Capítulo 1
Desarrollo web

1.1 Sitios web

Los sitios web son archivos que los usuarios descargan con sus navegadores desde ordenadores remotos. Cuando un usuario decide acceder a un sitio web, le comunica al navegador la dirección del sitio y el programa descarga los archivos, procesa su contenido y lo muestra en pantalla.

Debido a que los sitios webs son de acceso público e Internet es una red global, estos archivos deben estar siempre disponibles. Por este motivo, los sitios web no se almacenan en ordenadores personales, sino en ordenadores especializados diseñados para despachar estos archivos a los usuarios que los solicitan. El ordenador que almacena los archivos y datos de un sitio web se llama *servidor* y el ordenador que accede a esta información se llama *cliente*, tal como lo ilustra la Figura 1-1.

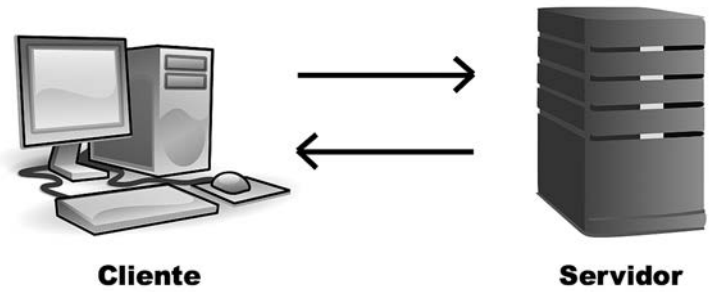

Figura 1-1: *Clientes y servidores*

Los servidores son muy similares a los ordenadores personales, con la diferencia de que están continuamente conectados a la red y ejecutando programas que les permiten responder a las solicitudes de los usuarios, sin importar cuándo se reciben o de donde proceden. Los programas más populares para servidores son Apache, para sistemas Linux, e IIS (Internet Information Server), creado por Microsoft para sistemas Windows. Entre otras funciones, estos programas son responsables de establecer la conexión entre el cliente y el servidor, controlar el acceso de los usuarios, administrar los archivos, y despachar los documentos y recursos requeridos por los clientes.

Archivos

Los sitios web están compuestos de múltiples documentos que el navegador descarga cuando el usuario los solicita. Los documentos que conforman un sitio web se llaman *páginas* y el proceso de abrir nuevas páginas *navegar* (el usuario navega a través de las páginas del sitio). Para desarrollar un sitio web, tenemos que crear un archivo por cada página que queremos incluir. Junto con estos archivos, también debemos incluir los archivos con las imágenes y cualquier otro recurso que queremos mostrar dentro de estas páginas (las imágenes y otros

medios gráficos se almacenan en archivos aparte). En la Figura 1-2 se representa cómo se muestran los directorios y archivos de un sitio web una vez que se suben al servidor.

Figura 1-2: *Archivos de un sitio web*

En el ejemplo de la Figura 1-2 se incluyen dos directorios llamados *imagenes* y *recursos,* y tres archivos llamados *contacto.html*, *index.html* y *news.html*. Los directorios se crearon para almacenar las imágenes que queremos mostrar dentro de las páginas web y otros recursos, como los archivos que contienen los códigos en CSS y JavaScript. Por otro lado, los archivos de este ejemplo representan las tres páginas web que queremos incluir en este sitio. El archivo index.html contiene el código y la información correspondiente a la página principal (la página que el usuario ve cuando entra a nuestro sitio web por primera vez), el archivo contacto.html contiene el código necesario para presentar un formulario que el usuario puede rellenar para enviarnos un mensaje y el archivo noticias.html contiene el código necesario para mostrar las noticias que queremos compartir con nuestros usuarios. Cuando un usuario accede a nuestro sitio web por primera vez, el navegador descarga el archivo index.html y muestra su contenido en la ventana. Si el usuario realiza una acción para ver las noticias ofrecidas por nuestro sitio web, el navegador descarga el archivo noticias.html desde el servidor y reemplaza el contenido del archivo index.html por el contenido de este nuevo archivo. Cada vez que el usuario quiere acceder a una nueva página web, el navegador tiene que descargar el correspondiente archivo desde el servidor, procesarlo y mostrar su contenido en la pantalla.

Los archivos de un sitio web son iguales que los archivos que podemos encontrar en un ordenador personal. Todos tiene un nombre seleccionado por el desarrollador y una extensión que refleja el lenguaje usado para programar su contenido (en nuestro ejemplo, los archivos tienen la extensión .html porque fueron programados en HTML). Aunque podemos asignar cualquier nombre que queramos a estos archivos, el archivo que genera la página inicial presenta algunos requisitos. Algunos servidores como Apache designan archivos por defecto en caso de que el usuario no especifique ninguno. El nombre utilizado con más frecuencia es *index*. Si un usuario accede al servidor sin especificar el nombre del archivo que intenta abrir, el servidor busca un archivo con el nombre *index* y lo envía de vuelta al cliente. Por esta razón, el archivo index es el punto de entrada de nuestro sitio web y siempre debemos incluirlo.

IMPORTANTE: los servidores son flexibles en cuanto a los nombres que podemos asignar a nuestros archivos, pero existen algunas reglas que debería seguir para asegurarse de que sus archivos son accesibles. Evite usar espacios. Si necesita separar palabras use el guion bajo en su lugar (_). Además, debe considerar que algunos caracteres realizan funciones específicas en la Web, por lo que es mejor evitar caracteres especiales como ?, %, #, /, y usar solo letras minúsculas sin acentos y números.

Lo básico: aunque *index* es el nombre más común, no es el único que podemos asignar al archivo por defecto. Algunos servidores designan otros nombres como *home* o *default*, e incluyen diferentes extensiones. Por ejemplo, si en lugar de programar nuestros documentos en HTML lo hacemos en un lenguaje de servidor como PHP, debemos asignar a nuestro archivo index el nombre *index.php*. El servidor contiene una lista de archivos y continúa buscando hasta que encuentra uno que coincida con esa lista. Por ejemplo, Apache primero busca por un archivo con el nombre index y la extensión .html, pero si no lo encuentra, busca por un archivo con el nombre index y la extensión .php. Estudiaremos HTML y PHP más adelante en este y otros capítulos.

Dominios y URL

Los servidores se identifican con un valor llamado *IP* (Internet Protocol). Esta IP es única para cada ordenador y, por lo tanto, trabaja como una dirección que permite ubicar a un ordenador dentro de una red. Cuando el navegador tiene que acceder a un servidor para descargar el documento solicitado por el usuario, primero busca el servidor a través de esta dirección IP y luego le pide que le envíe el documento.

Las direcciones IP están compuestas por números enteros entre 0 y 255 separados por un punto, o números y letras separadas por dos puntos, dependiendo de la versión (IPv4 o IPv6). Por ejemplo, la dirección 216.58.198.100 corresponde al servidor donde se encuentra alojado el sitio web de Google. Si escribimos esta dirección IP en la barra de navegación de nuestro navegador, la página inicial de Google se descarga y muestra en pantalla.

En teoría, podríamos acceder a cualquier servidor utilizando su dirección IP, pero estos valores son crípticos y difíciles de recordar. Por esta razón, Internet utiliza un sistema que identifica a cada servidor con un nombre específico. Estos nombres personalizados, llamados *dominios*, son identificadores sencillos que cualquier persona puede recordar, como *google* o *yahoo*, con una extensión que determina el propósito del sitio web al que hacen referencia, como *.com* (comercial) o *.org* (organización). Cuando el usuario le pide al navegador que acceda al sitio web con el dominio www.google.com, el navegador accede primero a un servidor llamado *DNS* que contiene una lista de dominios con sus respectivas direcciones IP. Este servidor encuentra la IP 216.58.198.100 asociada al dominio www.google.com, la devuelve al navegador, y entonces el navegador accede al sitio web de Google por medio de esta IP. Debido a que las direcciones IP de los sitios web siempre se encuentran asociadas a sus dominios, no necesitamos recordar la dirección de un servidor para acceder a él, solo tenemos que recordar el domino y el navegador se encarga de encontrar el servidor y descargar los archivos por nosotros.

Los sitios web están compuestos por múltiples archivos, por lo que debemos agregar el nombre del archivo al dominio para indicar cuál queremos descargar. Esta construcción se llama *URL* e incluye tres partes, tal como se describe en la Figura 1-3.

Figura 1-3: *URL*

La primera parte de la URL es una cadena de caracteres que representa el protocolo de comunicación que se utilizará para acceder al recurso (el protocolo creado para la Web se llama *HTTP*), el siguiente componente es el dominio del sitio web y el último componente es el nombre del recurso que queremos descargar (puede ser un archivo, como en nuestro ejemplo, o una ruta a seguir que incluye el directorio donde el archivo se encuentra almacenado (por ejemplo, http://www.ejemplo.com/imagenes/milogo.jpg). La URL en nuestro ejemplo le pide al navegador que utilice el protocolo HTTP para acceder al archivo contacto.html, ubicado en el servidor identificado con el domino www.ejemplo.com.

Las URL se utilizan para ubicar cada uno de los documentos en el sitio web y son, por lo tanto, necesarias para navegar por el sitio. Si el usuario no especifica ningún archivo, el servidor devuelve el archivo por defecto, pero de ahí en adelante, cada vez que el usuario realiza una acción para abrir una página diferente, el navegador debe incluir en la URL el nombre del archivo que corresponde a la página solicitada.

IMPORTANTE: una vez que ha conseguido el dominio para su sitio web, puede crear subdominios. Los subdominios son enlaces directos a directorios y nos permiten crear múltiples sitios web en una misma cuenta. Un subdominio se crea con el nombre del directorio y el dominio conectados por un punto. Por ejemplo, si su dominio es www.ejemplo.com y luego crea un subdominio para un directorio llamado *recursos*, podrá acceder directamente al directorio escribiendo en el navegador la URL http://recursos.ejemplo.com.

Lo básico: existen diferentes protocolos que los ordenadores utilizan para comunicarse entre ellos, y transferir recursos y datos. HTTP (HyperText Transfer Protocol) es el protocolo de comunicación que se utiliza para acceder a documentos web. Siempre tenemos que incluir el prefijo HTTP en la URL cuando el recurso al que estamos tratando de acceder pertenece a un sitio web, pero en la práctica esto no es necesario porque los navegadores lo hacen de forma automática. Existe otra versión disponible de este protocolo llamado *HTTPS*. La S indica que la conexión es encriptada por protocolos de encriptación como TLS o SSL. Los sitios web pequeños no necesitan encriptación, pero se recomienda utilizarla en sitios web que manejan información sensible.

Hipervínculos

En teoría, podemos acceder a todos los documentos de un sitio web escribiendo la URL en la barra de navegación del navegador. Por ejemplo, si queremos acceder a la página inicial en español del sitio web *For Masterminds*, podemos insertar la URL http://www.formasterminds.com/esindex.php, o bien insertar la URL http://www.formasterminds.com/escontact.php para abrir la página que nos permite enviar un mensaje a su desarrollador. Aunque es posible acceder a todos los archivos del sitio web usando este método, no es práctico. En primer lugar, los usuarios no conocen los nombres que el desarrollador eligió para cada archivo y, por lo tanto, estarán limitados a aquellos nombres que pueden adivinar o solo a la página principal que devuelve por defecto. En segundo lugar, los sitios web pueden estar compuestos por docenas o incluso miles de páginas web (algunos

sitios contienen millones) y la mayoría de los documentos serían imposibles de encontrar. La solución se encontró con la definición de hipervínculos. Los hipervínculos, también llamados *enlaces*, son referencias a documentos dentro de las páginas de un sitio web. Incorporando estos enlaces, una página puede contener referencias a otras páginas. Si el usuario hace clic con el ratón en un enlace, el navegador sigue esa referencia y el documento indicado por la URL de la referencia se descarga y muestra en pantalla. Debido a estas conexiones entre páginas, los usuarios pueden navegar en el sitio web y acceder a todos sus documentos simplemente haciendo clic en sus enlaces.

Lo básico: los enlaces son lo que transforma a un grupo de archivos en un sitio web. Para crear un sitio web, debe programar los documentos correspondientes a cada página e incluir dentro de las mismas los enlaces que establecen una ruta que el usuario puede seguir para acceder a cada una de ellas. Estudiaremos cómo incorporar enlaces en nuestros documentos en el Capítulo 2.

URL absolutas y relativas

Los hipervínculos son procesados por el navegador antes de ser usados para acceder a los documentos. Por esta razón, se pueden definir con URL absolutas o relativas. Las URL absolutas son aquellas que incluyen toda la información necesaria para acceder al recurso (ver Figura 1-3), mientras que las relativas son aquellas que solo declaran la parte de la ruta que el navegador tiene que agregar a la URL actual para acceder al recurso. Por ejemplo, si tenemos un hipervínculo dentro de un documento que referencia una imagen dentro del directorio imagenes, podemos crear el enlace con la URL http://www.ejemplo.com/imagenes/miimagen.png, pero también tenemos la opción de declararla como "imagenes/miimagen.png" y el navegador se encargará de agregar a esta ruta la URL actual y descargar la imagen.

Las URL relativas no solo pueden determinar una ruta hacia abajo, sino también hacia arriba de la jerarquía. Por ejemplo, si tenemos un documento dentro del directorio recursos en el ejemplo de la Figura 1-2 y queremos acceder a un documento en el directorio raíz, podemos crear una URL relativa usando los caracteres ../ al comienzo de la ruta. Si el documento que queremos acceder es noticias.html, la URL relativa sería ../noticias.html. Los dos puntos .. le indican al navegador que el documento al que queremos acceder se encuentra dentro del directorio padre del actual directorio (recursos, en nuestro ejemplo).

1.2 Lenguajes

Como mencionamos en la introducción, HTML5 incorpora tres características (estructura, estilo, y funcionalidad), con lo que integra tres lenguajes de programación independientes: HTML, CSS, y JavaScript. Estos lenguajes están compuestos por grupos de instrucciones que los navegadores pueden interpretar para procesar y mostrar los documentos al usuario. Para crear nuestros documentos, tenemos que aprender todas las instrucciones incluidas en estos lenguajes y saber cómo organizarlas.

HTML

HTML (*HyperText Markup Language*) es un lenguaje compuesto por un grupo de etiquetas definidas con un nombre rodeado de paréntesis angulares. Los paréntesis angulares delimitan la etiqueta y el nombre define el tipo de contenido que representa. Por ejemplo, la etiqueta **<html>** indica que el contenido es código HTML. Algunas de estas etiquetas son declaradas individualmente (por ejemplo, **
) y otras son declaradas en pares, que incluyen una de apertura y otra de cierre, como **<html></html> (en la etiqueta de cierre el nombre va precedido por una barra invertida). Las etiquetas individuales y las de apertura pueden incluir atributos para ofrecer información adicional acerca de sus contenidos (por ejemplo, **<html lang="es">**). Las etiquetas individuales y la combinación de etiquetas de apertura y cierre se llaman *elementos*. Los elementos compuestos por una sola etiqueta se usan para modificar el contenido que los rodea o incluir recursos externos, mientras que los elementos que incluyen etiquetas de apertura y cierre se utilizan para delimitar el contenido del documento, tal como ilustra la Figura 1-4.

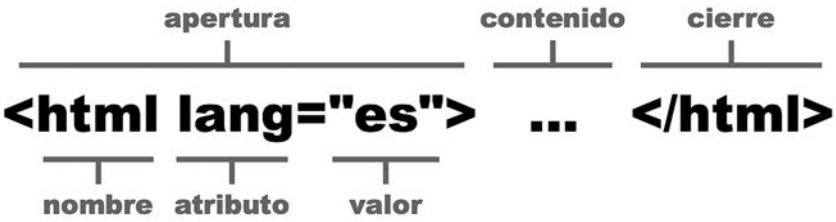

Figura 1-4: *Elemento HTML*

Se deben combinar múltiples elementos para definir un documento. Los elementos son listados en secuencia de arriba abajo y pueden contener otros elementos en su interior. Por ejemplo, el elemento **<html>** que se muestra en la Figura 1-4 declara que su contenido debe ser interpretado como código HTML. Por lo tanto, el resto de los elementos que describen el contenido de ese documento se deben declarar entre las etiquetas **<html>** y **</html>**. A su vez, los elementos dentro del elemento **<html>** pueden incluir otros elementos. El siguiente ejemplo muestra un documento HTML sencillo que incluye todos los elementos necesarios para definir una estructura básica y mostrar el mensaje HOLA MUNDO! en la pantalla.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Mi primer documento HTML</title>
  </head>
  <body>
    <p>HOLA MUNDO!</p>
  </body>
</html>
```

Listado 1-1: *Creando un documento HTML*

En el ejemplo del Listado 1-1 presentamos un código sencillo, pero con una estructura compleja. En la primera línea, se encuentra una etiqueta individual que declara el tipo de documento (**<!DOCTYPE html>**) seguida por una etiqueta de apertura **<html lang="es">**. Entre las etiquetas **<html>** y **</html>** se incluyen otros elementos que representan la cabecera y el cuerpo del documento (**<head>** y **<body>**), los cuales a su vez encierran más elementos con sus respectivos contenidos (**<title>** y **<p>**), demostrando cómo se compone un documento HTML. Los elementos se listan uno a continuación de otro y también dentro de otros elementos, de modo que se construye una estructura de tipo árbol con el elemento **<html>** como raíz.

Lo básico: en general, todo elemento puede ser anidado, convertirse en un contenedor o ser contenido por otros elementos. Los elementos exclusivamente estructurales como **<html>**, **<head>** y **<body>** tienen un lugar específico en un documento HTML, pero el resto son flexibles, tal como veremos en el Capítulo 2.

Como ya mencionamos, las etiquetas individuales y de apertura pueden incluir atributos. Por ejemplo, la etiqueta de apertura **<html>** declarada en el Listado 1-1 no está solo compuesta por el nombre **html** y los paréntesis angulares, sino también por el texto **lang="es"**. Este es un atributo con un valor. El nombre del atributo es **lang** y el valor **es** se asigna al atributo usando el carácter **=**. Los atributos ofrecen información adicional acerca del elemento y su contenido. En este caso, el atributo **lang** declara el idioma del contenido del documento (**es** por Español).

Lo básico: los atributos se declaran siempre dentro de la etiqueta de apertura (o etiquetas individuales) y pueden tener una estructura que incluye un nombre y un valor, como el atributo **lang** de la etiqueta **<html>**, o representar un valor por sí mismos, como el atributo **html** de la etiqueta **<!DOCTYPE>**. Estudiaremos los elementos HTML y sus atributos en el Capítulo 2.

CSS

CSS (*Cascading Style Sheets*) es el lenguaje que se utiliza para definir los estilos de los elementos HTML, como el tamaño, el color, el fondo, el borde, etc. Aunque todos los navegadores asignan estilos por defecto a la mayoría de los elementos, estos estilos generalmente están lejos de lo que queremos para nuestros sitios web. Para declarar estilos personalizados, CSS utiliza propiedades y valores. Esta construcción se llama *declaración* y su sintaxis incluye dos puntos después del nombre de la propiedad, y un punto y coma al final para cerrar la línea.

Figura 1-5: *Propiedad CSS*

En el ejemplo de la Figura 1-5, el valor **#FF0000** se asigna a la propiedad **color**. Si esta propiedad se aplica luego a un elemento HTML, el contenido de ese elemento se mostrará en color rojo (el valor **#FF0000** representa el color rojo).

Las propiedades CSS se pueden agrupar usando llaves. Un grupo de una o más propiedades se llama *regla* y se identifica por un nombre llamado *selector*.

```
body {
  width: 100%;
  margin: 0px;
  background-color: #FF0000;
}
```

Listado 1-2: *Declarando reglas CSS*

El Listado 1-2 declara una regla con tres propiedades: **`width`**, **`margin`** y **`background-color`**. Esta regla se identifica con el nombre **`body`**, lo que significa que las propiedades serán aplicadas al elemento **`<body>`**. Si incluimos esta regla en un documento, el contenido del documento se extenderán hacia los límites de la ventana del navegador y tendrán un fondo rojo.

Lo básico: existen diferentes técnicas para aplicar estilos CSS a elementos HTML. Estudiaremos las propiedades CSS y cómo incluirlas en un documento HTML en los Capítulos 3 y 4.

JavaScript

A diferencia de HTML y CSS, JavaScript es un lenguaje de programación. Para ser justos, todos estos lenguajes pueden ser considerados lenguajes de programación, pero en la práctica existen algunas diferencias en la forma en la que suministran las instrucciones al navegador. HTML es como un grupo de indicadores que el navegador interpreta para organizar la información, CSS puede ser considerado como una lista de estilos que ayudan al navegador a preparar el documento para ser presentado en pantalla (aunque la última especificación lo convirtió en un lenguaje más dinámico), pero JavaScript es un lenguaje de programación, comparable con cualquier otro lenguaje de programación profesional como C++ o Java. JavaScript difiere de los demás lenguajes en que puede realizar tareas personalizadas, desde almacenar valores hasta calcular algoritmos complejos, incluida la capacidad de interactuar con los elementos del documento y procesar su contenido dinámicamente.

Al igual que HTML y CSS, JavaScript se incluye en los navegadores y, por lo tanto, se encuentra disponible en todos nuestros documentos. Para declarar código JavaScript dentro de un documento, HTML ofrece el elemento **`<script>`**. El siguiente ejemplo es una muestra de un código escrito en JavaScript.

```
<script>
  function cambiarColor() {
    document.body.style.backgroundColor = "#0000FF";
  }
  document.addEventListener("click", cambiarColor);
</script>
```

Listado 1-3: *Declarando código JavaScript*

El código en el Listado 1-3 cambia el color de fondo del elemento **<body>** a azul cuando el usuario hace clic en el documento.

Lo básico: con el elemento **<script>** también podemos cargar código JavaScript desde archivos externos. Estudiaremos el elemento **<script>** en el Capítulo 2 y el lenguaje JavaScript en el Capítulo 6.

Lenguajes de servidor

Los códigos programados en HTML, CSS, y JavaScript son ejecutados por el navegador en el ordenador del usuario (el cliente). Esto significa que, después de que los archivos del sitio web se suben al servidor, permanecen inalterables hasta que se descargan en un ordenador personal y sus códigos son ejecutados por el navegador. Aunque esto permite la creación de sitios web útiles e interactivos, hay momentos en los cuales necesitamos procesar la información en el servidor antes de enviarla al usuario. El contenido producido por esta información se denomina *contenido dinámico*, y es generado por códigos ejecutados en el servidor y programados en lenguajes que fueron especialmente diseñados con este propósito (lenguajes de servidor). Cuando el navegador solicita un archivo que contiene este tipo de código, el servidor lo ejecuta y luego envía el resultado como respuesta al usuario. Estos códigos no solo se utilizan para generar contenido y documentos en tiempo real, sino también para procesar la información enviada por el navegador, almacenar datos del usuario en el servidor, controlar cuentas, etc.

Existen varios lenguajes disponibles para crear código ejecutable en los servidores. Los más populares son PHP, Ruby, y Python. El siguiente ejemplo es una muestra de un código escrito en PHP.

```
<?php
  $nombre = $_GET['minombre'];
  print('Su nombre es: '.$nombre);
?>
```

Listado 1-4: *Declarando código ejecutable en el servidor*

El código del Listado 1-4 recibe un valor enviado por el navegador, lo almacena en la memoria y crea un mensaje con el mismo. Cuando se ejecuta este código, se crea un nuevo documento que contiene el mensaje final, el archivo se envía de vuelta al cliente y finalmente el navegador muestra su contenido en pantalla.

IMPORTANTE: los lenguajes de servidor utilizan su propia tecnología, pero trabajan junto con HTML5 para llevar un registro de las cuentas de usuarios, almacenar información en el servidor, manejar bases de datos, etc. El tema va más allá del propósito de este libro. Para obtener más información sobre cómo programar en PHP, Ruby, o Python, descargue los contenidos adiconales en **www.marcombo.info**.

1.3 Herramientas

Crear un sitio web involucra múltiples pasos. Tenemos que programar los documentos en HTML, crear los archivos con los estilos CSS y los códigos en JavaScript, configurar el servidor

que hará que el sitio sea visible a los usuarios y transferir todos los archivos desde nuestro ordenador al servidor. Por fortuna existen muchas herramientas disponibles que nos pueden ayudar con estas tareas. Estas herramientas son muy fáciles de usar y la mayoría se ofrecen de forma gratuita.

IMPORTANTE: en esta sección del capítulo introducimos todas las herramientas que necesitará para crear sus sitios web y ofrecerlos a sus usuarios. Esto incluye las herramientas requeridas para programar y diseñar un sitio web, pero también otras que necesitará para configurarlo y probarlo antes de hacerlo público. La mayoría de los ejemplos de este libro no tienen que subirse a un servidor para poder trabajar adecuadamente y, por lo tanto, puede ignorar parte de esta información hasta que sea requerida por sus proyectos.

Editores

Los documentos HTML, así como los archivos CSS y JavaScript, son archivos de texto, por lo que podemos usar cualquier editor incluido en nuestro ordenador para crearlos, como el bloc de notas de Windows o la aplicación editor de texto de los ordenadores de Apple, pero también existen editores de texto especialmente diseñados para programadores y desarrolladores web que pueden simplificar nuestro trabajo. Estos editores resaltan texto con diferentes colores para ayudarnos a identificar cada parte del código, o listan los archivos de un proyecto en un panel lateral para ayudarnos a trabajar con múltiples archivos al mismo tiempo. La siguiente es una lista de los editores y de IDE (Integrated Development Environments) más populares disponibles para ordenadores personales y ordenadores de Apple.

- **Atom** (www.atom.io) es un editor gratuito, simple de usar y altamente personalizable (recomendado).
- **Brackets** (www.brackets.io) es un editor gratuito creado por Adobe.
- **KompoZer** (www.kompozer.net) es un editor gratuito con un panel de vista previa que facilita la búsqueda de partes específicas del documento a modificar.
- **Aptana** (www.aptana.com) es una IDE gratuita con herramientas que simplifican la administración de archivos y proyectos.
- **NetBeans** (www.netbeans.org) es una IDE gratuita con herramientas para administrar archivos y proyectos.
- **Sublime** (www.sublimetext.com) es un editor de pago con una versión gratuita de evaluación.
- **Komodo** (www.komodoide.com) es una IDE de pago que puede trabajar con una cantidad extensa de lenguajes de programación.
- **Dreamweaver** (www.adobe.com/products/dreamweaver.html) es una IDE de pago con tecnología WYSIWYG incorporada (Lo Que Ves Es Lo Que Obtienes) que nos permite ver los resultados de la ejecución del código en tiempo real.

Trabajar con un editor es simple: tenemos que crear un directorio en nuestro disco duro donde vamos a almacenar los archivos del sitio web, abrir el editor, y crear dentro de este directorio todos los archivos y directorios adicionales que necesitamos para nuestro proyecto. La Figura 1-6 muestra cómo se ve el editor Atom cuando se abre por primera vez.

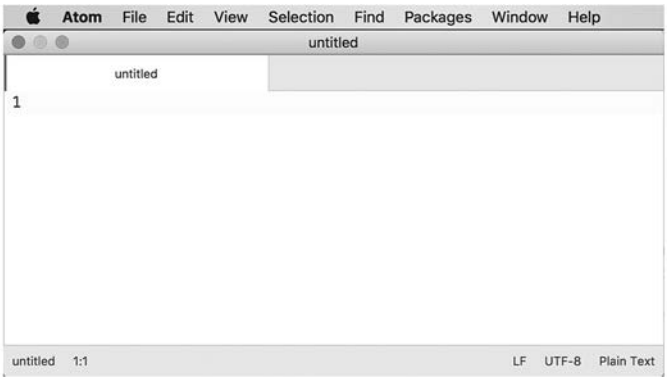

Figura 1-6: *Editor Atom con un archivo vacío*

Este editor tiene una opción en el menú **File** (Archivo) para abrir un proyecto (**Add Project Folder**). La opción se muestra en la Figura 1-7, número 1.

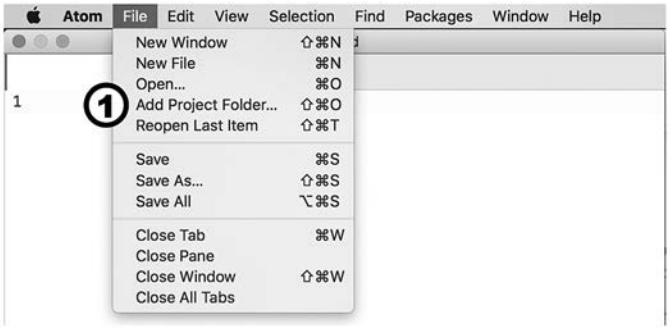

Figura 1-7: *Opción para agregar un proyecto*

Si hacemos clic en esta opción y luego seleccionamos el directorio creado para nuestro proyecto, el editor abre un nuevo panel a la izquierda con la lista de archivos dentro del directorio. La Figura 1-8, a continuación, muestra un directorio llamado **Test** creado para contener los archivos del ejemplo de la Figura 1-2.

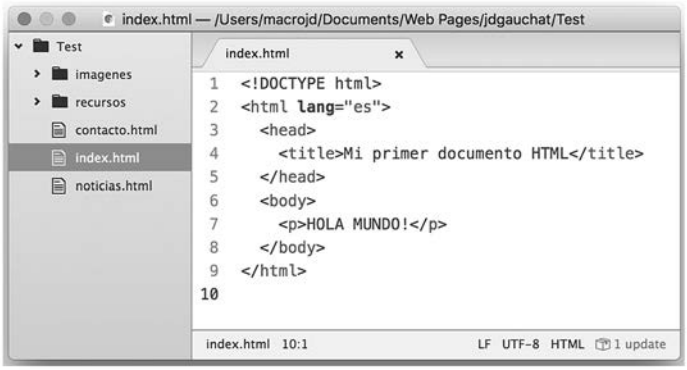

Figura 1-8: *Archivos del proyecto*

Hágalo usted mismo: visite **www.atom.io** para descargar el editor Atom. Una vez que el editor esté instalado en su ordenador, cree un nuevo directorio en su disco duro para almacenar los archivos de su sitio web. Abra Atom, vaya al menú **File** y seleccione la opción **Add Project Folder** (Figura 1-7, número 1). Seleccione el directorio que acaba de crear. Abra de nuevo el menú **File** y seleccione la opción **New File** (Nuevo Archivo). Copie el código del Listado 1-1 y grabe el archivo con el nombre index.html. Después de seleccionar el archivo en el panel de la izquierda, debería ver algo similar a lo que se muestra en la Figura 1-8.

IMPORTANTE: explicar cómo trabajar con Atom, u otro editor, va más allá del propósito de este libro, pero hemos incluido enlaces a cursos y vídeos en nuestro sitio web con más información. Para acceder a ellos, descargue los contenidos adicionales y haga clic en las opciones **Enlaces** y **Vídeos**.

Registro de dominios

Una vez que nuestro sitio web está listo para ser presentado en público, tenemos que registrar el dominio que los usuarios van a escribir en la barra de navegación para acceder a él. Como ya mencionamos, un domino es simplemente un nombre personalizado con una extensión que determina el propósito del sitio web. El nombre puede ser cualquiera que deseemos, y contamos con varias opciones para definir la extensión, desde extensiones con propósitos comerciales, como .com o .biz., a aquellas sin ánimo de lucro o personales, como .org, .net o .info, por no mencionar las extensiones regionales que incluyen un valor adicional para determinar la ubicación del sitio web, como .co.uk para sitios web en el Reino Unido o .eu para sitios web relacionados con la Union Europea.

Para obtener un domino para nuestro sitio web, tenemos que abrir una cuenta con un registrante y adquirirlo. La mayoría de los dominios requieren del pago de un arancel anual, pero el proceso es relativamente sencillo y hay muchas compañías disponibles que pueden hacerse cargo del trámite por nosotros. La más popular es GoDaddy (www.godaddy.com), pero la mayoría de las compañías que ofrecen servicios para desarrolladores también incluyen la posibilidad de registrar un dominio. Como dijimos, el proceso de registro es sencillo; tenemos que decidir el nombre y la extensión que vamos a asignar a nuestro dominio, realizar una búsqueda para asegurarnos de que el nombre que hemos elegido no está siendo utilizado y se encuentra disponible, y luego hacer el pedido (las compañías mencionadas con anterioridad facilitan todas las herramientas necesarias para este propósito).

Cuando el dominio está registrado, el sistema nos pide los nombres de servidores (nameservers) que queremos asociar al dominio. Estos nombres son cadenas de texto compuestas por un dominio y un prefijo, generalmente NS1 y NS2, que determinan la ubicación de nuestro sitio web (los nombres de servidor o nameservers los facilita el servidor en el que se almacena nuestro sitio web). Si aún no contamos con estos nombres, podemos usar los que ofrece la compañía y cambiarlos más adelante.

IMPORTANTE: la compañía que registra su dominio asigna nombres de servidor por defecto que ellos usan como destino provisional (también conocido como *aparcamiento* o *parking*). En principio puede asignar estos nombres y cambiarlos más adelante cuando su servidor esté listo. Algunas compañías ofrecen el registro de dominio junto con el alquiler de servidores y, por lo tanto, pueden encargarse de la configuración del dominio por nosotros si usamos sus servidores.

Alojamiento web

Configurar y mantener un servidor exige conocimientos que no todos los desarrolladores poseen. Por este motivo, existen compañías que ofrecen un servicio llamado alojamiento web (*web hosting*), que permite a cualquier individuo alquilar un servidor configurado y listo para almacenar, y operar uno o múltiples sitios web.

Existen diferentes tipos de alojamiento web disponible, desde aquellos que permiten que varios sitios web operen desde un mismo servidor (alojamiento compartido) hasta servicios más profesionales que reservan un servidor completo para un único sitio web (alojamiento dedicado), o distribuyen un sitio web extenso en muchos servidores (alojamiento en la nube), incluidas varias opciones intermedias.

La principal ventaja de tener una cuenta de alojamiento web es que todas ofrecen un panel de control con opciones para crear y configurar nuestro sitio web. Las siguientes son las opciones más comunes que nos encontraremos en la mayoría de estos servicios.

- **File Manager** es una herramienta web que nos permite administrar los archivos de nuestro sitio. Con esta herramienta podemos subir, bajar, editar o eliminar archivos en el servidor desde el navegador, sin tener que usar ninguna otra aplicación.
- **FTP Accounts** es un servicio que nos permite administrar las cuentas que usamos para conectarnos al servidor por medio de FTP. FTP (File Transfer Protocol) es un protocolo de comunicación diseñado para transferir archivos desde un ordenador a otro en la red.
- **MySQL Databases** es un servicio que nos permite crear bases de datos para nuestro sitio web.
- **phpMyAdmin** es una aplicación programada en PHP que podemos usar para administrar las bases de datos creadas para nuestro sitio web.
- **Email Accounts** es un servicio que nos permite crear cuentas de email con el dominio de nuestro sitio web (por ejemplo, info@midominio.com).

El panel de control más popular en el mercado es *cPanel*. La Figura 1-9 muestra el diseño y algunas de las opciones que ofrece.

Figura 1-9: *Opciones ofrecidas por cPanel*

El coste de una cuenta de alojamiento puede variar entre algunos dólares por una cuenta compartida hasta cientos de dólares al mes por un servidor dedicado. Una vez que abrimos la cuenta, la compañía nos envía un email con la información que necesitamos para acceder al panel de control y configurar el servidor. El sistema de la compañía generalmente crea todas las cuentas básicas que necesitamos, incluida una cuenta FTP para subir los archivos, tal como veremos a continuación.

Lo básico: además de las cuentas de alojamiento de pago, existe el alojamiento gratuito que podemos usar para practicar, pero estos servicios incluyen propaganda o imponen restricciones que impiden el desarrollo de sitios web profesionales. Siempre se recomienda comenzar con una cuenta de alojamiento compartido que puede costar unos 5 dólares al mes para aprender cómo trabaja un servicio de alojamiento profesional y estudiar todas las opciones que ofrece. Varias compañías incluyen en sus servicios este tipo de alojamiento. Las más populares en este momento son www.godaddy.com y www.hostgator.com.

Programas FTP

Como acabamos de mencionar, las cuentas de alojamiento web ofrecen un servicio para administrar los archivos del sitio web desde el navegador. Esta es una página web a la que podemos acceder desde el panel de control para subir, bajar y editar los archivos en el servidor. Es una herramienta útil, pero solo práctica cuando necesitamos realizar pequeñas modificaciones o subir unos pocos archivos. La herramienta aprovecha un sistema que se encuentra integrado en los navegadores y que trabaja con un protocolo llamado *FTP* (File Transfer Protocol) usado para transferir archivos desde un ordenador a otro en una red. Los navegadores incluyen este sistema porque lo necesitan para permitir a los usuarios descargar archivos pero, debido a que su principal propósito es descargar y mostrar sitios web en la pantalla, ofrecen una mala experiencia a la hora de manipular estos archivos. Por esta razón, los desarrolladores profesionales no utilizan el navegador sino programas diseñados específicamente para transferir archivos entre un cliente y un servidor usando el protocolo FTP.

El mercado ofrece varios programas FTP, incluidas versiones de pago y gratuitas. El programa gratuito más popular se llama *Filezilla* y se encuentra disponible en **www.filezilla-project.org**. Este programa ofrece varios paneles con información acerca de la conexión y los ordenadores que participan, incluidos dos paneles lado a lado con la lista de los archivos locales y remotos que podemos transferir entre ordenadores con solo arrastrarlos de un panel a otro.

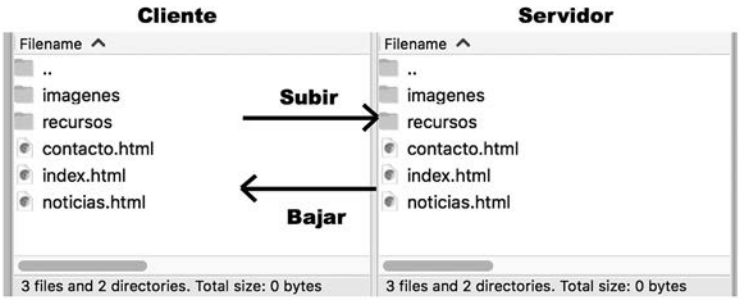

Figura 1-10: *Interfaz de Filezilla*

Cuando abrimos una cuenta de alojamiento, el sistema crea automáticamente una cuenta FTP para nuestro sitio web que incluye el nombre de usuario y clave requeridos para conectarse al servidor usando este protocolo (si el sistema no configura esta cuenta, podemos hacerlo nosotros mismos desde la opción **FTP Accounts** en el panel de control). Los valores que necesitamos para realizar la conexión son el host (IP o dominio), el usuario y la clave, además del puerto asignado por el servidor para conectarse por medio de FTP (por defecto, 21). Filezilla ofrece dos maneras de insertar esta información: una barra en la parte superior con la que realizar una conexión rápida (Figura 1-11, número 2) y un botón para almacenar múltiples conexiones de uso frecuente (Figura 1-11, número 1).

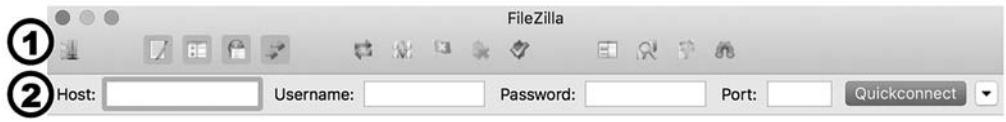

Figura 1-11: *Configuración de conexiones*

Si pulsamos el botón para almacenar o acceder a conexiones previas (número 1), Filezilla abre una ventana donde podemos administrar la lista de conexiones disponibles y especificar opciones adicionales de configuración. La ventana incluye botones para crear, renombrar y borrar una conexión (**New Site**, **Rename**, **Delete**), campos donde podemos seleccionar el protocolo que deseamos utilizar (FTP para una conexión normal y SFTP para una conexión segura), el modo de encriptación usado para transferir los archivos y el tipo de cuenta requerida (**Anonymous** para conexiones anónimas o **Normal** para conexiones que requieren usuario y clave). El programa también incluye paneles adicionales para una configuración más avanzada, tal como muestra la Figura 1-12.

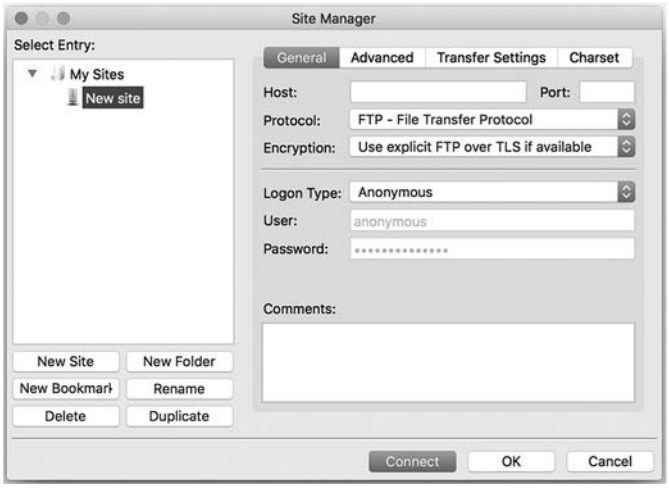

Figura 1-12: *Administrador de conexiones*

En una situación normal, para establecer la conexión tenemos que insertar el host (el IP o dominio de nuestro sitio web), seleccionar el tipo de cuenta como **Normal**, introducir el nombre de usuario y la contraseña, y dejar el resto de los valores por defecto. Una vez que los ordenadores se conectan, Filezilla muestra la lista de archivos en la pantalla. A la izquierda se

encuentran los archivos en el directorio seleccionado en nuestro ordenador (podemos seleccionar cualquier directorio que queramos en nuestro disco duro), y a la derecha se encuentran los archivos y directorios disponibles en el directorio raíz de nuestra cuenta de alojamiento, como muestra la Figura 1-13.

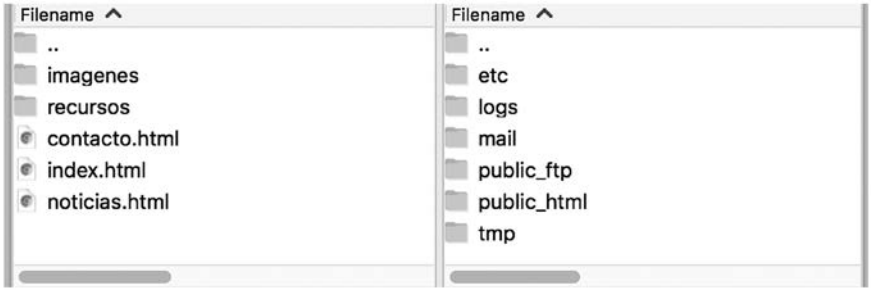

Figura 1-13: *Contenido del directorio raíz*

Cuando se crea una cuenta de alojamiento, el sistema incluye varios directorios y archivos para almacenar la información requerida por el servicio (almacenar emails, hacer un seguimiento de la actividad de los usuarios, etc.). El directorio en el que se deben almacenar los archivos de nuestro sitio web se llama *public_html*. Una vez que se abre este directorio, podemos comenzar a subir nuestros archivos arrastrándolos desde el panel de la izquierda al panel de la derecha (ver Figura 1-10).

IMPORTANTE: va más allá del propósito de este libro explicar cómo funciona Filezilla o cualquier otro programa de FTP, pero hemos incluido enlaces a cursos y vídeos en nuestro sitio web con más información. Para acceder a ellos, descargue los contenidos adicionales y haga clic en las opciones **Enlaces** y **Vídeos**.

MAMP

Los documentos HTML se pueden abrir directamente en un ordenador personal. Por ejemplo, si abrimos el archivo index.html con el documento creado en el Listado 1-1, la ventana del navegador muestra el texto HOLA MUNDO!, según ilustra la Figura 1-14 debajo (el resto de los elementos de este documento son estructurales y, por lo tanto, no producen resultados visibles).

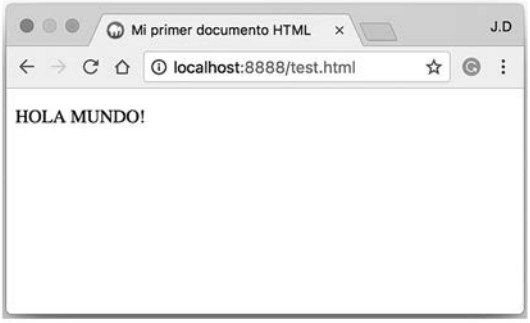

Figura 1-14: *Documento HTML en el navegador*

Lo básico: para abrir un archivo en el navegador, puede seleccionar la opción **Abrir Archivo** desde el menú del navegador o hacer doble clic en el archivo desde el explorador de archivos de Windows (o Finder en ordenadores Apple), y el sistema se encarga de abrir el navegador y cargar el documento.

Aunque la mayoría de los ejemplos de este libro se pueden probar sin subirlos a un servidor, abrir un sitio web completo en un ordenador personal no es siempre posible. Como veremos más adelante, algunos códigos JavaScript solo trabajan cuando se descargan desde un servidor, y tecnologías de servidor como PHP requieren ser alojadas en un servidor para funcionar. Para trabajar con estas clases de documentos existen dos alternativas: podemos obtener una cuenta de alojamiento web de inmediato y usarla para hacer pruebas, o instalar un servidor en nuestro propio ordenador. Esta última opción no hará que se pueda acceder a nuestro sitio web desde Internet, pero nos permite probarlo y experimentar con el código antes de subir la versión final a un servidor real.

Existen varios paquetes que instalan todos los programas necesarios para convertir nuestro ordenador en un servidor. Estos paquetes incluyen un servidor Apache (para despachar archivos web a través del protocolo HTTP), un servidor PHP (para procesar código PHP), y un servidor MySQL (para procesar bases de datos de tipo MySQL que podemos usar para almacenar datos en el servidor). El que recomendamos se llama *MAMP*. Es un paquete gratuito, disponible para ordenadores personales y ordenadores Apple, que podemos descargar desde **www.mamp.info** (la empresa también ofrece una versión comercial avanzada llamada *MAMP PRO*).

MAMP es fácil de instalar y usar. Una vez descargado e instalado, solo necesitamos abrirlo para comenzar a utilizar el servidor.

Figura 1-15: *Pantalla principal de MAMP*

MAMP crea un directorio dentro de su propio directorio llamado *htdocs* donde se supone que debemos almacenar los archivos de nuestro sitio web, pero si lo deseamos, podemos asignar un directorio diferente desde la opción **Preferences**. Esta opción abre una nueva ventana con varias pestañas para su configuración. La pestaña **Web Server** muestra el directorio actual que usa el servidor Apache y ofrece un botón para seleccionar uno diferente, tal como ilustra la Figura 1-16, número 1.

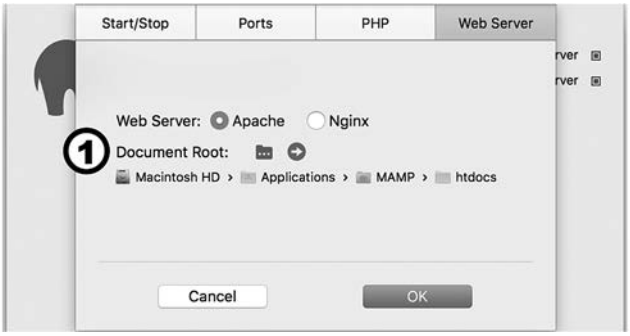

Figura 1-16: *Directorio del servidor Apache*

Después de seleccionar el directorio en el que se encuentran los archivos de nuestro sitio web, podemos acceder a ellos desde el servidor. Apache crea un dominio especial llamado *localhost* para referenciar al servidor y, por lo tanto, se puede acceder a nuestro sitio web desde la URL http://localhost/. Si queremos acceder a un archivo específico, solo tenemos que agregar el nombre del archivo al final de la URL, tal como hacemos con cualquier otro dominio (por ejemplo, http://localhost/contacto.html).

Hágalo usted mismo: visite **www.mamp.info** y descargue la versión gratuita de MAMP para su sistema (Windows o macOS). Instale el paquete y abra la aplicación. Seleccione la opción **Preferences** (Figura 1-15) y reemplace el directorio htdocs por el directorio que haya creado para su sitio web en el ejemplo anterior (Figura 1-8). Abra el navegador e inserte la URL http://localhost/. Si ha creado el archivo index.html como sugerimos anteriormente, debería ver el texto HOLA MUNDO! en la pantalla (Figura 1-14).

IMPORTANTE: el sistema operativo de Apple incluye su propia versión del servidor Apache, lo que obliga a MAMP a conectar Apache en un puerto diferente para evitar conflictos. Por esta razón, en un ordenador Mac tiene que especificar el puerto 8888 cuando intenta acceder al localhost (http://localhost:8888). Si lo desea, puede cambiar el puerto desde la configuración de MAMP. Haga clic en **Preferences**, seleccione la pestaña **Ports**, y presione el botón **Set Web & MySQL ports**.

Lo básico: la mayoría de los ejemplos de este libro se pueden ejecutar en un ordenador personal sin tener que instalar ningún servidor (le informaremos cuando esto no sea posible), pero si lo desea, puede instalar MAMP para asegurarse de que todo funcione correctamente como lo haría en un servidor real.