**Aplicación de Tecnologías Web**

Tema n.° 1

Introducción a las tecnologías Web

**Índice**

**Pág.**

[1.1. Conceptos, principios y fundamentos 3](#_Toc161054074)

[1.1.1. ¿Qué es el frontend? 3](#_Toc161054075)

[1.1.2. Conceptos básicos de HTML 5](#_Toc161054076)

[1.2. Elementos y atributos del HTML 6](#_Toc161054077)

[1.2.1. Atributo src 8](#_Toc161054078)

[1.2.2. Atributo href 9](#_Toc161054079)

[1.2.3. Atributos class e id 10](#_Toc161054080)

[1.2.4. Tablas HTML 12](#_Toc161054081)

[1.2.5. Listas HTML 14](#_Toc161054082)

[1.2.6. Formularios 16](#_Toc161054083)

[1.3. Display inline y block 20](#_Toc161054084)

[1.3.1. Display block 20](#_Toc161054085)

[1.3.2. Display inline 20](#_Toc161054086)

[Recursos complementarios 21](#_Toc161054087)

[Referencias 21](#_Toc161054088)

[Autoevaluación 22](#_Toc161054089)

# Conceptos, principios y fundamentos

En el ámbito del desarrollo web, es imprescindible dividir cualquier proyecto en dos componentes claramente definidos: el front-end, también conocido como lado cliente, y el backend, que corresponde al lado servidor. A pesar de su separación, estas dos partes se entrelazan de manera significativa, complementándose mutuamente.

* La interfaz de usuario, también conocida como front-end o "lado del cliente", constituye la porción de una aplicación web que los usuarios experimentan directamente. En términos generales, engloba todos los elementos visibles durante la interacción con la aplicación (Smith, 2018; Brown, 2020).
* El componente backend de una aplicación web asume principalmente la responsabilidad de procesar, adquirir, almacenar y garantizar todos los recursos o datos esenciales para el funcionamiento adecuado de la empresa (Johnson, 2019; García et al., 2021).

La colaboración armoniosa entre el front-end y el backend es esencial para asegurar el correcto funcionamiento de una aplicación web.

Este tópico aborda la definición del front-end y examina las tecnologías fundamentales presentes en esta faceta de las aplicaciones web. Se profundizará en la comprensión de qué es el HTML y su significativa relevancia, así como en la función de CSS.

## ¿Qué es el frontend?

Todas las aplicaciones web se separan en dos partes: el frontend y el backend. Los usuarios únicamente interactúan con el frontend o «lado del cliente» o frontal, ya que es en esta parte donde nos encontramos una gran cantidad de elementos visuales que le indican al usuario cómo usar la aplicación.

A continuación, se presentan algunas definiciones de front-end:

* El front-end, también conocido como "lado del cliente", constituye la interfaz de usuario de una aplicación web, comprendiendo todos los elementos visuales y funcionales con los cuales los usuarios interactúan directamente (Johnson, 2018).
* El front-end engloba el conjunto de tecnologías y prácticas destinadas a la creación y optimización de la experiencia del usuario en una aplicación web, abarcando aspectos visuales, de usabilidad y de interactividad (García et al., 2021).

El mundo del front-end agrupa una gran cantidad de tecnologías y de disciplinas. Al ser la parte del proyecto con la que el usuario interactúa, intervienen una gran cantidad de factores y no todos son necesariamente técnicos.

**Figura 1**

*Tecnologías Front-End*

Un desarrollador de front-end tendrá que tratar, con casi toda seguridad, con las siguientes tecnologías:

* **HTML:** Se trata de un lenguaje de marcas (y no de un lenguaje de programación) que permite al desarrollador definir la estructura de un documento que, posteriormente, será interpretado por un navegador web para presentar contenido al usuario.
* **CSS:** Es un lenguaje de diseño que sirve para añadir estilos a un documento HTML. Con este lenguaje podemos definir o crear el aspecto de un componente HTML.
* **JavaScript:** Se trata de un lenguaje de programación que aporta dinamismo a las aplicaciones web.

Estrechamente ligados al front-end, existen dos amplios conceptos muy importantes:

* **Experiencia de usuario:** Son todos aquellos factores que influyen y conforman la interacción del usuario con la aplicación.
* **Usabilidad:** Es la facilidad con la que los usuarios finales de la aplicación pueden hacer uso de ella y de todas las funcionalidades que ofrece.

## Conceptos básicos de HTML

El HTML (hyper text markup language) es el lenguaje utilizado para describir la estructura de una página web o de parte de ella. Un documento HTML está dividido en varios elementos, los cuales sirven para indicar al navegador web cómo debe mostrar el contenido.

La más reciente, HTML 5, fue publicada el 28 de octubre de 2014. Esta última versión incluye nuevas etiquetas y atributos para poder tratar con la web semántica (o web 3.0), lo que facilita el procesamiento de la información de los sitios web por máquinas y describe su contenido.

Durante toda esta sección se mostrarán diferentes ejemplos de código HTML cuyo resultado final se puede ver a través de un navegador web como Google Chrome o Mozilla Firefox. En este punto, el código que se va a mostrar y desarrollar será sencillo y no es, por tanto, necesario ningún tipo de IDE especializado. Algunas de las aplicaciones que se pueden usar para crear nuestros primeros documentos HTML son:

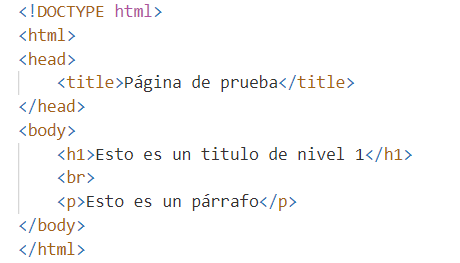
* Visual Studio Code
* Notepad
* Atom
* Sublime text

# Elementos y atributos del HTML

Un ejemplo sencillo de código HTML que posea diferentes elementos de este es el siguiente:

**Figura 2**

*Estructura básica de una página en HTML*



Un elemento HTML se indica por medio de un par de etiquetas de apertura y cierre (estas incluyen un «/» en el texto de la etiqueta) y entre ellas está su contenido.

Cada elemento aporta un significado concreto al documento y tiene una función determinada y diferenciada de la de otros. En el caso del código anterior, observamos los siguientes elementos HTML:

* **<html> .** Es el elemento raíz de cualquier documento HTML y siempre debe estar presente. Sirve para indicar al navegador que todo lo que esté dentro de él debe ser tratado como HTML.
* **<head> .** Este elemento ofrece metadatos acerca del documento HTML. Dentro de este podemos encontrar cosas como el título de la página o las hojas de estilo asociadas a ella.
* **<title> .** Indica el título de la página HTML. Este es el título que se verá en la pestaña del navegador web.
* **<body> .** Este elemento define el cuerpo del documento HTML. Se trata de un contenedor que incluye todo el contenido visible de la página (imágenes, encabezados, párrafos, tablas, etc.).
* **<h1> .** Se utiliza para crear headers o encabezados. Podemos encontrar variantes como h2, h3, h4, h5, h6, etc. Es útil para crear índices del contenido de la página.
* **<p> .** Este elemento define un párrafo de texto.

Existe otro elemento, <br>, que no cumple con la definición antes dada, ya que presenta una etiqueta de apertura, pero no de cierre. Lo cierto es que algunos elementos HTML pueden no tener contenido, como es el caso de esta etiqueta, cuya funcionalidad es incluir un salto de línea, y son denominados elementos vacíos (void elements). Un elemento vacío no precisa de etiqueta de cierre.

Además, al principio del documento, ¡encontramos una declaración de tipo <!DOCTYPE html>. Esta debe estar presente al comienzo de cualquier documento HTML, ya que sirve para indicar al navegador web de qué tipo de documento se trata.

Existen algunas consideraciones sobre los elementos HTML que se deben conocer:

* Aunque algunos elementos HTML se muestren correctamente en el navegador sin etiquetas de cierre, es una buena práctica incluir siempre etiquetas de cierre cuando trabajemos con elementos no vacíos.
* El HTML no es case sensitive (no distingue entre mayúsculas y minúsculas), al menos en lo que a elementos del lenguaje se refiere. Es decir, es lo mismo escribir <p> que <P> . Aquello que vaya dentro del párrafo (texto que mostrar), sí será case sensitive. El W3C recomienda el uso de minúsculas en el HTML, aunque no es obligatorio.

Los elementos se agrupan en categorías, dependiendo de su uso. Por otra parte, los atributos sirven para indicar información adicional sobre el elemento HTML que estamos tratando.

## Atributo src

El src se utiliza para indicar el path hacia un recurso externo al documento HTML. Es muy común ver este atributo junto al elemento HTML img , el cual indica que se debe mostrar una imagen. Un ejemplo sería el siguiente:

|  |
| --- |
| <img src="https://www.espe.edu.ec/wpcontent/uploads/2019/11/espe\_2021\_logo.svg"> |

Nótese que, como img es un elemento vacío, no es estrictamente requerida una etiqueta de cierre. Dentro de src podemos encontrar dos tipos de URL diferentes.

**Figura 3**

*Tipos de URL’s en HTML.*

## Atributo href

Se utiliza para especificar la URL de un enlace. Es muy común ver este atributo junto al elemento HTML a, el cual define un enlace a otra página, ya sea externa o interna. Un ejemplo sería el siguiente:

|  |
| --- |
| <a href="https://www.espe.edu.ec/">Visita ESPE</a> |

* Atributos width y height: se utilizan para especificar la anchura y altura de un elemento HTML, respectivamente. Es muy común ver este atributo junto al elemento HTML img. Un ejemplo sería el siguiente:

|  |
| --- |
| <img src=https://www.espe.edu.ec/wpcontent/uploads/2024/04/espe\_2024\_logo.svg width="250" height="250"> |

Este código haría que el navegador mostrase la imagen modificando sus dimensiones, estableciendo un ancho y un alto de 250 píxeles para la misma.

* Atributo alt: se utiliza junto al elemento HTML img . Este atributo especifica un texto alternativo para la imagen, que se mostrará si la imagen no puede ser cargada por cualquier motivo.

|  |
| --- |
| <img src=https://www.espe.edu.ec/wpcontent/uploads/2019/11/logo.svg width="250" height="250" alt="Logo ESPE"> |

Dado que la URL absoluta indicada no es válida, el navegador no podrá cargar la imagen.

## Atributos class e id

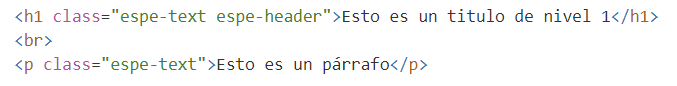
Class se utiliza para indicar que un elemento HTML pertenece a una determinada clase. Múltiples elementos HTML (sean o no del mismo tipo) pueden pertenecer a una misma clase. No debemos pensar en el concepto de clase de otros lenguajes de programación orientados a objetos.

En este contexto, las clases son utilizadas, principalmente, para referirnos a un grupo de elementos desde una hoja de estilos CSS. Este atributo se puede usar en cualquier elemento HTML y el nombre de la clase que indiquemos sí será case sensitive. Además, un elemento HTML puede pertenecer a varias clases a la vez.

Un ejemplo sería el siguiente:

**Figura 4**

*Atributos class e id*



En el ejemplo podemos ver que tanto el elemento <h1> como el elemento <p> pertenecen a la clase espe-text , aun siendo diferentes elementos HTML entre sí. Esto nos indica que, probablemente, comparten algún tipo de estilo o propiedad. Por otro lado, únicamente el elemento <h1> pertenece a la clase espe-header .

Por otra parte, el atributo id se utiliza para asignar un identificador único a un elemento HTML. Debido a esto, no se permite que dos elementos HTML tengan el mismo identificador. De nuevo, y al igual que en el caso anterior, es útil utilizar este identificador en una hoja de estilos de CSS, aunque también JavaScript puede hacer uso de él (y del atributo class ).



Por otro lado, existen dos elementos HTML que nos permiten organizar la información y maquetar el contenido de nuestro documento HTML: las tablas y las listas. Estos dos elementos suelen estar presentes en casi cualquier documento HTML, ya que permiten organizar la información y presentarla de una manera adecuada.

## Tablas HTML

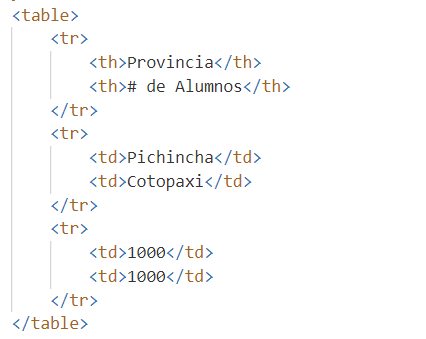
Al igual que en cualquier otro ámbito, están formadas por filas y columnas, lo que forma celdas. Las celdas pueden contener a su vez cualquier contenido HTML. Para definir una tabla en un documento HTML tendremos que hacer uso de las siguientes etiquetas:

* <table> . Indica que comienza la descripción de una tabla en HTML.
* <tr> . (Table row). Señala que comienza una fila de la tabla.
* <th> . (Table header). Indica que una celda es header de una columna.
* <td> . (Table data). Señala que se trata de una celda.

Un ejemplo es el siguiente:

**Figura 5**

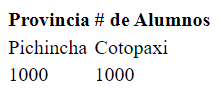
*Tablas HTML*



Este código se corresponde a una tabla que tiene dos columnas, dado que hay dos elementos <th> en la primera fila, que es la cabecera de la tabla. La primera columna contiene provincias en los que está presente ESPE y la segunda tiene la cantidad de alumnos en cada uno de esas provincias (estos datos son inventados). Al no haber aplicado estilos, la tabla se verá de la siguiente forma:

**Figura 6**

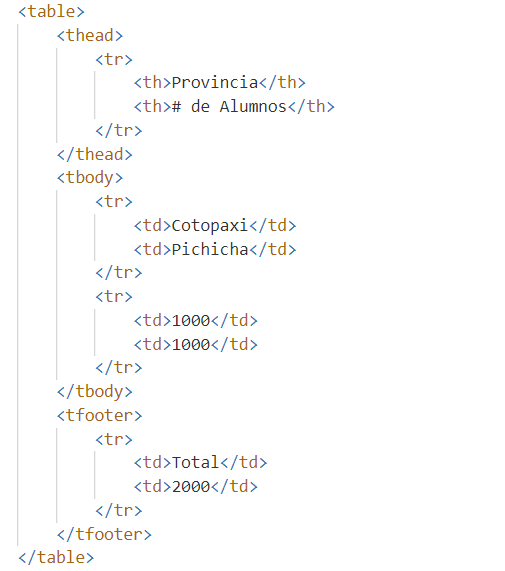
*Tabla sin estilos aplicados*



Para poder diferenciar de una forma más estructural el contenido de la tabla, existen otras tres etiquetas, cuya función es agrupar contenido para el header, el body y el footer. Son <thead> , <tbody> y <tfooter> .

**Figura 7**

*Estructura HTML de una tabla con footer.*



No aportan ninguna diferencia al *layout* de la tabla, salvo que mediante CSS se decida modificar sus estilos.

## Listas HTML

Las listas HTML permiten agrupar una serie de términos relacionados dentro una lista. Las listas pueden ser de **dos tipos.**

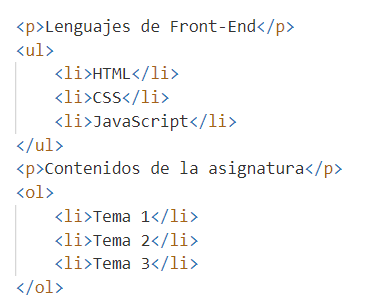
**Figura 8**

Tipos de lista en HTML. Fuente: elaboración propia.

Los elementos que están incluidos en la lista, ya sea ordenada o desordenada, se especifican con la etiqueta <li> (*list* ítem).

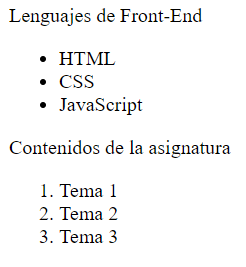
**Figura 9**

Código de las listas en HTML.



**Figura 10**

*Vista de las listas en HTML.*



Las **listas ordenadas** disponen de un atributo HTML llamado t y p e que permite indicar el tipo de numeración de la lista. Los posibles valores de este atributo son:

* type=“1” . Valor por defecto. Numeración decimal (1, 2, 3, etc.).
* type=“A” . Numeración con letras mayúsculas (A, B, C, etc.).
* type=“a” . Numeración con letras minúsculas (a, b, c, etc.).
* type=“I” . Numeración romana en mayúsculas (I, II, III, etc.).
* type=“i” . Numeración romana en minúsculas (i, ii, iii, etc.).

En cuanto a las **listas desordenadas,** también es posible modificar mediante CSS el carácter que se usa para indicar cada elemento de esta.

## Formularios

Por último, estudiaremos uno de los elementos HTML más utilizados: **los formularios.** Los formularios nos permiten indicarle al usuario qué información necesitamos para poder llevar a cabo una acción determinada. En la mayoría de los casos, la información proporcionada será enviada a un servidor web para su procesamiento. En este nivel no añadiremos ninguna funcionalidad a los formularios, ya que nos dedicaremos únicamente a su **diseño estructural.**

El elemento HTML que debemos utilizar para trabajar con los formularios es <form> , el cual se utiliza para crear el formulario. Este estará compuesto por todos los elementos HTML incluidos dentro de este, ya que es un contenedor.

Un formulario contendrá múltiples elementos HTML de formulario. Los **más utilizado**s son:

* <input>. Es el elemento HTML de formularios más utilizado y sirve para almacenar la información que introduce el usuario. Dispone de un atributo name que siempre debe tener valor y se recomienda que coincida con el valor del atributo id. Puede ser de varios tipos, según lo que se indique en su atributo type. Existen varios valores, aunque los más usados son:
* text. Cuadro de texto de una única línea.
* radio. Botón de tipo radio (seleccionar una opción de varias).
* checkbox . Botón de tipo checkbox (seleccionar una o ninguna opción de varias).
* submit . Botón para completar el formulario.
* password . Cuadro de texto de una única línea con formato de contraseña (no se ve el texto en claro).
* <label> . Define una etiqueta visual para una parte del formulario. Tiene un atributo llamado for , cuyo valor debe ser idéntico al del atributo id del elemento HTML <input> al que se quiere enlazar.

El **aspecto visual de los formularios,** al igual que el de cualquier otro elemento HTML, puede ser modificado mediante hojas de estilo CSS. A continuación, veremos un ejemplo de formulario que ilustra lo expuesto.

Se trata de un formulario de registro en una aplicación interna. Para ello es necesario aportar cierta información básica:

**Figura 11**

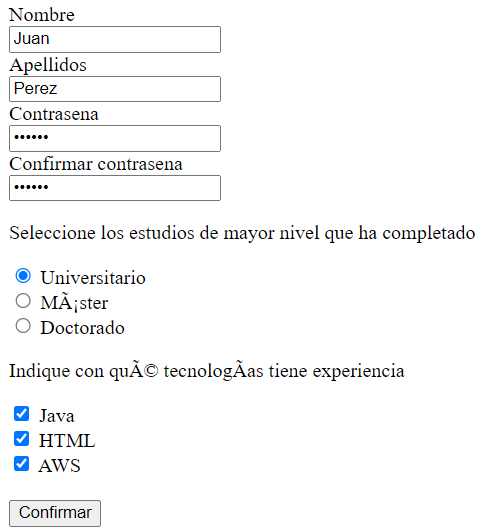
*Ejemplo de un formulario de registro.*



El ejemplo anterior (con datos ya incluidos en el formulario) se muestra así en un navegador web:

**Figura 12**

*Salida del ejemplo de la Figura 11*



Algunas observaciones:

* Varios elementos HTML incluyen un atributo *value* . El valor de este atributo es el que sería enviado al servidor. Por ejemplo, si se seleccionase el botón de tipo radio «Máster Universitario», lo que se enviaría al servidor sería «master».
* Para que un grupo de botones radio sea considerado como tal, todos los botones que se quiera que formen parte del grupo deben tener el mismo valor en su atributo name . En el caso del ejemplo, sería *«estudios»*.

# Display inline y block

Todos los elementos HTML disponen de un modo de visión o *display* por defecto, el cual dependerá del tipo de elemento que sea. Consideramos dos tipos: ***block*** e ***inline.***

## Display block

Los elementos HTML con este tipo de *display* siempre comenzarán en una nueva

línea y abarcarán, por defecto, el ancho máximo disponible de la pantalla. Estos elementos dispondrán siempre de márgenes superiores e inferiores, algo de lo que los elementos *inline* carecen.

Un ejemplo de elemento HTML con *display block* es <div> , aunque hay muchos más como <form> , <table> , <p> o <h1>-<h6> . Los ejemplos de código HTML mostrados anteriormente ilustran lo anterior.

## Display inline

Los elementos HTML con este tipo de *display* nunca comenzarán en una nueva línea, sino que estarán presentes en la misma línea en la que son incluidos, dentro de otros elementos que podrán ser de tipo *block* o también *inline*. Abarcarán el mínimo ancho de pantalla necesario.

Un ejemplo de elemento HTML con *display inline* es <span>, aunque hay muchos más como <a> , <b> , <br> o <label> . < p > Este elemento HTML se muestra con *display block* y < b > esto es un elemento HTML inline </ b ></ p >

# Recursos complementarios

* Manual de referencia de los elementos HTML

<https://www.w3schools.com/tags/>

<https://www.w3schools.com/tags/ref_attributes.asp>

# Referencias

Brown, A. (2020). *Desarrollo web moderno: estrategias y herramientas*. Editorial Digital Solutions.

García, M.; Rodríguez, A. & Pérez, S. (2021). Tecnologías Back-End: un enfoque integral. *Revista de Desarrollo Tecnológico*, 15(2), 45-63. doi:10.1234/rdt.2021.1234567890

Johnson, R. (2018). *Desarrollo de interfaces de usuario: principios y prácticas*. España: Editorial TechPress.

Johnson, R. (2019). Desarrollo de software empresarial: principios y prácticas. Editorial TechPress.

Smith, J. (2018). *Diseño de interfaces de usuario: principios y prácticas*. Editorial TechBooks.

# Autoevaluación

1. **¿Cuál es la etiqueta correcta para crear una lista no ordenada en HTML?**

<ul>

<ol>

<li>

<list>

1. **¿Qué tipo de lenguaje es HTML?**

Lenguaje de programación funcional.

Lenguaje de marcas.

Lenguaje de diseño gráfico.

Lenguaje de programación orientada a objetos.

1. **¿Qué es un elemento HTML vacío (o void element)?**

Un elemento HTML que no tiene ningún atributo, pero sí contenido.

Un elemento HTML que siempre tiene atributos, pero no contenido.

Un elemento HTML que puede o no tener atributos y no tiene contenido. No necesita etiqueta de cierre.

Un elemento HTML que puede o no tener atributos, y no tiene contenido. Necesita una etiqueta de cierre.

1. **¿Qué entidad es la encargada de mantener los estándares de la web y lenguajes como HTML y CSS?**

Google.

The World Wide Web Consortium (W3C).

Amazon Web Services

Facebook

1. **¿Cuál de los siguientes elementos HTML es correcto?**

<a> http://www.misitio.com </a> .

<a href=“ http://www.misitio.com ”> </a> .

<a value=“ http://www.misitio.com ”> </a> .

<a url=“ http://www.misitio.com ”> </a> .

1. **¿Cuál header de los siguientes tiene mayor tamaño?**

h3 .

h1 .

h5 .

h6 .

1. **Un elemento <div> , ¿es de tipo block o inline?**

Block, únicamente si tiene contenido.

Block, independientemente de su contenido.

Inline, independientemente de su contenido.

Inline, únicamente si no tiene contenido.

1. **¿Qué entendemos por front-end?**

La parte de una aplicación web visible únicamente por otras máquinas, no accesible para los usuarios.

La parte de una aplicación web con la que los usuarios interactúan directamente, por eso se la conoce también como el «lado del cliente».

Unos diseños en papel que dejan ver la apariencia que tendrá la aplicación.

Un prototipo de alta fidelidad.

1. **¿Cuál de los siguientes elementos se utiliza para definir encabezados en un documento HTML?**

<head>

<title>

<header>

<h1>

1. **¿Cuál de los siguientes elementos HTML se utiliza para definir una lista ordenada?**

<ul>

<li>

<ol>

<dl>