

HTML

HTML (*HyperText Markup Language*)



Desarrollador

- [World Wide Web Consortium](#)
 - [WHATWG](#)

Información general

<u>Extensión de archivo</u>	.html .htm
<u>Tipo de MIME</u>	text/html
<u>Type code</u>	TEXT
<u>Uniform Type Identifier</u>	public.html
Tipo de formato	<u>Lenguaje de marcado</u>
Extendido de	<u>SGML</u>
Extendido a	<u>XHTML</u>
<u>Estándar(es)</u>	ISO/IEC 15445 <u>W3C HTML 4.01</u> <u>W3C HTML5(borrador)</u>
<u>Formato abierto</u>	?

[\[editar datos en Wikidata\]](#)

HTML, sigla en inglés de **HyperText Markup Language** (lenguaje de marcas de hipertexto), hace referencia al [lenguaje de marcado](#) para la elaboración de [páginas web](#). Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del [World Wide Web Consortium \(W3C\)](#) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. Se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la [World Wide Web](#) (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.¹

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, [script](#), entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, [teléfonos inteligentes](#), [tabletas](#), [vipers](#) etc. No obstante, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática ([Firefox](#), [Chrome](#)) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial ([Internet Explorer](#)). Por lo que un navegador desactualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML. Así mismo, las páginas escritas en una versión anterior de HTML deberían ser actualizadas o reescritas, lo que no siempre se cumple. Es por ello que ciertos navegadores todavía mantienen la capacidad de interpretar páginas web de versiones HTML anteriores. Por estas razones, todavía existen diferencias entre distintos navegadores y versiones al interpretar una misma página web.

Índice

[\[ocultar\]](#)

- [1Primeras especificaciones de HTML](#)
- [2Marcador HTML](#)
 - [2.1Elementos](#)
 - [2.2Atributos](#)
- [3Códigos HTML básicos](#)
- [4Colores para HTML](#)
- [5Nociones básicas de HTML](#)
- [6Aprender HTML analizando páginas reales](#)
- [7Historia del estándar](#)
- [8Accesibilidad web](#)
- [9Entidades HTML](#)
- [10Véase también](#)
- [11Referencias](#)
- [12Bibliografía](#)
- [13Enlaces externos](#)

Primeras especificaciones de HTML^[editar]

[Tim Berners-Lee](#) (TBL) en 1991^{2 3} describe 18 elementos que incluyen el diseño inicial y relativamente simple de HTML. Trece de estos elementos todavía existen en HTML 4.⁴

Berners-Lee consideraba el HTML una ampliación de [SGML](#), pero no fue formalmente reconocida como tal hasta la publicación a mediados de 1993, por la [IETF](#) (en español: Grupo de Trabajo de Ingeniería de Internet), de una primera proposición para una especificación del HTML: el borrador del *Hypertext Markup Language* de Berners-Lee y [Dan Connolly](#), que incluía una [Definición de Tipo de Documento](#) SGML para definir la gramática.⁵ El borrador expiró a los seis meses, pero fue notable por su reconocimiento de la etiqueta propia del navegador [Mosaic](#) usada para insertar imágenes sin cambio de línea, que reflejaba la filosofía del IETF de basar estándares en prototipos con éxito.⁶ De la misma manera, el borrador competidor de Dave Raggett *HTML+ (Hypertext Markup Format)* (Formato de Marcaje de Hipertexto), de finales de 1993, sugería estandarizar características ya implementadas, como las tablas.⁷

Marcador HTML^[editar]

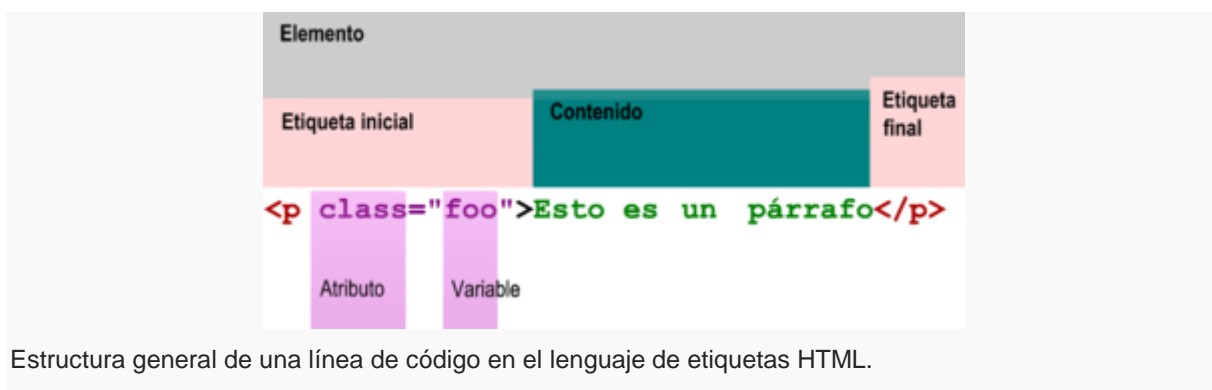
El HTML se escribe en forma de «etiquetas», rodeadas por [corchetes angulares](#) (<, >, /). El HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir o hacer referencia a un tipo de programa llamado [script](#), el cual puede afectar el comportamiento de [navegadores web](#) y otros procesadores de HTML.⁸

HTML también sirve para referirse al contenido del [tipo de MIME](#) text/html o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del [XML](#) (como [XHTML](#) 1.0 y posteriores) o en forma descendida directamente de [SGML](#) (como HTML 4.01 y anteriores).

HTML consta de varios componentes vitales, entre ellos los *elementos* y sus *atributos*, *tipos de data* y la *declaración de tipo de documento*.

Elementos^[editar]

Los elementos son la estructura básica de HTML. Los elementos tienen dos propiedades básicas: atributos y contenido. Cada atributo y contenido tiene ciertas restricciones para que se considere válido al documento HTML. Un elemento generalmente tiene una etiqueta de inicio (por ejemplo, <nombre-de-elemento>) y una etiqueta de cierre (por ejemplo, </nombre-de-elemento>). Los atributos del elemento están contenidos en la etiqueta de inicio y el contenido está ubicado entre las dos etiquetas (por ejemplo, <nombre-de-elemento atributo="valor">Contenido</nombre-de-elemento>). Algunos elementos, tales como
, no tienen contenido ni llevan una etiqueta de cierre. Debajo se listan varios tipos de elementos de marcado usados en HTML.



Estructura general de una línea de código en el lenguaje de etiquetas HTML.

El marcado *estructural* describe el propósito del texto. Por ejemplo, `<h2>Golf</h2>` establece «Golf» como un encabezamiento de segundo nivel, el cual se mostraría en un navegador de una manera similar al título «Marcado HTML» al principio de esta sección. El marcado estructural no define cómo se verá el elemento, pero la mayoría de los navegadores web han estandarizado el

formato de los elementos. Puede aplicarse un formato específico al texto por medio de [hojas de estilo en cascada](#).

El marcado *presentacional* describe la apariencia del texto, sin importar su función. Por ejemplo, `negrita` indica que los navegadores web visuales deben mostrar el texto en **negrita**, pero no indica qué deben hacer los navegadores web que muestran el contenido de otra manera (por ejemplo, los que leen el texto en voz alta). En el caso de `negrita` e `<i>itálica</i>`, existen elementos que se ven de la misma manera pero tienen una naturaleza más semántica: `énfasis fuerte` y `énfasis`. Es fácil ver cómo un [lector de pantalla](#) debería interpretar estos dos elementos. Sin embargo, son equivalentes a sus correspondientes elementos presentacionales: un lector de pantalla no debería decir más fuerte el nombre de un libro, aunque el nombre resalte en *itálicas* en una pantalla. La mayoría del marcado presentacional ha sido desechada con HTML 4.0, en favor de hojas de estilo en cascada.

El marcado *hipertextual* se utiliza para enlazar partes del documento con otros documentos o con otras partes del mismo documento. Para crear un enlace es necesario utilizar la etiqueta de ancla `<a>` junto con el atributo `href`, que establecerá la dirección URL a la que apunta el enlace. Por ejemplo, un enlace que muestre el texto de la dirección y vaya hacia nuestra Wikipedia podría ser de la forma `http://www.wikipedia.org`. También se pueden crear enlaces sobre otros objetos, tales como imágenes ``.

Atributos[\[editar\]](#)

La mayoría de los atributos de un elemento son pares nombre-valor, separados por un signo de igual «=» y escritos en la etiqueta de comienzo de un elemento, después del nombre del elemento. El valor puede estar rodeado por comillas dobles o simples, aunque ciertos tipos de valores pueden estar sin comillas en HTML (pero no en XHTML).^{9 10} De todas maneras, dejar los valores sin comillas es considerado poco seguro.¹¹ En contraste con los pares nombre-elemento, hay algunos atributos que afectan al elemento simplemente por su presencia¹² (tal como el atributo `ismap` para el elemento `img`).¹³

Códigos HTML básicos[\[editar\]](#)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Example</title>
5     <link rel="stylesheet" href="styl
6   </head>
7   <body>
8     <h1>
9       <a href="/">Header</a>
10    </h1>
11    <nav>
12      <a href="one/">One</a>
13      <a href="two/">Two</a>
14      <a href="three/">Three</a>
15    </nav>
```

Un ejemplo de código HTML con [coloreado de sintaxis](#).

- `<html>`: define el inicio del documento HTML, le indica al [navegador](#) que lo que viene a continuación debe ser interpretado como código HTML. Esto es así *de facto*, ya que en teoría lo que define el tipo de documento es el [DOCTYPE](#), que significa la palabra justo tras DOCTYPE el tag de raíz.

- `<script>`: incrusta un [script](#) en una web, o llama a uno mediante `src="url del script"`. Se recomienda incluir el [tipo MIME](#) en el atributo `type`, en el caso de [JavaScript](#) `text/javascript`.
- `<head>`: define la [cabecera](#) del documento HTML; esta cabecera suele contener información sobre el documento que no se muestra directamente al [usuario](#) como, por ejemplo, el título de la ventana del navegador. Dentro de la cabecera `<head>` es posible encontrar:
 - `<title>`: define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana.
 - `<link>`: para vincular el sitio a [hojas de estilo](#) o iconos. Por ejemplo: `<link rel="stylesheet" href="/style.css" type="text/css">`.
 - `<style>`: para colocar el estilo interno de la página; ya sea usando [CSS](#) u otros lenguajes similares. No es necesario colocarlo si se va a vincular a un archivo externo usando la etiqueta `<link>`.
 - `<meta>`: para metadatos como la autoría o la licencia, incluso para indicar parámetros http (mediante `http-equiv=""`) cuando no se pueden modificar por no estar disponible la configuración o por dificultades con server-side scripting.
- `<body>`: define el contenido principal o cuerpo del documento. Esta es la parte del documento html que se muestra en el navegador; dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como [color](#) de fondo y márgenes. Dentro del cuerpo `<body>` es posible encontrar numerosas etiquetas. A continuación se indican algunas a modo de ejemplo:
 - `<h1>` a `<h6>`: encabezados o títulos del documento con diferente relevancia.
 - `<table>`: define una tabla.
 - `<tr>`: fila de una tabla.
 - `<td>`: celda de una tabla (debe estar dentro de una fila).
 - `<a>`: [hipervínculo](#) o enlace, dentro o fuera del [sitio web](#). Debe definirse el parámetro de pasada por medio del atributo `href`. Por ejemplo: `Ejemplo` se representa como [ejemplo](#).¹⁴
 - `<div>`: división de la página. Se recomienda, junto con css, en vez de `<table>` cuando se desea alinear contenido.
 - ``: imagen. Requiere del atributo `src`, que indica la ruta en la que se encuentra la imagen. Por ejemplo: ``. Es conveniente, por accesibilidad, poner un atributo `alt="texto alternativo"`.
 - ``````: etiquetas para listas.
 - ``: texto en negrita (*etiqueta desaprobada. Se recomienda usar la etiqueta ``*).
 - `<i>`: texto en cursiva (*etiqueta desaprobada. Se recomienda usar la etiqueta ``*).
 - `<s>`: texto tachado (*etiqueta desaprobada. Se recomienda usar la etiqueta ``*).
 - `<u>`: Antes texto subrayado. A partir de HTML 5 define porciones de texto diferenciadas o destacadas del resto, para indicar correcciones por ejemplo (etiqueta desaprobada en HTML 4.01 y redefinida en HTML 5).^{15 16}

La mayoría de etiquetas deben cerrarse como se abren, pero con una barra (`</>`) tal como se muestra en los siguientes ejemplos:

- `<table><tr><td>Contenido de una celda</td></tr></table>`.
- `<script>Código de un script integrado en la página</script>`

Colores para HTML[\[editar\]](#)

Hay muchas formas de ponerle color a una página web con el lenguaje HTML, una de ellas es poner el siguiente código: `bgcolor="red"` y el fondo se hará de color rojo. Otro caso sería con la letra, veamos un ejemplo: **Texto de prueba** y así la letra se pondrá de color rojo.

Nociones básicas de HTML[[editar](#)]

El lenguaje HTML puede ser creado y editado con cualquier [editor de textos](#) básico, como puede ser [Gedit](#) en [GNU/Linux](#), el [Bloc de notas](#) de Windows, o cualquier otro editor que admita texto sin formato como [GNU Emacs](#), [Microsoft Wordpad](#), [TextPad](#), [Vim](#), [Notepad++](#), entre otros.

Existen, además, otros editores para la realización de sitios web con características [WYSIWYG](#) (*What You See Is What You Get*, o en español: «lo que ves es lo que obtienes»). Estos editores permiten ver el resultado de lo que se está editando en tiempo real, a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar sitios web, sino que una forma un tanto más simple, ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML, la cual va generando todo el código a medida que se va trabajando. Algunos ejemplos de editores WYSIWYG son [KompoZer](#), [Microsoft FrontPage](#) o [Adobe Dreamweaver](#).

Combinar estos dos métodos resulta muy interesante, ya que de alguna manera se ayudan entre sí. Por ejemplo, si se edita todo en HTML y el desarrollador olvida algún código o etiqueta, basta con dirigirse al editor visual o WYSIWYG y se continúa ahí la edición o viceversa, ya que hay casos en que resulta más rápido y fácil escribir directamente el código de alguna característica que el usuario desea adherir al sitio que buscar la opción en el programa mismo.

Existe otro tipo de editores HTML llamados [WYSIWYM](#) que dan más importancia al contenido y al significado que a la apariencia visual. Entre los objetivos que tienen estos editores es la separación del contenido y la presentación, fundamental en el diseño web.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma en la que debe aparecer en su [navegador](#) el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador.

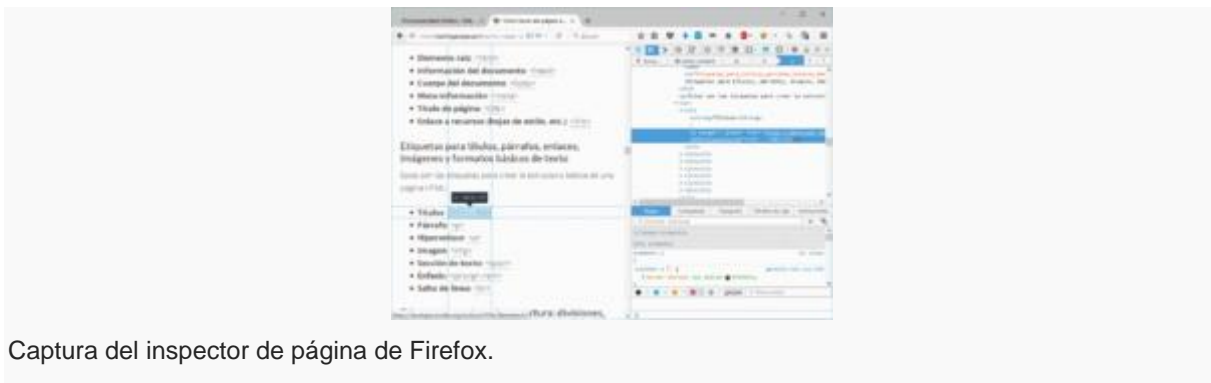
Toda etiqueta se identifica porque está encerrada entre los signos menor que y mayor que (<>), y algunas tienen atributos que pueden tomar algún valor. En general las etiquetas se aplicarán de dos formas especiales:

- Se abren y se cierran, como por ejemplo: negrita, que se vería en su [navegador web](#) como **negrita**.
- No pueden abrirse y cerrarse, como <hr />, que se vería en su [navegador web](#) como una línea horizontal.
- Otras que pueden abrirse y cerrarse, como por ejemplo <p>.
- Las etiquetas básicas o mínimas son:

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Ejemplo1</title>
  </head>
  <body>
    <p>Párrafo de ejemplo</p>
  </body>
</html>
```

Aprender HTML analizando páginas reales[[editar](#)]

Seleccionando la opción «ver código fuente» en el navegador, se puede ver realmente la información que está recibiendo el [navegador web](#) y cómo la está interpretando.



Captura del inspector de página de Firefox.

Por ejemplo: en [Internet Explorer](#) o en [Firefox](#), simplemente hay que desplegar el menú «ver» y luego elegir «código fuente», mientras que en [Chrome](#) presionar Ctrl+U. De esta forma, se abrirá el editor de texto configurado como predeterminado en el sistema con el código fuente de la página que se esté viendo en ese momento en el explorador. Otra forma más rápida consiste en hacer clic con el botón derecho del ratón en cualquier punto del área donde el navegador muestra la página web y elegir «ver código fuente».

Aparte de poder ver el código fuente HTML de una página web con las opciones antes descritas, Internet Explorer, Firefox y Google Chrome incorporan también unas herramientas conocidas como inspectores de página que se puede activar con F12.

Con estas herramientas es posible visualizar una página web y seleccionar dentro de ella un elemento concreto del cuál queremos conocer cuál es el código HTML con el que está hecho señalando el elemento en cuestión simplemente con el ratón. Al hacer esto, el código se mostrará en un área especial dentro del navegador en el que el usuario podrá ver el código HTML en cuestión (ver imagen), además, de las reglas [CSS](#) que aplican a ese código HTML en concreto. Este tipo de análisis resulta sumamente instructivo para aprender a programar en HTML.

Para el navegador Firefox, además, existe como alternativa a la herramienta nativa el plugin [Firebug](#), muy similar a la herramienta que Firefox incorpora por defecto.

Historia del estándar^[editar]

En 1989 existían dos técnicas que permitían vincular documentos electrónicos, por un lado los hipervínculos o enlaces (*hyperlinks* o *links*) y por otro lado un poderoso lenguaje de etiquetas denominado [SGML](#). Por entonces, [Tim Berners-Lee](#), quien trabajaba en el Centro Europeo de Investigaciones Nucleares [CERN](#) da a conocer a la prensa que estaba trabajando en un sistema que permitirá acceder a ficheros en línea que funcionaba sobre redes de computadoras o máquinas electrónicas basadas en el protocolo [TCP/IP](#). Inicialmente fue desarrollado para que se pueda compartir fácilmente información entre científicos de distintas universidades e institutos de investigación de todo el mundo.¹⁷

A principios de [1990](#), define por fin el HTML como un subconjunto del conocido [SGML](#) y crea algo más valioso incluso, el [World Wide Web](#).

[Tim Berners-Lee](#) creó el proyecto [World Wide Web](#) (Tejido o Telaraña Mundial), así como un sistema que facilitaba la lectura de información, mediante un programa de navegación. Sería el primer navegador web, llamado WorldWideWeb, y desarrollado durante la segunda mitad del año 1990; siendo tiempo después rebautizado como Nexus para evitar confusiones por su nombre que era igual al de la tecnología que representaba. Le siguieron otros dos navegadores: el Line Mode Browser y el [ViolaWWW](#). Este último, desarrollado en 1992, fue el primer navegador en popularizarse entre los primeros usuarios de la World Wide Web.¹⁸

[Pei-Yuan Wei](#) presentó el [ViolaWWW](#),¹⁹ que funcionaría en modo texto y sobre un sistema operativo [UNIX](#).

Los trabajos para crear un sucesor del HTML, denominado [HTML +](#), comenzaron a finales de [1993](#). HTML+ se diseñó originalmente para ser un superconjunto del HTML que permitiera

evolucionar gradualmente desde el formato HTML anterior. A la primera especificación formal de HTML+ se le dio, por lo tanto, el número de versión 2 para distinguirla de las propuestas no oficiales previas. Los trabajos sobre HTML+ continuaron, pero nunca se convirtió en un estándar, a pesar de ser la base formalmente más parecida al aspecto compositivo de las especificaciones actuales.

El borrador del [estándar](#) HTML 3.0 fue propuesto por el recién formado [W3C](#) en marzo de [1995](#). Con él se introdujeron muchas nuevas capacidades; por ejemplo, facilidades para crear tablas, hacer que el texto fluyese alrededor de las figuras y mostrar elementos matemáticos complejos. Aunque se diseñó para ser compatible con [HTML 2.0](#), era demasiado complejo para ser implementado con la tecnología de la época, y cuando el borrador del estándar expiró en septiembre de [1995](#), se abandonó debido a la carencia de apoyos de los fabricantes de [navegadores web](#). El [HTML 3.1](#) nunca llegó a ser propuesto oficialmente, y el estándar siguiente fue el [HTML 3.2](#), que abandonaba la mayoría de las nuevas características del [HTML 3.0](#) y, a cambio, adoptaba muchos elementos desarrollados inicialmente por los navegadores web [Netscape](#) y [Mosaic](#). La posibilidad de trabajar con fórmulas matemáticas que se había propuesto en el HTML 3.0 pasó a quedar integrada en un estándar distinto llamado [MathML](#).

En 1997, [HTML 4.0](#) se publicó como una recomendación del [W3C](#). [HTML 4.0](#) adoptó muchos elementos específicos desarrollados inicialmente para un navegador web concreto, pero al mismo tiempo comenzó a limpiar el HTML señalando algunos de ellos como «desaprobados» (*deprecated*, en inglés).

[HTML 4.0](#) implementa características como [XForms](#) 1.0 que no necesitan implementar motores de navegación que eran incompatibles con algunas páginas web HTML. En 2004 la [W3C](#) reabrió el debate de la evolución del HTML, y se dieron a conocer las bases para la versión [HTML5](#). No obstante, este trabajo fue rechazado por los miembros del [W3C](#) y se daría preferencia al desarrollo del XML.

[Apple](#), [Mozilla](#) y [Opera](#) anunciaron su interés en seguir trabajando en el proyecto bajo el nombre de [WHATWG](#),²⁰ que se basa en la compatibilidad con tecnologías anteriores.

En 2006, el [W3C](#) se interesó en el desarrollo de [HTML5](#), y en 2007 se unió al grupo de trabajo del [WHATWG](#) para unificar el proyecto.

Accesibilidad web^{[[editar](#)]}

Artículo principal: [Accesibilidad web](#)

El diseño en HTML, aparte de cumplir con las especificaciones propias del lenguaje, debe respetar ciertos criterios de [accesibilidad web](#), siguiendo unas pautas o las normativas y leyes vigentes en los países donde se regule dicho concepto. Se encuentra disponible y desarrollado por el [W3C](#) a través de las Pautas de Accesibilidad al Contenido Web 1.0 WCAG (actualizadas recientemente con la especificación 2.0),²¹ aunque muchos países tienen especificaciones propias, como es el caso de España con la Norma UNE 139803.²²

Entidades HTML^{[[editar](#)]}

Los caracteres especiales como signo de puntuación, letras con tilde o diéresis o símbolos de escritura del lenguaje se deben convertir en entidad HTML para mostrarse en un navegador. La siguiente es una lista de caracteres españoles y su correspondiente entidad HTML:

Carácter	Entidad HTML	Carácter	Entidad HTML
á	á	Á	Á

é	<code>&eacute;</code>	É	<code>&Eacute;</code>
í	<code>&iacute;</code>	Í	<code>&Iacute;</code>
ó	<code>&oacute;</code>	Ó	<code>&Oacute;</code>
ú	<code>&uacute;</code>	Ú	<code>&Uacute;</code>
ü	<code>&uuml;</code>	Ü	<code>&Uuml;</code>
ñ	<code>&ntilde;</code>	Ñ	<code>&Ntilde;</code>
¿	<code>&iexcl;</code>	¿	<code>&iquest;</code>

Véase también[\[editar\]](#)
