define la tabla.
<code>&lt;caption&gt;</code>	Título de una tabla.
<code>&lt;tr&gt;</code>	Representa una fila de celdas en una tabla.
<code>&lt;td&gt;</code>	Representa una celda de datos en una tabla.
<code>&lt;th&gt;</code>	Representa el encabezado de una tabla.

```
<table>
  <tr>
    <th>Encabezado 1</th>
    <th>Encabezado 2</th>
    <th>Encabezado 3</th>
  </tr>
  <tr>
    <td>Fila 1, Celda 1</td>
    <td>Fila 1, Celda 2</td>
    <td>Fila 1, Celda 3</td>
  </tr>
  <tr>
    <td>Fila 2, Celda 1</td>
    <td>Fila 2, Celda 2</td>
    <td>Fila 2, Celda 3</td>
  </tr>
</table>
```

# 6. Tablas de contenido

## 6.2 Atributos border, align y bgcolor

Para dar estilos a los elementos utilizaremos siempre CSS y no atributos HTML. No obstante, se van a mencionar estos atributos porque todavía se encuentran presentes en multitud de proyectos online.

Verás que en muchos códigos se le añade un borde a la tabla usando el atributo **border**, se alinea el contenido usando el atributo **align** o se utiliza el atributo **bgcolor** para darle un color de fondo. Sin embargo, estos atributos ya no se suelen utilizar en HTML5.

**¡NO USES LOS SIGUIENTES ATRIBUTOS! Utiliza hojas de estilo CSS**

```
<table border="2">
```

```
<table align="center">
```

```
<td bgcolor="#77DDEE">
```

# 6. Tablas de contenido

## 6.3 Agrupación de celdas

Para agrupar celdas en una sola, también llamado «combinación de celdas», utilizaremos los atributos **rowspan** (para agrupar verticalmente) y **colspan** (para agrupar horizontalmente).

Tabla II: Agrupación de celdas en horizontal y vertical

Encabezado 1	Encabezado 2	Encabezado 3	Encabezado 4
Campo 1	Campo 2	Campo 3	Campo 4
	Campo 6		Campo 8
Campo 10	Campo 11	Campo 12	
Campo 13	Campo 14		

```
<table border="1">
  <caption>Tabla II: Agrupación de celdas en vertical</caption>
  <tr>
    <th>Encabezado 1</th>
    <th>Encabezado 2</th>
    <th>Encabezado 3</th>
    <th>Encabezado 4</th>
  </tr>
  <tr>
    <td rowspan="2">Campo 1</td>
    <td>Campo 2</td>
    <td>Campo 3</td>
    <td>Campo 4</td>
  </tr>
  <tr>
    <td colspan="2" rowspan="2">Campo 6</td>
    <td>Campo 8</td>
  </tr>
  <tr>
    <td colspan="2" rowspan="2">Campo 12</td>
    <td>Campo 13</td>
    <td>Campo 14</td>
  </tr>
</table>
```

## 6. Tablas de contenido

### EJERCICIO PROPUESTO

Investiga el uso de las etiquetas `<colgroup>`, `<col>`, `<tbody>`, `<thead>` y `<tfoot>`.

### EJERCICIO BÁSICO TABLAS (Moodle)

Realiza el horario de clase combinando siempre que sea posible tanto filas como columnas

```
<table>
  <colgroup>
    <col span="2"
      style="background-color:
      lightblue;">
    <col style="background-color:
      lightgreen;">
  </colgroup>
  <tr>
    <td>Dato 1</td>
    <td>Dato 2</td>
    <td>Dato 3</td>
  </tr>
  <tr>
    <td>Dato 11</td>
    <td>Dato 22</td>
    <td>Dato 33</td>
  </tr>
</table>
```

# 7. Contenido incrustado

## 7.1 Etiquetas

`<img>`

Representa una imagen.

`<iframe>`

Representa un contexto anidado de navegación, es decir, un documento HTML embebido.

`<embed>`

Representa un punto de integración para una aplicación o contenido interactivo externo que por lo general no es HTML.

`<object>`

Representa un recurso externo, que será tratado como una imagen, un sub-dокументo HTML o un recurso externo a ser procesado por un plugin.

`<video>`

Representa un vídeo.

`<audio>`

Representa un sonido.

`<source>`

Permite a autores especificar recursos multimedia alternativos para los elementos multimedia como `<video>` o `<audio>`.

`<canvas>`

Representa un área de mapa de bits en el que se pueden utilizar scripts para renderizar gráficos.

`<svg>`

Define una imagen vectorial embebida.

# 7. Contenido incrustado

## 7.2. Etiqueta img

Los atributos más comunes en las etiquetas de imagen HTML son:

- **src**: especifica la ruta o URL de la imagen.
- **alt**: proporciona texto alternativo que se muestra si la imagen no se puede cargar.
- **title**: proporciona texto descriptivo que se muestra al pasar el cursor sobre la imagen.
- **width**: especifica el ancho de la imagen en píxeles o como un porcentaje del ancho disponible.
- **height**: especifica la altura de la imagen en píxeles o como un porcentaje de la altura disponible.

Además, puedes ver en proyectos más antiguos el uso de los siguientes atributos:

- **border**: especifica el ancho del borde alrededor de la imagen. **Ya no se utiliza, mejor usar estilos CSS.**
- **align**: especifica la alineación de la imagen con respecto al texto circundante. Valores: left | right| middle| top| bottom. **Ya no se utiliza, mejor usar estilos CSS.**

```

```

## 7. Contenido incrustado

### 7.3. Incrustar vídeos de youtube

1. Ve al vídeo o a la lista de reproducción de YouTube que quieras insertar.
2. Haz clic en **COMPARTIR**.
3. En la lista de opciones que se muestra, haz clic en **Insertar**.
4. Copia el código HTML del cuadro que aparece.

## **7. Contenido incrustado**

### **7.4. Incrustar un mapa de google**

1. Accede a las indicaciones, al mapa o a la imagen de Street View que quieras insertar.
2. En la parte superior izquierda, haz clic en Menú.
3. Haz clic en Compartir o insertar el mapa.
4. Haz clic en Insertar un mapa.
5. A la izquierda del cuadro de texto, haz clic en el icono de la flecha hacia abajo Flecha hacia abajo para elegir el tamaño que deseas.
6. Copia el texto en el cuadro y pégalo en el HTML de tu sitio web o blog.

# 7. Contenido incrustado

## Ejemplo

<https://codepen.io/Eniun/pen/NQboLE>

```
<h3>Etiqueta img</h3>


<h3>Etiqueta iframe</h3>
<iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d154214.5103364689!2d-
0.5427351669515812!3d38.35795447718404!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0xd62
(Alacant)%2C+Alicante%2C+Espa%C3%B1a!5e1!3m2!1ses!2sus!4v1564299065614!5m2!1ses!2sus" width="6
style="border:0" allowfullscreen></iframe>

<h3>Etiqueta iframe</h3>
<iframe width="600" height="400" src="https://www.youtube.com/embed/coy5h2w5xUg" frameborder=
encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
```

## **7. Contenido incrustado**

EJERCICIO sobre contenido incrustado (moodle)

## 8. Elementos de ordenación

### 8.1. Elementos de bloque

En la actualidad, gracias a las **hojas de estilo** podemos crear nuestros propios estilos para maquetar cualquier página web. Para ello, necesitaremos conocer **cómo se ordenan los elementos en un documento web**.

En un documento web lo más normal es ir posicionando los elementos de izquierda a derecha y de arriba a abajo dentro de la etiqueta `<body>`. En el caso de elementos en bloque o *block* como `<div>`, `<table>`, `<list>`, etc., estos **ocupan todo el ancho del contenedor o elemento padre**.

Elemento 1

Elemento 2

Elemento 3

**Etiquetas HTML en bloque:** `<address>` `<article>` `<aside>` `<blockquote>` `<canvas>` `<dd>` `<div>` `<dl>` `<dt>` `<fieldset>` `<figcaption>` `<figure>` `<footer>` `<form>` `<h1>`-`<h6>` `<header>` `<hr>` `<li>` `<main>` `<nav>` `<ol>` `<p>` `<pre>` `<section>` `<table>` `<ul>` `<video>`

## 8. Elementos de ordenación

### 8.2. Elementos en línea

Los elementos en línea o *inline* como `<span>`, `<a>`, `<img>`, etc., ocupan sólo el espacio delimitado por las etiquetas que definen el elemento en línea.

Elemento 1

Elemento 2

Elemento 3

**Etiquetas HTML en línea:** `<a>` `<abbr>` `<b>` `<button>` `<cite>` `<code>` `<dfn>` `<em>` `<i>` `<img>` `<input>` `<label>` `<q>` `<select>` `<small>` `<span>` `<strong>` `<sub>` `<sup>` `<textarea>` `<time>`

## 8. Elementos de ordenación

### 8.3. Elementos flotantes

El comportamiento de los elementos se puede modificar haciendo que floten. Cuando a un elemento html se le aplica un estilo con la propiedad de flotar o *float*, el **elemento sale del flujo normal y aparece posicionado a la izquierda o a la derecha de su contenedor**, donde el resto de elementos de la página se posicionarán alrededor.

Las propiedades de los elementos flotantes se verán a fondo en la siguiente unidad.

# **8. Elementos de ordenación**

## **8.1. Posicionamiento**

Los elementos pueden estar posicionados de forma absoluta o relativa.

- **Posicionamiento absoluto:** el elemento siempre se encuentra en el mismo lugar.
- **Posicionamiento relativo:** el elemento se posiciona según otros elementos.

## 9. Marcadores

### 9.1. Marcadores en una misma página

Al crear una página web muy larga y con muchos apartados, es útil crear ciertos enlaces que nos permitan saltar directamente a la parte de la página que nos interesa. Este tipo de hipervínculos se llaman marcadores o enlaces locales.

La estructura de los marcadores es básicamente la misma que al crear un hiperenlace a otra página. Lo primero que hay que hacer es definir con un nombre cada parte de la página a la que queremos enlazar mediante el atributo id.

```
<a id="nombre_marcador"></a>
```

Esta etiqueta puede contener en su interior un texto, una imagen o incluso podemos dejarla en blanco.

La etiqueta para crear un enlace a esa sección de la página es la siguiente:

```
<a href="#nombre_marcador">Ir al título</a>
```

<https://codepen.io/Eniun/pen/VwwvvVv>

# 10. Factores HTML clave para el SEO

## #1 Título de la página, etiqueta title

La **etiqueta title** es muy importante ya que los buscadores suelen utilizarla para el título del resultado de búsqueda.

- **Título descriptivo, claro y llamativo:** El título debe ser descriptivo, claro y que ofrezca una propuesta de valor atractiva con el fin de conseguir que el usuario sienta interés y decida hacer clic.
- **Tamaño de 70 caracteres:** De cara a los motores de búsqueda, la dimensión óptima del título es de unos 70 caracteres. Hay que tener mucho cuidado en no dejar el título en blanco, muy corto, muy largo o crear etiquetas de título duplicadas.
- **Palabras clave al principio:** el título debe contener las palabras más importantes al principio de la frase.

```
<head>
  <title>Título de la página</title>
  <meta name="title" content="Título
alternativo para SEO">
</head>
```

# 10. Factores HTML clave para el SEO

## #2 Descripción de la página, metadescripción o metaetiqueta description

Igual que con el título, la metaetiqueta description es muy importante puesto que los buscadores suelen utilizarla para el fragmento del resultado de búsqueda.

- **Metaetiqueta description con contenido claro y sugerente:** la descripción debe ser clara y sugerente con el fin de conseguir que el usuario sienta interés y decida hacer clic.
- **Tamaño de 160 caracteres:** de cara a los motores de búsqueda, la dimensión óptima de la descripción es de unos 160 caracteres. Hay que tener mucho cuidado en no dejar la descripción en blanco, muy corta, muy larga o crear etiquetas de metadescripción duplicadas.
- **Palabras clave incluidas en la metadescripción:** la metadescripción debe contener las palabras clave con las que queremos posicionar el documento web.

```
<meta name="description"  
content="Descripción de la WEB">
```

# 10. Factores HTML clave para el SEO

## #3 Etiquetas de encabezado

Como hemos visto en esta unidad, el lenguaje HTML dispone de las **etiquetas h1, h2, h3,... y h6** para especificar los encabezamientos, **siendo h1 el más importante y el h6 el menos importante**. Además, debemos tener en cuenta las siguientes sugerencias:

- Todas las páginas web deben disponer **mínimo de un título de encabezado h1** que debe aparecer al principio de todo el contenido. Lo más recomendable es que este título cuente con las palabras clave con las que queremos posicionar el documento web. Estas palabras clave deben coincidir con las de la etiqueta title y la etiqueta description. Esto favorece a la coherencia de la página web y en definitiva a mejorar su posicionamiento.
- Un documento web **no puede disponer de más de un título h1**.
- Si existen subsecciones en la página, es **recomendable añadir etiquetas h2, h3,... sin dar saltos entre las etiquetas y respetando su jerarquía**.
- Siempre que sea posible es recomendable añadir **una etiqueta de encabezado cada 300 o más palabras**.

# 10. Factores HTML clave para el SEO

## #4 Maquetación sin tablas

Originalmente los diseñadores de páginas web utilizaban las tablas y los espaciadores para crear y mantener la disposición de los elementos de un sitio web. Sin embargo, este tipo de diseño ha sido descartado para favorecer la reducción del tamaño de los archivos y porque al ser leído por un motor de búsqueda el orden del contenido de las celdas de las tablas puede resultar muy confuso.

Por tanto, siempre que sea posible es **aconsejable maquetar sin tablas (*tableless layout*)**. Por ello, para la maquetación debemos utilizar las etiquetas html correspondientes y hojas de estilo en cascada CSS.

# 10. Factores HTML clave para el SEO

## #5 Minificar el código HTML

La **velocidad de carga** es uno de los factores más influyentes en la experiencia que reciben los usuarios de un sitio web. Por este motivo, tiene una incidencia directa en las tasas de abandono: los usuarios tienden a cerrar la página cuando ven que tarda mucho en cargar.

De esta forma, para **reducir el tamaño de los ficheros HTML** hay que evitar los espacios en blanco y los comentarios innecesarios. Además, se puede minificar el código mediante herramientas específicas.

# 10. Factores HTML clave para el SEO

## #6 Número de enlaces internos/externos

Si el número de enlaces salientes hacia otras páginas web es elevado, tu sitio web puede perder popularidad ya que el zumo de popularidad se está repartiendo a páginas externas y no a tu propio dominio.

Por tanto, hay que **mantener un equilibrio entre número de enlaces internos y externos**. Intenta minimizar el número de enlaces salientes, sobre todo los situados al pie de página puesto que afectará a todas las páginas de tu web.

# 10. Factores HTML clave para el SEO

## #7 Anchor text en los enlaces entrantes

El texto de anclaje o *anchor text* es el texto visible en el que el usuario hace clic para entrar a una web. Las palabras clave contenidas en ese texto visible **intervienen en el cálculo que realizan los buscadores para el posicionamiento** por lo que deben **contener las palabras clave** que se quieren posicionar en esa página concreta.

# 10. Factores HTML clave para el SEO

## #8 Busca errores en tu documento HTML

Utiliza el validador HTML para encontrar posibles errores cometidos y solucionálos para mejorar tu posicionamiento. Además, asegúrate de **no utilizar etiquetas HTML antiguas y deprecadas**.

# 10. Factores HTML clave para el SEO

## #9 Incluye un favicon

Disponer de un **favicon** es otro de los elementos valorados por los buscadores para mejorar el posicionamiento online de una página web. Además, añadir un favicon a tu sitio es de gran utilidad porque le da un aspecto más serio y profesional, además de que aparece en la barra de navegación y los usuarios pueden ubicar tu página con mayor facilidad cuando tengan varias pestañas abiertas.

Para crear un favicon puedes utilizar la siguiente herramienta: <http://www.favicon.pro/es/>

```
<link rel="shortcut icon"  
      type="image/x-icon"  
      href="favicon.ico">
```

# 10. Factores HTML clave para el SEO

## #10 Incluye un mapa del sitio

Los **mapas de sitio** pueden mejorar el posicionamiento en buscadores de un sitio web ya que **aseguran que todas las páginas puedan ser encontradas**. Esto es especialmente importante si el sitio usa menús en JavaScript que incluyan enlaces HTML.

La mayoría de los motores de búsqueda no siguen un número infinito de enlaces desde una página, por lo que si el sitio es muy grande la existencia del mapa del sitio puede ser necesaria para que tanto los motores de búsqueda como los **usuarios accedan a todo el contenido web**.

<https://es.semrush.com/blog/crear-sitemap-web/>

## 10. Factores HTML clave para el SEO

### #11 Utiliza el atributo alt en todas tus imágenes

El nombre de alt viene de *alternate text* o texto alternativo, y es lo que utilizan los buscadores para **saber qué es lo que se visualiza en una imagen**. También es el texto que utiliza el navegador cuando la imagen no se puede cargar.

**Asegúrate de escribir bien el atributo alt:**

- Describe correctamente el contenido de la imagen.
- Incluye la palabra clave principal.
- Utiliza pocas palabras.
- En el caso de disponer de una imagen meramente decorativa, debemos incluir el atributo alt vacío.

# **10. Factores HTML clave para el SEO**

## **#12 Ten en cuenta la accesibilidad web**

Un sitio web accesible suele tener contenido de calidad, lo que es fundamental para el SEO, ya que los motores de búsqueda valoran el contenido útil y relevante. Además, tener un sitio web accesible mejora la experiencia del usuario, lo que puede aumentar el tiempo que los visitantes pasan en el sitio, reducir la tasa de rebote y mejorar el posicionamiento en los motores de búsqueda. Por este motivo, es recomendable tener en cuenta hacer [imágenes accesibles](#), [tablas accesibles](#) o [formularios accesibles](#), entre otros aspectos que se pueden estudiar en detalle en el curso de [accesibilidad web](#).

# 11. Test de validación HTML y otros útiles

## 1. Validador HTML, W3C

El **W3C** nos ofrece el **validador HTML** ([validator.w3.org](https://validator.w3.org)) que nos permite encontrar errores en nuestros ficheros. Este validador HTML nos permite validar nuestras páginas seleccionando los ficheros HTML a validar o eligiendo directamente la url publicada.

## 2. HTMLHint, extensión para Visual Studio Code

Las páginas web se pueden validar en Visual Studio Code mediante la extensión HTMLHint.

# 11. Test de validación HTML y otros útiles

## 3. Firefox developer edition

Se trata de un navegador creado específicamente para desarrolladores web que ofrece herramientas muy interesantes como por ejemplo: panel de tipografías, editor de formas, inspector, consola o depurador, entre otras. [mozilla.org/es-ES/firefox/developer/](http://mozilla.org/es-ES/firefox/developer/).

## 4. Web developer

Consiste en una **extensión disponible para Chrome, Firefox o IE** que nos ofrece **un control exhaustivo de varias partes de la web, tales como el CSS, los formularios o las imágenes**. Con Web Developer instalado podemos, por ejemplo, alterar casi cualquier elemento final de la web en tiempo real y así aplicarlo a los archivos finales.

# 11. Test de validación HTML y otros útiles

## 5. Seositecheckup

SEOSiteCheckup es una herramienta online que nos ayuda a analizar el SEO de un sitio web para conocer factores como las palabras clave, enlaces rotos, mapa de sitio, etiquetas alt en las imágenes, velocidad de carga, servidores y seguridad, usabilidad en dispositivos móviles, etc. Esta herramienta online ofrece funciones limitadas en su versión gratuita. [seositecheckup.com](http://seositecheckup.com)

## 6. CodePen

CodePen es una comunidad online para probar y mostrar fragmentos de código HTML, CSS y JavaScript. Funciona como un editor de código online y un entorno de aprendizaje de código abierto, donde se puede crear y probar fragmentos de código, llamados «pens». [Ver tutorial de CodePen.](#)

# 12. Metainformación en HTML

1. `<title>`: Define el título del documento, que se muestra en la barra de título del navegador y en los resultados de búsqueda.
2. `<meta charset="UTF-8">`: Especifica la codificación de caracteres utilizada en el documento.
3. `<meta name="description" content="Descripción del sitio">`: Es **utilizado por los motores de búsqueda** para mostrar información sobre el sitio en los resultados de búsqueda.
4. `<meta name="keywords" content="palabra clave1, palabra clave2">`: Especifica palabras clave relacionadas con el contenido del sitio web. Aunque no es tan relevante para los motores de búsqueda modernos, aún puede ser utilizado para conocer la temática del sitio web.
5. `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Define cómo se debe ajustar y escalar el contenido en dispositivos móviles. Es **importante para lograr un diseño responsive**.
6. `<meta name="author" content="Nombre del autor">`:
7. `<meta name="copyright" content="Año copyright, titular de los derechos">`: Indica los derechos de autor del documento.

# 13. Metaetiqueta viewport

Cuando trabajamos con una **web responsive** es necesario **definir un *viewport* adecuado**, de lo contrario es muy probable que la página no lea correctamente los *media queries* y que se vea en formato muy reducido, siendo necesario hacer zoom para ver el contenido.

Las propiedades más comunes utilizadas en «content» de la metaetiqueta *viewport* son:

1. «**width**»: Especifica el ancho inicial del *viewport*. Por ejemplo, «width=device-width» establece el ancho inicial del *viewport* para que coincida con el ancho del dispositivo.
2. «**initial-scale**»: Define el nivel de escala inicial del *viewport*. Por ejemplo, «initial-scale=1.0» establece que la página se mostrará inicialmente sin ninguna escala.
3. «**minimum-scale**» y «**maximum-scale**»: Estos atributos permiten establecer los límites mínimos y máximos de escala del *viewport*, controlando si el usuario puede hacer zoom en la página.
4. «**user-scalable**»: Este atributo permite habilitar o deshabilitar la capacidad del usuario para hacer zoom en la página. Si se establece en «yes», el usuario podrá hacer zoom; si se establece en «no», se deshabilitará el zoom.
5. **Height**: altura virtual de la pantalla o altura del *viewport*

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