

### 3.1. Estructura

```
<!DOCTYPE html>
<html lang="es">
...
</html>
```

```
<!DOCTYPE html>
<html lang="es">
<head>
...
</head>
...
</html>
```

```
<!doctype html>
<html>
<head>
    <meta charset="UTF-8">
        <title>Untitled Document</title>
</head>
<body>

</body>
</html>
```



#### Etiqueta

Representa la raíz de un documento HTML y sirve como contenedor para los demás elementos HTML.

#### Etiqueta head

Dentro del elemento “html”, es decir, entre `<html>` y `</html>`, se debe escribir el elemento “head” que, como iremos viendo a lo largo del tutorial, puede contener diversa información sobre la configuración del documento. En su interior se incluye la colección de metadatos sobre el documento, los enlaces a scripts y hojas de estilo.

```
<html>
    <head>
        ...
    </head>
</html>
```

## **Etiqueta <title>**

Entre <head> y </head> se pueden escribir otros elementos. Por ejemplo, el elemento “title”, el cual sirve para indicar el título del documento. Se muestra en la barra superior del navegador o en las pestañas de página.

```
<html>
  <head>
    <title>Ejemplo de 2 párrafos</title>
  </head>
</html>
```

## **Etiqueta <link>**

Enlaza documentos externos, por ejemplo CSS.

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
  <link rel="stylesheet" href="mystyle.css">
</head>
<body>
  <h1>Esto es un encabezado</h1>
  <p>Esto es un párrafo.</p>
</body>
</html>
```

## **Etiqueta <meta>**

Define los metadatos que no pueden ser definidos usando otro elemento HTML: set de caracteres, descripción de la página, autor, palabras clave, ...

Estos datos no se muestran por pantalla pero son usados por el navegador, por motores de búsqueda y por otros servicios web.

```
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML, CSS, JavaScript">
<meta name="author" content="John Doe">
```

## Etiqueta <style>

Usada para escribir código CSS interno sin necesitar un documento aparte. Es usado si el estilo es propio de esa página (no del site) y contiene pocas líneas o modificaciones.

```
<style>
  body {background-color: powderblue;}
  h1 {color: red;}
  p {color: blue;}
</style>
```

## Etiqueta <script>

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

```
<script>
  function myFunction() {
    document.getElementById("demo").innerHTML = "Hello JavaScript"
  }
</script>
```

## Etiqueta <body>

En un documento HTML, después del “head”, hay que escribir el elemento “body” y alberga todo el contenido (párrafos, imágenes, vídeos...) del documento, los cuales se mostrarán en el navegador. Sólo puede existir un elemento <body> en el documento.

```
<html>
  <head>
    <title>Ejemplo de 2 párrafos</title>
  </head>
  <body>
    ...
  </body>
</html>
```

Finalmente, la estructura básica de un documento HTML es la siguiente:

```
1  <!DOCTYPE html>
2  <html lang="es-ES">
3  <head>
4    <meta charset="utf-8">
5    <title>Ejemplo de 2 párrafos</title>
6  </head>
7  <body>
8    <p>Esto es un párrafo.</p>
9    <p>Esto es otro párrafo.</p>
10 </body>
11 </html>
```

- <!DOCTYPE html> en la primera línea. Esta es la forma de indicar al navegador donde se visualice el documento que este está escrito en la versión HTML5.
- **lang="es-ES"** en la etiqueta de inicio <html>. Esto sirve para especificar, mediante el atributo lang, el idioma del contenido del elemento "html". En este caso español (España). Otros posibles valores del atributo lang son: "es-AR" para español (Argentina), "es-CO" para español (Colombia), "es-MX" para español (Méjico), etc.
- <meta charset="utf-8"> dentro del elemento "head". Así, se indica la codificación de caracteres del documento, en esta ocasión UTF-8 (8-bit Unicode Transformation Format, Formato de Transformación Unicode de 8 bits). Obsérvese que, el elemento "meta" está vacío (no tiene contenido) y, en consecuencia, no necesita ser cerrado escribiendo </meta>.

De esta forma, hemos escrito la estructura básica de un documento HTML que muestra por pantalla dos párrafos.

## Ejercicio

Vas a crear un sitio web con todos los ejercicios que vas a realizar en esta Unidad.

1. Crea una carpeta con el nombre **EjerciciosHtml**.
2. Dentro, crea una página web y nómbrala como **index.html**. Edita este archivo.
3. Crea una estructura básica HTML que se **usará para todos los ejercicios** con los siguientes añadidos:
  1. Todas las páginas enlazarán a una futura página CSS llamada **miestilo.css**.
  2. Se cuidará de que se muestren correctamente los símbolos españoles (tildes, ñ, ...)
  3. Añade tu nombre como autor de la web.
4. Crea un encabezado con el título **Ejercicios de xxx** donde XXX será tu nombre y apellidos.
5. Debajo, inserta un subtítulo con el curso **1º DAM. IES Nervión. Curso 20xx-20xx**.
6. A continuación, insertarás un párrafo con el título de cada ejercicio que realicemos.

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

## 3.2. Elementos

Un elemento HTML se define mediante una etiqueta inicial. Si el elemento contiene otro contenido, termina con una etiqueta de cierre, donde el nombre del elemento está precedido por una barra diagonal. Ejemplos:

```
1 | <div> ... </div>
2 | <h1> .. </h1>
3 | <p> .. </p>
```

Los elementos `<p> .... </p>` y `<h1> ... </h1>` son elementos HTML.

Mientras que `<p>` y `</p>` son las etiquetas con las que construyo el elemento, `<p> Este es el párrafo </p>` es un bloque de párrafo con el texto indicado.

Hay algunos elementos HTML que no necesitan cerrarse, solo tienen etiqueta inicial, como `<img>`, `<br>`, `<hr>`, etc. Estos se conocen como **elementos vacíos**.

### Elementos anidados

En HTML esta permitido (y abundantemente usado) tener elementos HTML anidados (unos dentro de otros)

El único problema es siempre recordar cerrar los elementos (si hay que cerrarlos), y en el orden correcto.

```
1 | <!DOCTYPE html>
2 | <html>
3 |   <head>
4 |     <title> Ejemplo de código HTML</title>
5 |   </head>
6 |   <body>
7 |     <p>Bienvenido al <b>Curso de HTML</b></p>
8 |   </body>
9 | </html>
```

Las etiquetas de HTML pueden tener varios formatos, según el elemento que se trate:

## Etiquetas con contenido

Se inician en una etiqueta de **apertura**, luego llevan el **contenido**, y finalizan con una etiqueta de **cierre**:

```
<p> aquí va el contenido </p>
```

```
<h1> aquí va el contenido </h1>
```

## Etiquetas sin contenido

Se inician en una etiqueta de **apertura**, **no llevan contenido**, y finalizan en la misma etiqueta. No es obligatoria cerrarlas:

```
<br> (insistimos, aunque funcione <br/>, no esta bien según las reglas de w3.org)
```

```
<input type="submit" value="Acceder">
```

```

```

Las etiquetas que tienen prohibido cerrarse, son :

**area, base, basefont, br, col, frame, hr, img, input, isindex, link, meta, param**



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.  
<area>: Área de un mapa de imagen.  
<base>: URL base para enlaces relativos.

Estas etiquetas se utilizan para insertar elementos específicos en el documento HTML y no requieren una etiqueta de cierre separada.

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.2.1 Elementos para dar formato

Elemento	Descripción
p	Delimita los párrafos
h <i>i</i>	Encabezado de nivel <i>i</i> , donde <i>i</i> es un número entero entre 1 y 6, ambos inclusive. El tamaño de la letra del encabezado es mayor cuanto menor sea el valor de <i>i</i> . No deben de usarse estas etiquetas para formatear texto. Sólo estarán bien usadas para designar títulos de párrafos.
br	Define un salto de línea sin iniciar un nuevo párrafo. La etiqueta   es una etiqueta vacía, lo que significa que no tiene etiqueta de final.
b	Indica que el texto que está en ese elemento se le pondrá en negrita.
i	Indica que el texto que está en ese elemento se le pondrá en itálica ó cursiva.
u	Indica que el texto que está en ese elemento se le pondrá subrayado.
sup	Indica que el texto que está en ese elemento es un supraíndice.
sub	Indica que el texto que está en ese elemento es un subíndice.
strong	Indica que el texto que está en ese elemento estará resaltado. Habitualmente los navegadores resaltan el texto poniéndolo en negrita aunque podría haber algún navegador que resaltase el texto poniéndolo en cursiva y en naranja.

#### Ejemplo

Con el texto siguiente, ¿sabrías conseguir el mismo resultado?

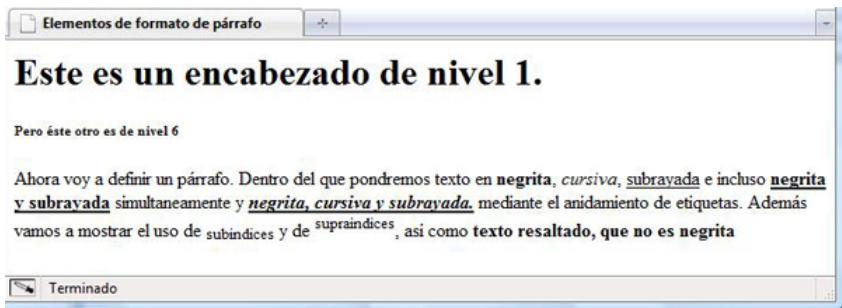
Este es un encabezado de nivel 1.

Pero éste otro es de nivel 6

Ahora voy a definir un párrafo. Dentro del que pondremos texto en negrita, cursiva, subrayada e incluso negrita y subrayada simultáneamente y negrita, cursiva y subrayada, mediante el anidamiento de etiquetas.

Además vamos a mostrar el uso de subíndices y de supraíndices, así como texto resaltado, que no es negrita.

Guarda el ejemplo como **ej1.html**



Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.2.2. Listas

Elemento	Descripción
ul	Delimita los elementos que forman una lista desordenada
ol	Delimita los elementos que forman una lista ordenada
li	Indica cada uno de los elementos de una lista
dl	Delimita los elementos que forman una lista de definición
dt	Cada uno de los términos que se definen de una lista de definición.
dd	Cada una de las definiciones de una lista de definición.

#### Listas sin orden

Las etiquetas **ul** usan la propiedad **list-style-type** para definir el estilo de la lista:

- **disc**. Para mostrar un círculo relleno (es el valor por defecto)
- **circle**. Muestra un círculo vacío.
- **square**. Muestra un cuadrado.
- **none**. No mostrará ningún símbolo en cada línea.

#### Listas ordenadas

La etiqueta **ol** pueden usar las siguientes propiedades:

- **start**. Para indicar con qué valor se comienza.
- **reversed**. Para ordenar de forma decreciente.
- **type="1"**. La lista se representará con números comenzando con el 1 (por defecto).
- **type="A"**. La lista se representará con letras mayúsculas.
- **type="a"**. La lista se representará con letras minúsculas.
- **type="I"**. La lista se representará con números romanos en mayúsculas.
- **type="i"**. La lista se representará con números romanos en minúsculas.

## Ejemplo

¿Sabrías reproducir la siguiente lista?

1. Primer elemento
  2. Segundo elemento
  3. Tercer elemento
- 

- I. Primer elemento
  - II. Segundo elemento
  - III. Tercer elemento
- 

6. Primer elemento
  7. Segundo elemento
  8. Tercer elemento
- 

- Primer elemento
  - Segundo elemento
  - Tercer elemento
- 

- Primer elemento
  - Segundo elemento
  - Tercer elemento
- 

- Primer elemento
- Segundo elemento
- Tercer elemento

Guarda el ejemplo como **ej2.html**



### 3.2.3. Tablas

Las tablas organizan la información de la página en filas y columnas. Cada vez se usan menos por su poca adaptabilidad (responsive) a los diferentes dispositivos existentes.

Elemento	Descripción
table	Delimita el contenido de una tabla.
caption	Para agregar un título a una tabla. Debe insertarse inmediatamente después de la etiqueta.
tr	Delimita cada una de las líneas de la tabla.
td	Delimita el contenido de cada celda de la tabla.
th	Delimita cada una de las celdas de la cabecera
colspan (Atributo de td y th)	Para hacer que una celda cubra más de una columna
rowspan (Atributo de td y th)	Para hacer un intervalo de celda más de una fila.

Las tablas pueden estructurarse en cabeza (**thead**), cuerpo (**tbody**) y pie (**tfoot**). Dentro de ellas se insertan las filas (**tr**) y las columnas (**td**) pero, también, celdas de cabecera (**th**) con formato.

Los elementos **td** son los contenedores de datos de la tabla. Pueden contener todo tipo de elementos HTML; texto, imágenes, listas, otras tablas, etc.

En ocasiones, tendremos la necesidad de unir celdas por algún motivo (encabezados, títulos, celdas con la misma información, ...).

Para ello, las celdas **td** cuentan con los atributos **rowspan** y **colspan**.

- **Rowspan** une filas
- **Colspan** une columnas

Hay que tener en cuenta que, al unir celdas, en las filas/columnas donde se consideran unidas a la anterior, no se representará su **td** correspondiente. En este caso, se creará el siguiente **td** si lo hubiera.

### Ejemplo

¿Sabrías crear esta tabla?

Lunes	Martes	Miércoles	Jueves	Viernes
Mates		Dibujo		Historia
Física	Lengua	Inglés	Francés	
	EF	Fútbol	Tenis	Mates

**Pista:** Una celda necesitará la etiqueta **p**.

Guarda el ejemplo como **ej3.html**

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.2.4. Enlaces y marcadores

Los enlaces se encuentran en casi todas las páginas web. Los enlaces permiten a los usuarios hacer clic en su camino de una página a otra. Los enlaces HTML son hipervínculos. Un enlace no tiene que ser texto. Puede ser una imagen o cualquier otro elemento HTML.

En HTML, los vínculos se definen con la etiqueta `<a>` :

```
1 | <a href="https://www.institutonervion.es" target="_blank">Visita r
```

El atributo **href** especifica la dirección de destino del enlace.

El atributo **target** especifica dónde abrir el documento vinculado y pude tener los siguientes valores:

- `_blank` - Abre el documento vinculado en una nueva ventana o pestaña
- `_self` - Abre el documento vinculado en la misma ventana / pestaña que se hizo clic (esto es el predeterminado)
- `_parent` - Abre el documento vinculado en el marco padre
- `_top` - Abre el documento vinculado en todo el cuerpo de la ventana
- `framename` - Abre el documento vinculado en un marco con nombre

#### Marcadores

Al crear una página web con un contenido extenso, existen ciertos enlaces que permiten saltar directamente a una parte concreta de la página. Este tipo de enlaces o hipervínculos se llaman **marcadores** o **enlaces locales**.

Un ejemplo claro se puede visualizar en la web [www.wikipedia.es](http://www.wikipedia.es) <https://es.wikipedia.org/wiki/Marillion> en cualquiera de sus artículos (por ejemplo, **aquí** ) donde se utiliza en el menú para acceder a los diferentes apartados.

Para crear marcadores se debe contar con **una etiqueta** de tipo enlace `<a>` para el origen y otra, de cualquier tipo, para el destino (el marcador).

- En el origen se declara como referencia **href** un nombre utilizado como marcador, precedido por el símbolo `#` (almohadilla).

- En el destino, se debe declarar el atributo **id** o el atributo **name** con dicho nombre de marcador pero **sin** la almohadilla. Esta etiqueta puede contener en su interior un texto, una imagen o incluso podemos crear una en blanco. Lo importante es el **id**.

```
<a href="#nombre_marcador">Soy el origen que se pulsa</a>
```

...

```
<etiqueta id="nombre_marcador">Soy el marcador que se mostrará al
```



La respuesta es sí. Simplemente se declara como referencia el nombre del marcador en la etiqueta enlace.

```
<a href="sierradePINARES.html#quinto_pino">Pulsa para acceder al c
```



## Ejercicio

Descarga [este archivo](#) (html - 5122 B) y modifícalo para poder acceder a los diferentes apartados así como poder regresar al Inicio cuando se acabe de leer cada uno de ellos.

Guarda el ejemplo como **ej4.html**

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.2.5. Imágenes

Las imágenes se definen con la etiqueta **<img>**. La etiqueta **<img>** está vacía y **no tiene etiqueta de cierre**.

El atributo **src** especifica la URL (dirección web) de la imagen.

El atributo **alt** proporciona un texto alternativo para una imagen, si por alguna razón el usuario no puede verla (debido a una conexión lenta, un error en el atributo src, ...) se mostrará el valor del atributo alt. Puedes probar a insertar una etiqueta imagen sin **src** pero con **alt**. ¿Qué ocurrirá?

```
1 | 
```

La altura **height** y la anchura **width** se deben especificar mediante CSS y no en la propia etiqueta. Aunque, a modo de ejemplo, nos puede ser ilustrativa.

#### Figure

La etiqueta **<figure>** puede usarse para agrupar elementos tales como imágenes, vídeos o incluso texto que forma parte de una ilustración del contenido principal.

La etiqueta **<figcaption>**, usada de forma conjunta con la etiqueta **<figure>**, proporciona una leyenda a los elementos así agrupados, por ejemplo para añadir una leyenda a un grupo de imágenes.

Ejemplo de Figure en w3schools <[https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5\\_figure](https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_figure)>

#### Ejercicio

Crea una tabla con **dos filas y tres columnas** donde la columna central será **una única celda**.

Busca 5 imágenes de una misma temática y coloca la más representativa en la celda central con un tamaño de 250px de alto y no más de 500px de ancho.

En las celdas laterales, coloca el resto de imágenes con dimensiones de 100px de alto por 200px de ancho.

Adicionalmente, puedes crear una línea en la parte superior de la tabla para escribir el título de lo que se muestra. Además, intenta centrar las imágenes.

Guarda el ejemplo como **ej5.html**

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.2.6. Audio y vídeo

#### Incrustado (embedded)

Existen etiquetas que incorporan (embed) elementos externos a en una página web. Estos elementos pueden ser imágenes, vídeos, audios, mapas, fórmulas, etc...

- **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>. Por ejemplo, audios en diferentes idiomas.
- **canvas**, Representa un área de mapa de bits en el que se pueden utilizar scripts para renderizar gráficos.
- **svg**. Define una imagen vectorial embebida.

El sonido y el video han tenido y tienen un papel muy importante en el diseño de sitios web. Ofrecen servicios de descarga de una muestra de un número musical o del último video de moda.

Antes de incorporar este tipo de archivos, hay que tener en cuenta que formatos sin comprimir pueden ser muy grandes y nada recomendables.

Existen formatos de archivos para la Web con **algoritmo de compresión** complejos que mejoran en gran medida esta cuestión: MP3, OGG, MPA, WMA...

Incluso estos formatos permiten iniciar la reproducción descargando sólo parte del archivo mientras se muestra el contenido al usuario, y continuar con la reproducción en un segundo plano.

Se recomienda **no reproducir los archivos de audio y video automáticamente** cuando un usuario visita un sitio Web.

La sintaxis de la etiqueta es la siguiente.

```

<audio autoplay controls loop muted preload="valor" src="valor">
    Texto para navegadores que no soportan audio...
</audio>

```

Entre los atributos de la etiqueta encontramos los siguientes:

Atributo	Valor	Descripción
autoplay	autoplay	El audio comenzará a reproducirse tras su carga.
controls	controls	Especifica que se mostrarán los controles (play, stop, ...)
loop	loop	El audio se repetirá indefinidamente una vez acabe.
muted	muted	El audio estará en silencio.
preload	auto metadata none	Especifica el comportamiento del audio una vez se cargue la página.
src	URL	Origen del audio

Ejemplo de la etiqueta audio en w3schools <[https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5\\_audio](https://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_audio)>

## Ejercicio

Usa la página <https://lamusicagratis.com/> <<https://lamusicagratis.com/>> para descargar una canción libre de derechos de autor.

Crea una página donde se muestre el nombre del autor, así como el texto que acompaña la canción que has descargado. Formatea correctamente dicho texto evitando que parezca un copia-pegar.

Finalmente, muestra el reproductor con la canción descargada reproduciéndose automáticamente. Añade los datos de Autoría, Licencia, el enlace donde encontrarla así como el de descarga.

Guarda el ejemplo como **ej6.html**

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.2.7. Formularios

Los **formularios** ocupan un lugar predominante en el diseño y explotación de una aplicación o un sitio Web. Es, en efecto, **la única forma de recibir** de vuelta la **información** que proviene directamente del usuario final, **estructurada**, además, según las necesidades del diseñador.

El propósito último de los formularios es el procesamiento automático de los datos devueltos. Por ello es preciso mencionar, del lado del **servidor**, los lenguajes de programación de gestión de bases de datos como, por ejemplo, **PHP** o **MySQL**.

La etiqueta **<form>** ... **</form>** tiene como única función la de declarar al navegador que debe incluir un formulario. Esta etiqueta englobará los elementos o campos de formulario, como por ejemplo: casillas de texto, casillas de marcar, listas desplegables, etc.

```
<html>
<head>
<title>Html5</title>
<meta charset="UTF-8">
</head>
<body>
<form action="">
Elementos del formulario
</form>
</body>
</html>
```

Los atributos usuales de la etiqueta

son:

- **name**: Para asignar un nombre (name) al formulario.
- **action**: Se especificará aquí el destino donde será enviada la información del formulario. El destino normalmente es una llamada a otra página, donde se procesarán los datos, alojada en un servidor. Por ejemplo: action = "http://www.servidor/procesamiento.php". También puede ser a sí mismo dejando **action** vacío.
- **target**: Al igual que en los enlaces, **target** hace que el destino se muestre en la **misma página**, una **nueva** o un **frame** (ver **enlace** para más información).

- **method:** Especifica el método **HTTP** ( **GET** o **POST** ) que se utilizará al enviar los datos del formulario. El método predeterminado al enviar datos del formulario es **GET**, sin embargo, cuando se utiliza, los datos del formulario enviado serán visibles.
- **enctype:** Especifica en qué formato informático se transmitirán los datos del formulario. El valor por defecto es **application/x-www-form-urlencoded**. Para enviar archivos, este campo debe valer **multipart/form-data** y, por último, para el envío de datos a una dirección de correo electrónico mediante el protocolo **mailto**, el valor será **text/plain**.

La etiqueta debe, obligatoriamente, cerrarse. En caso de olvidar la etiqueta de cierre el navegador no mostrará ningún campo del formulario.

Puedes usar la página [w3schools.com/action\\_page.php](https://www.w3schools.com/action_page.php)

<[https://www.w3schools.com/action\\_page.php](https://www.w3schools.com/action_page.php)> como destino del formulario y observar si los resultados son los esperados o no.

## GET y POST

### GET

GET es la opción por defecto de un formulario. Este método agrega datos de formulario a la URL en pares de nombre / valor mostrándose en el momento del envío/recepción.

**¡Nunca use GET para enviar datos confidenciales! (será visible en la URL)**

GET es mejor para datos no seguros, como cadenas de consulta en Google o el número de página que se está visitando.

### POST

Utilice siempre POST si los datos del formulario contienen información confidencial o personal.

El método POST no muestra los datos del formulario enviado en el campo de dirección de página.

POST no tiene limitaciones de tamaño y puede usarse para enviar grandes cantidades de datos.

### fieldset

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

La etiqueta **<fieldset>** actúa como un contenedor que agrupa campos del formulario. Ejemplo:

```
<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>
```

En este ejemplo, la etiqueta **<fieldset>** envuelve el grupo de elementos de formulario relacionados con la información de contacto. La etiqueta **<legend>** se utiliza como una leyenda o título para el **fieldset**, proporcionando una descripción del 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.

### 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">
```

En este ejemplo, la etiqueta `<label>` se utiliza para crear una etiqueta para el campo de entrada de nombre de usuario. 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.

## 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>`. Veamos algunos ejemplos comunes de cómo se utiliza la etiqueta `<input>` en un formulario:

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

```
<input type="text" id="nombre" name="nombre" placeholder="T
```

**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.

```
<input type="password" id="contrasena" name="contrasena" pl
```

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

```
<input type="checkbox" id="suscripcion" name="suscripcion">  
<label for="suscripcion">Acepto los términos y condiciones<
```

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

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



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

```
<input type="button" value="Enviar" onclick="enviarFormularios()>
```



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

```
<input type="email" id="correo" name="correo" placeholder="Introduce tu correo electrónico">
```



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

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



Estos son solo algunos ejemplos de cómo se puede utilizar la etiqueta `<input>` en un formulario HTML.

## button

La etiqueta `<button>` en HTML se utiliza para crear un botón. Puede ser utilizado para activar una acción o comportamiento específico al ser pulsado.

```
<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>
```

## 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.

Ejemplo básico de cómo utilizar la etiqueta **<select>** junto con la etiqueta **<option>** para crear un menú desplegable:

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

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" selected>Opción 2</option>
  <option value="option3">Opción 3</option>
</select>
```

En este ejemplo, **la opción «Opción 2» estará seleccionada** de forma predeterminada cuando se muestre el menú desplegable.

La etiqueta **<select>** en HTML también admite el atributo **multiple**, que permite al usuario seleccionar múltiples opciones del menú desplegable.

Al agregar el atributo **multiple** a la etiqueta **<select>**, se permite la selección de varias opciones manteniendo presionada la tecla Ctrl (en Windows) o Cmd (en Mac) mientras se hacen clic en las opciones deseadas. Ejemplo:

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

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>
    <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>
```

En este ejemplo, hemos creado una lista desplegable con dos grupos: «Frutas» y «Verduras». Cada grupo está definido por la etiqueta **<optgroup>**, que tiene un atributo **label** que especifica el nombre del grupo. Dentro de cada grupo, hemos incluido etiquetas **<option>** que representan las opciones individuales, como «Manzana», «Naranja», «Zanahoria», etc.

Cuando se renderiza en un navegador, esto creará una lista desplegable con las opciones agrupadas visualmente bajo las etiquetas de grupo. Esto facilita que los usuarios encuentren y seleccionen las opciones que desean de manera más organizada.

## 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.

A continuación se puede ver un ejemplo de cómo se utiliza la etiqueta **<datalist>**:

```
<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>
```

En este ejemplo, hemos creado un campo de entrada de texto con la etiqueta **<input>** y le hemos asignado un atributo **list** que corresponde al **id** de la etiqueta **<datalist>**. El **<datalist>** contiene opciones (**<option>**) que representan las sugerencias disponibles para el campo de entrada.

Cuando un usuario comienza a escribir en el campo de entrada, verá una lista desplegable con las opciones del **<datalist>**. Puede seleccionar una de las opciones o seguir escribiendo. Esto es especialmente útil para formularios en los que los usuarios deben elegir una opción de una lista predefinida, como una lista de países o productos.

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

## 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>
```

En este ejemplo, el atributo **rows** define el número de filas visibles del área de texto, mientras que el atributo **cols** establece el número de columnas visibles.

El texto dentro de la etiqueta **<textarea>** se mostrará inicialmente dentro del área de texto y los usuarios podrán editarla y seleccionarla.

[https://www.w3schools.com/html/html\\_form\\_elements.asp](https://www.w3schools.com/html/html_form_elements.asp)  
[<https://www.w3schools.com/html/html\\_form\\_elements.asp>](https://www.w3schools.com/html/html_form_elements.asp)

## Ejemplo

```
<form action="https://www.w3schools.com/action_page.php" method="post">  
  <fieldset>  
    <legend>Datos personales</legend>  
    <label for="nombre">Nombre y apellidos:</label>  
    <input name="nombre" id="nombre" type="text"><br>  
    <label for="socio">Socio:</label>  
    <select name="socio" id="socio">  
      <!--Etiqueta option-->  
      <option value="value1">Sí<option>  
      <option value="value2" selected>No<option>  
    </select><br><br>  
    <label>Edad:</label>  
    <input name="edad" type="checkbox" value="20-39"> 20-39  
    <input name="edad" type="checkbox" value="40-59"> 40-59  
    <input name="edad" type="checkbox" value="60-79"> 60-79<br>  
    <label for="email">Email:</label>  
    <input id="email" name="email" type="email"><br><br>  
    <label for="estudios">Estudios:</label>  
    <select name="estudios" id="estudios">  
      <optgroup label="Estudios Universitarios">  
        <option>Doctorado</option>  
        <option>Máster</option>  
        <option>Grado</option>  
      </optgroup>  
      <optgroup label="Ciclo Formativo">
```

```

        <option>Grado Superior</option>
        <option>Grado Medio</option>
    </optgroup>
</select><br><br>
<label for="navegador">Elige tu navegador favorito: </label>
<input name="navegador" id="navegador" list="browsers">
<datalist name="browsers" id="browsers">
    <option value="Chrome">
    <option value="Firefox">
    <option value="Internet Explorer">
    <option value="Opera">
    <option value="Safari">
    <option value="Microsoft Edge">
</datalist><br><br>
<label for="mensaje">Escribe tu mensaje: </label>
<textarea rows="10" cols="50" placeholder="Mensaje" id="mensaje">
</fieldset>
<input type="submit" name="enviar" value="Enviar">
</form>

```

En el navegador se mostrará un formulario que se podrá llenar como en el ejemplo siguiente:

Datos personales

Nombre y apellidos: Juan García Pérez

Socio: No

Edad:  20-39  40-59  60-79

Email: juan.garcia@perez.com

Estudios: Grado

Elige tu navegador favorito: Firefox

Hola Mundo.  
Esto es una prueba.

Escribe tu mensaje:

Enviar

La página [www.w3schools.com/action\\_page.php](http://www.w3schools.com/action_page.php) [recibirá todos los datos y podrá tratarlos como el programador desee. En este caso, mostrando los datos enviados:](http://www.w3schools.com/action_page.php)

# Submitted Form Data

Your input was received as:

```
nombre=Juan García Pérez&socio=value2&edad=40-59&
email=juan.garcia@perez.com&estudios=Grado&navegador=Firefox&
mensaje=Hola Mundo. Esto es una prueba.&enviar=Enviar
```

The server has processed your input and returned this answer.

**Note:** This tutorial will not teach you how servers are processing input. Processing input is explained in our [PHP tutorial](#).

## Atributos usados en formularios

Cada etiqueta tiene una serie de atributos propios aunque coinciden un muchos de ellos. Puedes consultar los atributos de INPUT **aquí** <<https://oregoom.com/html/input/>>

Los más comunes son:

### name

El atributo name se utiliza para identificar un campo de entrada en particular. Es especialmente útil cuando se procesan los datos del formulario en el servidor, ya que actúa como clave para los valores ingresados.

```
<input type="text" name="nombre_usuario" placeholder="Ingrese su nombre">
```

### value

El atributo value define el valor predeterminado de un campo de entrada. Este valor se envía al servidor cuando se envía el formulario.

```
<input type="text" name="nombre" value="John Doe">
```

### placeholder

El atributo placeholder muestra un texto de sugerencia en el campo de entrada cuando está vacío. Este texto desaparece cuando el usuario comienza

a escribir en el campo.

```
<input type="text" name="correo" placeholder="Ingrese su correo electrónico">
```

## required

El atributo `required` indica que un campo de entrada es obligatorio y debe completarse antes de enviar el formulario.

```
<input type="text" name="nombre" placeholder="Ingrese su nombre" required>
```

## disabled

El atributo `disabled` desactiva un campo de entrada, lo que significa que los usuarios no pueden interactuar con él ni modificar su valor. Los campos desactivados no se envían con el formulario.

```
<input type="text" name="nombre" value="John Doe" disabled>
```

## readonly

El atributo `readonly` hace que un campo de entrada sea de solo lectura. A diferencia de los campos desactivados, los campos de solo lectura sí se envían con el formulario, pero los usuarios no pueden modificar su valor.

```
<input type="text" name="id_usuario" value="12345" readonly>
```

## id

El atributo `id` asigna un identificador único al elemento

[REDACTED]. Este identificador se puede utilizar para seleccionar y manipular el elemento mediante JavaScript y para aplicar estilos específicos mediante CSS. Los valores de los atributos `id` deben ser únicos en todo el documento.

```
<input type="text" id="nombre_usuario" name="nombre_usuario" placeholder="Ingrese su nombre de usuario">
```

## class

El atributo `class` permite asignar una o varias clases al elemento . Las clases son útiles para aplicar estilos CSS y manipular elementos que comparten características comunes mediante JavaScript. Puedes asignar múltiples clases separándolas con espacios.

```
<input type="text" class="formulario-entrada entrada-nombre"  
name="nombre" placeholder="Ingrese su nombre">
```

## Ejercicio

La Web **juanpalomotologuisayolocomo.es** tiene un formulario en el que, los visitantes a la página, rellenan una información inicial que luego en una segunda página se pide el resto para confeccionar así una receta. El formulario inicial pide la siguiente información:

- Nombre de la receta (llámala **nombre**)
- Correo del autor (llámala **correo**)
- Momento para comer: desayuno, almuerzo, merienda, cena, postre, ... Usa checkboxes para que se puedan elegir varios. (llámalo **momento**)
- Ingredientes necesarios escogidos de una lista entre carne, pescado, verdura y fruta. Se debe poder elegir varios. (llámalo **ingredientes**)
- Tiempo de preparación aproximado elegido de dos radios: uno para las horas de 0 a 2; y otro para los minutos en franjas de 15 minutos. (llámalo **horas** y **minutos**)

Crea un diseño usandofieldset, label, optgroup, etc.

Para probar la información que se envía, usa la página

[https://www.w3schools.com/action\\_page.php](https://www.w3schools.com/action_page.php)

[<https://www.w3schools.com/action\\_page.php>](https://www.w3schools.com/action_page.php)

Para ver cómo se trataría la información en otra página con formulario, usa

<https://lm.ciclo.iesnervion.es/> <<https://lm.ciclo.iesnervion.es/>>

### 3.2.8. Comentarios

En un documento HTML, los comentarios pueden escribirse entre los caracteres “`<!--`” y “`-->`”.

Por ejemplo:

**`<!-- Esto es un comentario escrito en un documento HTML -->`**

Los comentarios no se mostrarán en el navegador donde se visualice el documento HTML.

Sin embargo, al ver el código fuente del documento en un navegador, sí se mostrarán los comentarios.

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.3. Atributos

Los atributos son elementos de configuración de las etiquetas. Muchos de ellos son imprescindibles debido a que, en su ausencia, la etiqueta no realizaría su función correctamente (por ejemplo los enlaces). Otros, en cambio, proporcionan más información a la etiqueta e incluso formato.

Para esto último, es recomendable el uso de CSS por limpieza de código y facilidad en el mantenimiento y modificación de futuras revisiones de código.

Los siguientes atributos son comunes a todas las etiquetas:

Atributo	Acción
title	Su valor se muestra al pasar el ratón por encima
id	Indica el identificador único de un elemento. No debe haber más de un elemento con el mismo identificador
class	Indica un identificador de clase. Se usa para agrupar varios elementos con un identificador común, generalmente, para aplicarles conjuntamente estilos CSS
style	Permite añadir un estilo específico al elemento, con notación CSS
width	Ancho del contenido. Típico de etiquetas contenedoras
height	Alto del contenido. Típico de etiquetas contenedoras

```
1 <p title="Title se muestra como un tooltip">
2   Texto con tooltip.
3 </p>
4 <p id="identificador1">Esto es un párrafo con identificador</p><br>
5 <h1 style="color:blue;">Esto es un título azul</h1>
6 
```

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

## 3.4. Contenedores

Varias etiquetas HTML sirven para “contener” dentro de ellas a otras etiquetas. Se pueden clasificar en dos tipos:

### Contenedores de bloque (blocks)

Se inician en una linea nueva, esto es, cogen todo el ancho que tengan disponible.

Ejemplo:

```
<div>  
<h1> .. <h6>  
<p>
```

Más información: <https://www.mclibre.org/consultar/htmlcss/html/html-bloques.html> <<https://www.mclibre.org/consultar/htmlcss/html/html-bloques.html>>

### Contenedores de línea (inlines)

NO inician una linea nueva. Cogen solo el ancho que necesitan. Ejemplo:

```
<span>  
<a>  
<b>...<i>...<u>...  
<image>
```

Más información:

[https://developer.mozilla.org/es/docs/Web/HTML/Inline\\_elements](https://developer.mozilla.org/es/docs/Web/HTML/Inline_elements)  
<[https://developer.mozilla.org/es/docs/Web/HTML/Inline\\_elements](https://developer.mozilla.org/es/docs/Web/HTML/Inline_elements)>

Etiquetas span y div

Es habitual confundir el uso de span y div. Con lo antedicho, los **div** (tambien se llaman vulgarmente capas o cajas) son apropiados para agrupar elementos (incluso es recomendable su uso), y los **span** se suele usar para aplicar estilos.

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.4.1. La etiqueta DIV

La etiqueta `<div>` es un elemento de bloque en HTML5 que se utiliza para crear un contenedor de contenido en una página web. Esta etiqueta puede contener otros elementos de HTML como texto, imágenes, enlaces, formularios, etc.

La principal función de la etiqueta `<div>` es organizar y agrupar elementos dentro de una página web para darle una estructura clara y estética. Además, se pueden aplicar estilos CSS a un contenedor `<div>` para personalizar su aspecto y comportamiento en la página.

Es importante destacar que la etiqueta `<div>` no tiene un propósito semántico específico, sino que se utiliza como un elemento genérico para agrupar contenido en la página. Por lo tanto, se recomienda utilizar etiquetas específicas para contenido específico, como por ejemplo `<header>`, `<nav>`, `<article>`, `<footer>`, etc.

La sintaxis básica de la etiqueta `<div>` es la siguiente:

```
<div> Contenido a agrupar </div>
```

#### Ejemplo

index.html

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>Ejemplo de Composiciones con Etiquetas DIV</title>
    <meta charset="utf-8">

    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <div id="encabezado">
        <h1>Mi Sitio Web</h1>
        <p>Este es el encabezado de la página.</p>
    </div>

    <div id="contenido">
        <div id="columna-izquierda">
            <h2>Columna Izquierda</h2>
            <p>Contenido de la columna izquierda.</p>
```

```
</div>
<div id="columna-derecha">
    <h2>Columna Derecha</h2>
    <p>Contenido de la columna derecha.</p>
</div>
</div>

<div id="enlaces">
    <h2>Enlaces Relacionados</h2>
    <ul>
        <li><a href="#">Enlace 1</a></li>
        <li><a href="#">Enlace 2</a></li>
        <li><a href="#">Enlace 3</a></li>
    </ul>
</div>

<div id="pie-de-pagina">
    <p>Pie de página de la página web.</p>
</div>
</body>
</html>
```

## style.css

```
/* body */
body {
    font-family: Arial, sans-serif;
}

/* Estilo en bloques */
#encabezado {
    background-color: #333;
    color: #fff;
    padding: 20px;
    text-align: center;
}

#contenido {
    width: 80%;
    margin: 0 auto;
    display: flex;
    flex-wrap: wrap;
    justify-content: space-between;
```

```
}

#columna-izquierda, #columna-derecha {
    width: 48%;
    background-color: #f7f7f7;
    margin-bottom: 20px;
    padding: 10px;
    box-sizing: border-box;
}

/* Estilos para la composición en columnas */
#columna-izquierda {
    background-color: #f2f2f2;
}

#columna-derecha {
    background-color: #e2e2e2;
}

/* Estilos para la composición de elementos relacionados */
#enlaces {
    background-color: #333;
    color: #fff;
    text-align: center;
    padding: 10px;
}

#enlaces ul {
    list-style: none;
    padding: 0;
}

#enlaces li {
    margin-bottom: 5px;
}

#enlaces a {
    color: #fff;
    text-decoration: none;
}

/* Estilos para el pie de página */
#pie-de-pagina {
    background-color: #777;
    color: #fff;
    text-align: center;
```

```
padding: 10px;  
}
```

## Tipos de composiciones

### Composición en bloques

En esta composición, se utilizan etiquetas `<div>` para dividir el contenido en secciones separadas, como encabezado, barra lateral, contenido principal y pie de página. Esto permite una organización estructurada y facilita la aplicación de estilos a cada sección.

Ejemplo:

```
<div id="header">...</div>  
<div id="sidebar">...</div>  
<div id="main-content">...</div>  
<div id="footer">...</div>
```

### Composición en columnas

Las etiquetas `<div>` se usan para crear diseños de múltiples columnas en una página web. Puedes dividir el contenido en columnas y aplicar estilos específicos a cada una.

Ejemplo:

```
<div class="column">  
    <p>Columna 1</p>  
</div>  
<div class="column">  
    <p>Columna 2</p>  
</div>  
<div class="column">  
    <p>Columna 3</p>
```

## Composición de elementos relacionados

Las etiquetas `<div>` pueden utilizarse para agrupar elementos relacionados, como un conjunto de enlaces o botones.

Ejemplo:

```
<div class="links">
  <a href="#">Enlace 1</a>
  <a href="#">Enlace 2</a>
  <a href="#">Enlace 3</a>
</div>
```

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)

### 3.4.2. DIV y CSS

Una vez que tienes una página web realizada con una composición por bloques mediante etiquetas <div>, hay varias cosas que puedes hacer para mejorar y ampliar la funcionalidad de tu sitio web:

- **Añadir contenido y secciones:** Puedes seguir añadiendo más contenido y secciones a tu página web utilizando etiquetas <div>. Crea secciones adicionales para otros temas, servicios o información que deseas presentar en tu sitio.
- **Mejorar el diseño y la estética:** Aplica estilos CSS para mejorar el diseño y la estética de tu página. Asegúrate de que el diseño sea coherente y atractivo, y utiliza colores, fuentes y elementos visuales que se alineen con la identidad de tu sitio.
- **Hacer que la página sea receptiva (responsive):** Asegúrate de que tu página sea compatible con dispositivos móviles y se ajuste adecuadamente a pantallas de diferentes tamaños. Utiliza CSS para crear un diseño adaptable.

Obra publicada con [Licencia Creative Commons Reconocimiento No comercial Compartir igual 4.0](#)