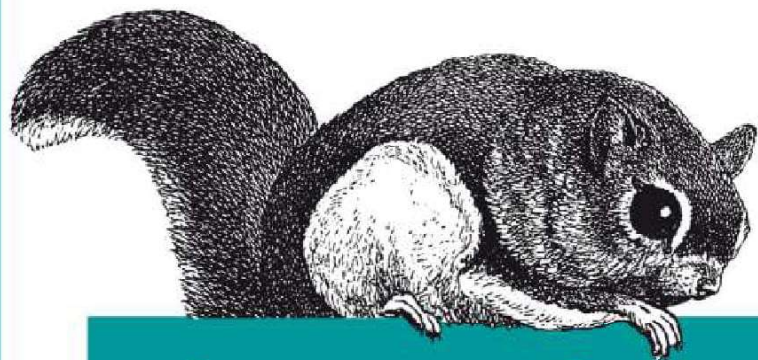


O'REILLY®

5a edición
Incluye aplicaciones
web y móviles



Robin Nixon



Aprender PHP, MySQL y JavaScript

CON JQUERY, CSS Y HTML5



Marcombo

Descargado en: eybooks.com

QUINTA EDICIÓN

Aprender PHP, MySQL y JavaScript

Con jQuery, CSS y HTML5

Acceda a www.marcombo.info
para descargar gratis
contenido adicional
complemento imprescindible de este libro

Código:

PHP1

QUINTA EDICIÓN

Aprender PHP, MySQL y JavaScript

Con jQuery, CSS y HTML5

Robin Nixon



Quinta edición original publicada en inglés por O'Reilly Media, Inc. con el título: *Learning PHP, MySQL & JavaScript, 5th Edition* © 2018 Robin Nixon

Título de la edición en español: *Aprender PHP, MySQL y JavaScript*

Quinta edición en español, año 2019

© 2019 MARCOMBO, S.A.
www.marcombo.com

Diseñador de portada: Karen Montgomery
Ilustrador: Rebecca Demarest
Traducción: Francisco Martínez Carreno
Correctora: Meritxell Peleato García
Directora de producción: M.^a Rosa Castillo Hidalgo

«Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra solo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necesita fotocopiar o escanear algún fragmento de esta obra».

ISBN: 978-84-267-2715-2

D.L.: B-16735-2019

ISBN Colombia: 978-958-778-612-5

Impreso en Servicepoint
Printed in Spain

Para Julie

Contenido

Prefacio.....	xxiii
1. Introducción al contenido dinámico de la web	1
HTTP y HTML: los conceptos básicos de Berners-Lee.....	2
Procedimiento de solicitud/respuesta	2
Ventajas de PHP, MySQL, JavaScript, CSS y HTML5	5
MariaDB: el clon de MySQL.....	6
Utilización de PHP.....	7
Utilización de MySQL.....	7
Utilización de JavaScript	9
Utilización de CSS.....	10
Y luego está HTML5	11
El servidor web Apache	11
Uso de dispositivos móviles.....	12
Sobre el software libre	12
Todo junto	13
Preguntas	14
2. Configuración de un servidor de desarrollo	17
¿Qué son WAMP, MAMP o LAMP?.....	18
Instalación de AMPPS en Windows.....	18
Prueba de la instalación.....	22
Acceso a la carpeta principal (Windows).....	24
WAMP alternativos	25
Instalación de AMPPS en macOS	26
Acceso a la carpeta principal (macOS)	27
Instalación de LAMP en Linux	28
Trabajar de forma remota	28
Inicio de sesión	29
Utilización de FTP	29
Utilización del editor de programas	30
Utilización del EDI	31
Preguntas	33
3. Introducción a PHP	35
Inclusión de PHP en HTML	35
Ejemplos de este libro	37
Estructura de PHP	37

Utilización de comentarios	37
Sintaxis básica.....	38
Variables.....	39
Operadores	44
Asignación de valores a variables	47
Comandos de varias líneas	49
Tipificación de variables.....	51
Constantes.....	52
Constantes predefinidas	53
Diferencia entre los comandos echo y print	54
Funciones.....	54
Ámbito de aplicación de variables	55
Preguntas	61
4. Expresiones y control de flujo en PHP	63
Expresiones	63
¿TRUE o FALSE?.....	63
Literales y variables.....	65
Operadores	66
Prioridades de los operadores	66
Asociatividad	68
Operadores relacionales	70
Condicionales.....	73
La declaración if	74
La declaración else.....	75
La declaración elseif.....	77
La declaración switch	78
El operador ?	81
Bucles.....	82
Bucles while.....	83
Bucles do...while	84
Bucles for	85
Salida del bucle	87
Declaración continue	88
Conversión implícita y explícita	88
Enlaces dinámicos en PHP	90
Enlaces dinámicos en acción	90
Preguntas	91
5. Funciones y objetos en PHP	93
Funciones en PHP	94
Definición de función	95
Devolución de un valor	95
Devolución de una matriz	97

Paso de argumentos por referencia	98
Devolución en variables globales.....	99
Recapitulación sobre el ámbito de aplicación de las variables	100
Inclusión y requisición de archivos	100
La declaración include	100
Utilización de include_once.....	101
Utilización de require y require_once	101
Compatibilidad de las versiones PHP	102
Objetos en PHP	102
Terminología.....	103
Declaración de clases	104
Creación de objetos.....	105
Acceso a objetos	105
Clonación de objetos.....	107
Constructores	108
Destrucción	108
Métodos de escritura	109
Declaración de propiedades	110
Declaración de constantes	110
Ámbito de las propiedades y de los métodos	111
Métodos estáticos.....	112
Propiedades estáticas	113
Herencia	114
Preguntas	117
6. Matrices en PHP	119
Introducción	119
Matrices indexadas numéricamente	119
Matrices asociativas	121
Asignación mediante la palabra clave array.....	122
Bucle foreach...as	123
Matrices de varias dimensiones.....	125
Uso de funciones en matrices	128
is_array.....	128
count	128
sort	128
shuffle	129
explode.....	129
extract.....	130
compact.....	131
reset.....	132
end.....	132
Preguntas	132

7. PHP práctico	135
Uso de printf	135
Ajustes de la precisión	136
Relleno de cadenas	138
Uso de sprintf	139
Funciones de fecha y hora	139
Constantes de fecha	142
Uso de la verificación de fecha	142
Manejo de archivos	143
Verificación de la existencia de un archivo	143
Creación de archivos	143
Lectura de archivos	145
Copia de archivos	146
Movimiento de archivos	146
Eliminación de archivos	147
Actualización de archivos	147
Bloqueo de archivos debido a accesos múltiples	148
Lectura de archivos completos	150
Carga de archivos	151
Llamadas al sistema	156
¿XHTML o HTML5?	158
Preguntas	158
 8. Introducción a MySQL	 161
Fundamentos de MySQL	161
Resumen de términos de bases de datos	162
Acceso a MySQL mediante la línea de comandos	162
Inicio de la interfaz de la línea de comandos	163
Uso de la interfaz de la línea de comandos	167
Comandos MySQL	168
Tipos de datos	173
Índices	183
Creación de un índice	183
Consulta de bases de datos MySQL	189
Unión de tablas	199
Uso de operadores lógicos	201
Funciones MySQL	202
Acceso a MySQL mediante phpMyAdmin	202
Preguntas	203
 9. Dominio de MySQL	 205
Diseño de bases de datos	205
Claves principales: las claves de las bases de datos relacionales	206

Normalización	207
Primera forma normal	208
Segunda forma normal	210
Tercera forma normal	212
Cuándo no utilizar la normalización	214
Relaciones	215
Uno a uno	215
Uno a muchos	216
Muchos a muchos	217
Bases de datos y anonimato	218
Transacciones	219
Motores de almacenamiento de transacciones	219
Uso de BEGIN	220
Uso de COMMIT	221
Uso de ROLLBACK	221
Uso de EXPLAIN	222
Copias de seguridad y restauración	223
Uso de mysqldump	223
Creación de archivos de copias de seguridad	225
Restauración del archivo de la copia de seguridad	227
Descarga de datos en formato CSV	227
Planificación de copias de seguridad	228
Preguntas	228

10. Acceso a MySQL mediante PHP231

Consultas de la base de datos MySQL con PHP	231
El proceso	231
Creación del archivo de inicio de sesión	232
Conexión a la base de datos MySQL	233
Un ejemplo práctico	239
La matriz \$_POST	242
Eliminación de un registro	243
Visualización del formulario	243
Consulta de la base de datos	244
Ejecución del programa	245
MySQL práctico	246
Creación de una tabla	247
Descripción de una tabla	247
Eliminación de una tabla	248
Adición de datos	249
Recuperación de datos	250
Actualización de datos	251
Borrado de datos	251

Uso de AUTO_INCREMENT	252
Realización de consultas adicionales	253
Prevencción de intentos de piratería	254
Pasos que puedes seguir	255
Uso de marcadores de posición	256
Prevencción de la inyección de HTML	259
Uso procedimental de mysqli	260
Preguntas	262
11. Gestión de formularios	263
Creación de formularios	263
Extracción de los datos enviados	265
Valores por defecto	266
Tipos de entradas	267
Desinfección de entradas	274
Programa de ejemplo	276
Mejoras en HTML5	279
Atributo autocomplete	279
Atributo autofocus	279
Atributo placeholder	279
Atributo required	280
Atributos de sustitución	280
Atributos width y height	280
Atributos min y max	280
Atributo step	281
Atributo form	281
Atributo list	281
Tipo de entrada color	281
Tipos de entradas number y range	282
Selectores de fecha y hora	282
Preguntas	282
12. Cookies, sesiones y autenticación	283
Uso de cookies en PHP	283
Configuración de cookies	285
Acceso a cookies	286
Eliminación de cookies	286
Autenticación HTTP	286
Almacenamiento de nombres de usuario y contraseñas	290
Programa de ejemplo	292
Uso de sesiones	295
Inicio de sesión	296
Finalización de sesión	298
Configuración del tiempo de espera	299

Seguridad de sesión.....	300
Preguntas.....	303
13. Exploración de JavaScript	305
Texto JavaScript y HTML	305
Uso de scripts en el encabezamiento de documentos	307
Navegadores antiguos y no estándar.....	307
Inclusión de archivos JavaScript.....	308
Depuración de errores en JavaScript.....	309
Uso de comentarios	310
Signos de punto y coma	310
Variables	310
Variables de cadena de caracteres.....	311
Variables numéricas	311
Matrices	312
Operadores	312
Operadores aritméticos	313
Operadores de asignación	313
Operadores de comparación.....	314
Operadores lógicos.....	314
Asignación creciente, decreciente y abreviada	314
Concatenación de cadenas	315
Caracteres de escape	315
Escritura de variables	316
Funciones	317
Variables globales	317
Variables locales.....	317
Modelo de objetos del documento	318
Otro uso del símbolo \$	320
Uso del DOM	321
Sobre document.write.....	322
Uso de console.log	322
Uso de alert.....	322
Escritura en elementos	322
Uso de document.write.....	323
Preguntas	323
14. Expresiones y control de flujo en JavaScript	325
Expresiones	325
Literales y variables	326
Operadores	327
Prioridad de operadores	328
Asociatividad	328
Operadores relacionales	329

Declaración with	332
Uso de onerror	333
Uso de try...catch	334
Condicionales.....	335
Declaración if	335
Declaración else	335
Declaración switch.....	336
Operador ?	338
Bucles.....	338
Bucles while.....	338
Bucles do...while.....	339
Bucles for.....	340
Salida del bucle.....	341
Declaración continue	341
Conversión explícita.....	342
Preguntas	343
15. Funciones, objetos y matrices de JavaScript	345
Funciones JavaScript	345
Definición de función	345
Devolución de un valor	347
Devolución de una matriz	349
Objetos JavaScript.....	350
Declaración de clase	350
Creación de objetos.....	351
Acceso a objetos	352
La palabra clave prototype	352
Matrices JavaScript	355
Matrices numéricas	355
Matrices asociativas	357
Matrices de varias dimensiones	358
Métodos de uso de matrices	359
Preguntas	364
16. Validación de JavaScript y PHP y tratamiento de errores.....	367
Validación de la entrada de usuario con JavaScript	367
Documento validate.html (Parte 1)	367
Documento validate.html (Parte 2)	370
Expresiones regulares.....	373
Concordancia mediante metacaracteres	373
Concordancia de caracteres difusos	374
Agrupación mediante paréntesis	375
Clase de caracteres.....	376
Indicación del intervalo.....	376

Negación	376
Otros ejemplos más complicados.....	377
Resumen de metacaracteres	379
Modificadores generales	381
Uso de expresiones regulares en JavaScript	382
Uso de expresiones regulares en PHP	382
Nueva visualización del formulario después de la validación PHP	383
Preguntas	389
17. Uso de comunicaciones asíncronas.....	391
¿Qué es la comunicación asíncrona?.....	392
Uso de XMLHttpRequest	392
Tu primer programa asíncrono	394
Uso de GET en lugar de POST	399
Envío de solicitudes XML	401
Uso de frameworks para la comunicación asíncrona.....	406
Preguntas	406
18. Introducción a CSS.....	407
Importación de hojas de estilo.....	408
Importación de CSS desde HTML.....	408
Ajuste de estilo integrados	409
Uso de ID.....	409
Uso de clases.....	409
Uso del punto y coma.....	410
Reglas CSS.....	410
Asignaciones múltiples	410
Uso de comentarios.....	411
Tipos de estilos	412
Estilos por defecto.....	412
Estilos de usuario	412
Hoja de estilo externas	413
Estilos internos.....	413
Estilos en línea	414
Selectores CSS	414
Selector de tipo	414
Selector de descendiente	414
Selector de hijo	415
Selector de ID	416
Selector de clase.....	417
Selector de atributo	418
Selector universal.....	418
Selección por grupo	419

Cascada CSS	419
Creadores de hojas de estilo	420
Métodos de hojas de estilo	420
Selectores de hojas de estilo	421
Diferencia entre los elementos div y span	423
Medidas	425
Fuentes y tipografía	427
font-family	427
font-style	428
font-size	428
font-weight	429
Tratamiento de estilos de texto	429
Decoración	429
Espaciado	430
Alineación	430
Transformación	430
Sangrado	430
Colores CSS	431
Cadenas reducidas para determinar el color	432
Degradados	432
Elementos de posicionamiento	434
Posicionamiento absoluto	434
Posicionamiento relativo	434
Posicionamiento fijo	435
Pseudoclases	437
Reglas abreviadas	439
El modelo de caja y el diseño	440
Fijación de márgenes	440
Aplicación de bordes	442
Ajuste de relleno	443
Contenidos del objeto	445
Preguntas	445

19. CSS avanzado con CSS3.....447

Selectores de atributos	448
Partes coincidentes de las cadenas	448
Propiedad box-sizing	449
Fondos CSS3	450
Propiedad background-clip	450
Propiedad background-origin	452
Propiedad background-size	452
Uso de auto Value	453
Múltiples fondos	453

Bordes CSS3	455
Propiedad border-color	455
Propiedad border-radius	456
Sombras de caja.....	459
Desbordamiento de elementos	460
Diseño en varias columnas	460
Colores y opacidad	462
Colores HSL	462
Colores HSLA.....	463
Colores RGB.....	463
Colores RGBA.....	463
Propiedad opacity.....	464
Efectos de texto.....	464
Propiedad text-shadow	464
Propiedad text-overflow	464
Propiedad word-wrap	465
Fuentes web	466
Fuentes de la web de Google	467
Transformaciones.....	468
Transformaciones 3D	469
Transiciones	470
Propiedades de las transiciones.....	470
Duración de las transiciones	471
Retardo en las transiciones.....	471
Tiempo de transición.....	471
Sintaxis abreviada	472
Preguntas	474
20. Acceso a CSS desde JavaScript	475
Revisión de la función getElementById	475
La función O	475
La función S	476
La función C	477
Inclusión de funciones	478
Acceso a las propiedades de CSS desde JavaScript	478
Algunas propiedades de uso frecuente.....	479
Otras propiedades.....	480
JavaScript en línea.....	482
Palabra clave this	482
Anexión de eventos a objetos en un script.....	483
Anexión a otros eventos.....	484
Adición de nuevos elementos.....	485
Eliminación de elementos	486

Alternativas para añadir y eliminar elementos.....	487
Uso de interrupciones.....	488
Uso de setTimeout.....	488
Cancelación del tiempo de espera.....	489
Uso de setInterval.....	489
Uso de interrupciones en animaciones.....	491
Preguntas.....	493

21. Introducción a jQuery.....495

¿Por qué jQuery?.....	495
Inclusión de jQuery.....	496
Elección de la versión adecuada.....	496
Descarga.....	498
Uso de una red de entrega de contenido.....	498
Personalización de jQuery.....	499
Sintaxis de jQuery.....	499
Un sencillo ejemplo.....	500
Cómo evitar conflictos entre bibliotecas.....	501
Selectores.....	501
Método css.....	502
Selector de elemento.....	502
Selector de ID.....	503
Selector de clase.....	503
Combinación de selectores.....	503
Tratamiento de eventos.....	504
En espera de que el documento esté preparado.....	505
Funciones y propiedades de eventos.....	506
Eventos de enfoque y desenfoque.....	507
Palabra clave this.....	508
Eventos click y dblclick.....	508
Evento keypress.....	509
Programación amable.....	511
Evento mousemove.....	511
Otros eventos del ratón.....	514
Métodos alternativos del ratón.....	515
Evento submit.....	516
Efectos especiales.....	517
Ocultación y presentación.....	518
Método toggle.....	519
Desvanecimiento de entrada y salida.....	520
Elementos deslizantes hacia arriba y hacia abajo.....	521
Animaciones.....	522
Detención de animaciones.....	525

Tratamiento del DOM	526
Diferencia entre los métodos text y html	527
Métodos val y attr.....	527
Adición y eliminación de elementos	529
Aplicación dinámica de clases	531
Modificación de dimensiones.....	531
Métodos width y height.....	532
Métodos innerWidth e innerHeight.....	534
Métodos outerWidth y outerHeight	534
Atravesar el DOM	535
Elementos padre.....	535
Elementos hijo	539
Elementos hermanos	539
Selección de elementos anteriores y posteriores.....	541
Atravesar selecciones jQuery	543
Método is	544
Uso de jQuery sin selectores.....	546
Método \$.each.....	546
Método \$.map	547
Uso de la comunicación asíncrona.....	548
Uso del método POST	548
Uso del método GET	549
Complementos.....	549
Interfaz de usuario de jQuery	550
Otros complementos	550
Preguntas	550
22. Introducción a jQuery Mobile.....	553
Inclusión de jQuery Mobile.....	554
Primeros pasos	555
Páginas enlazadas.....	557
Enlace síncrono.....	557
Enlace en un documento de varias páginas.....	558
Transiciones de página.....	558
Botones de diseño	562
Gestión de listas	565
Filtrado de listas.....	566
Divisores de listas	568
¿Y ahora qué?.....	571
Preguntas	571
23. Introducción a HTML5	573
El lienzo	573
Geolocalización.....	575

Audio y vídeo	577
Formularios	578
Trabajadores de la web	579
Microdatos	579
Preguntas	579
24. El lienzo HTML5	581
Creación y acceso al lienzo	581
Función toDataURL.....	583
Especificación del tipo de imagen.....	584
Método fillRect	585
Método clearRect	585
Método strokeRect	585
Combinación de estos comandos	585
Método createLinearGradient	587
Método addColorStop detallado	589
Método createRadialGradient	590
Uso de patrones para el relleno.....	592
Escritura de texto en el lienzo	593
Método strokeText.....	594
Propiedad textBaseLine	594
Propiedad font.....	595
Propiedad textAlign	595
Método fillText.....	596
Método measureText	596
Dibujo de líneas.....	597
Propiedad lineWidth	597
Propiedades lineCap y lineJoin	597
Propiedad miterLimit.....	599
Uso de rutas	600
Métodos moveTo y lineTo	600
Método stroke	600
Método rect	601
Áreas de relleno.....	601
Método clip	603
Método isPointInPath.....	606
Trabajo con curvas	607
Método arc	607
Método arcTo	610
Método quadraticCurveTo	611
Método bezierCurveTo	612
Tratamiento de imágenes	613
Método drawImage	613

Redimensionado de imágenes	614
Selección del área de la imagen	614
Copias del lienzo.....	616
Adición de sombras.....	616
Edición a nivel de píxel	618
Método getImageData	618
Método putImageData.....	621
Método createImageData	621
Efectos gráficos avanzados	621
Propiedad globalCompositeOperation	621
Propiedad globalAlpha.....	624
Transformaciones	624
Método scale	625
Método save y restore	626
Método rotate	626
Método translate	627
Método transform.....	628
Método setTransform.....	630
Preguntas	631
25. Audio y vídeo en HTML5.....	633
Sobre los códecs	634
Elemento <audio>	635
Compatibilidad con navegadores que no son HTML5	638
Elemento <video>	639
Códecs de vídeo	640
Compatibilidad con navegadores más antiguos	643
Preguntas	645
26. Otras características de HTML5	647
La geolocalización y el servicio GPS	647
Otros métodos de localización	648
Geolocalización y HTML5.....	649
Almacenamiento local.....	652
Uso del almacenamiento local	653
Objeto localStorage.....	653
Trabajadores de la web	655
Arrastrar y soltar	658
Mensajería entre documentos	660
Otras etiquetas HTML5	664
Preguntas	665
27. Todo junto.....	667
Diseño de una aplicación de red social	668

Sobre el sitio web	668
functions.php	668
Funciones	669
header.php	671
setup.php	673
index.php	675
signup.php	676
Comprobación de la disponibilidad de nombres de usuario	677
Inicio de sesión	677
checkuser.php	680
login.php	681
profile.php	683
Adición del texto "About Me"	684
Adición de la imagen del perfil	684
Procesamiento de la imagen	684
Visualización del perfil actual	685
members.php	688
Visualización del perfil de usuario	688
Incorporación y eliminación de amigos	689
Listado de todos los miembros	689
friends.php	692
messages.php	695
logout.php	698
styles.css	700
javascript.js	702
A. Soluciones a las preguntas de los capítulos.....	705
B. Recursos en línea.....	729
C. Palabras vacías en FULLTEXT de MySQL.....	733
D. Funciones MySQL	737
E. Selectores, objetos y métodos en jQuery.....	747
Índice.....	769

Prefacio

La combinación de PHP y MySQL es el enfoque más conveniente para el diseño de páginas web dinámicas utilizando bases de datos, y además está a la altura de los desafíos de los frameworks integrados (como Ruby on Rails) que son más difíciles de aprender. Debido a sus orígenes de software libre (a diferencia de la competencia Microsoft.NET Framework), se puede implementar libremente y por lo tanto es una opción muy popular para el desarrollo web.

Cualquier desarrollador potencial que utilice la plataforma Unix/Linux o incluso Windows/Apache necesitará dominar estas tecnologías. Y, combinadas con las tecnologías asociadas de JavaScript, jQuery, CSS y HTML5, podrá crear sitios web con el nivel de los estándares de la industria como Facebook, Twitter y Gmail.

Lectores

Este libro está orientado a personas que desean aprender a crear sitios web efectivos y dinámicos. Es decir, para administradores de sitios web o diseñadores gráficos que ya crean sitios web estáticos, pero que desean llevar sus habilidades al siguiente nivel, así como para estudiantes de enseñanza secundaria, universitarios, recién graduados y personas autodidactas.

De hecho, cualquiera que esté preparado para aprender los fundamentos del diseño web interactivo asimilará un conocimiento profundo de las tecnologías básicas de PHP, MySQL, JavaScript, CSS y HTML5, y también aprenderá los fundamentos de las bibliotecas de jQuery y jQuery Mobile.

Conocimientos previos

El libro se dirige a personas que tienen unos conocimientos básicos de HTML y que pueden al menos crear un sitio web estático sencillo, pero no tienen por qué tener conocimientos previos de PHP, MySQL, JavaScript, CSS o HTML5, aunque si los tienen, progresarán mucho más rápidamente a medida que avancen en el contenido del libro.

Organización del libro

Los capítulos del libro siguen un orden específico, primero hacen una introducción de las tecnologías básicas de las que tratan y luego te guían en la instalación de las mismas en un servidor de desarrollo web para poder trabajar con los ejemplos.

En la primera sección, se explican las ideas fundamentales del lenguaje de programación PHP y se presentan los conceptos básicos de sintaxis, matrices, funciones y programación orientada a objetos.

Luego, una vez que conozcas PHP, continuarás con una introducción al sistema de bases de datos MySQL, con la que lo aprenderás todo, desde cómo están estructuradas las bases de datos MySQL hasta cómo generar consultas complejas.

Después, aprenderás a combinar PHP y MySQL para empezar a crear tus propias páginas web dinámicas integrando formularios y otras características HTML. A continuación, se presentarán los aspectos prácticos de PHP y el desarrollo de MySQL mediante el aprendizaje de una serie de funciones útiles, cómo administrar cookies y sesiones, y cómo mantener un alto nivel de seguridad.

En los capítulos que siguen a lo anterior, adquirirás una sólida formación en JavaScript, desde sencillas funciones de simulación y gestión de eventos hasta el acceso al Modelo de Objetos del Documento, la validación en el navegador y la gestión de errores. También obtendrás una introducción completa sobre el uso de la popular biblioteca jQuery para JavaScript.

Con una comprensión de estas tres tecnologías esenciales, aprenderás a realizar llamadas a AJAX en segundo plano y a convertir tus sitios web en entornos dinámicos.

En los dos siguientes capítulos aprenderás todo sobre el uso de CSS para diseñar y maquetar tus páginas web, antes de descubrir cómo las bibliotecas jQuery pueden hacer tu trabajo de desarrollo mucho más fácil. Luego pasarás a la sección final sobre las características interactivas incorporadas a HTML5, incluidos la geolocalización, el audio, el vídeo y el lienzo. Después de esto, podrás reunir todo lo que has aprendido en un completo conjunto de programas con los que se crea un sitio web de red social completamente funcional.

En el recorrido, encontrarás muchos consejos sobre buenas prácticas de programación y trucos que pueden ayudarte a encontrar y resolver errores de programación difíciles de detectar. También hay muchos enlaces a sitios web que contienen más detalles sobre los temas tratados.

Libros de consulta

Una vez que hayas aprendido a desarrollar páginas web con PHP, MySQL, JavaScript, CSS y HTML5, estarás listo para alcanzar las habilidades del siguiente nivel mediante los libros de referencia de O'Reilly que figuran a continuación:

- *Dynamic HTML: The Definitive Reference* de Danny Goodman
- *PHP in a Nutshell* de Paul Hudson
- *MySQL in a Nutshell* de Russell Dyer
- *JavaScript: The Definitive Guide* de David Flanagan

- *CSS: The Definitive Guide* de Eric A. Meyer and Estelle Weyl
- *HTML5: The Missing Manual* de Matthew MacDonald

Convenciones que se utilizan en el libro

En el libro se utilizan las siguientes convenciones tipográficas:

Texto sin formato

Indica los títulos de los menús, las opciones y los botones.

Itálica

Indica nuevos términos, URL, direcciones de correo electrónico, nombres de archivos, extensiones de archivos, nombres de rutas, directorios y utilidades Unix. También se utiliza para nombres de bases de datos, tablas y columnas.

`Anchura constante`

Indica comandos y opciones de línea de comandos, variables y otros elementos de código, etiquetas HTML y contenido de los archivos.

Negrita de anchura constante

Muestra el resultado de programas y se utiliza para resaltar las secciones de código que se tratan en el texto.

Itálica de ancho constante

Muestra el texto que se debe sustituir por los valores suministrados por el usuario.



Este elemento se refiere a un consejo, sugerencia o nota general.



Este elemento indica una advertencia o precaución.

Uso de los ejemplos de código

En *www.marcombo.info* hay material suplementario (ejemplos de código, ejercicios, etc.) disponible para su descarga. Para acceder a él, sigue los pasos de la primera página del libro.

Existe también una página web para este libro, en la que aparecen listados de erratas, ejemplos, y cualquier información adicional. Puedes acceder a esta página en *http://bit.ly/lpmjch_5e*.

El propósito de este manual es ayudarte a hacer tu trabajo. En general, si se ofrecen códigos como ejemplos en el libro, puedes usarlos en tus programas y documentación. No es necesario que te pongas en contacto con nosotros para pedir permiso, a menos que reproduzcas una parte importante del código. Por ejemplo, escribir un programa que utiliza varios fragmentos de código no requiere pedir permiso. Vender o distribuir un CD-ROM con ejemplos de libros requiere pedir permiso. Responder a una pregunta citando este libro y el código de ejemplo no requiere pedir permiso. La incorporación de una importante cantidad de código de ejemplos de este libro a la documentación de tu producto requiere pedir permiso.

Capítulo 1

Introducción al contenido dinámico de la web

La World Wide Web es una red en constante evolución que ha ido mucho más lejos de lo que fue su concepción a principios de la década de 1990, cuando se creó con el propósito de resolver un problema específico. Los experimentos más avanzados del CERN (Laboratorio Europeo de Física de Partículas, ahora más conocido como el operador del Gran Colisionador de Hadrones) generaban ingentes cantidades de datos, hasta tal punto que resultaba difícil hacerlos llegar a los científicos de todo el mundo que participaban en las investigaciones.

En aquel momento, Internet ya estaba en funcionamiento, y a la red estaban conectados varios cientos de miles de ordenadores, por lo que Tim Berners-Lee (miembro del CERN) ideó un método para navegar entre ellos mediante una estructura de hiperenlaces, la cual llegó a conocerse como Protocolo de Transferencia de Hipertexto, o HTTP. También creó un lenguaje de marcado llamado Hypertext Markup Language, o HTML. Para integrarlos, desarrolló el primer navegador y el primer servidor web.

Hoy en día estamos acostumbrados a disponer de estas herramientas, pero en aquellas fechas el concepto era revolucionario. Hasta aquel momento, la conectividad más avanzada que tenían a su alcance los usuarios que disponían de un módem en casa era la de realizar una llamada y conectarse a un tablón de anuncios, alojado en un ordenador, a través del que el usuario podía comunicarse e intercambiar datos solo con otros usuarios de ese servicio. Por consiguiente, era necesario que el usuario fuera miembro de muchos sistemas de tableros de anuncios para poder comunicarse electrónicamente de manera efectiva con sus colegas y amigos.

Pero la contribución de Berners-Lee cambió todo aquello de golpe y, a mediados de la década de 1990, había tres grandes navegadores gráficos que competían por la captación de 5 millones de usuarios. Pronto se hizo evidente, sin embargo, que algo faltaba. Es cierto que las páginas de texto con hipervínculos que nos llevan a otras páginas fue un concepto brillante, pero los resultados no reflejaban el potencial de los ordenadores y de Internet para satisfacer de forma inmediata las necesidades particulares de cada usuario con contenidos que cambian dinámicamente. Utilizar la web era una experiencia árida y poco atractiva, incluso aunque hubiéramos tenido ¡texto en movimiento y GIF animados!

Los carritos de la compra, los motores de búsqueda y las redes sociales han alterado sin duda la forma en la que utilizamos la web. En este capítulo, echaremos un breve vistazo a los diversos componentes que forman la web y al software que ayuda a hacer de su uso una experiencia rica y dinámica.



Es necesario empezar a utilizar algunos acrónimos de forma más o menos inmediata. He tratado de explicarlos claramente antes de hacer uso de ellos, pero no hay que preocuparse demasiado por lo que representan o lo que significan sus nombres, porque los detalles los trataremos a medida que vayamos avanzando en el contenido del libro.

HTTP y HTML: los conceptos básicos de Berners-Lee

HTTP es un estándar de comunicación que gobierna las peticiones y respuestas que se envían entre el navegador, que se ejecuta en el ordenador del usuario final, y el servidor web.

El servidor tiene como función aceptar una petición del cliente e intentar responderle en un archivo de manera efectiva, por lo general mediante la entrega de la página web que ha solicitado. Este es el motivo por el que se utiliza el término *servidor*. El equivalente por naturaleza del servidor es el *cliente*, término que se aplica tanto al navegador web como al ordenador en el que se ejecuta.

Entre el cliente y el servidor puede haber otros equipos, como enrutadores, proxies, pasarelas, etc. Cumplen diferentes funciones para garantizar que las solicitudes y las respuestas se transfieran correctamente entre el cliente y el servidor. Habitualmente se utiliza Internet como medio para enviar esta información. Algunos de estos dispositivos intermedios también pueden ayudar a acelerar la respuesta de Internet, almacenan páginas o información de forma local en lo que se denomina una caché y, a continuación, sirven este contenido a los clientes directamente desde la caché, en lugar de tener que transferirlo desde el servidor de origen.

Un servidor web normalmente puede manejar múltiples conexiones de forma simultánea, y cuando no está comunicándose con un cliente, se dedica a escuchar para detectar una conexión entrante. Cuando llega una conexión, el servidor envía una respuesta para confirmar su recepción.

Procedimiento de solicitud/respuesta

En su nivel más básico, el proceso de solicitud/respuesta consiste en una pregunta que formula el navegador web al servidor web para que este le envíe una página web, y el servidor le envía la página. El navegador se encarga de mostrar la página (ver Figura 1-1).

1. Introducción al contenido dinámico de la web

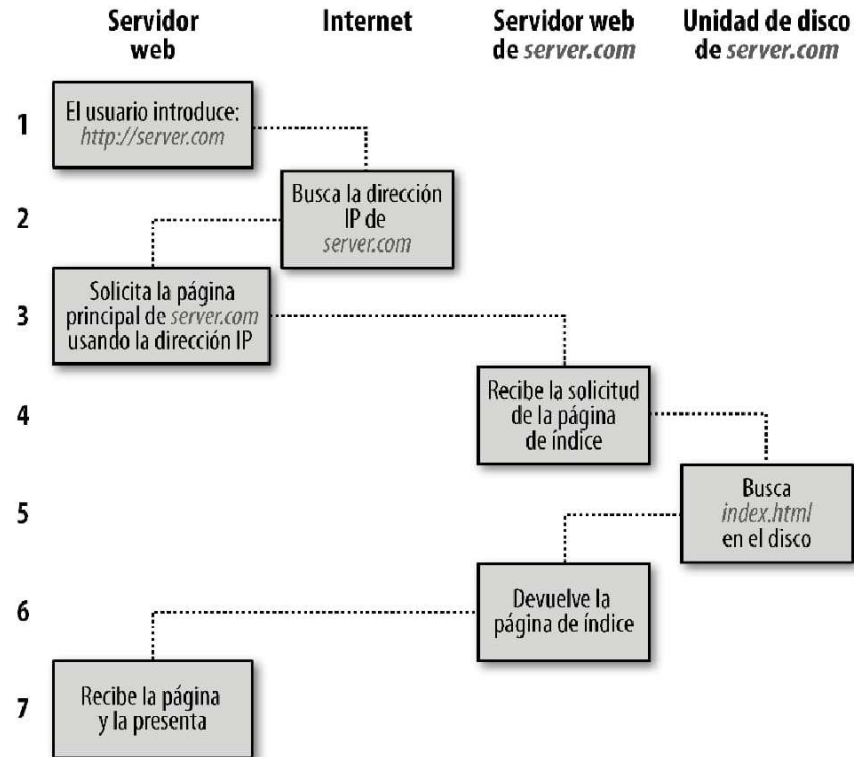


Figura 1-1. Secuencia básica de solicitud/respuesta entre cliente/servidor

Los pasos en la secuencia de solicitud y respuesta son los siguientes:

1. Introduces `http://server.com` en la barra de direcciones del navegador.
2. El navegador busca la dirección del protocolo Internet (IP) de `server.com`.
3. El navegador emite la solicitud de la página de inicio de `server.com`.
4. La solicitud viaja por Internet y llega al servidor web de `server.com`.
5. El servidor web, una vez recibida la petición, busca la página web en su disco.
6. El servidor web recupera la página de su disco y la envía al navegador.
7. El navegador muestra la página web.

Para una página web normal, este proceso también se lleva a cabo para cada objeto dentro de la página, ya sea un gráfico, un vídeo integrado o un archivo flash, o incluso una plantilla CSS.

En el paso 2, observamos que el navegador busca la dirección IP de `server.com`. Cada máquina conectada a Internet tiene una dirección IP (incluido nuestro ordenador) pero generalmente accedemos a los servidores web utilizando un nombre, como por ejemplo `google.com`. Como probablemente ya sabes, el navegador consulta un servicio adicional

Aprender PHP, MySQL y JavaScript

de Internet llamado Servicio de Nombres de Dominio (DNS) para encontrar la dirección IP asociada al servidor y, posteriormente, la utiliza para comunicarse con el ordenador.

Para páginas web dinámicas, el procedimiento es un poco más complicado, porque puede ser una mezcla de PHP y MySQL. Por ejemplo, supongamos que hacemos clic sobre la imagen de un impermeable. A continuación, PHP creará una petición usando el lenguaje de base de datos estándar, SQL (muchos de cuyos comandos aprenderás en este libro) y la enviará al servidor MySQL. El servidor MySQL devolverá la información del impermeable que hemos seleccionado, el código PHP lo encerrará todo en HTML, y el servidor lo enviará al navegador (ver la Figura 1-2).

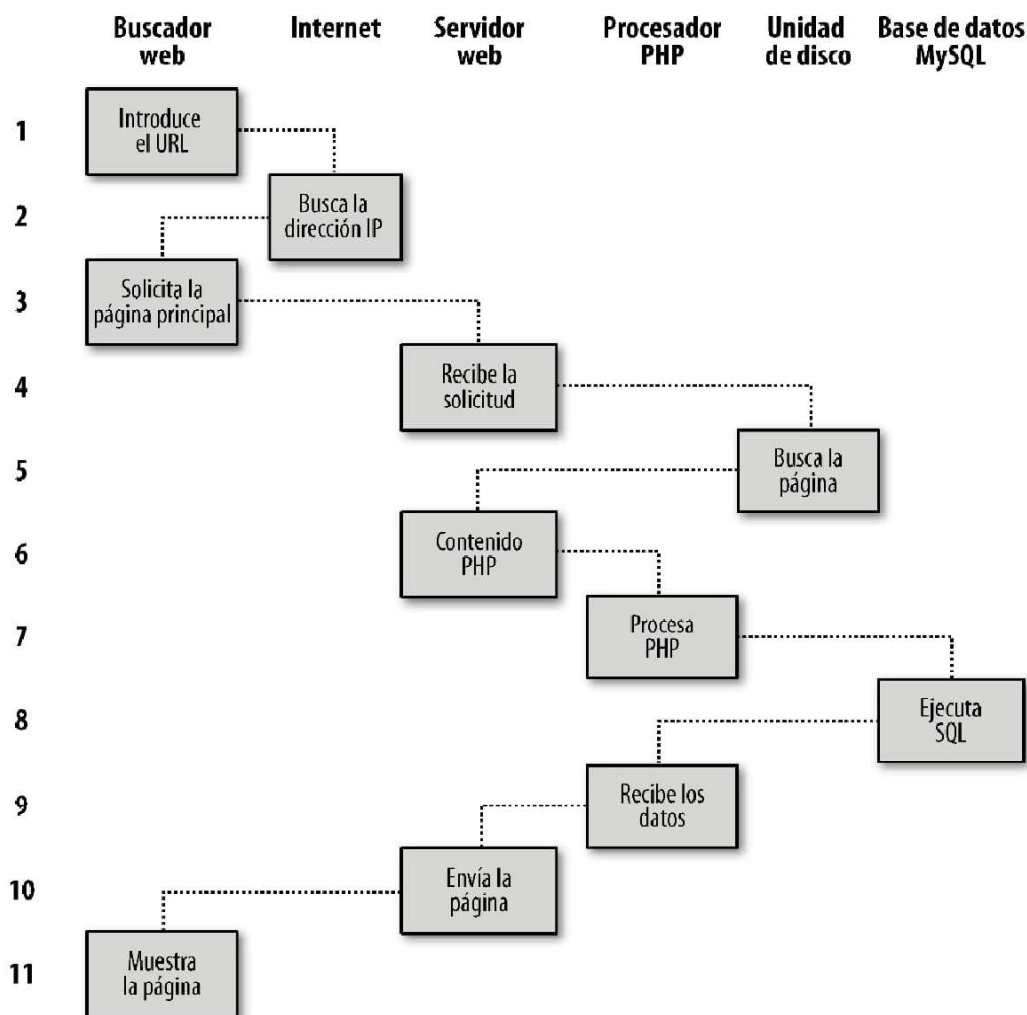


Figura 1-2. Secuencia dinámica de solicitud/respuesta entre cliente/servidor

Los pasos son los siguientes:

1. Tecleas *http://server.com* en la barra de direcciones del navegador.
2. El navegador busca la dirección IP de *server.com*.

1. Introducción al contenido dinámico de la web

3. El navegador envía una solicitud de la página de inicio del servidor web, a esa dirección.
4. La solicitud viaja por Internet y llega al servidor web *server.com*.
5. El servidor web, una vez recibida la petición, recupera la página de inicio de su disco duro.
6. Con la página de inicio ahora en memoria, el servidor web se da cuenta de que es un archivo que incorpora PHP y pasa la página al intérprete de PHP.
7. El intérprete de PHP ejecuta el código PHP.
8. Algunos de los códigos PHP contienen sentencias SQL, que el intérprete PHP ahora pasa al motor de base de datos MySQL.
9. La base de datos MySQL devuelve los resultados de las sentencias al intérprete PHP.
10. El intérprete PHP devuelve los resultados del código PHP ejecutado, junto con el archivo de la base de datos MySQL, al servidor web.
11. El servidor web devuelve la página al programa cliente que la ha solicitado, y este la muestra al usuario.

Aunque es útil conocer este proceso para saber cómo funcionan los tres elementos juntos, en la práctica no es necesario que nos preocupemos por ellos, porque todos se ejecutan automáticamente.

Las páginas HTML, devueltas al navegador en cada uno de los ejemplos, pueden contener código en JavaScript, que interpretará localmente el cliente, el cual podría iniciar otra petición, de la misma manera que lo podrían hacer los objetos integrados o las imágenes.

Ventajas de PHP, MySQL, JavaScript, CSS y HTML5

Al principio de este capítulo, introduje el mundo de la Web 1.0, pero no había transcurrido mucho tiempo cuando se dieron prisa en crear la Web 1.1, que incorporaba el desarrollo de mejoras en el navegador como Java, JavaScript, JScript (una variante con pocos cambios de JavaScript de Microsoft) y ActiveX. Por lo que respecta al servidor, se estaban realizando progresos en la interfaz de pasarela común (CGI) con lenguajes de scripting (secuencia de comandos) como Perl (una alternativa al lenguaje PHP) y scripts, insertando el contenido de un archivo (o la salida de la ejecución de un programa local) en otro, de forma dinámica.

Una vez calmados los ánimos, tres tecnologías principales se diferenciaban de las demás. Aunque Perl seguía siendo un lenguaje de programación con un gran número de seguidores, la simplicidad de PHP y los enlaces integrados al programa de base de datos MySQL habían ganado la partida: contaban con más del doble del número de usuarios. Y JavaScript, que se había convertido en una parte esencial de la ecuación para manipular

Aprender PHP, MySQL y JavaScript

dinámicamente Cascading Style Sheets (CSS) y HTML, ahora asumía la tarea aún más difícil de manejar el lado de cliente de la comunicación asíncrona (el intercambio de datos entre el cliente y el servidor después de que se haya cargado una página web). Mediante el uso de la comunicación asíncrona, las páginas web realizan el tratamiento de datos y envían solicitudes a los servidores web en segundo plano, sin que el usuario de la web sea consciente de que esto esté sucediendo.

Sin duda la naturaleza simbiótica de PHP y MySQL ayudó a impulsar a ambos, pero ¿qué fue lo que más atrajo a los desarrolladores de estos dos lenguajes de programación? La respuesta más sencilla es la facilidad con la que se pueden utilizar para crear rápidamente elementos dinámicos en sitios web. MySQL es un sistema de base de datos rápido y potente, pero al mismo tiempo fácil de usar, que ofrece casi todo lo que un sitio web necesita para encontrar y servir datos a los navegadores. Cuando PHP se alía con MySQL para almacenar y recuperar estos datos, tenemos los componentes fundamentales necesarios para poder desarrollar sitios de redes sociales, lo que marca el comienzo de la Web 2.0.

Y cuando también se incorporan al conjunto JavaScript y CSS, disponemos de la receta para crear sitios web altamente dinámicos e interactivos, especialmente porque ahora hay una amplia gama de sofisticados entornos de funciones JavaScript a los que se puede recurrir para acelerar de forma real el desarrollo web, como la conocida biblioteca jQuery, que ahora es probablemente el recurso más habitual que utilizan los programadores para acceder a las funciones de comunicación asíncrona.

MariaDB: el clon de MySQL

Después de que Oracle comprara Sun Microsystems (los propietarios de MySQL), surgió en la comunidad de desarrolladores la preocupación de que MySQL no siguiera siendo totalmente de software libre, así que, para asegurar esto, MariaDB se escindió de MySQL para mantenerse como software libre bajo licencia GNU GPL. El desarrollo de MariaDB está liderado por algunos de los desarrolladores que crearon MySQL y su compatibilidad es extremadamente alta con MySQL. Por lo tanto, podemos encontrar instalado MariaDB en algunos servidores en lugar de MySQL, pero en lo que respecta a los ejemplos de este libro, no hay que preocuparse, todos ellos funcionan igualmente bien tanto con MySQL como con MariaDB, sistema que se basa en el mismo código base que MySQL Server 5.5. A todos los efectos se pueden intercambiar y no se aprecia ninguna diferencia.

De todos modos, muchos de los temores iniciales parecen haberse disipado, ya que MySQL sigue siendo de software libre, y Oracle solamente cobra por el soporte y las ediciones que proporcionan características adicionales como la georreplicación y el escalado automático. Sin embargo, a diferencia de MariaDB, la comunidad ya no gestiona MySQL, así que si tenemos en cuenta que MariaDB siempre estará ahí por si alguna vez se necesita, proporcionará mucha tranquilidad a los desarrolladores y, probablemente, garantice que el mismo MySQL siga siendo de software libre.

Utilización de PHP

Con PHP, integrar la actividad dinámica en páginas web es una cuestión sencilla. Cuando le asignamos a las páginas la extensión *.php*, tienen acceso instantáneo a este lenguaje de scripting. Desde el punto de vista del desarrollador, todo lo que hay que hacer es escribir código como el siguiente:

```
<?php
echo " Today is " . date("l") . ". ";
?>
Here's the latest news.
```

La apertura `<?php` le dice al servidor web que permita al programa PHP interpretar el código que sigue a continuación hasta la etiqueta `?>`. Fuera de este constructor, todo se envía al cliente directamente como HTML. Entonces, el texto `Here's the latest news.` se envía al navegador, y dentro de las etiquetas PHP, la función integrada `date` (fecha), muestra el día en curso de la semana de acuerdo con la hora del sistema del servidor.

La salida final de las dos partes será:

```
Today is Wednesday. Here's the latest news.
```

PHP es un lenguaje flexible, y algunos prefieren colocar el constructor PHP directamente al lado del código PHP, así:

```
Today is <?php echo date("l"); ?>. Here's the latest news.
```

Hay incluso más posibilidades de formatear y dar salida a la información, que explicaré en los capítulos correspondientes a PHP. Lo importante es que con PHP, los desarrolladores web tienen un lenguaje de programación que, aunque no es tan rápido como la compilación del código en C o un lenguaje similar, es increíblemente rápido y también se integra perfectamente con el marcado HTML.



Si tienes la intención de introducir los ejemplos PHP de este libro en un programa editor para estudiarlos conmigo, debes acordarte de agregar `<? php` delante y `?>` después de ellos, para tener la seguridad de que el intérprete PHP los procesa. Para hacerlo más fácil, tal vez desees preparar un archivo llamado *example.php* con esas etiquetas en su lugar correspondiente.

Al utilizar PHP, tenemos un control ilimitado sobre nuestro servidor web. Ya sea que necesitemos modificar HTML sobre la marcha, procesar una tarjeta de crédito, añadir detalles de usuario a una base de datos o buscar información de un sitio web de terceros. Podemos hacerlo todo en los mismos archivos PHP en los que reside el propio HTML.

Utilización de MySQL

Por supuesto, no tiene mucho sentido cambiar a una salida HTML dinámica a menos que también tengamos un medio para rastrear la información que los usuarios proporcionan

Aprender PHP, MySQL y JavaScript

a nuestro sitio web a medida que lo utilizan. En los primeros tiempos de la web, muchos sitios utilizaban archivos de texto "plano" para almacenar datos como nombres de usuarios y contraseñas. Pero este enfoque podía causar problemas si el archivo no estaba adecuadamente bloqueado contra la corrupción de múltiples accesos simultáneos. Además, un archivo plano puede crecer hasta cierto punto antes de que sea difícil de manejar, por no mencionar la dificultad de intentar fusionar archivos y realizar búsquedas complejas en un periodo de tiempo razonable.

Ahí es donde las bases de datos relacionales con consultas estructuradas se convierten en esenciales. Y MySQL, al poderse utilizar e instalar libremente en un gran número de servidores web de Internet, viene magníficamente para la ocasión. Es un sistema de gestión de bases de datos robusto y excepcionalmente rápido que utiliza comandos en inglés.

El nivel más alto de la estructura MySQL es una base de datos, dentro de la cual pueden existir una o más tablas que contienen los datos. Por ejemplo, supongamos que estamos trabajando en una tabla llamada `users` (usuarios), dentro de la cual hemos creado columnas para `surname` (apellido) `firstname` (nombre) y `email` (correo electrónico), y ahora deseamos añadir otro usuario. Un comando que podemos usar para hacerlo es el siguiente:

```
INSERT INTO users VALUES('Smith', 'John', 'jsmith@mysite.com');
```

Anteriormente habremos utilizado otros comandos para crear la base de datos y la tabla, y para configurar todos los campos correctamente, pero el comando SQL `INSERT` en este caso muestra lo sencillo que puede ser añadir nuevos datos a una base de datos. SQL es un lenguaje diseñado a principios de los años 70 que recuerda a uno de los lenguajes de programación más antiguos, COBOL. Sin embargo, se adapta muy bien a las consultas de bases de datos, por lo que sigue utilizándose después de todo este tiempo.

Es igualmente fácil buscar datos. Supongamos que tenemos la dirección de correo electrónico de un usuario y necesitamos buscar el nombre de esa persona. Para ello, podemos generar una consulta MySQL como la siguiente:

```
SELECT surname,firstname FROM users WHERE  
email='jsmith@mysite.com';
```

MySQL devolverá `Smith, John` y cualesquiera otros pares de nombres y apellidos que puedan estar asociados con esa dirección de correo electrónico en la base de datos.

Como es de esperar, podemos hacer mucho más con MySQL de lo que hacen los comandos básicos `INSERT` y `SELECT`. Por ejemplo, se pueden combinar conjuntos de datos relacionados para reunir información relacionada, pedir resultados mediante órdenes muy variadas, hacer coincidencias parciales cuando solo conocemos una parte de la cadena que estamos buscando, devolver solo el *n*-ésimo resultado, etc.

Con PHP, podemos hacer todas estas llamadas directamente a MySQL sin que tener que acceder a la interfaz de línea de comandos MySQL. Esto significa que podemos guardar los resultados en matrices para procesar y realizar búsquedas múltiples, cada una dependiente de los resultados devueltos de las anteriores, para profundizar en el elemento de datos que necesitamos.

1. Introducción al contenido dinámico de la web

Para conseguir un mayor rendimiento, como veremos más adelante, hay funciones adicionales incorporadas a MySQL a las que podemos invocar para ejecutar eficientemente operaciones habituales dentro de MySQL, en lugar de crearlas a partir de múltiples llamadas de PHP a MySQL.

Utilización de JavaScript

La más antigua de las tres tecnologías básicas expuestas en este libro, JavaScript, se creó para permitir el acceso de la programación a todos los elementos de un documento HTML. En otras palabras, proporcionaba un medio para la interacción dinámica del usuario, como la comprobación de la validez de la dirección de correo electrónico en formularios de entrada y la visualización de indicaciones como "¿Querías realmente decir eso?" (sin embargo, no se puede confiar en esta tecnología para temas de seguridad, que siempre se deben llevar a cabo en el servidor web).

Combinado con CSS (ver la siguiente sección), JavaScript es el responsable que hay detrás de las páginas web dinámicas que cambian ante nuestros propios ojos, a diferencia del caso en el que el servidor devuelve una página nueva.

Sin embargo, JavaScript también puede resultar difícil de usar, debido a algunas diferencias importantes entre las formas en las que los diferentes diseñadores de navegadores han elegido implementarlo. Esto sucedió principalmente cuando algunos fabricantes trataron de incluir funcionalidades adicionales a sus navegadores a expensas de la compatibilidad con sus rivales.

Afortunadamente, los desarrolladores han entrado en razón y se han dado cuenta de la necesidad de una compatibilidad total entre sí, por lo que hoy en día es menos exigente tener que optimizar nuestro código para diferentes navegadores. Sin embargo, aún quedan millones de usuarios que utilizan navegadores heredados, y este seguirá siendo probablemente el caso durante muchos años. Afortunadamente, hay soluciones para los problemas de incompatibilidad; más tarde, en este libro, veremos las bibliotecas y técnicas que nos permiten poder ignorar con total seguridad estas diferencias.

Por ahora, veamos cómo usar JavaScript básico, aceptado por todos los navegadores:

```
<script type="text/javascript">
  document.write("Today is " + Date() );
</script>
```

Este fragmento de código le dice al navegador web que interprete todo dentro de las etiquetas `<script>` como JavaScript, y el navegador lo hace escribiendo el texto `Today is` en el documento, junto con la fecha, mediante la función JavaScript `Date`. El resultado será algo así:

Today is Sun Jan 01 2017 01:23:45



A menos que necesitemos especificar una versión concreta de JavaScript, se puede normalmente omitir `type="text/javascript"` y solo usar `<script>` para iniciar la interpretación de JavaScript.

Aprender PHP, MySQL y JavaScript

Como se mencionó anteriormente, JavaScript se desarrolló originalmente para ofrecer un control dinámico sobre los diversos elementos dentro de un documento HTML, y ese sigue siendo su principal cometido. Pero cada vez más, JavaScript se utiliza para la comunicación asíncrona, el proceso de acceso al servidor web en segundo plano.

La comunicación asíncrona es lo que permite que las páginas web empiecen a parecerse a los programas independientes, porque no hay que volverlas a cargar en su totalidad para mostrar un nuevo contenido. En lugar de eso, una llamada asíncrona puede incorporar y actualizar un solo elemento en una página web, como cambiar tu fotografía en un sitio de redes sociales o reemplazar un botón en el que se hace clic para responder a una pregunta. Este tema se trata con detalle en el Capítulo 17.

Después, en el Capítulo 21, echamos un vistazo al entorno de trabajo de jQuery, que podemos usar para evitar tener que reinventar la rueda cuando necesitamos un código rápido de navegador cruzado para manipular nuestras páginas web. Por supuesto, también hay otros entornos de trabajo disponibles, pero jQuery es, con diferencia, el más popular. Debido a su mantenimiento continuo, es extremadamente fiable y es una herramienta importante en el kit de utilidades de muchos desarrolladores web experimentados.

Utilización de CSS

CSS es el compañero indispensable de HTML, con el que se asegura que el texto HTML y las imágenes se presentan de forma coherente y adecuada en la pantalla del usuario. Con la aparición del estándar CSS3 en los últimos años, CSS ofrece ahora un nivel de interactividad dinámica anteriormente soportada solo por JavaScript. Por ejemplo, no solo se puede cambiar el estilo de cualquier elemento HTML para modificar sus dimensiones, colores, bordes, espaciado, etc., sino que ahora también podemos añadir transiciones animadas y transformaciones a las páginas web, con solo unas pocas líneas de CSS.

El uso de CSS puede ser tan simple como insertar algunas reglas entre las etiquetas `<style>` y `</style>` en el encabezamiento de una página web, así:

```
<style>
  p {
    text-align:justify;
    font-family:Helvetica;
  }
</style>
```

Estas reglas cambian la alineación predeterminada del texto de la etiqueta `<p>` para que los párrafos contenidos en ella estén totalmente justificados y utilicen la fuente helvética.

Como veremos en el Capítulo 18, existen muy variadas formas de diseñar reglas CSS; las podemos incluir directamente dentro de etiquetas o guardar un conjunto de reglas en un archivo externo, que se carga por separado. Esta flexibilidad no solo nos permite estilizar nuestro HTML con precisión, sino que también podemos, por ejemplo, proporcionar la funcionalidad integrada de movimiento de los objetos cuando el ratón pasa sobre ellos. También aprenderás a acceder a todas las funciones de un elemento desde JavaScript y HTML.

Y luego está HTML5

A pesar de la utilidad de todas estas incorporaciones a los estándares web, no fueron suficientes para desarrolladores cada vez más ambiciosos. Por ejemplo, todavía no existía una forma sencilla de manipular gráficos en un navegador web sin recurrir a complementos como Flash. Y lo mismo ocurría con la inserción de audio y vídeo en las páginas web. Además, varias inconsistencias molestas se habían colado en HTML durante su evolución.

Por lo tanto, para aclarar todo esto y llevar Internet más allá de la Web 2.0 y a su próxima iteración, se creó un nuevo estándar para HTML con el fin de subsanar todas estas deficiencias: *HTML5*. Su desarrollo comenzó ya en 2004, cuando la Fundación Mozilla y Opera Software (desarrolladores de dos populares webs) redactaron el primer borrador, pero no fue hasta principios de 2013, año en el que se envió el borrador definitivo al World Wide Web Consortium (W3C), el organismo internacional que gobierna los estándares de calidad de la web.

El desarrollo de HTML5 ha necesitado varios años, pero ahora estamos en una fase muy sólida y estable de la versión 5.1 (desde 2016). Es un ciclo interminable de desarrollo. Sin embargo, es seguro que con el tiempo aumentará su funcionalidad. Algunas de las mejores características de HTML5 para el manejo y visualización de los media incluyen el `<audio>`, el `<video>` y los elementos `<canvas>` y `(lienzo)`, que añaden sonido, vídeo y gráficos avanzados. Todo lo que necesitas saber sobre estos y todos los demás aspectos de HTML5 se trata con todo detalle a partir del Capítulo 23.



Una de las pequeñas cosas que me gusta de la especificación HTML5 es que la sintaxis XHTML ya no es necesaria para los elementos de auto cierre. Antes, se podía visualizar un salto de línea con el elemento `<br>`. Luego, para asegurar la compatibilidad futura con XHTML (el sustituto que se había planeado para HTML, lo que nunca ocurrió), se cambió a `<br />`, en el que se agregó un carácter de cierre / (ya que se esperaba que todos los elementos incluyeran una etiqueta de cierre con este carácter). Pero ahora las cosas han cambiado por completo, y se puede usar cualquier versión de este tipo de elementos. Por lo tanto, en aras de la brevedad y de tener que pulsar menos teclas, en este libro he vuelto al estilo anterior de `<br>`, `<hr>`, etc.

El servidor web Apache

Además de PHP, MySQL, JavaScript, CSS y HTML5, hay un sexto héroe en la web dinámica: el servidor web. En el caso de este libro, este es el servidor web Apache. Hemos discutido un poco sobre lo que hace un servidor web durante el intercambio HTTP entre servidor/cliente, pero hace muchas más cosas entre bastidores.

Por ejemplo, Apache no solo sirve archivos HTML, sino que maneja una amplia gama de archivos, desde imágenes y archivos Flash hasta archivos de audio MP3, RSS (Really Simple Syndication), etc. Y estos objetos no tienen que ser archivos estáticos como las

imágenes GIF. Todos ellos pueden estar generados por programas como scripts PHP. Así es: PHP puede incluso crear imágenes y otros archivos para nosotros, ya sea sobre la marcha o por adelantado para servirlos más tarde.

Para hacer esto, normalmente tiene módulos bien precompilados en Apache o PHP o bien se los llama durante la ejecución. Uno de estos módulos es la biblioteca GD (Graphics Draw), que PHP usa para crear y manejar gráficos.

Apache también soporta una amplia gama de módulos propios. Además de PHP, lo más importante para nuestros propósitos como programadores web son los módulos que se encargan de la seguridad. Otros ejemplos son el módulo Rewrite, que permite al servidor web manejar un rango de tipos de URL y reescribirlos adaptándolos a sus propios requerimientos internos, y el módulo Proxy, que se puede utilizar para servir páginas que se solicitan con frecuencia, desde una caché, para agilizar la carga del servidor.

Más adelante, en el libro, veremos cómo usar algunos de estos módulos para mejorar las características que proporcionan las tres principales tecnologías.

Uso de dispositivos móviles

Actualmente vivimos en un mundo de dispositivos informáticos móviles interconectados, y el concepto de desarrollar sitios web solo para ordenadores de mesa se ha vuelto bastante anticuado. En lugar de esto, los desarrolladores ahora pretenden desarrollar sitios y aplicaciones web sensibles que se adapten al entorno en el que se están ejecutando.

Así que como novedad en esta edición, explico cómo se pueden crear fácilmente este tipo de productos solo con las tecnologías detalladas en este libro, junto con la potente tecnología jQuery Mobile de funciones sensibles JavaScript. Con ella, podrás centrarte en el contenido y usabilidad de tus sitios y aplicaciones web, sabiendo que la forma en que se muestran se optimizará automáticamente para una amplia gama de dispositivos informáticos diferentes: una cosa menos de la que preocuparte.

Para mostrar cómo hacer pleno uso de sus capacidades, en el capítulo final de este libro se crea un sencillo sitio web de ejemplo de red social. Para ello utilizamos jQuery Mobile, para que tengas capacidad de respuesta y estés seguro de que se presenta adecuadamente en cualquier dispositivo, desde la pantalla de un pequeño teléfono móvil a una tablet o a un ordenador de mesa.

Sobre el software libre

Las tecnologías que se tratan en este libro son de software libre: a cualquier persona se le permite leer y cambiar el código. Se ha debatido a menudo si este hecho es o no la razón por la que estas tecnologías son tan populares, sea como sea, PHP, MySQL y Apache *son* las tres herramientas que se usan con más frecuencia en sus categorías. Lo que se puede decir de manera definitiva, sin embargo, es que el hecho de que sean de software libre significa que se han desarrollado en la comunidad por equipos de

1. Introducción al contenido dinámico de la web

programadores que escriben las características que ellos mismos quieren y necesitan, y el código original está disponible para que todos tengamos acceso al mismo e incluso podamos modificarlo. Los errores se pueden encontrar rápidamente y las infracciones de seguridad se pueden prevenir antes de que ocurran.

Hay otra ventaja: todos estos programas son gratuitos. No hay que preocuparse por tener que comprar licencias adicionales si tienes que ampliar tu sitio web y añadir más servidores, y no necesitas estudiar el presupuesto antes de decidir si deseas actualizar estos productos a las últimas versiones.

Todo junto

La verdadera belleza de PHP, MySQL, JavaScript (a veces con la ayuda de jQuery u otros entornos de trabajo), CSS y HTML5 es la maravillosa forma en que todos ellos trabajan juntos para crear contenido web dinámico: PHP gestiona el trabajo principal en el servidor web, MySQL gestiona los datos y la combinación de CSS y JavaScript se encarga de la presentación de la página web. JavaScript también puede hablar con tu código PHP en el servidor web siempre que necesites actualizar algo (ya sea en el servidor o en la página web). Y con las nuevas y potentes funciones de HTML5, como el lienzo, el audio, el vídeo y la geolocalización, puedes hacer que tus páginas web sean muy dinámicas, interactivas y estén repletas de multimedia.

Sin usar código de programación, vamos a resumir el contenido de este capítulo estudiando el proceso de combinar algunas de estas tecnologías con la característica de comunicación asíncrona cotidiana que utilizan muchos sitios web: comprobar si ya existe un nombre de usuario deseado en el sitio cuando un usuario se está registrando para crear una nueva cuenta. Un buen ejemplo de esto se puede ver en Gmail (ver la Figura 1-3).

A screenshot of a web browser window showing the Gmail sign-up page. The browser's address bar displays a URL from accounts.google.com. The page content includes a heading 'One account is all you need' with a subtext 'One free account gets you into everything Google.' Below this are icons for various Google services. A section titled 'Take it all with you' mentions switching between devices. On the right side, there is a registration form with the following fields: 'Name' (split into 'First name' with 'Samuel' and 'Last name' with 'Smith'), 'Choose your username' (with 'samsmith@gmail.com' and a message 'That username is taken. Try another.'), 'Create a password' and 'Confirm your password' (both empty), and 'Birthday' (with dropdowns for 'Month', 'Day', and 'Year').

Figura 1-3. Gmail usa la comunicación asíncrona para verificar la disponibilidad de nombres de usuario

Aprender PHP, MySQL y JavaScript

Los pasos a seguir en este proceso asíncrono serán parecidos a los siguientes:

1. El servidor genera el código HTML para crear el formulario web, en el que se pide la información necesaria, como el nombre de usuario, nombre, apellido y dirección de correo electrónico.
2. Al mismo tiempo, el servidor añade código JavaScript a HTML para monitorizar el cuadro de entrada de nombre de usuario y comprobar dos cosas: si se ha tecleado texto y si la entrada ha dejado de estar seleccionada debido a que el usuario hizo clic en otro cuadro de entrada.
3. Una vez que se ha introducido el texto y se ha deseleccionado el campo, el código JavaScript pasa, en segundo plano, el nombre de usuario que se escribió, a un script PHP en el servidor web y espera una respuesta.
4. El servidor web busca el nombre de usuario y responde a JavaScript en cuanto a si ese nombre ya existe.
5. JavaScript entonces coloca una indicación al lado del cuadro de entrada del nombre de usuario para mostrar si el nombre está disponible, tal vez una marca de verificación verde o un gráfico con una cruz roja, junto con alguna indicación de texto.
6. Si el nombre de usuario no está disponible y el usuario sigue enviando el formulario, JavaScript interrumpe la presentación y vuelve a hacer hincapié (quizás con un gráfico más grande y/o un cuadro de aviso) en que el usuario necesita elegir otro nombre de usuario.
7. Opcionalmente, una versión mejorada de este proceso podría incluso examinar el nombre de usuario y sugerir una alternativa que esté disponible en ese momento.

Todo esto se lleva a cabo discretamente, en segundo plano, y hace que la experiencia del usuario sea cómoda e impecable. Sin comunicación asíncrona, se tendría que enviar todo el formulario al servidor, que entonces devolvería el HTML y resaltaría cualquier error. Sería una solución viable, pero ni de lejos tan ordenada o placentera como el procesamiento de los campos del formulario sobre la marcha.

Sin embargo, la comunicación asíncrona se puede utilizar para mucho más que una simple verificación y procesamiento de entradas. Vamos a explorar muchas otras cosas que podemos hacer con ella más adelante en este libro.

En este capítulo, he presentado una completa introducción a las tecnologías básicas de PHP, MySQL, JavaScript, CSS y HTML5 (así como Apache), y has aprendido cómo funcionan juntas. En el Capítulo 2, veremos cómo puedes instalar tu propio servidor de desarrollo web en el que practicar todo lo que vas a aprender.

Preguntas

1. ¿Cuáles son los cuatro componentes (como mínimo) necesarios para crear una página web completamente dinámica?

1. Introducción al contenido dinámico de la web

2. ¿Qué significa HTML?
3. ¿Por qué el nombre MySQL contiene las letras SQL?
4. PHP y JavaScript son lenguajes de programación que generan resultados dinámicos para páginas web. ¿Cuál es su principal diferencia, y por qué los usarías a los dos?
5. ¿Qué significa CSS?
6. Enumera tres nuevos elementos principales introducidos en HTML5.
7. Si encuentras un error (que es raro) en una de las herramientas de software libre, ¿cómo crees que podrías arreglarlo?
8. ¿Por qué un framework como jQuery Mobile es tan importante para desarrollar sitios web y aplicaciones web modernas?

Consulta "Respuestas del Capítulo 1" en la página 705 del Apéndice A para comprobar las respuestas a estas preguntas.

Capítulo 2

Configuración del servidor de desarrollo

Si deseas desarrollar aplicaciones de Internet pero no tienes tu propio servidor de desarrollo, tendrás que subir cada modificación que hagas a un servidor situado en algún lugar de la web antes de que puedas probarla.

Incluso en el caso de una conexión de banda ancha de alta velocidad, esto puede suponer una importante pérdida de tiempo de desarrollo. Sin embargo, en un ordenador local las pruebas pueden ser tan sencillas como guardar una actualización (que normalmente es cuestión de hacer clic una vez en un icono) y luego pulsar el botón Refresh (Actualizar) en el navegador.

Otra ventaja de disponer de un servidor de desarrollo es que cuando programas y haces pruebas, no tienes que preocuparte de errores embarazosos o por problemas de seguridad. Sin embargo, cuando tu aplicación está en un sitio web público, tienes que estar al tanto de lo que la gente puede ver o hacer con ella. Es mejor solucionarlo todo mientras todavía la tienes instalada en un equipo de casa o de la oficina, los cuales están presumiblemente protegidos por cortafuegos y otras medidas de seguridad.

Una vez que tengas tu propio servidor de desarrollo, te preguntarás cómo has podido sobrevivir sin uno. Es muy fácil crear uno. Solo tienes que seguir los pasos que se detallan en las siguientes secciones y utilizar las instrucciones apropiadas para PC, Mac o para el sistema Linux.

En este capítulo, trataremos de la experiencia web solo el lado del servidor, como se describe en el Capítulo 1. Pero para probar los resultados de tu trabajo, especialmente cuando más adelante, en este libro, empecemos a usar JavaScript, CSS y HTML5, lo ideal sería tener funcionando los principales navegadores web en algún sistema que te resulte cómodo. Siempre que sea posible, en la lista de navegadores deben figurar al menos Microsoft Edge, Mozilla Firefox, Opera, Safari y Google Chrome. Si quieres tener la seguridad de que tus sitios se vean bien en dispositivos móviles, debes tratar de organizar el acceso a una amplia gama de dispositivos iOS y Android.

¿Qué son WAMP, MAMP o LAMP?

WAMP, MAMP y LAMP son abreviaturas de "Windows, Apache, MySQL y PHP", "Mac, Apache, MySQL y PHP" y "Linux, Apache, MySQL y PHP".

Cada una de estas abreviaturas describe una configuración funcional que se utiliza para desarrollar páginas web dinámicas en Internet.

WAMP, MAMP y LAMP se presentan en forma de paquetes en los que los programas están agrupados, lo que evita tener que instalarlos y configurarlos por separado. Esto significa que puedes descargar e instalar un solo programa y seguir algunas indicaciones sencillas para poner en marcha el servidor de desarrollo web de forma rápida y con las mínimas molestias.

Durante la instalación, se crean varias configuraciones por defecto. Las configuraciones de seguridad de una instalación de este tipo no serán tan estrictas como las de un servidor web de producción, ya que su desarrollo está optimizado para uso local. Por estas razones, no debes instalar estas configuraciones como un servidor de producción.

Sin embargo, para desarrollar y probar sitios web y aplicaciones, debería ser suficiente utilizar una de estas instalaciones.



Si decides no utilizar WAMP/MAMP/LAMP para crear tu sistema de desarrollo, debes saber que puede que tengas que dedicar mucho tiempo a la descarga e integración de las distintas partes, y es posible que requiera mucha investigación para configurarlo todo completamente. Pero, sin embargo, si ya tienes todos los componentes instalados e integrados entre sí, estos deben funcionar con los ejemplos de este libro.

Instalación de AMPPS en Windows

Hay varios servidores WAMP disponibles, cada uno de los cuales ofrece configuraciones ligeramente diferentes. De las varias opciones de software libre y gratuito, una de las mejores es AMPPS. Puedes descargar la aplicación haciendo clic en el botón de la página de inicio (<http://ampps.com/>), del sitio web, que se muestra en la Figura 2-1.

Te recomiendo que descargues siempre la última versión estable (mientras escribo esto, es la 3.8, que tiene un tamaño aproximado de 128 MB). Los distintos instaladores de Windows, macOS y Linux aparecen en la página de descargas.

2. Configuración del servidor de desarrollo

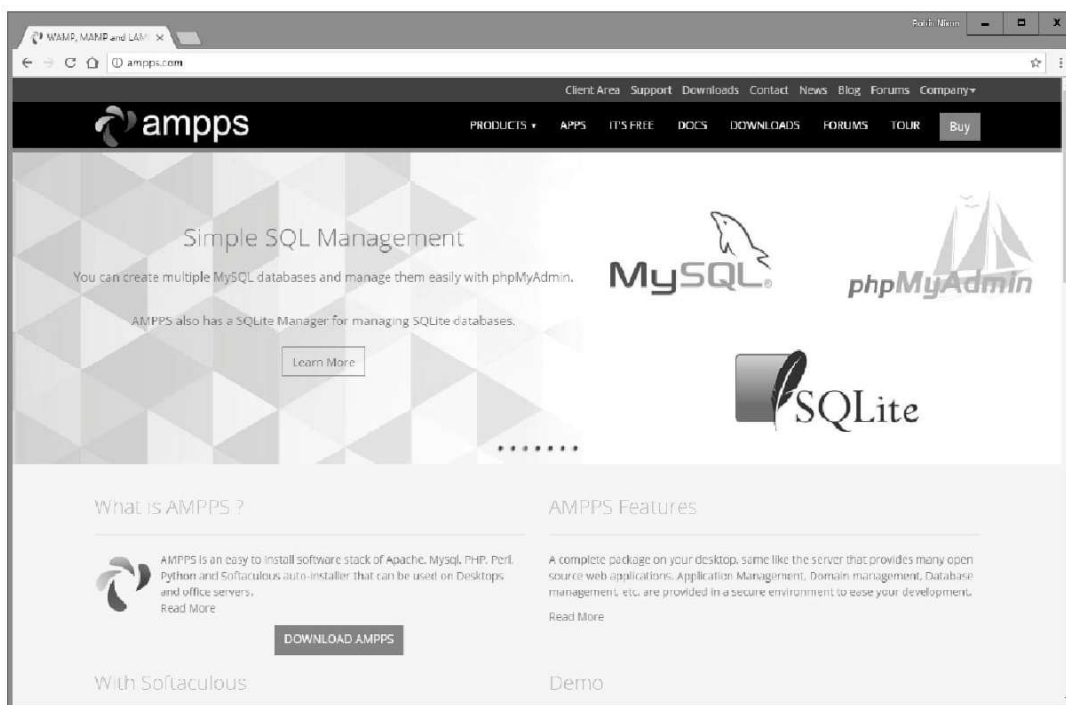


Figura 2-1. Sitio web de AMPPS



Durante la vigencia de esta edición, algunas de las pantallas y opciones que se muestran aquí pueden cambiar. Si es así, usa el sentido común para proceder de la manera más parecida posible a la secuencia de acciones que se indican.

Una vez que hayas descargado el instalador, ejecútalo y se abrirá una ventana como la que se muestra en la Figura 2-2. Sin embargo, antes de llegar a esa ventana, si utilizas un programa antivirus o tienes activado el control de cuentas de usuario de Windows, es posible que primero se te muestren uno o más avisos y tengas que hacer clic en Yes y/o OK para continuar con la instalación.

Haz clic en Next (Siguiente), después de lo cual debes aceptar el acuerdo. Vuelve a hacer clic en Next y, luego, otra vez para pasar la pantalla de información. Ahora tendrás que confirmar el lugar en el que se va a instalar AMPPS. Probablemente te sugerirá algo como lo siguiente, en función de la letra del disco duro principal de tu ordenador, pero lo puedes cambiar si lo deseas:

C:\Program Files (x86)\Ampps



Figura 2-2. Ventana de apertura del instalador

Una vez que hayas decidido dónde instalar AMPPS, haz clic en Next, elige un nombre de carpeta en el menú Start (Inicio) y vuelve a hacer clic en Next. Puedes elegir qué iconos desees instalar, como se muestra en la Figura 2-3. En la pantalla siguiente, haz clic en el botón Install (Instalar) para iniciar el proceso.

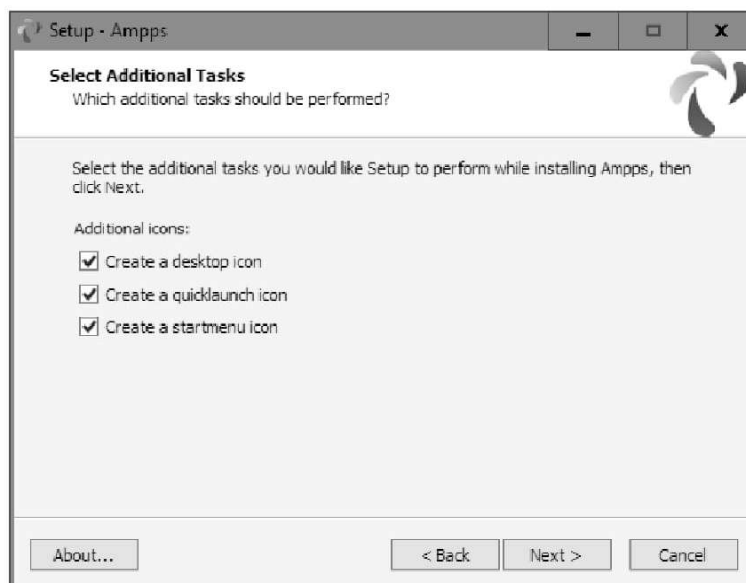


Figura 2-3. Elección de los iconos que desees instalar

La instalación durará unos minutos, después de los cuales verás la pantalla de finalización de la Figura 2-4, y puedes hacer clic en Finish (Terminar).

2. Configuración del servidor de desarrollo



Figura 2-4. AMPPS ya está instalado

Lo último que debes hacer es instalar C++ Redistributable Visual Studio, si aún no lo has hecho. Visual Studio es un entorno en el que realizaremos trabajo de desarrollo. Se abrirá una ventana para preguntarte sobre la instalación, como se muestra en la Figura 2-5. Haz clic en Yes para iniciar la instalación o en No si estás seguro de que ya la tienes.

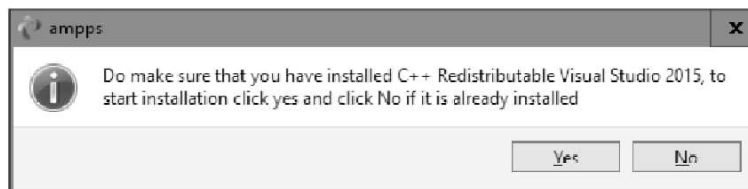


Figura 2-5. Instala C++ Redistributable Visual Studio si aún no lo tienes

Si decides continuar con la instalación, tendrás que aceptar los términos y condiciones que se muestran en la ventana emergente que aparece y, a continuación, hacer clic en Install. La instalación debería ser bastante rápida. Haz clic en Close (Cerrar) para finalizar.

Una vez instalado AMPPS, debería aparecer en la parte inferior derecha de tu escritorio la ventana de control que se muestra en la Figura 2-6. También puedes acceder a esta ventana mediante el acceso directo de la aplicación AMPPS en el menú Start (Inicio) o en el escritorio, si has permitido que se hayan creado estos iconos.



Figura 2-6. Ventana de control de AMPPS

Antes de continuar, te recomiendo que te familiarices con la documentación de AMPPS (<http://ampps.com/wiki>). Una vez que la hayas digerido, si todavía tiene alguna dificultad, hay un enlace de Support (Soporte) en la parte inferior de la ventana de control que te llevará al sitio web de AMPPS, donde puedes abrir un ticket con el problema.



Puedes observar que la versión por defecto de PHP en AMPPS es 5.6. En otras secciones de este libro detallo algunos de los cambios más importantes en PHP 7. Si deseas probarlos, haz clic en el botón Options (Opciones) (nueve casillas de color blanco en un cuadrado) dentro de la ventana de control de AMPPS, en la parte superior izquierda, y luego selecciona Change PHP Version (Cambiar versión de PHP), con lo cual aparecerá un nuevo menú desde el cual podrás elegir una versión desde la 5.6 a la 7.1.

Prueba de la instalación

Lo primero que hay que hacer llegados a este punto es verificar que todo funciona correctamente. Para ello, introducimos uno de los dos siguientes URL en la barra de direcciones del navegador:

```
localhost  
127.0.0.1
```

2. Configuración del servidor de desarrollo

Se abrirá una pantalla de introducción, en la que tendrás la oportunidad de proteger AMPPS proporcionando una contraseña (consultar la Figura 2-7). Te recomiendo que no marques la casilla y hagas clic en el botón Submit (Enviar) para continuar sin establecer una contraseña.

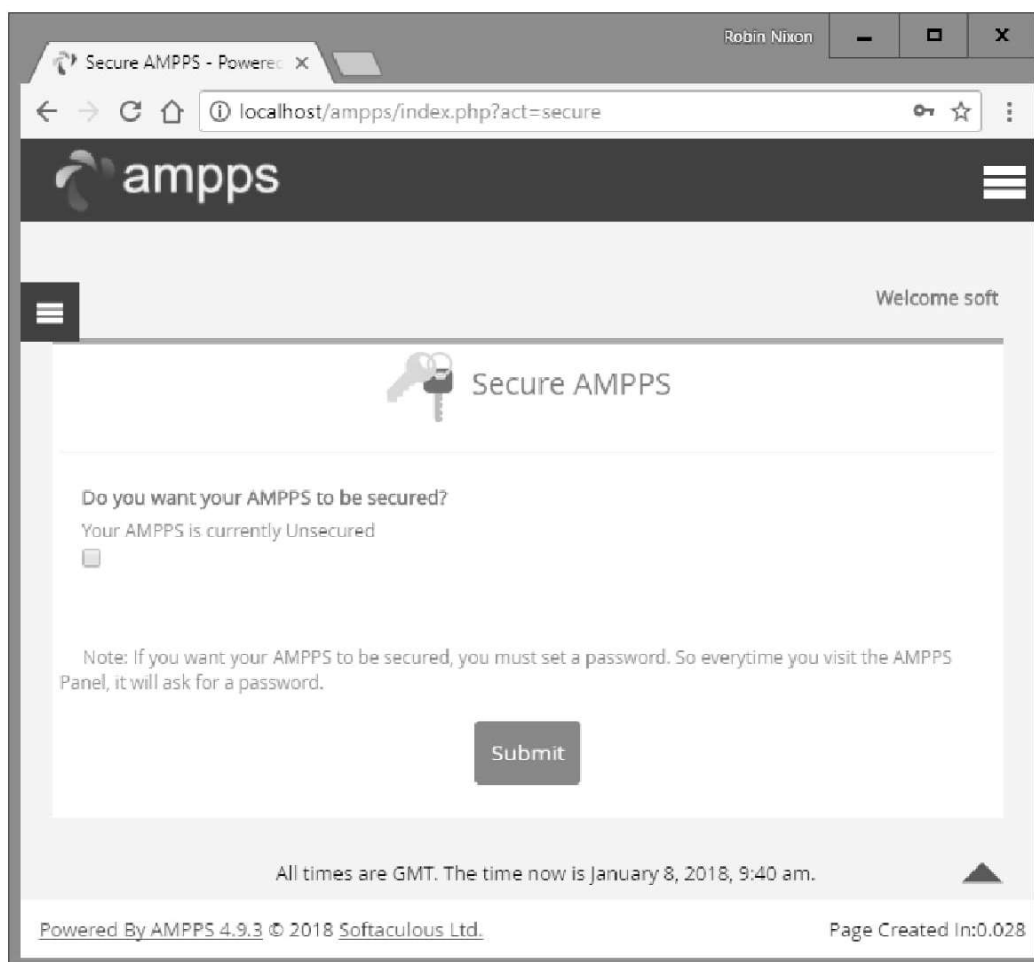


Figura 2-7. Pantalla inicial de la configuración de seguridad

Una vez hecho esto, te llevará a la página de control principal en *localhost/ampps/* (de ahora en adelante supongo que accedes a AMPPS a través de *localhost* en lugar de *127.0.0.1*). Desde aquí puedes configurar y controlar todos los aspectos de la pila AMPPS, así que toma nota de esto para futuras referencias, o quizás puedas poner un marcador en tu navegador.

A continuación, para ver la carpeta principal (descrita en la siguiente sección) de tu nuevo servidor web Apache, escribe lo siguiente en la barra del navegador:

```
localhost
```

Esta vez, en lugar de ver la pantalla inicial de la configuración de seguridad, deberías ver algo similar a la Figura 2-8.

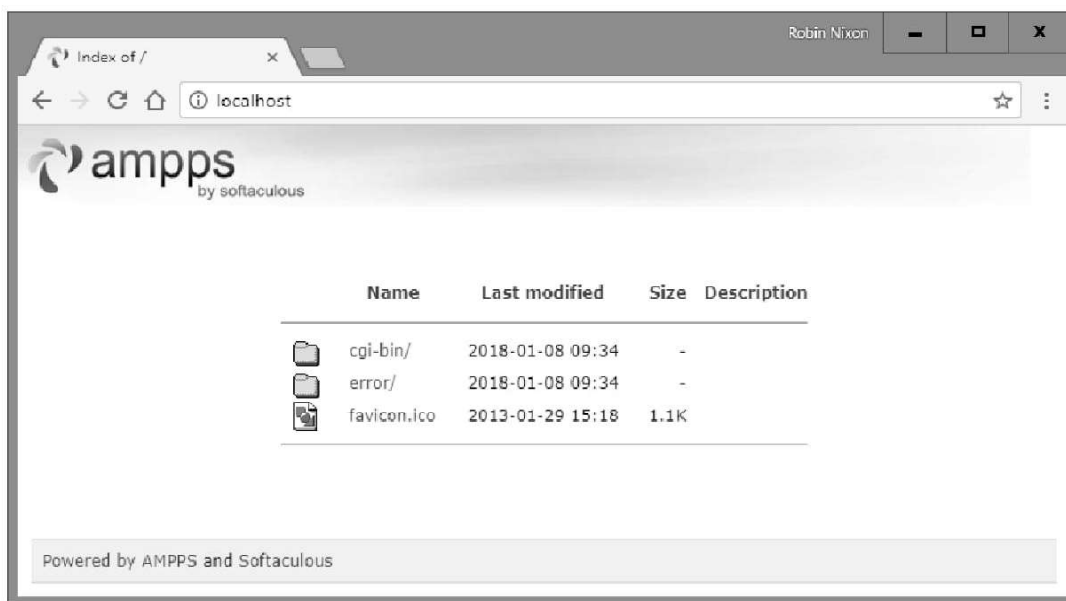


Figura 2-8. Visualización de la carpeta principal

Acceso a la carpeta principal (Windows)

La *carpeta principal* es el directorio que contiene los principales documentos web de un dominio. Este directorio es el que utiliza el servidor cuando se escribe en el navegador un URL básico sin ruta, como *http://yahoo.com* o, para tu servidor local, *http://localhost*.

Por defecto, AMPPS utilizará la siguiente ubicación como carpeta principal:

```
C:\Program Files (x86)\Ampps\www
```

Para tener la seguridad de que tienes todo configurado correctamente, debes crear ahora el obligado archivo "Hello World". Por lo tanto, crea un pequeño archivo HTML con las líneas que se detallan a continuación, con el bloc de notas de Windows o cualquier otro programa o editor de texto, pero no un procesador de texto enriquecido como Microsoft Word (a menos que lo guardes como texto plano):

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>A quick test</title>
  </head>
  <body>
    Hello World!
  </body>
</html>
```

Una vez que lo hayas escrito, guarda el archivo en el directorio de la carpeta principal, con el nombre de archivo *test.html*. Si utilizas el bloc de notas, ten en cuenta que el valor de la casilla "Save as type" ("Guardar como") cambia de "Text Documents (*.txt)" ("Documentos de texto (*.txt)") a "All Files (*.*)" ("Todos los archivos (*.*)").

2. Configuración del servidor de desarrollo

Ahora puedes llamar a esta página en tu navegador; escribe el siguiente URL en la barra de direcciones (ver la Figura 2-9):

`localhost/test.html`

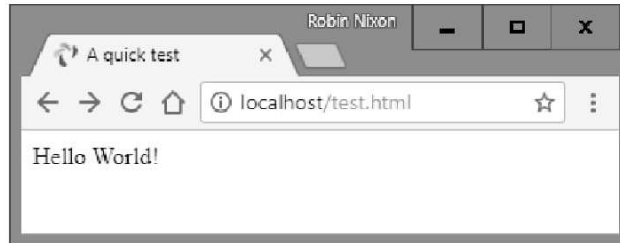


Figura 2-9. Tu primera página web



Recuerda que cargar una página web en un navegador web desde la carpeta principal (o una subcarpeta de la misma) es diferente a cargarla desde el sistema de archivos de tu ordenador. En el primer caso se asegura el acceso a PHP, MySQL y a todas las características de un servidor web, mientras que en el segundo caso simplemente cargará el archivo en el navegador, el cual hará todo lo posible para mostrar la web, pero será incapaz de procesar cualesquiera instrucciones PHP u otras instrucciones del servidor. Por lo tanto, en general deberías ejecutar los ejemplos tecleando el preámbulo *localhost* en la barra de direcciones de tu navegador, a menos que estés seguro de que el archivo no depende de la funcionalidad del servidor web.

WAMP alternativos

Cuando el software se actualiza, a veces funciona de manera diferente a como esperamos que lo haga, e incluso se pueden introducir errores. Así que, si encuentras dificultades que no puedes resolver en AMPPS, puede que prefieras elegir alguna de las otras soluciones disponibles en la web.

Podrás seguir utilizando todos los ejemplos de este libro, pero tendrás que seguir las instrucciones que se facilitan con cada WAMP, que pueden no ser tan fáciles de seguir como las de la guía anterior.

He aquí una selección de algunos de los mejores, en mi opinión:

- EasyPHP
- XAMPP
- WAMPsServer
- Glossword WAMP



Actualizaciones de AMPPS

A lo largo de la vida de esta edición del libro, es muy probable que los desarrolladores de AMPPS realicen mejoras en el software y, por lo tanto, las pantallas de instalación y el método de uso pueden evolucionar con el tiempo, como también puede ocurrir con las versiones de Apache, PHP o MySQL. Así que, por favor, no supongas que algo está mal si las pantallas y el funcionamiento son diferentes a lo explicado aquí. Los desarrolladores de AMPPS ponen el máximo cuidado para que sea fácil de usar, así que solo tienes que seguir las indicaciones y consultar la documentación del [sitio web](http://ampps.com) (<http://ampps.com>).

Instalación de AMPPS en macOS

AMPPS también está disponible en macOS, y puedes descargarlo desde el sitio web (<https://apachefriends.org/>), como se mostraba anteriormente en la Figura 2-1 (cuando escribo esto, la versión actual es la 3.8 y su tamaño es de unos 270 MB).

Si el navegador no lo abre automáticamente una vez descargado, haz doble clic en archivo *.dmg* y, a continuación, arrastra la carpeta AMPPS a tu carpeta *Applications* (consulta la Figura 2-10).

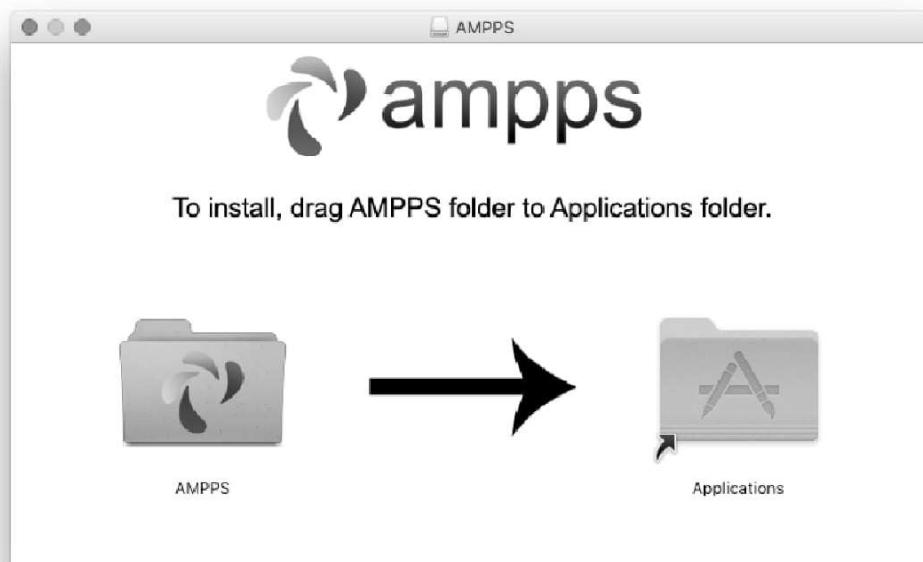


Figura 2-10. Arrastra la carpeta AMPPS a Applications (Aplicaciones)

Ahora abres la carpeta *Applications* de la forma habitual y haz doble clic en el programa AMPPS. Si la configuración de seguridad impide que se abra el archivo, mantén pulsada la tecla Control y haz clic en el icono una vez. Aparecerá una nueva ventana que te