

Material Imprimible

Curso de CSS para desarrollo Web

Módulo 1: Introducción a CSS

Contenidos:

- Qué es y para qué sirve CSS
- Propiedades CSS Sintaxis CSS
- Selector Id. # Selector class. Selector Universal
- Insertar hojas de Estilos - CSS Externo

Introducción a CSS

- CSS es un lenguaje que describe el estilo de un documento HTML.
- CSS describe cómo deben mostrarse los elementos HTML.
- CSS significa C ascading S tyle S heets (Hojas de estilos en cascada)
- CSS describe cómo se deben mostrar los elementos HTML en la pantalla, papel o en otros medios
- CSS ahorra mucho trabajo. Puede controlar el diseño de varias páginas web a la vez
- Las hojas de estilo externas se almacenan en archivos .CSS

¿Qué es CSS?

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "*documentos semánticos*"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para *marcar* los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

Breve historia de CSS

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores.

El organismo [W3C](#) (World Wide Web Consortium), encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron nueve propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (*Cascading HTML Style Sheets*) y la SSP (*Stream-based Style Sheet Proposal*).

La propuesta CHSS fue realizada por Håkon Wium Lie y SSP fue propuesto por Bert Bos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (*Cascading Style Sheets*).

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como "CSS nivel 1".

A principios de 1997, el W3C decide separar los trabajos del grupo de HTML en tres secciones: el grupo de trabajo de HTML, el grupo de trabajo de DOM y el grupo de trabajo de CSS.

El 12 de Mayo de 1998, el grupo de trabajo de CSS publica su segunda recomendación oficial, conocida como "CSS nivel 2". La versión de CSS que utilizan todos los navegadores de hoy en día es CSS 2.1, una revisión de CSS 2 que aún se está elaborando (la última actualización es del 8 de septiembre de 2009). Al mismo tiempo, la siguiente recomendación de CSS, conocida como "CSS nivel 3", continúa en desarrollo desde 1998 y hasta el momento sólo se han publicado borradores.

La adopción de CSS por parte de los navegadores ha requerido un largo periodo de tiempo. El mismo año que se publicó CSS 1, Microsoft lanzaba su navegador Internet Explorer 3.0, que disponía de un soporte bastante reducido de CSS. El primer navegador con soporte completo de CSS 1 fue la versión para Mac de Internet Explorer 5, que se publicó en el año 2000. Por el momento, ningún navegador tiene soporte completo de CSS 2.1.

Soporte de CSS en los navegadores

El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

Internamente los navegadores están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor. Desde el punto de vista del diseñador CSS, la versión de un motor es mucho más importante que la versión del propio navegador.

La siguiente tabla muestra el soporte de CSS 1, CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los usuarios:

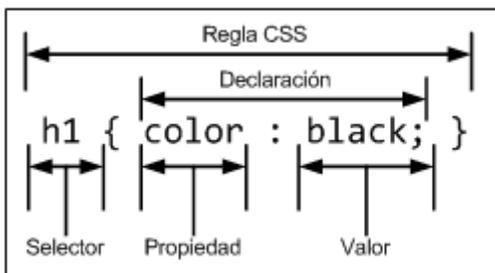
Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Google Chrome	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Internet Explorer	Trident	Completo desde la versión 7.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades a partir de la versión 10.0 del navegador
Firefox	Gecko	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Safari	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades

Opera	Presto	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades
-------	--------	---	----------	--

Los navegadores Firefox, Chrome, Safari y Opera son los más avanzados en el soporte de CSS, ya que incluyen muchos elementos

Glosario básico

CSS define una serie de términos que permiten describir cada una de las partes que componen los estilos CSS. El siguiente esquema muestra las partes que forman un estilo CSS muy básico:



Componentes de un estilo CSS básico

Los diferentes términos se definen a continuación:

- **Regla:** cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de "selectores", un símbolo de "llave de apertura" (`{`), otra parte denominada "declaración" y por último, un símbolo de "llave de cierre" (`}`).
- **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.
- **Declaración:** especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.
- **Propiedad:** característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
- **Valor:** establece el nuevo valor de la característica modificada en el elemento.

Un archivo CSS puede contener un número ilimitado de reglas CSS, cada regla se puede aplicar a varios selectores diferentes y cada declaración puede incluir tantos pares propiedad/valor como se desee.

El estándar CSS 2.1 define 115 propiedades, cada una con su propia lista de valores permitidos. Por su parte, los últimos borradores del estándar CSS 3 ya incluyen 239 propiedades.

Sintaxis CSS

Un conjunto de reglas CSS consta de un selector y un bloque de declaración:

El selector apunta al elemento HTML que desea diseñar.

El bloque de declaración contiene una o más declaraciones separadas por punto y coma. Cada declaración incluye un nombre de propiedad CSS y un valor, separados por dos puntos.

Una declaración CSS siempre termina con un punto y coma, y los bloques de declaración están rodeados por llaves.

Selectores

Para crear diseños web profesionales, es imprescindible conocer y dominar los selectores de CSS. Una regla de CSS está formada por una parte llamada "selector" y otra parte llamada "declaración".

La declaración indica *"qué hay que hacer"* y el selector indica *"a quién hay que hacérselo"*. Por lo tanto, los selectores son imprescindibles para aplicar de forma correcta los estilos CSS en una página.

A un mismo elemento HTML se le pueden aplicar varias reglas CSS y cada regla CSS puede aplicarse a un número ilimitado de elementos. En otras palabras, una misma regla puede aplicarse sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

El estándar de CSS 2.1 incluye una docena de tipos diferentes de selectores, que permiten seleccionar de forma muy precisa elementos individuales o conjuntos de elementos dentro de una página web.

No obstante, la mayoría de páginas de los sitios web se pueden diseñar utilizando solamente los cinco selectores básicos.

El selector de elementos CSS

El selector de elementos selecciona elementos HTML en función del nombre del elemento

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p>Todo el párrafo será afectado por el estilo. </p>
<p id="para1">Acá tambien!</p>
<p>Y acá!</p>

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

El selector de id. De CSS

El selector de id utiliza el atributo id de un elemento HTML para seleccionar un elemento específico.

La identificación de un elemento es única dentro de una página, por lo que el selector de identificación se usa para seleccionar un elemento único.

Para seleccionar un elemento con una identificación específica, escriba un carácter hash (#), seguido de la identificación del elemento.

```
<!DOCTYPE html>
<html>
<head>
```

```
<style>
#para1 {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<p id="para1">Hola Mundo!</p>
<p>Este párrafo no contiene efectos de estilo.</p>

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

El selector de clase CSS

El selector de clase selecciona elementos HTML con un atributo de clase específico. Para seleccionar elementos con una clase específica, escriba un carácter de punto (.), seguido del nombre de la clase.

```
<!DOCTYPE html>
<html>
<head>
<style>
.center {
  text-align: center;
  color: red;
}
</style>
</head>
<body>

<h1 class="center">Rojo y alineación centrado de encabezado</h1>
<p class="center">Rojo y alineación centrado del párrafo.</p>
```

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

El selector universal de CSS

El selector universal se indica mediante un asterisco (*). Se utiliza para seleccionar todos los elementos HTML de la página.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
* {  
  text-align: center;  
  color: blue;  
}  
</style>  
</head>  
<body>
```

```
<h1>Hola Mundo!</h1>
```

```
<p>Todos los elementos de la página tendrán efectos del estilo.</p>
```

```
<p id="para1">Acá también!</p>
```

```
<p>Y Acá!</p>
```

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

```
////////////////////////////////////
```

El selector de agrupación CSS

El selector de agrupación selecciona todos los elementos HTML con las mismas definiciones de estilo.

Mire el siguiente código CSS (los elementos h1, h2 y p tienen las mismas definiciones de estilo):

```
h1 {  
  text-align: center;  
  color: red;  
}
```

```
h2 {  
  text-align: center;  
  color: red;  
}
```

```
p {  
  text-align: center;  
  color: red;  
}
```

Será mejor agrupar los selectores para minimizar el código.
Para agrupar selectores, separe cada selector con una coma.

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

Selector	Ejemplo	Descripción
<u>.class</u>	.intro	Selecciona todos los elementos con la class="intro"
<u>#id</u>	#nombre	Selecciona todos los elementos con id="nombre"

<u>*</u>	*	Selecciona todos los elementos
<u>element</u>	p	Selecciona todos los elementos <p>
<u>element,element...</u>	div, p	Selecciona todos los elementos <div> y todos los elementos <p>

Tres formas de insertar CSS

Hay tres formas de insertar una hoja de estilo:

- CSS externo
- CSS interno
- CSS en línea

CSS externo

Con una hoja de estilo externa, puede cambiar el aspecto de un sitio web completo cambiando solo un archivo.

Cada página HTML debe incluir una referencia al archivo de hoja de estilo externo dentro del elemento <link>, dentro de la sección de encabezado (HEAD).

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>Esto es un encabezado</h1>
<p>Esto es un párrafo<p>
```

```
</body>
```

```
</html>
```

Se puede escribir una hoja de estilo externa en cualquier editor de texto y se debe guardar con una extensión .css.

El archivo externo .css no debe contener ninguna etiqueta HTML.

```
body {  
  background-color: lightblue;  
}
```

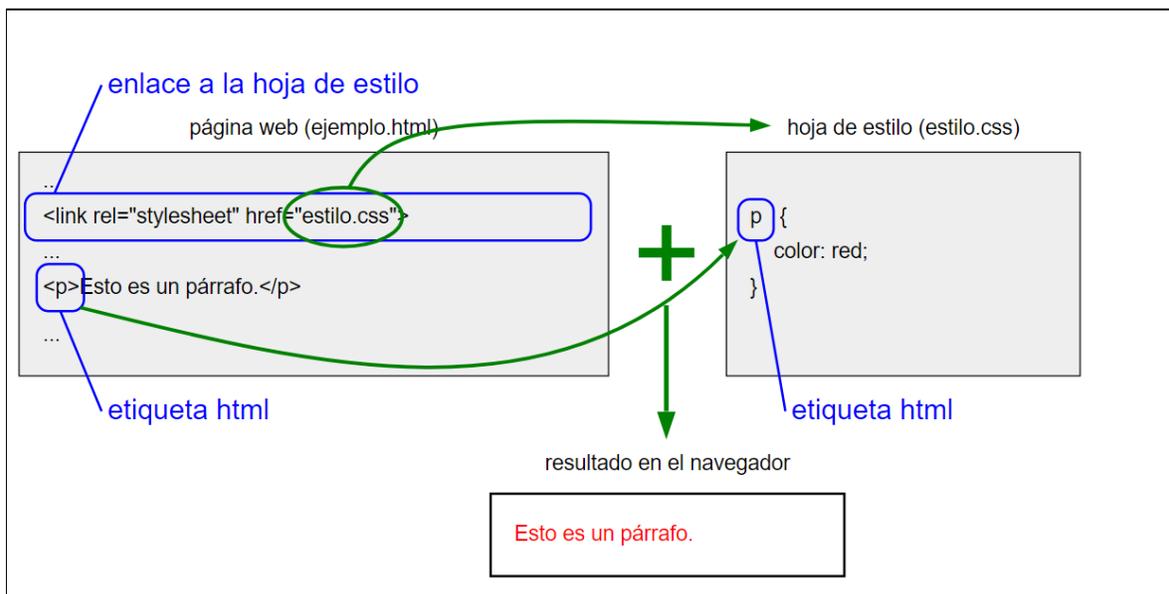
```
h1 {  
  color: navy;  
  margin-left: 20px;  
}
```

Qué es una hoja de estilo

En las primeras versiones del HTML, el código fuente de una página web contenía tanto la información (el contenido) como la forma de representarse (el diseño o formato). Actualmente, estos dos aspectos se pueden separar. La página web (el documento html) sólo debe contener información, mientras que el formato se debe definir en las llamadas hojas de estilo (en inglés, CSS, Cascading Style Sheets, es decir, Hojas de Estilo en Cascada).

El funcionamiento de las hojas de estilo es el siguiente:

- En la página web (archivo .html) se escriben las etiquetas que definen categorías o elementos.
- En la hoja de estilo (archivo .css) se escribe cómo queremos que sea el estilo de presentación de las etiquetas (color, tamaño, fuente, bordes, márgenes, posición, etc).
- En la página web se escribe qué hoja de estilo queremos utilizar.



Las ventajas de utilizar hojas de estilo son muchas, sobre todo permiten hacer un diseño consistente y fácil de modificar. Si varias páginas web hacen referencia a la misma hoja de estilo, para cambiar la apariencia de todas las páginas es suficiente con hacer los cambios en un único lugar, en la hoja de estilo.

10 razones para usar CSS

A continuación, describimos diez ventajas de usar Hojas de Estilo en Cascada (Cascade Style Sheets o "CSS"):

1.- Separación del contenido y presentación.

Las hojas de estilo generalmente se encuentran en archivos separados del código principal (html, por ejemplo). Esto nos va a permitir que en un equipo de trabajo, programador y diseñador puedan realizar sus tareas de forma independiente aunque paralela, sin correr el riesgo de que haya interferencias entre ambos, y ello no alterará el resultado final.

2.- Flexibilidad.

Podríamos comparar las hojas de estilo con la ropa que guardamos en nuestros cajones. Nosotros somos los mismos, pero dependiendo de la temporada variamos nuestra apariencia.

En el caso de las hojas de estilo sucede lo mismo: podemos cambiar en cualquier momento alguna parte o la totalidad del diseño de nuestras páginas con sólo modificar nuestra hoja de estilo, sin que ello suponga modificar el contenido.

3.- Unificación del diseño de las páginas del sitio.

Un sitio web, ya sea dinámico o estático, suele estar formado por unas cuantas páginas. Mantener una misma apariencia se puede volver una tarea pesada y tediosa si tenemos que copiar y pegar código cada vez que creamos una página nueva, o que deseemos modificar una misma cosa en todas.

Enlazando a cada una de nuestras páginas nuestras hojas de estilo, agilizamos este proceso y minimizamos el trabajo.

4.- Optimización de los tiempos de carga y de tráfico en el servidor.

Al haber dividido contenido y apariencia obtenemos archivos más ligeros, y esto nos reporta dos beneficios: por un lado, reducimos notablemente los tiempos de carga del sitio en el navegador. A esto debemos unir la capacidad de éste para mantener nuestra hoja de estilo en caché.

Por otro lado, reducimos el volumen de tráfico de nuestro servidor, que siempre es de agradecer, tanto si disfrutamos de servicios gratuitos en que solemos tener cuotas muy reducidas, como si pagamos por tener alojamiento propio.

5.- Precisión o elasticidad.

Desde el momento en que usemos CSS, el tamaño y posicionamiento de los elementos que formen nuestras páginas podrá ser exacto. Podremos indicarle al navegador en qué píxel debe colocar ésta o aquella imagen, o qué alto y ancho deberá mostrar.

Pero al mismo tiempo, podremos emplear medidas variables o relativas que nos permitan expandir el contenido hasta ocupar la totalidad de la ventana de navegación a nuestro antojo, o contraerla a sólo una parte de la misma, con independencia de la resolución de pantalla del usuario.

6.- Accesibilidad y estructuración.

La combinación de CSS y marcadores descriptivos va a posibilitar que nuestra página se vea correctamente con o sin hoja de estilos, puesto que en cualquier caso, la información se mantendrá estructurada y ordenada.

Esto supone que podrá ser accesible sin ningún tipo de problemas tanto por navegadores antiguos o sin soporte para CSS, como para personas con algún tipo discapacidad.

7.- Limpieza del código fuente.

Si escribimos una hoja de estilo independiente, el código fuente de nuestra web va a resultar menos farragoso y agilizaremos las tareas de localización de las líneas que busquemos.

8.- Compatibilidad y continuidad.

Las reglas establecidas por la especificación CSS-1 fijaron los estándares del diseño, y se mantienen y respetan en la CSS-2. Es de prever que en el nivel 3 sucederá lo mismo con respecto a su predecesor.

Pero lo realmente interesante es que los navegadores que no soporten CSS-3 no tendrán problemas a la hora de asimilar el contenido CSS puesto que siempre les quedará la compatibilidad de CSS-2, o la CSS-1 en su caso. La compatibilidad de las especificaciones CSS anteriores estará siempre garantizada.

9.- Estandarización frente a especificaciones propietarias.

La adopción de estándares por la W3C ofrece la ventaja de la compatibilidad del código entre los diferentes navegadores web. El uso de “soluciones propietarias”, como es el caso de muchas etiquetas o patrones usadas por Microsoft.

En su interpretación de lo que es y no es CSS, dificulta la creación de páginas web, porque supone tener que escribir dos códigos distintos para obtener un mismo resultado, en función del tipo de navegador que use el visitante.

El uso del estándar CSS de la W3C evitará visualizaciones incorrectas de nuestras páginas en distintos navegadores.

10.- Permite la diferenciación de estilos para imprimir / visualizar en pantalla.

El uso de CSS nos va a permitir también maquetar separadamente el contenido de nuestra web para ser mostrado en pantalla o para ser impreso.

Tengamos en cuenta que las necesidades y propiedades de un folio de papel y de un monitor nunca van a ser las mismas, y gracias a CSS podremos determinar cómo queremos que se imprima lo que mostramos en la pantalla, manteniendo siempre una apariencia limpia, ordenada y agradable visualmente.

Espero que estas pequeñas reflexiones hayan servido para animarte a dar el gran salto y vestir a tu web con un bonito traje CSS hecho a medida. Y si aún no te animas a usar CSS en tu sitio, te invitamos a conocer [CSS Reinicia](#), una iniciativa que busca que sitios web en nuestro idioma se reinventen gracias a esta tecnología.

10 excelentes ejemplos de uso de CSS3

Si la página se encuentra en inglés, simplemente, haga clic derecho con su mouse y elija “traducir a español”

Las técnicas relacionadas con CSS3 son cada vez más populares en Internet. Su uso está cada vez más extendidos en sitios web y hay muchos creativos que están

experimentando hasta donde se puede llegar con esta técnica que promete revolucionar la manera en que se diseña en Internet.

Recorriendo la web, podemos maravillarnos con diferentes ejemplos de inspiración, creatividad e inventiva.

A continuación, veremos 10 excelentes ejemplos de uso de CSS3.

1. Un reloj análogo en CSS3.

Ejemplo: <http://www.paulrhayes.com/experiments/clock/#clock>

Explicación del código: www.paulrhayes.com/2009-03/an-analogue-clock-using-only-css

2. Fotos con estilo polaroid utilizando CSS3.

Ejemplo: <http://www.zurb.com/playground/css3-polaroids/>

Explicación del código: <http://www.zurb.com/article/305/easily-turn-your-images-into-polaroids-wi>

3. Cartas que se apilan con CSS3.

Ejemplo: http://designlovr.com/examples/dynamic_stack_of_index_cards/

Explicación del código: <http://designlovr.com/use-css3-to-create-a-dynamic-stack-of-index-cards/>

4. Creando cubos (y animándolos) con las posibilidades 3D de CSS3.

Ejemplo: <http://www.paulrhayes.com/experiments/cube/multiCubes.html>

Explicación del código: <http://www.paulrhayes.com/2009-04/3d-cube-using-css-transformations/>

5. Filtro para las imágenes con CSS.

Ejemplos: <http://www.cssportal.com/css-image-filters/> (para ver el código se debe hacer click sobre las imágenes que tienen filtro en los ejemplos y se mostrará el estilo de cada caso en una ventana modal)

6. Globos de diálogo con CSS3

Ejemplo: <http://nicolasgallagher.com/pure-css-speech-bubbles/demo/>

Explicación del código: <http://nicolasgallagher.com/pure-css-speech-bubbles/>

7. DropDown Menú con CSS3.

Ejemplo: <http://www.webdesignerwall.com/demo/css3-dropdown-menu/>

Explicación del código: <http://www.webdesignerwall.com/tutorials/css3-dropdown-menu/>

8. Animaciones usando transformaciones

Ejemplo y código: <http://www.the-art-of-web.com/css/css-animation/>

9. Efectos de sombra en texto. Un excelente trabajo realizado utilizando grises.

Ejemplo: <http://line25.com/wp-content/uploads/2009/letterpress/demo/demo.html>

Explicación del código: <http://line25.com/tutorials/create-a-letterpress-effect-with-css-text-shadow>

10. Asombrosos efectos de Overlay y uso de bordes de imágenes con CSS3

Ejemplo y explicación del código: <http://www.zurb.com/playground/awesome-overlays>

Fuentes:

<https://uniwebsidad.com/libros/css/capitulo-1/soporte-de-css-en-los-navegadores>

<http://www.maestrosdelweb.com/usarcss/>

<https://damiandeluca.com.ar/10-excelentes-ejemplos-de-uso-de-css3>

<https://www.mclibre.org/consultar/amaya/css/css-que-es.html>