

## Prácticas

Estilo:  ▼ Tamaño letra:  ▼ Interlineado:  ▼

# Desarrollo de Aplicaciones Web

## Práctica 2: HTML

### 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.

### 2. Recursos

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

- **W3Schools** [\(1\)](#): cursos de aprendizaje y guías de referencia de diversas tecnologías empleadas en la programación web.
- **W3C** [\(2\)](#): organismo internacional que desarrolla las especificaciones de las principales tecnologías que se emplean en la programación web.
- **XHTML Cheat Sheet** [\(3\)](#): resumen en una sola página de lo más importante de XHTML.
- **HTML Cheat Sheet** [\(4\)](#): resumen en una sola cara de lo más importante de (X)HTML.
- **HTML5 Cheat Sheet** [\(5\)](#): resumen de HTML5 de la famosa revista Smashing Magazine. Es del año 2009, así que seguramente faltan algunas de las últimas novedades.
- **HTML Tags** [\(6\)](#): 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** [\(7\)](#): servicio de validación para diferentes lenguajes de marcado (HTML, XHTML, SVG, y otros).
- **Html Validator** [\(8\)](#): complemento para Mozilla Firefox que permite comprobar el número de errores que posee una página web que se visualiza.
- **WDG HTML Validator** [\(9\)](#): herramienta de validación de XHTML que permite validar múltiples páginas web (hasta un máximo de 100) a partir de una dirección.

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

- **Browsershots** [\(10\)](#): permite comprobar cómo se visualiza una página web en decenas de navegadores y en distintos sistemas operativos.
- **Total Validator Service** [\(11\)](#): 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.
- **BrowserCam** [\(12\)](#): permite comprobar la compatibilidad de una página web con diferentes navegadores (incluso con navegadores de PDA y teléfonos móviles). Es un servicio de pago.

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

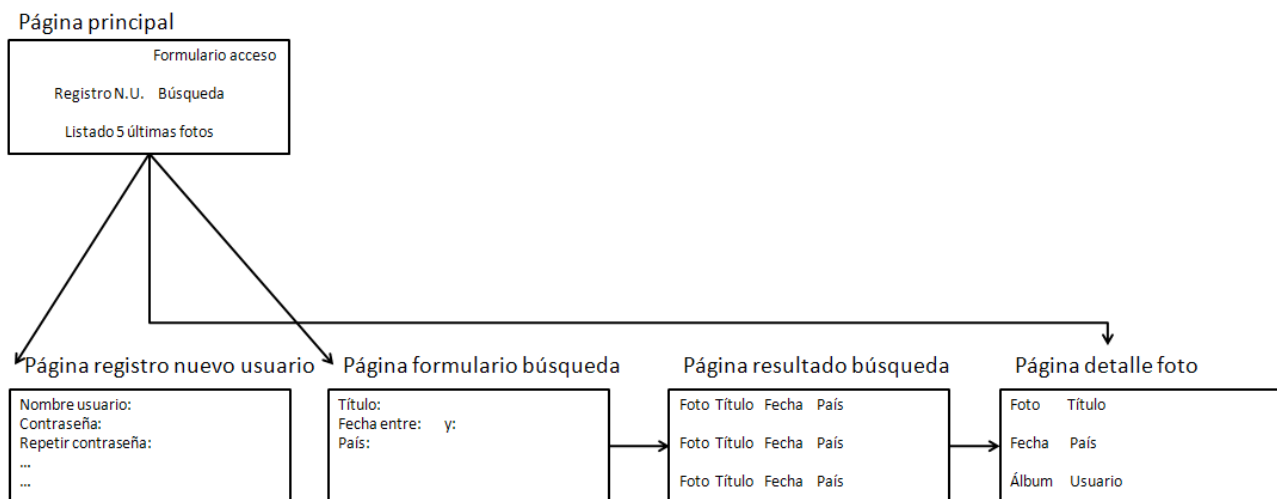
- **Notepad++** [\(13\)](#): editor gratuito de código fuente que soporta varios lenguajes de programación. Entre otras características, posee sintaxis coloreada, envoltura de sintaxis y autocompletado.
- **Sublime Text** [\(14\)](#): excelente editor compatible con múltiples lenguajes de programación 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** [\(15\)](#): editor de HTML disponible para diferentes sistemas operativos.
- **Visual Studio Code** [\(16\)](#): editor de Microsoft disponible para diferentes sistemas operativos.
- **Real-time HTML Editor** [\(17\)](#): editor WYSIWYG de HTML en página web.
- **Online-HTML-Editor.org** [\(18\)](#): editor WYSIWYG de HTML en página web que permite abrir y guardar una página web.
- **HTML Tidy** [\(19\)](#): limpia y organiza el código HTML. También se puede descargar el ejecutable y el código fuente para ejecutarlo en local.

### 3. ¿Qué tengo que hacer?

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:



**Figura 1:** Diagrama de páginas que componen el sitio web

#### 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.

#### 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.

#### **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.

#### **Página detalle foto**

Muestra toda la información sobre una foto seleccionada en la página anterior (foto, título, fecha, país, álbum de fotos y usuario al que pertenece).

Por ahora, estas páginas contienen datos estáticos que en prácticas posteriores se sustituirán por datos reales obtenidos de una base de datos. El modelo de datos que se debe emplear 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 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 registrado y otra para usuario no registrado, eso ya se programará más adelante. Pero la página principal sí que se tiene que realizar con sus dos alternativas: puede cambiar la barra de menú, el enlace de entrar o salir, etc. Cuando se inicie sesión como usuario registrado, se debe cambiar a la versión de la página como usuario registrado; cuando se pulse en salir o cerrar sesión, se debe cambiar a la versión de la página como usuario no registrado.

**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 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.**

## **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.**

## 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** [\(20\)](#) 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** [\(21\)](#).
- **Diez razones para aprender y usar los estándares web** [\(22\)](#).
- **Developing With Web Standards - Recommendations and best practices** [\(23\)](#).

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** [\(24\)](#).
- **HTML - XHTML** [\(25\)](#).
- **HTML5 Tutorial** [\(26\)](#).

¿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** [\(27\)](#).

En HTML5 puedes emplear dos sintaxis [\(28\)](#):

- 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>  
<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** [\(29\)](#)) 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** [\(30\)](#)) y los DTD de las tres versiones (**Strict** [\(31\)](#), **Transitional** [\(32\)](#) y **Frameset** [\(33\)](#)). ¿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** [\(34\)](#)).

Cuando escribas el código HTML a mano, utiliza la herramienta **W3C Markup Validation Service** [\(35\)](#) para comprobar que el código es correcto y la herramienta **HTML Tidy** [\(36\)](#) 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** [\(37\)](#) 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** [\(38\)](#) 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>.

### 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.**
- **El fichero de la página principal de un sitio web suele tener el nombre index.html. Sólo como curiosidad, los servidores web suelen tener configurada la siguiente lista de páginas por defecto para la página principal [\(39\)](#):** 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.
- **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” [\(40\)](#). Recuerda esto durante toda la asignatura.**

---

(1) <http://www.w3schools.com>

(2) <http://www.w3.org>

(3) <http://floeLe.flyspray.org/htmlcheatsheet-ncr.pdf>

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

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

(6) <http://html-tags.info/>

(7) <http://validator.w3.org>

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

(9) <http://htmlhelp.com/tools/validator/>

(10) <http://browsershots.org>

(11) <http://www.totalvalidator.com/>

(12) <http://www.browsercam.com/>

(13) <http://notepad-plus.sourceforge.net/es/site.htm>

(14) <http://www.sublimetext.com/>

(15) <http://brackets.io/>

(16) <https://code.visualstudio.com/download>

(17) <http://htmledit.squarefree.com/>

(18) <http://online-html-editor.org/>

(19) <http://cgi.w3.org/cgi-bin/tidy>

- (20) <http://www.w3csites.com/>
- (21) <http://www.webstandards.org/about/mission/es/>
- (22) <http://www.estandaresyaccesibilidad.com/2006/01/diez-razones-para-aprender-y-usar-los.html>
- (23) [http://www.456bereastreet.com/lab/developing\\_with\\_web\\_standards/](http://www.456bereastreet.com/lab/developing_with_web_standards/)
- (24) <http://www.w3schools.com/html/default.asp>
- (25) [http://www.w3schools.com/html/html\\_xhtml.asp](http://www.w3schools.com/html/html_xhtml.asp)
- (26) [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp)
- (27) <http://olgacarreras.blogspot.com/2007/02/plantilla-base-xhtml.html>
- (28) W3C HTML5 1.6 HTML vs XHTML: <http://www.w3.org/TR/html5/introduction.html\#html-vs-xhtml>
- (29) <http://www.w3schools.com/tags/default.asp>
- (30) <http://www.w3.org/TR/xhtml1/>
- (31) <http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>
- (32) <http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
- (33) <http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd>
- (34) <http://www.w3.org/TR/html5/>
- (35) <http://validator.w3.org>
- (36) <http://cgi.w3.org/cgi-bin/tidy>
- (37) <http://htmlhelp.com/tools/validator/>
- (38) [http://www.456bereastreet.com/lab/web\\_development\\_mistakes/es/](http://www.456bereastreet.com/lab/web_development_mistakes/es/)
- (39) El orden de configuración es importante porque el servidor web va a intentar encontrar los archivos en el orden indicado.
- (40) [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)