

# Especificaciones y ejemplos XHTML



**Tutorial desarrollado por:**  
[Ismael Caballero, Febrero, 2007.](#)

**Contacto:** [icaballero@carixma.com](mailto:icaballero@carixma.com)



## 1- Introducción

En este tutorial, pretendo mostrar los conceptos generales de **XHTML** para su utilización en el desarrollo de soluciones web.

Para ello se describirán los apartados más importantes y varios ejemplos para su uso en páginas web.

En la parte final de este tutorial, se podrán consultar y descargar varios documentos, con las especificaciones completas de [W3C](#) (World Wide Web Consortium) sobre XHTML 1.0 (en inglés y español) y de la versión 1.1 en inglés.

Destacar que parte de la información para la realización de este tutorial y el documento en español de la especificación XHTML 1.0 que anteriormente he mencionado, han sido recogidos de la traducción realizada por [Sidar.org](#) que se encuentra en:

<http://www.sidar.org/recur/desdi/traduc/es/xhtml/xhtml11.htm>

Como requisitos previos recomendados, para el seguimiento de este tutorial, son conocer el lenguaje [HTML](#) y las hojas de estilos ([CSS](#)).

## 2- Definición de XHTML

[XHTML](#) significa lenguaje de marcado de hipertexto extendido.

Se utiliza para generar documentos y contenidos de [hipertexto](#) generalmente publicados en la Web. Es una reformulación del lenguaje HTML que ahora pasa a ser compatible con el estándar [XML](#), considerándose que se incluye en la familia de los [Lenguajes de Marcado](#).

XHTML es una familia de módulos y tipos de documentos que reproduce, engloba y extiende HTML 4.0.

XHTML mantiene una separación entre el contenido y el diseño. Es decir, que uno no afecte al otro, y se puedan modificar independientemente.

El XHTML está diseñado para que sea funcional no sólo a los navegadores sino a varios dispositivos Web (teléfonos móviles, portátiles, PDAs, etc.).

El fichero de un documento XHTML, en sí, es un archivo de texto plano generalmente con alguna de estas extensiones: **.html**, **.htm**, o bien **.xhtml**. Los navegadores (IE, Firefox, Mozilla, Opera, etc.) interpretan estos archivos y procesan el código para mostrarlos en pantalla.

El logotipo que indica que un sitio web aplica esta tecnología y cumple las especificaciones del W3C es el siguiente:



Los desarrolladores que migren aplicaciones hacia XHTML apreciarán las siguientes mejoras:

- Los documentos XHTML son conformes a las reglas de los documentos XML. Como tales, son fácilmente visualizados, editados y validados con herramientas XML estándar.
- Los documentos XHTML pueden escribirse para que funcionen igual o mejor que en las aplicaciones de usuario conformes a HTML 4.0.
- Los documentos XHTML pueden usar aplicaciones (Ej. scripts y applets) que se basen ya sea en el Modelo del Objeto Documento de HTML o XML [\[DOM\]](#).
- Las soluciones implementadas con XHTML podrán ser utilizadas en distintas plataformas y distintos dispositivos (teléfonos, pdas, etc.) sin necesidades de incluir software o soluciones adicionales.

### 3- Diferencias con HTML 4.0.

#### 3.1 Los documentos deben ser "gramaticalmente correctos"

Esencialmente significa que todos los elementos deben tener etiquetas de cierre, incluso las etiquetas únicas, como por ejemplo: `<br />`, `img`, `<hr />`, etc. y que todos los elementos deben estar anidados siguiendo el orden correcto.

**Por ejemplo:** es correcto (elementos anidados): `<p>he aquí un <em>texto</em> enfatizado.</p>`  
 Pero es incorrecto (elementos solapados): `<p>he aquí un <em>texto</p> enfatizado</em>`

#### 3.2 Los nombres de etiquetas y atributos son sensibles a las minúsculas y a las mayúsculas.

Los documentos XHTML deben usar minúsculas para los nombres de todos los elementos y atributos HTML. Esta diferencia es necesaria porque XML es sensible a minúsculas y mayúsculas por ejemplo: `<li>` y `<LI>` son etiquetas diferentes.

#### 3.3 Los elementos no vacíos requieren etiquetas de cierre

Con HTML 4.0, basado en SGML, en algunos elementos podía omitirse la etiqueta de cierre, de tal manera que la apertura de los elementos que les sucedían implicaba dicho cierre. Esta omisión no está permitida en XHTML.

**Por ejemplo:** es correcto (elementos cerrados): `<p>he aquí un párrafo.</p><p>aquí hay otro párrafo.</p>`  
 Pero es incorrecto (elementos no cerrados): `<p>he aquí un párrafo.<p>aquí hay un párrafo.`

#### 3.4 Los valores de los atributos deben ir entre comillas

Todos los valores de atributos deben ir entrecomillados, incluso aquellos que son numéricos.

**Por ejemplo:** es correcto (valores de atributo entrecomillados): `<table rows="3">`  
 Pero es incorrecto (valores de atributo no entrecomillados): `<table rows=3>`

#### 3.5 Minimización de atributos

XML no soporta la minimización de atributos. Los pares atributo-valor deben escribirse en toda su extensión. Los nombres de atributos como `compact`, `selected` o `checked` no pueden aparecer en elementos sin que sea especificado su valor.

**Por ejemplo:** es correcto (atributos no minimizados):  
`<select name="sexo">`  
`<option value="hombre" select="selected">Hombre</option>`  
`<option value="mujer" >Mujer</option>`  
`</select>`

Pero es incorrecto (atributos minimizados):  
`<select name="sexo">`  
`<option value="hombre" selected>Hombre</option>`  
`<option value="mujer" >Mujer</option>`  
`</select>`

### 3.6 Elementos vacíos

Los elementos vacíos deben bien tener una etiqueta de cierre bien terminar su etiqueta de apertura con `/>`. **Por ejemplo:** son correctas (etiquetas vacías cerradas): `<br /><hr />`, e incorrectas: `<br><hr>`

**NOTA:** Es importante notar el espacio que queda entre el fin de la palabra y la barra, en la versión resumida, de esta manera se conserva compatibilidad con algunos agentes de usuario que no reconocen la etiqueta sin ese espacio.

### 3.7 Elementos script y style

En XHTML, los elementos `script` y `style` se declaran como elementos con contenido `#PCDATA`. Como resultado, `<` y `&` serán tratados como comienzos de etiquetado, y entidades como `&lt;` y `&amp;` serán reconocidas como referencias a las entidades `<` y `&` respectivamente por el procesador XML. Englobar el contenido del elemento `script` o `style` dentro de una sección marcada como `CDATA` evita el procesamiento de estas entidades.

```
<script>

<![CDATA[

... contenido no procesado del script ...

]]>

</script>
```

Una alternativa es usar documentos externos de estilo y escritura de código (includes).

## 5- Características principales de XHTML

Los documentos XHTML, al ser un documento XML, para estar conformes, deben ajustarse a los siguientes puntos:

- Deben poder validarse con alguna de las tres DTDs siguientes:
  1. [XHTML-1.0-Strict](#)
  2. [XHTML-1.0-Transitional](#)
  3. [XHTML-1.0-Frameset](#)
- El elemento raíz del documento debe ser `<html>`.
- El elemento raíz del documento debe indicar el espacio nominal XHTML usando el atributo `xmlns` (XMLNAMES). El espacio nominal para XHTML es `http://www.w3.org/1999/xhtml`
- Debe haber una declaración DOCTYPE en el documento antes del elemento raíz. El identificador público incluido en la declaración DOCTYPE debe hacer referencia a alguna de las tres DTD comentadas anteriormente, de la siguiente forma:

Para **XHTML-1.0-Strict**: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"`

`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">`

Para **XHTML-1.0-Transitional**: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"`

`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">`

Para **XHTML-1.0-Frameset**: `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"`

`"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">`

He aquí un ejemplo de un pequeño documento XHTML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```

<!--Este es un comentario y no sera tomado en cuenta por el navegador -->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Título de la pagina</title>
</head>
<body>
<p>Primer documento XHTML, es decir un, Hola mundo</p>
</body>
</html>

```

Analicemos el código línea a línea:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Es una declaración XML, dónde especificamos el atributo **encoding** con el valor [UTF-8](#).

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

```

Este es el encabezado que deberían llevar todos los documentos XHTML **1.0 estrictos** acordes a las [especificaciones de la W3C](#). Es un aviso para que el agente de usuario sepa que tipo de documento va a interpretar. Para ser más específicos diremos que al ser un documento de tipo *strict.dtd*, los elementos utilizados serán sólo los incluidos en el lenguaje XHTML 1.0. Si utilizáramos etiquetas antiguas y declaráramos el documento cómo estricto, no podríamos [validar nuestra página XHTML](#).

```
<!--Este es un comentario y no sera tomado en cuenta por el navegador -->
```

Los comentarios comienzan con `<!--` y terminan con `-->` (los saltos de línea no los quiebran). No son tomados en cuenta por el agente de usuario, es decir no se interpretan (por lo que no se representan) y se utilizan frecuentemente para hacer aclaraciones en el código.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

Aquí damos comienzo al documento XHTML, todo documento debe llevar esta etiqueta de apertura (`<html>`) y al finalizar debemos escribir la etiqueta de cierre que en el ejemplo se encuentra al final del código (`</html>`). Además aparecen 3 atributos, el primero especifica el tipo de documento, y el segundo y el tercero deben llevar como valores el código del lenguaje en el que será escrito. Para la lengua castellana este valor es: **"es"**. Puedes consultar los [códigos de otros idiomas](#).

```
<head>
```

```
<title>Título de la pagina</title>
```

```
</head>
```

Con la etiqueta de apertura `<head>` comienza lo que se llama el encabezado de las páginas web, aquí se dan ciertas especificaciones y otra *META* información acerca del documento que ya veremos con más detalle. Pero entre ellas está el elemento utilizado en el ejemplo, mediante las etiquetas `<title></title>`, que encierran el título de la página. Los navegadores suelen mostrarlo cómo título de la ventana. También es utilizada por los buscadores cuando presentan resultados de los documentos a sus usuarios.

```
<body>
```

```
<p>Mi primer documento XHTML</p>
```

```
</body>
```

El cuerpo que comienza con `<body>` contiene todo el contenido que será representado por el agente de usuario. En nuestro ejemplo sólo utilizamos una etiqueta más, `<p>` que sirve para encerrar párrafos.

Cerrando el cuerpo (`</body>`) y el documento XHTML (`</html>`) conseguimos nuestra primer página.

A continuación, mostraremos algunos ejemplos de utilización de etiquetas y estructuras que podemos utilizar en nuestros desarrollos webs empleando XHTML, son similares a las empleadas en nuestros códigos HTML, pero debemos aplicar las reglas antes descritas, para que los **documentos XML finales, estén bien formados**:

## Encabezados títulos

Los encabezados o títulos se pueden obtener mediante 6 etiquetas diferentes: **h1**, **h2**, **h3**, **h4**, **h5** y **h6** que tienen una jerarquía de importancia en la manera en que se representan por los navegadores. Esto significa que **h1** será la etiqueta de más relevancia y de mayor tamaño, el cual irá descendiendo hasta la etiqueta **h6** que será la más pequeña.

Por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

```

<head>
  <title>Encabezados y títulos</title>
</head>
<body>
  <h1>Encabezado con etiqueta h1</h1>
  <h2>Encabezado con etiqueta h2</h2>
  <h3>Encabezado con etiqueta h3</h3>
  <h4>Encabezado con etiqueta h4</h4>
  <h5>Encabezado con etiqueta h5</h5>
  <h6>Encabezado con etiqueta h6</h6>
</body>
</html>

```

## Enlaces

Los enlaces son las estructuras que nos permiten navegar por la Web, es decir que vayamos saltando por diferentes páginas de un mismo sitio, o de sitios externos. La etiqueta utilizada para crear un vínculo es **a** (*anchor*).

Por ejemplo: `<a href="http://www.carixma.com">Ver Carixma</a>`

La estructura de los enlaces permite incluir varios atributos, uno destacado es: **title**, que sirve para especificar una pequeña descripción del sitio que estamos enlazando. Los navegadores suelen desplegar un cartelito con el valor de este atributo cuando el usuario señala con el mouse el enlace. Esto es de gran ayuda a los usuarios, que decidirán si seguir el enlace o no.

Ejemplo: `<a href="subdirectorio/colegios.html" title="Lista de Colegios privados de la zona sur">Colegios Privados</a>`

Otros atributos interesantes son: **class**, **name**, **style**, **target**, ... que permite identificar, personalizar y configurar el estilo y tipo del enlace.

## Imágenes

La etiqueta para insertar imágenes es **img**. Su principal atributo es **src** (de source). Este atributo lleva como valor la dirección (relativa o absoluta) de la ubicación de la imagen.

Ejemplo: ``

Otro atributo interesante es **align**, que permite alinear las imágenes, cuyos valores pueden ser: **right**, **left**, **center**, etc.

Por ejemplo: ``

## Tablas

Las tablas son una interesante herramienta para la representación de la información de forma tabular, mediante filas y columnas.

**Ejemplo:**

```

<table>
  <caption>Elementos XHTML</caption>
  <thead>
    <tr>
      <th>ELEMENTO</th>
      <th>TIPO</th>
      <th>SEMANTICA</th>

```

```

</tr>

</thead>

<tbody>

  <tr>

    <td>em</td>

    <td>línea</td>

    <td>mayor énfasis</td>

  </tr>

  <tr>

    <td>strong</td>

    <td>línea</td>

    <td>mayor fuerza</td>

  </tr>

  <tr>

    <td>pre</td>

    <td>bloque</td>

    <td>texto preformateado</td>

  </tr>

</tbody>

</table>

```

El elemento **table** da comienzo a la tabla. Dentro tenemos a **caption** que encierra el título que llevará la tabla.

A partir de aquí definimos dos partes principales: **thead** dónde se ubica el encabezado, y **tbody** que encierra a las celdas del cuerpo.

**tr** es el encargado de contener filas, el encabezado tiene una sola fila, cuyas celdas de encabezado se contienen mediante tres etiquetas del elemento **th**.

Las celdas se definen mediante el uso del elemento **td**.

Cabe destacar el atributo *summary* dentro del elemento **table**, este nos permite agregar algo de semántica o detalle, al ponerle como valor un resumen del contenido de la tabla para que pueda ser reconocido por robots (ej. buscadores, sistemas de accesibilidad, etc.) u otros agentes de usuario.

Por ejemplo: `<table summary="Elementos del lenguaje XHTML">...</table>`

## Formularios

Los formularios nos dan la posibilidad de recibir información por parte del usuario de manera dinámica.

Hay dos métodos de envío de información mediante formularios que trataremos aquí: **post** y **get**.

### Método get

El método **get** envía información al servidor como parte de la *URL*. El servidor analizará la *URL* pedida por el agente de usuario y separará la información relevante para su posterior procesamiento.

Ejemplo: `http://www.carixma.com/busqueda.asp?palabra=xhtml`, siendo el signo `?` es el delimitador que nos permite separar la consulta de la *URL*.

### Método post

Este método envía la información como parte de la petición del agente de usuario al servidor web (esta petición también incluye la *URL*, sin embargo nuestra información no formará parte de ella) y no se le muestra directamente al usuario.

El elemento principal y obligatorio de todo formulario es **form** y quizá sus dos atributos más importantes son *method* y *action*. El primero especifica el método de envío (**get** o bien **post**), y en el segundo pondremos la *URL* del recurso que contendrá la secuencia de comandos encargada de hacer algo con esa información.

Sin embargo un formulario sin otros elementos no nos sirve de mucho. Hagamos entonces un recorrido por los elementos más comunes:

- . Elemento Input
- . Elemento Select
- . Elemento TextArea

Ahora indicaremos algunos ejemplos y detalle de estos elementos, empezando por **input**, que al especificarle su atributo **type** nos permite construir diferentes tipos de controles.

- . Campos de Texto (text)
- . Campos Ocultos (hidden)
- . Campos Contraseña (password)
- . Campos Radio Button (radio)
- . Campos Checkbox (checkbox)
- . Campos Button (button)
- . Campos Submit (submit)
- . Campos Reset (reset)

Ejemplos:

**<form>**

```
<p>Nombre: <input type="text" name="nombre" name="carixma"/></p>
```

```
<p><input type="hidden" name="estado" value="activo"/></p>
```

```
<p><select name="sexo">
<option value="hombre" select="selected">Hombre</option>
<option value="mujer">Mujer</option>
</select></p>
```

```
<p> <textarea cols="50" rows="4" name="Observaciones">Aquí puedes incluir tus observaciones...</textarea> </p>
```

**</form>**

## 5- Consulta y Descarga de Documentación

He recopilado y unificado la especificación de la traducción realizada por el sitio web [Sidar.org](http://www.sidar.org) de la especificación XHTML del W3C que se encuentra en: <http://www.sidar.org/recur/desdi/traduc/es/xhtmll/xhtmll11.htm> en un documento PDF para su consulta o descarga, también, es posible consultar y descargar las especificaciones en inglés sobre XHTML del W3C.

1. Puedes consultar o descargar la [especificación completa de XHTML 1.0 del W3C en español en formato PDF](#).
2. Puedes consultar o descargar la [especificación completa de XHTML 1.0 del W3C en inglés en formato PDF](#).
3. Puedes consultar o descargar la [especificación completa de XHTML 1.1 del W3C en inglés en formato PDF](#).

## 6- Conclusión

En este tutorial he pretendido describir brevemente y mostrar algunos ejemplos de utilización de XHTML, tecnología que supone la actualización del HTML 4.0 y que permite separar el contenido y el diseño de nuestros proyectos web.

Si se desea profundizar en esta tecnología, en el apartado anterior se incluyen varios documentos con las especificaciones completas del W3C, donde se incluyen todas las definiciones, propiedades, ejemplos, etc.

Quiero destacar las siguientes referencias con información y utilidades muy interesantes sobre XHTML y utilizadas en tutorial.

- Validador XHTML del W3C: <http://validator.w3.org/>
- <http://manual-xhtml.blogspot.com/> (manual realizado por [emilio.astarita@gmail.com](mailto:emilio.astarita@gmail.com))
- <http://www.sidar.org/recur/desdi/traduc/es/xhtml/xhtml11.htm>
- <http://www.w3.org/TR/xhtml1/>
- <http://www.w3.org/TR/xhtml11/>
- Wikipedia: <http://es.wikipedia.org/wiki/Portada>

Además de señalar varias referencias de sitios web que aplican XHTML:

- <http://www.w3csites.com/>
- <http://www.red.es/>
- <http://www.humanlevel.com/portfolio.asp>
- <http://www.canal-si.com/>
- <http://www.inmostrategy.com/>
- <http://www.juliotrespalacios.com/>
- <http://www.procumasa.com/>

Por último, en este tutorial he puesto la mejor intención para presentar esta tecnología tan actual y ampliamente utilizada en el desarrollo de sitios web, pero puedo haberme equivocado en algún aspecto de este tutorial, si alguien desea aportar datos, o colaborar, o corregirme, estaré encantado de recibir y estudiar su respuesta en [icaballero@carixma.com](mailto:icaballero@carixma.com)

## [SOBRE EL AUTOR...](#)