

# Desarrollo de Aplicaciones Web

## Práctica 2: HTML (1)

### 1. Objetivos

- Aprender el lenguaje de marcado HTML y la importancia de escribir código correcto y conforme a la especificación oficial.
- Conocer algunas herramientas que ayudan a escribir código HTML correcto y compatible con distintos navegadores.
- Aprender a publicar un sitio web en Internet.

### 2. Recursos

¿Qué etiquetas existen en HTML? ¿Qué atributos existen y qué valores pueden tomar? ¿Cómo se emplean las etiquetas?

- **W3Schools**<sup>1</sup>: cursos de aprendizaje y guías de referencia de diversas tecnologías empleadas en la programación web.
- **W3C**<sup>2</sup>: organismo internacional que desarrolla las especificaciones de las principales tecnologías que se emplean en la programación web.
- **XHTML Cheat Sheet** <sup>3</sup>: resumen en una sola página de lo más importante de XHTML.
- **HTML Cheat Sheet** <sup>4</sup>: resumen en una sola cara de lo más importante de (X)HTML.
- **HTML5 Cheat Sheet** <sup>5</sup>: resumen de HTML5 de la famosa revista Smashing Magazine. Es del año 2009, así que seguramente faltan algunas de las últimas novedades.
- **HTML Tags**<sup>6</sup>: resumen de las etiquetas con sus atributos y posibles valores.

¿Cómo puedo saber que el código HTML que he escrito es correcto?

- **W3C Markup Validation Service**<sup>7</sup>: servicio de validación para diferentes lenguajes de marcado (HTML, XHTML, SVG, y otros).
- **Html Validator**<sup>8</sup>: complemento para Mozilla Firefox que permite comprobar el número de errores que posee una página web que se visualiza.
- **Validity**<sup>9</sup>: complemento para Google Chrome que permite comprobar el número de errores que posee una página web que se visualiza.

¿Cómo puedo comprobar que la página web se visualiza correctamente en diferentes navegadores?

- **Browsershots**<sup>10</sup>: permite comprobar cómo se visualiza una página web en decenas de navegadores y en distintos sistemas operativos.

---

<sup>1</sup><http://www.w3schools.com>

<sup>2</sup><http://www.w3.org>

<sup>3</sup><http://floele.flyspray.org/htmlcheatsheet-ncr.pdf>

<sup>4</sup><http://www.addedbytes.com/cheat-sheets/html-cheat-sheet/>

<sup>5</sup><http://www.smashingmagazine.com/2009/07/06/html-5-cheat-sheet-pdf/>

<sup>6</sup><http://html-tags.info/>

<sup>7</sup><http://validator.w3.org>

<sup>8</sup><http://addons.mozilla.org/es-ES/firefox/addon/249>

<sup>9</sup><https://chrome.google.com/webstore/detail/validity/bb1cmjbjbohdfglopkidebfccilipeif?hl=es>

<sup>10</sup><http://browsershots.org>

- **Total Validator Service**<sup>11</sup>: ofrece diversos servicios, como buscar enlaces rotos o comprobar el nivel de accesibilidad. También ofrece la posibilidad de generar una captura de pantalla de cómo se visualiza en distintos navegadores.

¿Existe alguna herramienta que me pueda ayudar a escribir el código HTML?

- **Notepad++**<sup>12</sup>: editor gratuito de código fuente que soporta varios lenguajes de programación. Entre otras características, posee sintaxis coloreada, envoltura de sintaxis y autocompletado.
- **Sublime Text**<sup>13</sup>: excelente editor compatible con múltiples lenguajes de programación y disponible para Windows, OS X y Linux; se puede probar de forma gratuita, pero para un uso continuo hay que adquirir una licencia.
- **Brackets**<sup>14</sup>: editor de HTML disponible para diferentes sistemas operativos.
- **Visual Studio Code**<sup>15</sup>: editor de Microsoft disponible para diferentes sistemas operativos.
- **Real-time HTML Editor**<sup>16</sup>: editor WYSIWYG de HTML en página web.
- **Online-HTML-Editor.org**<sup>17</sup>: editor WYSIWYG de HTML en página web que permite abrir y guardar una página web.
- **HTML Tidy**<sup>18</sup>: limpia y organiza el código HTML. También se puede descargar el ejecutable y el código fuente para ejecutarlo en local.

¿Qué es un servidor web? ¿Cómo se publica un sitio web en un servidor web?

- **MDN - What is a web server?**<sup>19</sup>: qué son los servidores web, cómo funcionan y por qué son importantes.
- **MDN - Publishing your website**<sup>20</sup>: un sitio web se debe publicar en un servidor web para que la gente lo pueda encontrar.

## 3. ¿Qué tengo que hacer?

### 3.1. Las primeras páginas de la aplicación web

En esta práctica vamos a empezar el desarrollo de un sitio web de gestión de álbumes de fotos llamado “PI - Pictures & Images”. Con cada práctica el sitio web crecerá tanto en número de páginas como en funcionalidad.

El objetivo es crear un sistema gestor de álbumes de fotos que admita múltiples usuarios. Un usuario se tendrá que registrar para poder emplear el sistema. Una vez registrado podrá crear todos los álbumes que quiera y en cada álbum podrá publicar todas las fotos que quiera.

En esta práctica tienes que crear un sitio web compuesto de cinco páginas web; en la Figura 1 se incluye un diagrama que resume las páginas que se tienen que crear y las relaciones que existen entre ellas:

**Página principal** Contiene enlaces a las distintas páginas que componen el sitio web, un formulario (nombre de usuario y contraseña) para acceder como usuario registrado y un listado con un resumen (foto, título, fecha, país) de las últimas cinco fotos que se han introducido. Cada foto debe enlazar con la **Página detalle foto**. Para verificar el comportamiento del prototipo, una de las fotos debe enlazar con la **Página detalle foto** y otra de las fotos debe enlazar a la página con un mensaje de error que indique que no se puede acceder al detalle de la foto si el usuario no está registrado.

<sup>11</sup><http://www.totalvalidator.com/>

<sup>12</sup><http://notepad-plus.sourceforge.net/es/site.htm>

<sup>13</sup><http://www.sublimetext.com/>

<sup>14</sup><http://brackets.io/>

<sup>15</sup><https://code.visualstudio.com/download>

<sup>16</sup><http://htmledit.squarefree.com/>

<sup>17</sup><http://online-html-editor.org/>

<sup>18</sup><http://cgi.w3.org/cgi-bin/tidy>

<sup>19</sup>[https://developer.mozilla.org/en-US/docs/Learn/Common\\_questions/What\\_is\\_a\\_web\\_server](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server)

<sup>20</sup>[https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web/Publishing\\_your\\_website](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/Publishing_your_website)

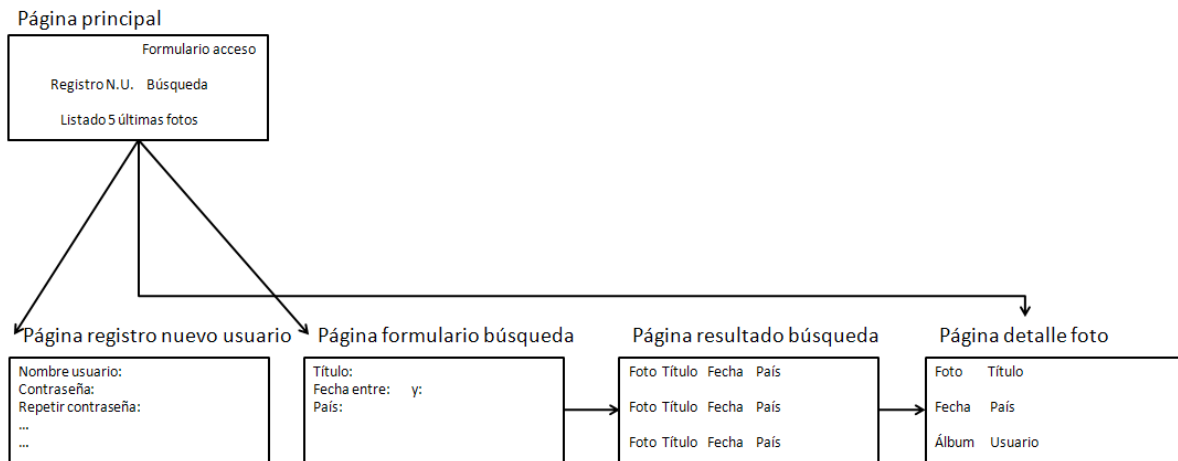


Figura 1: Diagrama de páginas que componen el sitio web

**Página con el formulario de registro como nuevo usuario** Contiene un formulario con los datos necesarios para registrarse (nombre de usuario, contraseña, repetir contraseña, dirección de email, sexo, fecha de nacimiento, ciudad y país de residencia, foto).

**Página con el formulario de búsqueda** Contiene un formulario para realizar una búsqueda de fotos con los siguientes criterios: título, fecha y país. El usuario puede buscar por solo uno de los campos o los puede combinar para formar una consulta en la que se deben cumplir todos los campos indicados por el usuario.

**Página con el listado resultado de una búsqueda** Contiene un listado con un resumen (foto, título, fecha y país) de las fotos que cumplen los criterios de una búsqueda. Cada foto debe enlazar con la **Página detalle foto**. Para verificar el comportamiento del prototipo, una de las fotos debe enlazar con la **Página detalle foto** y otra de las fotos debe enlazar a la página con un mensaje de error que indique que no se puede acceder al detalle de la foto si el usuario no está registrado.

**Página detalle foto** Muestra toda la información (foto, título, fecha, país, álbum de fotos y usuario al que pertenece) sobre una foto seleccionada en la **Página principal** o en la **Página con el listado resultado de una búsqueda**.

Por ahora, estas páginas contienen datos estáticos que en prácticas posteriores se sustituirán por datos reales obtenidos de una base de datos. Esto significa que, por ejemplo, si se tiene que mostrar una lista desplegable con países, se deben colar algunos valores estáticos a modo de ejemplo. El modelo de datos que se debe emplear en la base de datos se presentará en una próxima práctica.

Sin embargo, la navegación entre las páginas debe funcionar correctamente: por ejemplo, cuando se pulse en el botón “Buscar” se debe navegar a la página con el listado resultado de una búsqueda, que mostrará siempre el mismo resultado; cuando se pulse sobre una foto, se debe navegar a la página con el detalle de la foto, pero se puede mostrar el mismo detalle para todas las fotos.

Cuando realmente funcione la práctica, habrá páginas que se podrán unificar en una sola. Por ejemplo, la **Página con el formulario de búsqueda** y la **Página con el listado resultado de una búsqueda** pueden ser la misma página (el formulario se invoca a sí mismo), pero ahora no pueden ser la misma página (más exactamente, no pueden ser el mismo fichero) porque se deben representar los dos estados posibles: página sin resultado y página con resultado. Y también al revés: la **Página principal** contiene un formulario para acceder como usuario registrado, pero también se puede realizar como una página independiente.

No hay que realizar dos versiones de cada página, una para usuario autenticado y otra para usuario no autenticado, eso ya se programará más adelante. Pero la página principal sí que se tiene que realizar con sus dos alternativas (habrá dos ficheros) para mostrar las diferencias entre estar o no estar autenticado: puede cambiar la barra de menú, el enlace de entrar o salir, el enlace al perfil del usuario, un enlace rápido para subir/añadir/publicar foto, etc. Cuando se inicie sesión como usuario registrado, se debe cambiar a la versión de la página como usuario autenticado; cuando se pulse en salir o cerrar sesión, se debe

cambiar a la versión de la página como usuario no autenticado. Al existir dos alternativas (dos ficheros), también se puede simular el comportamiento del enlace a **Página detalle foto**. Cuando el usuario no esté autenticado, se debe mostrar una página de aviso indicando que no es posible acceder al detalle de la foto.

**Muy importante: tienes que cumplir exactamente lo que se te pide en cada práctica porque las prácticas son acumulativas, si te desvías de lo que se te pide en esta práctica, es posible que tengas graves problemas en prácticas posteriores.**

**Muy importante: en esta práctica no te tienes que preocupar por la presentación visual de las páginas web. La presentación visual la definirás en las próximas prácticas con la tecnología CSS.**

### 3.2. Publicación de un sitio web

Si dispones de un alojamiento web contratado, puedes usarlo para publicar el sitio web que estás construyendo. Si no lo quieres usar para la práctica de la asignatura o no dispones de un alojamiento web contratado, sigue leyendo lo siguiente.

Localiza varias ofertas de alojamiento gratuito y crea una tabla resumen que contenga los principales factores que se explican en el curso en línea que debes evaluar a la hora de seleccionar un alojamiento: sistema operativo, almacenamiento, transferencia mensual, etc. **Un factor muy importante para esta asignatura es que el alojamiento gratuito ofrezca la posibilidad de crear bases de datos con MySQL y programar con PHP.**

Confeciona un informe con la tabla resumen y una explicación de cuál es la mejor oferta desde tu punto de vista.

Publica el sitio web de esta práctica en el servicio de alojamiento gratuito con la mejor oferta: si el servicio de alojamiento ofrece la publicación mediante FTP normal o mediante FTP online (a través de un administrador de archivos en página web), utiliza ambas opciones para acostumbrarte a utilizar ambos métodos.

Una vez publicado el sitio web comprueba que los enlaces funcionan correctamente y todas las imágenes se ven. Si algo falla, seguramente te habrás equivocado con los nombres de los ficheros.

Indica en el informe el URL de tu sitio web.

**Importante: el fichero de la página principal de un sitio web suele tener el nombre `index.html`. Cuando lleguemos a PHP se cambiará por `index.php`. Sólo como curiosidad, los servidores web suelen tener configurada la siguiente lista de páginas por defecto para la página principal<sup>21</sup>: `index.htm`, `index.html`, `index.php`, `index.php3`, `index.php5`, `index.php4`, `index.shtml`, `default.htm`, `default.html`, `index.py`, `default.shtml`, `index.pl`, `index.cgi` y `home.html`.**

## 4. ¿Cómo lo hago?

Antes de ponerte a escribir el código HTML, realiza un diagrama de las páginas web que componen tu sitio web. Este diagrama puede tener tanto detalle como tú quieras, pero hay que encontrar un equilibrio entre utilidad y legibilidad: un diagrama muy simple y con poco contenido será poco útil, mientras que un diagrama muy complejo y con mucha información será difícil de entender y no supondrá una ayuda.

Crea una página web “esqueleto” a partir de la cual puedas crear todas las páginas de tu sitio web. En esta página incluye todos los elementos que se repitan en todas las páginas (cabecera y pie de página, barra de navegación). Asegúrate que el código de esta página web es correcto.

En el pie de página incluye los nombres de los autores de la práctica, un aviso de copyright con el año y alguna información más.

Sólo con HTML no se puede reutilizar código, así que tendrás que repetir fragmentos de código en las distintas páginas. Cuando se llegue a la parte de PHP se podrá reutilizar el código.

En la actualidad existen diferentes versiones de HTML: HTML 4.01, XHTML 1.0 y HTML5. **Para realizar la práctica debes utilizar HTML5.**

**Muy importante: en esta práctica no tienes que usar JavaScript, todo lo que se plantea se puede realizar sin JavaScript, solo hay que leer y entender el enunciado de la práctica. Y claro, saber utilizar bien HTML.** En concreto, en cursos anteriores hay estudiantes que han tenido las siguientes dudas:

---

<sup>21</sup>El orden de configuración es importante porque el servidor web va a intentar encontrar los archivos en el orden indicado.

1. Estaba leyendo la práctica y he visto esto “por ejemplo, cuando se pulse en el botón ‘Buscar’ se debe navegar a la página con el listado resultado de una búsqueda”. ¿Hay alguna manera de hacer un botón que redirija a otra página sin JavaScript o tenemos que usar un enlace para ello?

Respuesta: Por supuesto que se puede hacer que un botón haga eso, no hay que hacer nada especial, existe un botón que hace eso. Y poner un botón dentro de un enlace es un terrible error.

2. En la parte del registro se piden el país y la ciudad de residencia, había pensado en hacerlo con un select pero claro sin JavaScript no puedo cargar todos los países y ciudades. ¿Pongo algunos de ejemplo en el select o puedo dejarlo como un input text?

Respuesta: Todos los países pueden estar en el `<select>` de forma estática, no necesitas JavaScript para eso. Si consigues una lista de países en Internet, puedes generar el código HTML necesario de varias formas, por ejemplo, con Microsoft Excel. Pero de todas formas, por ahora solo es necesario que pongas algunos países, con 10 países es suficiente para hacer pruebas. Y las ciudades... ni con JavaScript podrías tener la lista de todas las ciudades del mundo (¿Google las tendrá?, puede ser, ¿y las tiene actualizadas al día?), ahí lo más conveniente es un cuadro de texto.

3. Me ha surgido la siguiente duda: en la práctica explicas que hay que hacer una versión, de `index.html`, para el usuario registrado y otra para el usuario no registrado. ¿Tengo que enlazarlas entre sí de alguna manera, o simplemente tener las dos versiones hechas de manera independiente? Lo digo porque al no tener bases de datos que comprueben si un usuario está registrado/logueado o no, no podría hacer la comprobación para realizar un `logout`, por ejemplo.

Respuesta: El enunciado de la práctica dice:

Por ahora, estas páginas contienen datos estáticos que en prácticas posteriores se sustituirán por datos reales obtenidos de una base de datos. [...] Sin embargo, la navegación entre las páginas debe funcionar correctamente: por ejemplo, cuando se pulse en el botón “Buscar” se debe navegar a la página con el listado resultado de una búsqueda, que mostrará siempre el mismo resultado;

Lo mismo se tiene que hacer con el “login”: ahora mismo no se puede realizar la comprobación (no hay programación, no hay base de datos con los usuarios), pero sí que se puede hacer que la navegación entre las páginas funcione cuando se pulse el botón “Entrar”.

## 5. Recomendaciones

El objetivo de la asignatura es construir una aplicación web mediante el desarrollo de un prototipo que se irá refinando y ampliando en las diferentes prácticas. Antes de ponerte delante del ordenador a escribir páginas web puedes hacer (y es muy recomendable) un *mockup* (modelo a escala, maqueta) de cómo piensas que será el sitio web. Un *mockup* lo puedes hacer en papel, con una herramienta de diseño gráfico o con el mismo Microsoft PowerPoint.

El lenguaje HTML es muy sencillo de emplear: no es un lenguaje de programación, como mucha gente cree, sino un lenguaje de marcado compuesto de etiquetas (o elementos) que pueden contener atributos que poseen un valor. Las etiquetas no se pueden escribir de cualquier forma, sino que existen una serie de reglas que definen un orden entre las etiquetas.

Si escribes mal el código HTML, *normalmente* la página web se visualizará más o menos bien en la mayoría de los navegadores porque están preparados para admitir errores, pero en algunos casos la presentación de la página puede variar enormemente de un navegador a otro. Por tanto, para asegurar que una página web se visualiza correctamente en la mayoría de los navegadores es imprescindible escribir un código correcto.

Cumplir los estándares cada vez tiene más importancia. El sitio **W3C Sites**<sup>22</sup> recopila miles de sitios web que cumplen las recomendaciones del W3C. ¿Por qué son importantes los estándares? Algunas lecturas que te lo explican:

- **WaSP: Luchando por los estándares**<sup>23</sup>.
- **Diez razones para aprender y usar los estándares web**<sup>24</sup>.

<sup>22</sup><http://www.w3csites.com/>

<sup>23</sup><http://www.webstandards.org/about/mission/es/>

<sup>24</sup><http://www.estandaresyaccesibilidad.com/2006/01/diez-razones-para-aprender-y-usar-los.html>

- **Developing With Web Standards - Recommendations and best practices**<sup>25</sup>.

En la actualidad existen diferentes versiones de HTML: HTML 4.01, XHTML 1.0 y HTML5. Conviene que las conozcas todas: para un desarrollo nuevo, lo más conveniente es utilizar la última versión, pero es muy probable que tengas que trabajar también con páginas web desarrolladas con versiones antiguas de HTML.

Para empezar a aprender HTML consulta algún curso de aprendizaje, como por ejemplo los disponibles en W3Schools:

- **HTML**<sup>26</sup>.
- **HTML - XHTML**<sup>27</sup>.
- **HTML5 Tutorial**<sup>28</sup>.

¿De dónde puedes copiar la estructura base de una página web y que sea correcta? Mira el código de la página del W3C y copia lo mínimo necesario para crear una página web correcta. Por ejemplo, el siguiente código lo puedes utilizar para crear páginas web basadas en XHTML 1.0 Strict:

```
<?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">
<!-- La cabecera -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="generator" content="Bloc de notas" />
<meta name="author" content="Sergio Luján Mora" />
<meta name="keywords" content="XHTML, web" />
<meta name="description" content="Plantilla base de una página creada con XHTML" />
<title>Plantilla base de XHTML</title>
</head>
<!-- El cuerpo -->
<body>
<p>
Bla, bla, bla.
</p>
</body>
</html>
```

XHTML 1.0 está disponible en tres versiones o variantes llamadas *Strict*, *Transitional* y *Frameset*. Si quieres saber más sobre la estructura básica de una página web basada en XHTML 1.0, consulta el artículo **Plantilla base XHTML**<sup>29</sup>.

En HTML5 puedes emplear dos sintaxis<sup>30</sup>:

- **HTML**: la recomendada para la mayoría de los autores de páginas web. Cuando se envía una página web con esta sintaxis se debe emplear el tipo MIME `text/html`.
- **XHTML**: una aplicación de XML, la sintaxis es más estricta que la anterior. Se recomienda si se quiere hacer uso de software para XML. Cuando se envía una página web con esta sintaxis se debe emplear el tipo MIME `application/xhtml+xml`.

El siguiente código lo puedes utilizar para crear páginas web basadas en XHTML5 (HTML5 escrito con la sintaxis XML):

```
<!DOCTYPE html>
<html lang="es">
<!-- La cabecera -->
<head>
```

<sup>25</sup>[http://www.456bereastreet.com/lab/developing\\_with\\_web\\_standards/](http://www.456bereastreet.com/lab/developing_with_web_standards/)

<sup>26</sup><http://www.w3schools.com/html/default.asp>

<sup>27</sup>[http://www.w3schools.com/html/html\\_xhtml.asp](http://www.w3schools.com/html/html_xhtml.asp)

<sup>28</sup>[http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)

<sup>29</sup><http://olgacarreras.blogspot.com/2007/02/plantilla-base-xhtml.html>

<sup>30</sup>W3C HTML5 1.6 HTML vs XHTML: <http://www.w3.org/TR/html5/introduction.html#html-vs-xhtml>

```

<meta charset="utf-8" />
<meta name="generator" content="Bloc de notas" />
<meta name="author" content="Sergio Luján Mora" />
<meta name="keywords" content="HTML5, web" />
<meta name="description" content="Plantilla base de una página creada con HTML5" />
<title>Plantilla base de HTML5</title>
</head>
<!-- El cuerpo -->
<body>
<p>
Bla, bla, bla.
</p>
</body>
</html>

```

Una vez hayas aprendido los conceptos básicos, consulta la guía de referencia de W3Schools (**HTML Reference**<sup>31</sup>) para conocer todas las etiquetas y atributos. Te vendrá muy bien recordar esta página web cuando estés escribiendo código HTML y te surja una duda.

En esta práctica sólo tienes que definir la estructura y contenido de las páginas web, el estilo visual lo definiremos en prácticas posteriores (utilizando únicamente HTML es imposible cambiar el tipo de letra o el color de un texto en una página web). Emplea las etiquetas de HTML que poseen contenido semántico, como `<h1>`, `<p>`, `<ul>`, etc. Fíjate que las etiquetas `<center>` o `<font>` no existen ni en XHTML 1.0 ni en HTML5, aunque las puedes encontrar en muchas páginas web “mal hechas”. No utilices tablas para maquetar una página web. Para todo ello emplearemos la tecnología CSS que utilizaremos en una próxima práctica, **en esta práctica no debes definir la presentación visual de las páginas web**.

Si quieres conocer más de XHTML, consulta la especificación oficial del W3C (**XHTML 1.0 The Extensible HyperText Markup Language**<sup>32</sup>) y los DTD de las tres versiones (**Strict**<sup>33</sup>, **Transitional**<sup>34</sup> y **Frameset**<sup>35</sup>). ¿Qué es un DTD? Es una especificación que define la estructura y los elementos que componen un documento.

Si quieres conocer más de HTML5, consulta la especificación oficial del W3C (**HTML5: A vocabulary and associated APIs for HTML and XHTML**<sup>36</sup>).

Cuando escribas el código HTML a mano, utiliza la herramienta **W3C Markup Validation Service**<sup>37</sup> para comprobar que el código es correcto y la herramienta **HTML Tidy**<sup>38</sup> para obtener una mejor apariencia visual del código fuente.

La herramienta de validación del W3C sólo permite validar las páginas web de una en una. Si necesitas validar todo un sitio web, la herramienta **WDG HTML Validator**<sup>39</sup> permite validar múltiples páginas a partir de una única dirección.

Cuando se empieza a trabajar con una nueva tecnología es muy normal cometer muchos errores. ¿Cuales son los errores más comunes? El artículo **Errores comunes del desarrollo web**<sup>40</sup> te lo explica.

El uso del atributo `placeholder` en lugar de la etiqueta `<label>` asociada a un control de un formulario es una malísima idea. Se puede usar `placeholder`, pero siempre como un complemento a la etiqueta `<label>`.

Cuando valides una página web que tenga un formulario con un control de tipo `<input type="date">` te puede aparecer el siguiente warning: “The date input type is not supported in all browsers. Please be sure to test, and consider using a polyfill”. No te preocupes, no es un error, es una advertencia, te está avisando de que ese control puede no visualizarse y funcionar correctamente en algunos navegadores web.

#### **Importante:**

- **Almacena cada práctica en un directorio independiente; al final de la asignatura debes tener un directorio por cada práctica semanal. Cuando comiences una nueva práctica, copia y renombra el directorio de la práctica anterior.**

<sup>31</sup><http://www.w3schools.com/tags/default.asp>

<sup>32</sup><http://www.w3.org/TR/xhtml1/>

<sup>33</sup><http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>

<sup>34</sup><http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>

<sup>35</sup><http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd>

<sup>36</sup><http://www.w3.org/TR/html5/>

<sup>37</sup><http://validator.w3.org>

<sup>38</sup><http://cgi.w3.org/cgi-bin/tidy>

<sup>39</sup><http://htmlhelp.com/tools/validator/>

<sup>40</sup>[http://www.456bereastreet.com/lab/web\\_development\\_mistakes/es/](http://www.456bereastreet.com/lab/web_development_mistakes/es/)

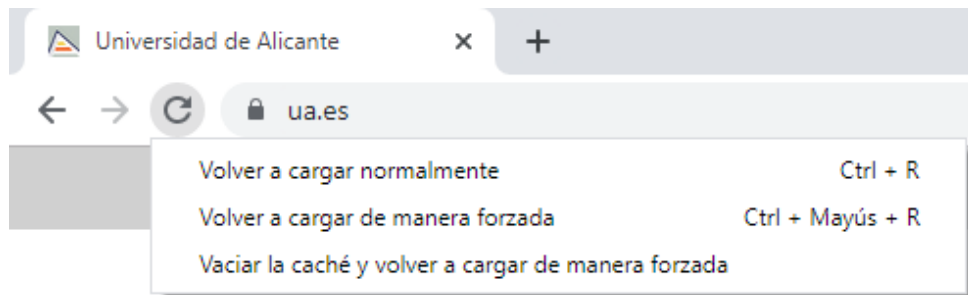


Figura 2: Opciones para volver a cargar una página web en Google Chrome

- Se recomienda usar el juego de caracteres UTF-8.
- Ten cuidado con los nombres de los ficheros, utiliza únicamente letras del alfabeto inglés y números, no uses espacios en blanco y emplea únicamente minúsculas.
- Ten cuidado con la caché del navegador, consulta una explicación sobre los problemas y soluciones en el artículo “Ayuda:Cómo limpiar la caché”<sup>41</sup>. En Google Chrome, cuando se muestran las herramientas para inspeccionar el código de una página, se puede pulsar con el botón derecho sobre el icono de recargar y aparece un menú para forzar la recarga y el vaciado de la caché, tal como se puede ver en la Figura 2. Recuerda esto durante toda la asignatura.

<sup>41</sup>[https://es.wikipedia.org/wiki/Ayuda:C%C3%B3mo\\_limpiar\\_la\\_cach%C3%A9](https://es.wikipedia.org/wiki/Ayuda:C%C3%B3mo_limpiar_la_cach%C3%A9)